

問題解決能力育成の観点から捉えた中学校技術・家庭科技術分野における 試行錯誤によるプログラミングの学習指導に関する研究課題の展望

Prospects on Research Issues on Teaching Methods of Programming Learning by Trial and Error in the Junior High School Technology Education from the Perspective of Cultivating Problem-Solving Skills.

中尾 尊 洋* 森 山 潤**
NAKAO Takahiro MORIYAMA Jun

本稿の目的は、中学校技術・家庭科技術分野（以下、技術科）のプログラミング学習において、問題解決能力育成の観点からプログラミング学習において効果的に試行錯誤を支援する授業の方法を検討するために、関連する先行研究を整理し、今後の研究課題を展望することである。外国や日本の情報教育に関する諸政策を踏まえたプログラミング教育の位置づけを明らかにし、STEM、STEAM教育やComputational Thinkingの概念、日本のプログラミング教育の動向を踏まえて、問題解決能力を育成する観点から捉えた具体的なプログラミング学習の方法について先行研究をレビューした。その結果、プログラミング教育は、世界的に重要な位置づけとなっており、その目的はプログラムを構築する力の育成にとどまらず、問題を解決するための情報活用能力の育成にあることが明らかとなった。また、具体的なプログラミング学習の実践からは、実社会の問題を解決する製品を簡略的にモデル化し、工夫して制御する内容が多く実践されていた。一方、プログラム自体を試行錯誤して構築する実践に関しては十分とは言えず、中学生の段階では自らプログラムを構築することが困難であると認識されていることが明らかとなった。これらのことより、プログラミングと試行錯誤との親和性について、十分な根拠を示す必要があること、プログラミングにおける試行錯誤に対して「はいまわり」を防ぐ支援を講じるための基礎資料が十分ではないこと、という2つの研究課題について展望した。

キーワード：中学校技術・家庭科技術分野、試行錯誤、問題解決能力、プログラミング学習、研究課題

Key words : technology education, trial and error, problem-solving ability, learning of computer programing, research issues

1. はじめに

本稿の目的は、プログラミング学習において、効果的に試行錯誤を支援する授業方法を確立するため、関連する先行研究を整理し、今後の研究課題を展望することである。

情報技術の進展により、社会や生活は変化してきた。AI技術の進歩により、それまで人間が行ってきた労働の一部を機械に任せられるようになった。また、社会全体でデジタル化が進み、あらゆるデータが集積されることで、ビッグデータとして活用でき、様々な政策や創造等の場面でデータに基づいた意思決定が行われるようになってきている。このような変化の背景として、ドイツでは、2011年にインダストリー4.0構想を公表し、政府が中心となって、情報化を推進してきている。アメリカでは、2012年にIIC (Industrial Internet Consortium)¹⁾という民間のIoT推進団体が発足し、製造、エネルギー、輸送、行政等の幅広い分野でのIoT (Internet of Things)の実現を加速させようとしてきた²⁾。我が国においても、サイバー空間とフィジカル空間を高度に融合させたシステムにより、経済発展と社会的課題の解決

を両立する人間中心の社会としてSociety5.0を実現させようとしており³⁾、現在では諸外国と同様に、ビッグデータを利用した新たな付加価値の創造、AIの自律学習による判断やロボット技術の向上等、第4次産業革命が達成されている⁴⁾。今後ますます発展し、Society5.0が実現されていく社会においては、情報を適切に扱い活用する重要な資質・能力である情報活用能力の育成が必要と考えられる。

情報活用能力は、1986年の臨時教育審議会第二次答申にて示された概念で、「情報及び情報手段を主体的に活用していくための個人の基礎的な資質」と示された⁵⁾。さらにこの答申では、読み、書き、算盤と並び、リテラシーとして情報活用能力を扱うことが提言されている。また、小学校段階における論理的思考力や創造性、問題解決能力等の育成とプログラミング教育に関する有識者会議では、情報技術の発展における社会の変化を捉え、小学校段階からプログラミング教育を導入することが提言された⁶⁾。その結果、2017年に告示された学習指導要領では小学校においてプログラミング教育の内容が盛り込まれ⁷⁾、小学校、中学校、高等学校と継続的

* 兵庫教育大学大学院（博士課程）教科教育実践学専攻生活・健康系教育連合講座

令和4年7月13日受理

** 兵庫教育大学大学院学校教育研究科人間発達教育専攻生活・健康・情報系教育コース 教授

にプログラミング教育が行われることとなり、各校種ごとに目標とすべき資質・能力が示された⁸⁾。その学習内容においては、発達段階に応じたものとされており、小学校では、身近な生活の中で気付きを促したり、各教科等で身に付いた思考力をプログラミング的思考につなげたりすること、中学校、高等学校では、それぞれの学校段階における子供たちの抽象的思考の発達に応じて、構造化された内容を体系的に教科学習として学ばせることが示されている⁸⁾。いわば、小学校では、プログラミングがどのように生かせるのかを体験する段階、中学校では、課題の解決のために処理の手順を可視化・統合化して設計する等問題解決的な思考を育む段階、高等学校では、目的に応じたアルゴリズムを考え、コンピュータやネットワークを利用したプログラミングを通して、モデル化やシミュレーションの実行とともに問題の適切な解決方法を考え出す段階と考えられる。この時、プログラミングが体験として扱われる小学校段階と違い、問題解決の手段として扱われる中学校段階では、実際にプログラムを思考して作成する学習が導入されるため、命令の選択等に関して試行錯誤する必要性が生じ、学習内容の難易度について困難感が大きくなるのではないかと推察される。

そこで本稿では、プログラミングについて中学校で主として扱われる技術・家庭科技術分野（以下、技術科）におけるプログラミング学習の効果的な試行錯誤の支援を検討するために、関連する先行研究を整理し、今後の研究課題を展望することとした。

2. プログラミング教育の位置づけ

2.1. Computational Thinking を踏まえたプログラミング教育の動向

プログラミングとは機械の動作を自動化させるために、機械用の言語を用いた指示の記述を行うことである。指示の記述にはコンピュータを用いるため、プログラミング教育とはコンピュータの操作に関する学習と考えがちである。例えば、中尾らは、プログラミング未経験の大学生を対象に、プログラミング学習に対するイメージとコンピュータ親和度との関連について調査し、コンピュータに対して肯定的な学生ほどプログラミングへの意欲が高いことを報告しており⁹⁾、コンピュータに対するイメージがプログラミングへの意欲に関連することを示唆している。

しかし、Wing による、Computational Thinking（以下、CT）の概念¹⁰⁾では、プログラミング教育はコンピュータの操作に関する学習に限らない教育的意義を有していることが示されている。林の報告では、CTには Computer Science（以下、CS）を喧伝する意図があり、CSがコンピュータのみに適用される意識を変えるために端的な定義記述を明確にせず、CS研究者や実践者のグランドビジョンとなるべく提起された概念とされている¹¹⁾。また同報告に、Wingが社会のあらゆる場でコンピューティングの概念と方法が活用される未来を見

通して、コンピュータ科学者の思考の仕方を伝えることが重要と考えていたことにも触れられている。このことはCTに関する様々な解釈を生みやすくしているといえよう。しかし、そのことで、CTの概念を矮小化して捉えてしまう懸念もあり、日本におけるプログラミング的思考の概念が十分ではないとする議論もある¹²⁾。

こうした中で、プログラミング教育の意義については様々な考察が行われている。例えば、山本らは、プログラミング教育に関する教育的意義を探究力や論理的思考力等の7項目に整理し、効果が期待できることを指摘している¹³⁾。また、松らは、プログラミング教育において獲得すべき資質として、抽象化・モデル化出来る能力や課題解決（創造）する能力等の5項目を挙げ、プログラミング教育の具体的なモデルケースを提案している¹⁴⁾。このようなCTやプログラミング教育の意義に関する考察から、プログラミング教育は、コンピュータに関連する技術を身につけさせること以上に、論理的思考力や創造力を高め、問題解決するための資質・能力を身につけさせるための教育として認識されるようになってきている。

2.2. プログラミング教育と STEM/STEAM 教育

STEM教育はアメリカ合衆国において提唱された。Science, Technology, Engineering, Mathematicsの頭文字をつなげたものであり、科学技術の進展が予測される中、科学技術を牽引する人材の育成や市民の科学技術に関するリテラシーを向上させることを目指した教育運動である¹⁵⁾。元々、アメリカ合衆国における科学技術の進展に伴って変化していく社会における市民性の向上を目指したものであったが、近年では、STEMにArtsのAを含んだSTEAM教育として教科横断的な概念を強くして世界中で推進されつつある。ただし、STEAM教育の概念に関しては様々な解釈があり¹⁶⁾、黒田らは、アメリカ合衆国と異なり、日本ではSTEM教育の議論が充分行われないうまま、STEAM教育が国策として導入され、教科横断的な学習との区別が不明瞭な状態であると指摘している¹⁷⁾。また、川原田らは、STEAMのAにあたるリベラルアーツについて、初等中等高等教育段階を一貫したリベラルアーツ教育を推進する主張を指示しており、STEAM教育の要件であるサイエンスとアーツをエンジニアリングとデザイン方略で架橋する最適解の導出を重視した実践の重要性を指摘している¹⁸⁾。

これらの議論はあるものの、STEM/STEAM教育は科学技術の高度化における新しい学習の在り方についてフレームワークを示したものといえる。ヤング吉原らは、STEAM教育によってイノベーション人材を育成する重要性について述べており、STEAMの目的、マインドセット、メソッドとして、それぞれ「ヒューマニズム」、「イノベーターのマインドセット」、「デザイン思考」を挙げている¹⁹⁾。このことは、人間を中心とした新しい社会を築くことを中心的な課題とし、その解決策として、科学技術による新しい価値やサービスの創出が必要

であることを示唆している。そのためには、プログラミングやデータ解析の知識を持ちつつ新たな課題を発見できる力が必要であり、これらを実現するメソッドとしてデザイン思考が重視されていると考えられる。また、デザイン思考においては、「Empathize (共感)」、「Define (定義づけ)」、「Ideate (発想)」、「Prototype (試作)」、「Test (試験)」という5つのプロセスを経ることが提案されており、これらを繰り返すことが新しいアイデアを発想する上で重要と考えられている²⁰⁾。つまり、今後の社会では、デザイン思考による問題解決に必要と考えられるこれらのプロセスのように、プロトタイプにより実際に形を作りだして評価するプロセスが重要と考えられ、Society5.0を目指す社会に対してのリテラシーとしてプログラミング教育は重要な位置づけにあるといえる。このようにSTEM/STEAM教育の観点からプログラミング教育の位置づけを捉えると、創造性を高めるための実践的な能力の育成に対してプログラミング教育への重要性が高まっていると考えられる。

2.3. 日本の学校教育におけるプログラミング教育

このような潮流の中で、我が国における情報教育にも大きな変化が見られるようになってきた。「教育の情報化に関する手引き (追補版)」では、「世の中の様々な事象を情報とその結び付きとして捉え、情報及び情報技術を適切かつ効果的に活用して、問題を発見・解決したり自分の考えを形成したりしていくために必要な資質・能力」を3観点8要素に整理し、情報活用能力と定義している⁸⁾。この情報活用能力は、小学校プログラミング教育の導入における議論の中で、問題発見・解決能力と並ぶ「学習の基盤となる資質・能力」のひとつと位置づけられ、小学校、中学校、高等学校にわたって段階的に育成していくことが示された⁶⁾。

例えば、小学校段階においては、特定の教科を設定せず、様々な教科においてプログラミング的思考を育成することが目標とされている。プログラミング的思考は、「自分が意図する一連の活動を実現するために、どのような動きの組合せが必要であり、ひとつひとつの動きに対応した記号を、どのように組み合わせたらいいのか、記号の組合せをどのように改善していけば、より意図した活動に近づくのか、といったことを論理的に考えていく力」⁶⁾とされている。つまり、プログラミング作業時のように論理的に物事を組み立てる思考方法の育成が目指されている。一方、発達段階が考慮され、コンピュータに関する専門的な知識等は求められてはいない²¹⁾。

中学校段階では主に技術・家庭科技術分野(以下、技術科)の内容「D 情報の技術」にプログラミングが導入されている。技術科の目標は、「生活の営みに係る見方・考え方や技術の見方・考え方を働かせ、生活や技術に関する実践的・体験的な活動を通して、よりよい生活の実現や持続可能な社会の構築に向けて、生活を工夫し創造する資質・能力」を育成することとされており²²⁾、プログラミングにおいても、その目標を達成する手段とし

て実施される。具体的な内容としては、2008年告示の学習指導要領より計測・制御に関する内容が盛り込まれ²³⁾、2017年告示の学習指導要領では、ロボティクスやフィジカル・コンピューティングにもつながる内容に発展している²²⁾。また、同学習指導要領より、「ネットワークを利用した双方向性のあるコンテンツのプログラミング」が新たに導入された。この内容は、急速に発展してきたネットワーク技術に対応するものとなっている。このように技術科におけるプログラミングの内容は、社会や生活における技術的な諸問題を解決する方法のひとつとして位置づけられており、小学校における内容に比べて、より高度な課題設定が想定されている。

高等学校においては、2018年告示高等学校学習指導要領により共通必修修科目として情報Ⅰが新設され、2022年度より全面実施となった²⁴⁾。情報Ⅰでは「情報に関する科学的な見方・考え方を働かせ、情報技術を活用して問題の発見・解決を行う学習活動を通して、問題の発見・解決に向けて情報と情報技術を適切かつ効果的に活用し、情報社会に主体的に参画するための資質・能力」を育成することが目指されており、ここでも問題解決能力を育成することが示されている。具体的な内容としては、「情報社会の問題解決」、「コミュニケーションと情報デザイン」、「コンピュータとプログラミング」、「情報通信ネットワークとデータの活用」の大きく4つに分けられている。このうち、プログラミングに関する内容は主に「コンピュータとプログラミング」において設定され、コンピュータの仕組みやシミュレーション、アルゴリズム等、小、中学校の学習内容を発展させた内容となっている。

このように、校種を超えて情報活用能力の育成が一貫して目指され、小学校、中学校、高等学校と難易度を段階的に上げながら学習を進めていくカリキュラムが設定された。しかし、それを実現するための問題点も指摘されている。例えば、山崎らは、校種間の連携を図るためには、音楽、図画工作、美術、家庭、技術、情報といった教科が重要な役割を果たすが、これらの教科担当教員が減少していることを問題視し、教員養成においてプログラミング教育の充実を図る必要性を指摘している²⁵⁾。また、安本らは、校種間の接続に配慮した実践研究として、扱うプログラミング言語に、小学校においてScratch、中学校においてPythonを用い²⁶⁾、高等学校での実践にスムーズに移行できるように配慮した実践を行っている。しかし、中学校では例題プログラムの修正や穴埋めによるものとしており、中学校段階での、テキストベースのプログラミング言語を用いたプログラミングの困難さが認識されている。校種間の連携を踏まえて、中学校段階におけるプログラミング教育の充実と効果的な学習方法の必要性が一層高まっていると考えられる。

以上、プログラミング教育は、科学技術の発展に付随する重要なリテラシーを育成する教育として重視されている。そのため、問題解決能力の育成という文脈に位

置づけられ、情報活用能力の育成を目的として導入されている。しかし、その実践に際しては、次に挙げる問題点が指摘される。

3. 問題解決能力育成の視点から見たプログラミング教育の授業とその問題点

3.1. 問題解決能力育成の文脈におけるプログラミングの問題点

以上に述べてきたように、プログラミング教育は問題解決能力育成の文脈に位置づけられており、学習基盤としての情報活用能力を育成する手段として位置づけられている。よって、プログラミング教育では、単にプログラムを作成させればよいのではなく、実社会における問題に対して適切な解決に導くために、どこで、どのようにプログラムを活用するのか、解決に適したプログラムとはどのようなものなのかを判断させる実践的な活動が求められている。プログラミングを問題解決に位置づけた実践的な研究を見ると、例えば木村らは、技術科における統合的な問題解決の事例として、自動ドア、お掃除ロボット、植物工場等を題材とし、その一部を開発させる指導過程を提案している²⁷⁾。この中では、発達段階に応じた問題の難易度が設定され、必要とされるプログラム自体も複雑なアルゴリズムを伴わないように配慮されている。また、森石らは、実社会における問題を解決させる製品をモデル化できるように、多様なセンサやアクチュエータが接続できるインターフェースを製作し、Logo Writer で操作させる教材を開発しているが、生徒にとってプログラミングが困難と判断される動作については、あらかじめモジュールを作成しておき、利用させている²⁸⁾。

これらの実践では、実社会における問題の複雑さに対して、中学生の段階でプログラムによる解決方法を考え出させることが困難と捉えられていることがわかる。よって、実社会の問題を切り分け、ある程度のサイズでパッケージ化したり、問題の一部のみを扱ったりすることで、解決が困難と考えられる部分については触れずに授業を構築している。プログラムに関しても複雑になる場合については、アルゴリズムをモジュール化することで、アルゴリズムの内容を理解できなくても動作させることができるよう、単純な構造にして構築させる方法が採られている。このような実践は、問題解決の文脈がプログラミング以外の場面であり、問題を解決する手段としてプログラミングが用いられている。

一方、新宮は、プログラミングには、プログラムを作ったり理解したり誤りを修正したりという作業があり、プログラミング自体が問題解決として捉えられると指摘している²⁹⁾。このような観点から、森山らは、中学生を対象としたプログラミングの思考過程を分析し、問題を理解する過程や理解した問題を命令や文法に沿って変換する過程の重要性を指摘している³⁰⁾。また、宮川らは、生徒が課題の内容を論理的に分析し、プログラム言語を用いて実現するという問題解決の過程が重要で

あることを指摘し、オブジェクト指向イベントドリブン型のプログラミングにおける問題解決過程の分析により、「考えながら作る」や「作りながら考える」といったプログラミングの問題解決が展開されやすいことを示唆している³¹⁾。これらの指摘は、プログラミングが困難であることを理由にモジュール化してしまうのではなく、アルゴリズムを考えること自体を問題解決として捉えることの重要性を示唆している。

3.2. 写経プログラミング学習とその問題点

プログラミングそのものに対する学習（以下、プログラミング学習）の方法として、ある程度のアルゴリズムをモジュールとして構築する方法以外に、写経プログラミングといわれるサンプルプログラムをそのまま入力させる教示的な授業が行われていることが多い。岡崎らは、プログラミング体験イベントにおいて最も多い形式として講義型を挙げており、この特徴として、逐次的にプログラミングを教示し、達成を確認しつつ次の教示が行えることを指摘している³²⁾。このような方法が多く用いられている理由としては、プログラミング言語が未習得の状態ではプログラムの構築が困難と考えられていることが推察される。そもそもプログラムは、プログラミング言語のルールに基づいて命令等を設定し、手順を組み立てたものである。人間が日常的に用いている自然言語との違いが大きく、プログラムを動作させるためには、言語のルールや命令の扱い方を理解する必要がある。近年では、自然言語を活かしたコードの記述を可能とする新しいプログラミング言語を開発しようとする動きも見られるものの³³⁾、基本的には論理的に命令の順序を組み立てる必要があり、目的とする動作に対して適切な順序を探る必要がある。目的とする動作に対する適切な順序を探るためには、命令の種類や意味、構文を知らなければならない。このような特徴から、プログラミング学習に関する様々な実践事例では、プログラムを構築する困難さを捉えており、教示的な指導が多いものと推察される。この場合のプログラミング学習の方法として、提示されたプログラムをコンピュータに打ち込む活動を主とした、いわゆる写経プログラミングを行うことが考えられる³⁴⁾、³⁵⁾、³⁶⁾。写経プログラミングによってプログラムを入力し実際に動作させることは、未知の状態からプログラミングの経験を積ませ、動作と構築された命令の手順を対比させることによって命令の種類や意味、構文について理解を促す方法といえる。

しかし、写経プログラミングについては、情意面や認知面に関する長所と短所が指摘されており科学的に是非が結論づけられていない³⁷⁾。このような状況から、例えば亀井らは、写経プログラミング学習の方法は、場合によっては思考を伴わない単調な作業となるため理解が不十分になる可能性を指摘し、意欲を向上させるツールを開発している³⁸⁾。これらの議論を見ると、写経プログラミングの目的はプログラミングの力を向上させることであり、そのためプログラミングの技術を習

得することに対しての効果が問われている。学校教育においては、前述の通り、問題解決能力育成の文脈にプログラミングが位置づけられており、情報活用能力を向上させることが目的である。問題を命令や文法に沿って変換する過程を重視する観点から、命令や文法の意味を理解せずに写経的に入力させることは、問題解決能力育成の観点からは、必ずしも十分とは言えないのではないかと考えられる。

3.3. 試行錯誤させることの重要性

モジュール化されたアルゴリズムによるプログラミングや写経プログラミングは、アルゴリズムの内容を理解していなくても作成可能である。いわば、完成されたパッケージであるが故に、自分の期待する動作に対して、どのような命令をどのような手順で用いればよいのかを考えなくとも、動作させることが可能となる。このことは、実動作と命令の手順との対応に関して、パッケージ内の動作については考える必要が無く、理解を深めにくい状況になりやすい（ブラックボックス化）。問題に対して様々な条件を踏まえて最適解を導くことが重要である問題解決の観点からは、むしろプログラムのパッケージの内容にどのような意図が含まれているのかを理解していくことこそ重要な学習になると考えられる。その際、技術的なプロジェクトとして目標拡散的、または方法拡散的に課題を設定することで問題解決を具体化することが重要とされる技術科の学習においては³⁹⁾、目的とする動作に対する、命令やその手順へのアプローチを拡散的に試行することが重要と考えられる。このような点を踏まえると、プログラミング学習においては、自分の考えで命令の手順を組み換えたり、使う命令を入れ替えたりすることによって、実動作がどのように変化するかを体験させる試行錯誤によってプログラムを構築させることが重要ではないかと考えられる。

4. プログラミングにおける試行錯誤

4.1. プログラミングにおける試行錯誤の捉え方

試行錯誤とは、一般的に「課題が困難な時、何回もやってみて、失敗を重ねながらもだんだんと目的に迫っていくという仕方」と捉えられている⁴⁰⁾。試行錯誤と学習との関係については、古くから研究が行われている。Thorndikeは、行動主義の視点から試行錯誤を捉え、報酬が刺激と反応の連合を強めると指摘した⁴¹⁾。無秩序に行動しても偶然に解決できた際にそれが報酬となり、学習されるというものである。これは、観察可能な行動のみに焦点が当てられており、理解や推論、思考などの認知過程の研究が行われていなかった当時の限界といえる⁴²⁾。

近年では認知科学の研究も盛んになり、問題解決や創造性に関する認知的な分析等、様々な研究が行われている。中でも洞察問題解決に関しては、答えがわからず行き詰まっていた状態から、問題の解法が突然にひらめ

く体験となるため、感動を伴い⁴³⁾、学習において目指す問題解決と考えることができる。洞察問題解決では、WallasやOhlssonによって、「行き詰まり」、「あため」、「ひらめき」、「検証」の4段階の過程が明らかにされている⁴⁴⁾、⁴⁵⁾。このうち特に「行き詰まり」の際には、問題の解決に向けて様々な試行が行われ、解決に向けての制約の強度を弱めていくことが示されている⁴⁶⁾。寺井らも洞察について、突然表象が変化したかのように感じるが、外部からのフィードバックを受けながら、制約が漸進的に緩和されてゆくプロセスの存在を示している⁴⁷⁾。

プログラミングは、バグと呼ばれるエラー等によって「行き詰まり」が生じ、試行を通じて解決に向けての制約の強度を弱めることができる点で洞察問題解決と捉えることができる。つまり、プログラミングにおける試行錯誤とは、無秩序な行動による偶然の解決を捉えているのではなく、「行き詰まり」の際に解決に向けての制約の強度を弱める行動と捉えられる。これに関して、Cinii Reserchにおいて「プログラミング」、「試行錯誤」というキーワードで検索したところ113件の論文が該当した。これらの中には、プログラミングを含む問題解決における試行錯誤に着目したもの、プログラミング自体の試行錯誤に着目したもの、プログラミングによる試行錯誤を期待したもの等があり、小学校、中学校、高等学校、特別支援学校、大学と様々な校種に対して実践的に研究が行われている。これらの研究において試行錯誤という言葉は、試行を通じて「行き詰まり」の制約が漸進的に緩和されてゆく文脈で使われている。

また、デザイン思考研究においても試行錯誤が着目されている。例えば野口らは、試行錯誤について、「何かを求めているが、その答えが簡単には得られず、何度も仮の解を出しては、その中からもっとも適切な解を選び出し、そのつど評価のフィルターに掛けながら次第に求める解に近づく思考」とし、人間が自然の中で生き抜くために「時間の先取り」能力として発達させた、過去の経験を予測のために役立てる推論力、その中からある一般的なルールを抽出する抽象力、の典型としている⁴⁸⁾。試行錯誤の過程については、失敗をくり返ししながら目的・手段関係の連鎖を構築して問題を解決していく過程と捉え、それをなくしたり、減らしたりするのではなく、より深い思考をもたらず試行錯誤に転換させることが創造的思考を高めるうえで必要であると論じている⁴⁹⁾。つまり、試行錯誤には目的への方向性があり、拡散的に解決方法を探索したり、失敗からの情報を得て解決方法を収束的に思考したりする過程が含まれ、そのことが創造的な問題解決の力を育成すると考えられている。このような視点での研究として池田らは、高校生を対象とした探究的な学習の実践において、発散と収束を繰り返すデザイン思考プロセスを活用した報告をしている⁵⁰⁾。また、試行錯誤の支援については、試行の差分を時系列に並べることで、思考のパターンに内在していたが気づけなかった思考の癖に気づくきっかけを与えること

が出来ると指摘されている⁵¹⁾。ここで述べられている、思考のパターンに内在していたが気づけなかった思考の癖によって、同じ試行のくり返しや、意味の無い試行が誘引され、いわゆる「はいまわり」となり得ると考えられる。この状況は、いつまでも個人的な試行錯誤をくり返し、うろついていることにしか過ぎない⁵²⁾状況と考えられる。

一方、試行錯誤の視点からプログラミング学習を捉えると、*tinkering*によるプログラミングが考えられる⁵³⁾。*tinkering*は、直接的な経験、反復、実験、発見を通じて問題にアプローチして解決する遊び心を含んだ方法に関する考え方と解釈されている⁵⁴⁾。原形である*tinker*は「いじくりまわす」という意味に和訳され、*tinkering*はあれこれといじくりまわしているイメージで捉えられ、遊びと学びの融合による新しい学習の機会として期待されている。山崎らは、*tinkering*をSTEAM教育の概念に位置づけ、初等教育段階におけるコンピューティング教育の活動において重要な考え方であることを指摘している⁵⁵⁾。様々な命令を選択したり、あれこれ順番を並び替えたりと、プログラムをいじくりまわして試行錯誤をすることで、命令やその手順を構築した結果がどのような動作につながるのかを深く理解し、問題解決能力を高めることが期待される⁵⁶⁾。

以上の試行錯誤に関する先行研究からは、プログラミング学習における試行錯誤の捉え方として次のように整理できる。プログラミングにおける試行錯誤を行動主義の観点から捉えると、刺激と反応を強化させるための報酬が得られることが重要となるため、試行錯誤の過程で得られる試行の差分に関する重要な情報を見落とすことになる。よって、洞察問題解決のように、行き詰まりの状態から解決に向けての制約の強度を弱める行動として試行錯誤を捉えるべきと考えられる。このような捉え方をすることで、プログラムの完成にむけて*tinkering*のように命令や手順をいじくりまわして試行錯誤することに対して、試行の差分から自らの試行パターンに内在している思考の癖に気づかせ、「行き詰まり」をもたらしている制約の強度を弱めているという意味を持たせることになる。「教育の情報化に関する手引き(追補版)」では、プログラミングにおいて試行錯誤することが求められている⁸⁾が、その意図は試行錯誤によって、上述した自らの思考の癖を自覚的にし、拡散的な探索と収束的な思考を促し、創造的な問題解決の力を育成することにあると言える。プログラミング教育において問題解決能力を育成することは重要な目的である。そのために、プログラミング学習において試行錯誤させることで、より効果的な問題解決能力を育成できるのではないかと考えられる。

4.2. 試行錯誤に関する授業実践の先行研究

次に、プログラミングにおける試行錯誤の観点から、授業実践に関する先行研究を概観した。まず、試行錯誤は問題解決の過程において表出すると考えられること

から、プログラミングによる問題解決に着目した先行研究についてレビューした。こうした内容の先行研究は非常に多く、様々な実践例が報告されている。例えば、黒田らは、小学校において生活の課題に対してレゴ WeDo⁵⁷⁾を用いて解決する方法を試行錯誤させ、モデルを作成して解決方法を考案する実践を行い、技術リテラシーの向上に効果的であることを報告している⁵⁸⁾。萱津らは小学生対象のプログラミング体験教室における簡易的なゲーム作りの実践を行い、プログラミングの興味、他者と協力する力、創造力等を育むことに効果的であることを報告している⁵⁹⁾。山本らは、中学校において教育用マインドストーム NXTを活用して、身の回りの自動制御された製品に関する模型を作成させ、プログラムで制御させる実践を通して、学習意欲が高まり、技術を評価することが出来たと報告している⁶⁰⁾。これらの先行研究は、問題解決の手段としてのプログラミングが、創造性や学習意欲等の情意面に効果的に働いていることを報告している。

また、技術科における実践研究に着目すると、例えば萩嶺らは、プログラムによる計測・制御の学習における題材タイプの違いによる、情意面や技術的な見方・考え方の傾向を報告し、用いる教材によって学習の効果に影響を及ぼすことを明らかにしている⁶¹⁾。藤田らは、小型コンピュータを用いた教材を開発し、計測・制御の学習においてプログラミングのプランニングの困難さを報告し、教材の効果を検討している⁶²⁾。これらは、プログラミングによる問題解決を実践する際の教材に対して効果を検討したものと見える。また、古平らは、自律型ロボット教材を開発するとともに授業計画を立案、その有用性を検討した結果、資質・能力の育成に対して効果的であることを報告している⁶³⁾。針谷らは、市販のロボット教材を用いた計測・制御のプログラミングに関する授業計画を立案し、効果を検討している⁶⁴⁾。これらは、問題解決を授業計画として具体化し、その効果を検討したものと見える。また、増田らは、状態遷移図を用いて⁶⁵⁾、青木らはシミュレーションツールを用いて⁶⁶⁾、それぞれ、問題解決を目指したプログラミングの支援を検討し、その効果を検討している。このように技術科の先行研究を見ても、プログラミングによる問題解決能力の育成に着目したものが多く、技術科は、技術的な問題を題材として問題解決能力を育成することを重視しており、その観点から、技術科における実践研究では、問題解決の全体像を捉え、教材、学習計画、思考を支援するツール等の効果について検討するものが多くなっていると考えられる。しかしこれらは、問題解決全体における試行錯誤を捉えているものであり、プログラミングそのものを問題解決と捉えたときの試行錯誤について検討されているわけではない。

そこで、プログラミング学習において試行錯誤を支援する観点での先行研究をレビューすると、次のような研究が挙げられる。例えば、北島らは、何らかの間違いが含まれることを前提に、コーディングを介することで

バグの存在をコンピュータに教えてもらい、またそのフィードバックをもとにバグを特定し、修正していくという試行錯誤を通して、論理的な思考力を子どもが獲得していくものであると指摘し、それを支援する教材を開発している⁶⁷⁾。知見らは、大学生のプログラミング学習の実践において試行錯誤によってプログラミングさせ、その際に失敗情報に基づいて内省させる実践により、学習の効果について検討したところ、アチーブメントテストにより知識が得られていることが確認され、アンケートの記述からは肯定的なコメントが得られたことを報告している⁶⁸⁾。これらの先行研究では、プログラミングの試行錯誤によって論理的な思考やプログラミングの知識に効果が示されていることを報告している。しかし、これらは中学生を対象としているものではなく、また、試行錯誤を具体的に分析して支援の方法を検討したものではない。

一方、プログラミングに限らず、技術科の学習全般から試行錯誤に着目すると、例えば佐田らは、問題表象と修正行動との関連を検討し、結果において、作業中の問題表象により、新たなプランとそれに基づく修正行動の関係が推測されたことを報告している⁶⁹⁾。つまり、試行錯誤は教師に促されているのではなく、技術的な作業の過程において得られる問題表象を契機に促されるものであると考えられる。このことを捉えると、技術的な活動経験が試行錯誤を生起させると考えられるため、プログラミングにおいても試行錯誤が生起すると考えられる。しかし、どのような技術的な活動が試行錯誤と親和性が高いのかは明確になってはいない。

以上、試行錯誤に関連する先行研究から、プログラミングを問題解決に位置づけられた先行研究は多いものの、中学校技術科における授業において、プログラミング自体を問題解決と位置づけ、試行錯誤のプロセスを明らかにした上での効果的な支援の方法については十分に検討されていないことが明らかになった。また、技術的な活動経験が試行錯誤に対して何らかの効果を与えると考えられるが、自立的に試行錯誤を促す観点から、どのような活動経験が試行錯誤に対して親和性が高いのかを検討することも重要であることが明らかになった。

5. 研究課題の展望

5.1. 問題点の整理

以上、ここまでプログラミング教育に関する先行研究および試行錯誤に関する研究のレビューを行った。ここからは、レビューを通してプログラミング教育における問題の所在について検討し、研究課題を展望する。

プログラミング教育が行われる目的として、情報活用能力や Society5.0 を実現させる人材の育成やそのような社会を生き抜くリテラシーを育成することが挙げられた。教育においては、STEM/STEAM 教育や Computational Thinking の概念が重視され、問題解決にプログラミング教育を位置づけることが重要であるこ

とが指摘され、単にプログラムを作成できる力を育成するだけでは十分とはいえないことが把握された。このような背景から、現在行われているプログラミング教育の実践では、プログラミングによって問題を解決する学習が効果的と考えられ、問題に対してどのようにプログラムを適用するのかを試行錯誤させる実践事例が多く研究されていることが確認されたものの、中核となるプログラムの命令や構文について試行錯誤させる実践については十分に検討されているとはいえないことが明らかになった。さらに、問題解決において自立的に試行錯誤させる観点から、技術的な活動経験と試行錯誤との関連性を検討する必要性が明らかになった。

5.2. 研究課題の展望

これらのことを踏まえると、試行錯誤を取り入れたプログラミング学習の指導方法に関する研究課題として次の2点が考えられる。

第1に、プログラミングと試行錯誤との親和性について、十分な根拠を示す必要があることである。情報活用能力を育成するために、プログラミング教育への期待は高まっている。その実践場面といえるプログラミング学習に対して様々な研究も行われている。しかし、プログラミング学習において試行錯誤させることの効果については十分な検討が行われていない。これに対して、実証的な調査等によって根拠を示すことで、プログラミング学習における試行錯誤の重要性を示していくことが求められる。

第2に、プログラミングにおける試行錯誤に対して「はいまわり」を防ぐ支援を講じるための基礎資料が十分ではないことである。プログラミング学習に関する研究のうち、実社会の問題に関連させ、プログラミングを通して解決方法を試行錯誤して工夫させる内容の実践研究は充実している。しかし、プログラムを構築する際の試行錯誤場面において、どのような思考が働いているのか、どのように「はいまわり」のような現象が起きているのか、起きているのであれば何がその原因となるのか等は必ずしも明らかではない。

これらの問題に対してアプローチすることで、技術科でこれまで行われていた写経的なプログラミングやパッケージ化されたアルゴリズムの選択のような学習方法に加えて、効果的に試行錯誤させてプログラムを構築させる指導方略を確立することができると期待される。

6. おわりに

本稿では、技術科において効果的な試行錯誤によってプログラムを構築させる指導方略の開発に向けて、社会的な意義を踏まえたプログラミング教育に対する期待を整理し、学校現場において実践されているプログラミング学習に関する先行研究のレビューをすることで、その現状を検討した。その結果、多くの授業実践では実社会や身近な問題に関連する題材を扱い、モデル化して

解決する実践が行われており、問題を解決するための情報活用能力の育成に効果が示されていることが明らかとなった。しかし、プログラムの構築自体を試行錯誤させる研究については十分とは言えず、プログラミングと試行錯誤との関連性についての調査やプログラミングで試行錯誤させる際の「はいまわり」への危惧と対策の重要性を指摘した。

今後は以上の指摘を検討するために、①プログラミングと試行錯誤との親和性について検討すること、②プログラミングにおける試行錯誤の効果について検討すること、③試行錯誤における「はいまわり」が生起する場面やそうではない場面の思考過程を検討すること、④「はいまわり」への支援を講じた授業実践の効果について検討すること等が必要と考えられる。

なお、筆者らは①に対して中学生の技術的な活動経験と試行錯誤に対する意識との関連性について検討している⁷⁰⁾。また、②に対して、中学生に対して試行錯誤によるプログラミングを行かせた際の効果について検討している⁷¹⁾。これらの研究が上述した研究課題の解決に寄与することを期待する。

参考文献

- 1) IIC: "Industry IoT Consortium", <https://www.iiconsortium.org/>
- 2) 野中洋一・福本勲・山本宏・高梨千賀子: インダストリアル IoT に関する日米独の最新動向, 研究 技術計画, vol.33, no.4, pp.299-314 (2018)
- 3) 内閣府: "Society5.0", https://www8.cao.go.jp/cstp/society5_0/index.htm
- 4) 内閣府: "第4次産業革命のインパクト", https://www5.cao.go.jp/keizai3/2016/0117nk/n16_2_1.html
- 5) 文部科学省: "情報教育に関連する資料", 教育課程部会情報ワーキンググループ資料8, https://www.mext.go.jp/b_menu/shingi/chukyo/chukyo3/059/siryo/_icsFiles/fieldfile/2015/11/11/1363276_08_1.pdf
- 6) 小学校段階における論理的思考力や創造性、問題解決能力等の育成とプログラミング教育に関する有識者会議: "小学校段階におけるプログラミング教育の在り方について(議論の取りまとめ)", http://www.mext.go.jp/b_menu/shingi/chousa/shotou/122/attach/1372525.htm
- 7) 文部科学省: 小学校学習指導要領(平成29年告示), 東洋館出版社(2018)
- 8) 文部科学省: "教育の情報科に関する手引き(追補版)", https://www.mext.go.jp/a_menu/shotou/zyouhou/detail/mext_00117.html
- 9) 中尾茂子・安達一寿: プログラミング学習に対するイメージとコンピュータ親和度との関連, 教育情報研究, vol.10, no.4, pp.3-10 (1994)
- 10) M. Wing Jeannette・中島秀之: Computational Thinking 計算論的思考, 情報処理, vol.56, no.6, pp.584-587 (2015)
- 11) 林向達: Computational Thinking に関する言説の動向, 日本教育工学会研究報告集, vol.18, no.2, pp.165-172 (2018)
- 12) 阪東哲也・黒田昌克・福井昌則・森山潤: 我が国の初等中等教育におけるプログラミング教育の制度化に関する批判的検討, 兵庫教育大学学校教育学研究, vol.30, pp.173-184 (2017)
- 13) 山本利一・本郷健・本村猛能・永井克昇: 初等中等教育におけるプログラミング教育の教育的意義の考察, 教育情報研究, vol.32, no.2, pp.3-11 (2014)
- 14) 松秀樹・難波宏治: プログラミング教育の意義に関する研究, 園田学園女子大学論文集, vol.52, pp.49-60 (2018)
- 15) Bybee Rodger: What Is STEM Education?, Science, vol.329, no.5995, p.996 (2010)
- 16) 辻合華子・長谷川春生: STEAM 教育における "A" の概念について, 科学教育研究, vol.44, no.2, pp.93-103 (2020)
- 17) 黒田昌克・森山潤: STEM/STEAM 教育の観点から見た小学校プログラミング教育の在り方に関する研究課題の展望, 兵庫教育大学学校教育学研究, vol.33, pp.189-200 (2020)
- 18) 川原田康文・磯部征尊・上野朝大・大森康正・山崎貞登: STEAM 教育とエンジニアリング・デザイン教育を重視した小・中学校を一貫したプログラミング学習, 上越教育大学研究紀要, vol.40, no.1, pp.307-317 (2020)
- 19) ヤング吉原麻里子・木島里江: 世界を変える STEAM 人材: シリコンバレー「デザイン思考」の核心, 朝日新聞出版(朝日新書) (2019)
- 20) HASSO PLATTNER Institute of Design at Stanford: "An Introduction to Design Thinking Process Guide", <https://www.web.stanford.edu/~mshanks/MichaelShanks/files/509554.pdf>
- 21) 文部科学省: 小学校プログラミング教育の手引き(第三版) (2020)
- 22) 文部科学省: 中学校学習指導要領(平成29年告示) 解説 技術・家庭科編, 開隆堂(2018)
- 23) 文部科学省: 中学校学習指導要領解説 技術・家庭編, 文部科学省, (2008)
- 24) 文部科学省: 高等学校学習指導要領(平成30年告示), 東山書房, (2018)
- 25) 山崎貞登・大森康正・磯部征尊・上野朝大: プログラミング教育の小・中・高各校種間連携・一貫教育推進のための技術・情報教育課程と専門職能発達体系の改革, 上越教育大学研究紀要, vol.37, no.1, pp.217-227 (2017)
- 26) 安本太一・磯部征尊・梅田恭子・鎌田敏之・齋藤ひとみ・松永豊: 小学校・中学校・高等学校間の接続を考慮したプログラミングの授業の提案, 情報教育シンポジウム論文集, no.2021, pp.36-43 (2021)
- 27) 木村真人・山本利一・鈴木航平・軽部禎文: 技術に

- よる統合的な問題解決を支援する「情報の技術」の指導過程の提案, 埼玉大学紀要, vol.69, no.1, pp.381-390 (2020)
- 28) 森石峰一・横山宏・魚井宏高: 中学校の技術・家庭科の「プログラムによる計測・制御」で利用する学習教材の開発と試用, 教育システム情報学会誌, vol.29, no.4, pp.190-200 (2012)
- 29) 新宮英夫: スキルの認知心理学 行動のプログラムを考える, 川島書店 (1993)
- 30) 森山潤: 中学生のプログラミングにおける思考過程の構造解析, 日本産業技術教育学会誌, vol.38, no.4, pp.255-262 (1996)
- 31) 宮川洋一・森山潤・松浦正史: オブジェクト指向イベントドリブン型のプログラミングにおける問題解決過程の構造分析: プログラム作成能力との関連に焦点を当てて, 教育情報研究, vol.22, no.2, pp.3-11 (2006)
- 32) 岡崎善弘・大角茂之・倉住友恵・三島知剛・阿部和広: プログラミングの体験形式がプログラミング学習の動機づけに与える効果, 日本教育工学会論文誌, vol.41, no.2, pp.169-175 (2017)
- 33) 高野志歩・田村みゆ・富岡真由・秋信有花・倉光君郎: 擬似コードから考える自然言語を活かしたプログラミング言語, 情報教育シンポジウム論文集, no.2021, pp.147-151 (2021)
- 34) 岡本雅子: 模倣の重要性に着目した初学者向けプログラミング教育の研究, 京都大学, (2014)
- 35) 岡本雅子・村上正行・吉川直人・喜多一: 「視覚的顕在化」に着目したプログラミング学習教材の開発と評価, 日本教育工学会論文誌, vol.37, no.1, pp.35-45 (2013)
- 36) 岡本雅子・喜多一: プログラミングの「写経型学習」における初学者のつまづきの類型化とその考察, 滋賀大学教育学部附属教育実践総合センター紀要, vol.22, pp.49-53 (2014)
- 37) 岡本雅子: べた語義: 写経プログラミングをめぐる終わりそうもない論争, 情報処理, vol.59, no.1, pp.81-81 (2017)
- 38) 亀井亮汰・吉塚大浩・篠埜功・古宮誠一: 写経型学習の欠点を補う摂動を用いた理解度確認問題生成手法 — 二項演算子の事例に基づく有効性評価, コンピュータソフトウェア, vol.38, no.1, pp.1_111-1_139 (2021)
- 39) 森山潤・菊地章・山崎貞登・兵庫教育大学大学院連合学校教育学研究科共同研究プロジェクト (P) 研究グループ: イノベーション力育成を図る中学校技術科の授業デザイン: 子どもが小さなエンジニアになる教室, ジアース教育新社 (2016)
- 40) 西尾実・岩淵悦太郎・水谷静夫: 岩波国語辞典第4版, 岩波書店 (1986)
- 41) Thorndike L Edwor: Educational Psychology (Vols.1 and3), Columbia University Press (1913)
- 42) 米国学術研究推進会議: 授業を変える — 認知心理学のさらなる挑戦, 北大路書房 (2002)
- 43) 箱田裕司・都築誉史・川畑秀明・萩原滋: 認知心理学, 有斐閣 (2010)
- 44) Graham Wallas: The Art of Thought, Solis Press (2014)
- 45) Ohlsson S: Information processing explanations of insight and related phenomena, Advances in the Psychology of Thinking, vol.1, no.1, pp.1-44 (1992)
- 46) 鈴木宏昭・開一夫: 洞察問題解決への制約論的アプローチ, 心理学評論, vol.46, no.2, pp.211-232 (2003)
- 47) 寺井仁・三輪和久・古賀一男: 仮説空間とデータ空間の探索から見た洞察問題解決過程, 認知科学, vol.12, no.2, pp.74-88 (2005)
- 48) 野口尚孝・永井由佳里: デザイン発想支援システムは何を支援しうるのか?, 人工知能学会全国大会論文集, vol.18, 2E1-03, pp.1-3 (2004)
- 49) 野口尚孝: デザイン行為における試行錯誤過程の意味について, 日本デザイン学会研究発表大会概要集, vol. 56, pp.A10-A10 (2009)
- 50) 池田努・柚木泰彦: 探究的な学習におけるデザイン思考プロセスの活用, 日本デザイン学会研究発表大会概要集, vol.66, p.254 (2019)
- 51) 野口尚孝: 試行錯誤支援の考え方について, 第六回知識創造支援システムシンポジウム報告書, pp.75-83 (2009)
- 52) 矢川徳光: 新教育への批判 反コア・カリキュラム論, 刀江書院 (1950)
- 53) 藤原伸彦・阪東哲也・曾根直人・長野仁志・山田哲也・伊藤藤介: ティンカリングとしてのプログラミング, 鳴門教育大学情報教育ジャーナル, vol.16, pp.21-26 (2019)
- 54) Sylvia Libow Martinez・Gary Stager: Invent To Learn — Making, Tinkering, and Engineering in the classroom 2nd edition. Constructing Modern Knowledge Press (2019)
- 55) 山崎貞登・松田孝・二宮裕之・久保田善彦・磯部征尊・川原田康文・大森康正・上野朝大: Society5.0を支えるSTEAM/STREAM教育の推進に向けた小学校教育課程の教科等構成の在り方と学習指導形態, 上越教育大学研究紀要, vol.39, no.2, pp.525-538 (2020)
- 56) Wilkinson K・Petrich・金井哲夫: ティンカリングをはじめよう アート, サイエンス, テクノロジーの交差点で作って遊ぶ, オライリー・ジャパン (2015)
- 57) Education LEGO: “LEGO WeDo2.0”, <https://legoedu.jp/wedo2/>
- 58) 黒田昌克・森山潤: 技術リテラシー育成の観点から日常生活の問題を解決する学習活動を取り入れた小学校プログラミング教育の実践とその効果, 日本産業技術教育学会誌, vol.61, no.4, pp.305-313 (2019)
- 59) 萱津理佳・矢澤星奈: 初等中等教育段階におけるプログラミング教育の考察: プログラミング体験教室の実践から [研究ノート], 長野県短期大学紀要, vol.71, pp.13-22 (2016)
- 60) 山本利一・齋藤雅宏: プログラムによる計測・制

- 御を学習する指導過程の提案: 自動制御模型の製作とプログラムによる制御学習, 教育情報研究, vol.27, no.1, pp.25-32 (2011)
- 61) 萩嶺直孝・宮川洋一・森山潤: 中学校技術科「プログラムによる計測・制御」の学習における題材タイプの違いによる生徒の反応の差異, 日本産業技術教育学会誌, vol.55, no.3, pp.181-190 (2013)
- 62) 藤田眞一・加賀江孝信・三浦吉信: 小型コンピュータを用いたプログラムによる計測・制御学習の効果, 日本産業技術教育学会誌, vol.55, no.3, pp.191-198 (2013)
- 63) 古平真一郎・坂本弘志・針谷安男: 自律型ロボット教材を用いた「プログラムによる計測・制御」学習の授業実践に基づく学習効果の検証, 日本産業技術教育学会誌, vol.51, no.4, pp.285-292 (2009)
- 64) 針谷安男・飯塚真弘・山菅和良: プログラムによる計測・制御学習の授業実践とその学習効果の検証, 日本産業技術教育学会誌, vol.52, no.3, pp.205-214 (2010)
- 65) 増田麻人・大村基将・片田宗一郎・紅林秀治: 状態遷移図を利用したプログラムによる計測・制御教材の開発, 日本産業技術教育学会誌, vol.57, no.2, pp.93-101 (2015)
- 66) 青木浩幸・西ヶ谷浩史・紅林秀治: 創造的な計測・制御学習のためのプログラミング環境と授業モデル, 日本産業技術教育学会誌, vol.57, no.4, pp.223-230 (2015)
- 67) 北島茂樹・山中脩也・喜田綾芽: コンピュータとの対話を通じたプログラミング教育の授業デザインの提唱とその実践, 2018 PC Conference, pp.352-355 (2018)
- 68) 知見邦彦・樋山淳雄・宮寺庸造: 失敗知識を利用したプログラミング学習環境の構築, 電子情報通信学会論文誌, D-I, 情報・システム, I- 情報処理, D-I, vol.88, no.1, pp.66-75 (2005)
- 69) 左田和幸・松浦正史: 技術的な課題の問題解決過程における修正行動に関する研究, 日本産業技術教育学会誌, vol.37, no.3, pp.205-212 (1995)
- 70) 中尾尊洋・森山潤: 問題解決における試行錯誤に対する中学生の意識と技術的な活動経験との関連性, 日本産業技術教育学会誌, vol.63, no.3, pp.315-324 (2021)
- 71) 中尾尊洋・森山潤: 中学生の LED 制御プログラム作成過程における 試行錯誤の効果に関する探索的検討, 日本産業技術教育学会誌, vol.62, no.2, pp.133-140 (2020)

(全ての URL は 2022 年 7 月 4 日参照)