

プログラミング教育におけるプログラミング的思考教材の効果検証 EFFECT VERIFICATION OF PROGRAMMATIC THINKING MATERIALS IN PROGRAMMING EDUCATION

田崎 誠人[†]
Masato Tasaki

三木 良雄[‡]
Yoshio Miki

1. はじめに

近年, chatGPT に代表される生成 AI など, 人工知能が飛躍的な進化を遂げている. それらは我々の生活を豊かにする一方で, 人類の脅威であるともいえる. これらからの社会を生きるためには, 情報通信技術を活用し, さらに自らの力で考える思考力がますます重要になる.

2020 年度から初等教育において小学校プログラミング教育が開始された. これはプログラミング的思考を育成することを中核とし, 上述した情報通信技術を活用する能力を育成することも同時に目指している. [1]

プログラミング的思考とは, 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, 一つ一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[1]とされている. プログラミング的思考を習得することで, 社会に出てから, 型どおりの作業をこなすのではなく, 定型化・言語化されていない曖昧な問題を自ら頭を使って整理し, 解決策を見出すことができるようになる.

本研究では, 情報処理学会 第 86 回全国大会で提案したプログラミング的思考に関する新規教材[2]を発展させ, その効果を分析し, 教材の有効性について定量的に検証するものである.

2. 提案した教材とその検証

前述したように, プログラミング的思考の教材を以前提案した. 本章ではその教材の解説を行う.

プログラミング的思考の既存教材として, 文部科学省が学習指導要領に示しているもののほか, アルゴリズムなどについて学ぶものがある.

本章で解説するプログラミング的思考の教材は, 純粋な数理最適化以外にも, 実社会で経営者などが抱える問題を包含しているとされるオペレーションズリサーチ(OR)を題材とし, 通常あらかじめ設定されている問題の定義を明確にし, 変化させることで, 物事の解決までのアプローチに対し複数の選択や新たな選択肢を持つことができる思考を育成するものである.

2.1 ハンドベルモデル

上述したような課題を解決するため, ハンドベルモデルを提案した. 以下の図 1 にハンドベルモデルについての図解を示す. 通常の間では, 問に対して解は 1 つに絞られており, 一本の道筋を通り単一の解にたどり着く. それに対し, 提案するハンドベルモデルでは, 解決までを以下の 3

[†]工学院大学大学院 工学研究科 情報学専攻 Informatics Program, Kogakuin Graduate School of Engineering

[‡]工学院大学 情報学部 情報科学科 Kogakuin University, Department of Information Science, Schools of Informatics

つのフェーズに分ける.

①最初に解決する対象(問題・目的)を設定・提供する. 従来の教育手法と同様に通常の問題を与える. これ以降の問を潤滑に導入するために必要なフェーズである.

②制約や目標といった前提条件を明確にする. 第一段階で提供した問の前提条件と共に, 目的を達成するための別の前提条件を提供する.

③自ら前提条件を考え, 思考を発散させる. 第二段階では提供”されて”いた前提条件を受講者自らが考え, それに対し自らが解を出すサイクルを繰り返す.

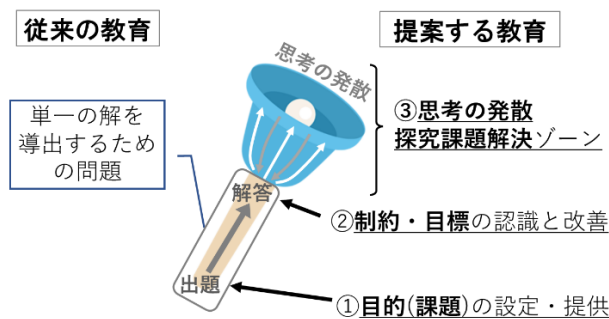


図 1 ハンドベルモデル

ここで目標とは最適化する数値・状態や終了の条件. 制約とは用いてよい手段や数, 動作における制約のことを指す. また, 解決する問題にはあらかじめ設定されている目的があり, その目的を達成するために動かしてはならない制約がある. [2]

2.2 有効性の検証

2.1 のハンドベルモデルの効果を確認するために初等・中等教育課程の児童と生徒, 計 30 名を対象に効果検証を行った. 効果検証は, 処置の前後を比較する前後比較試験で実施した. ハンドベルモデルの教材として, OR の代表的な例題である巡回セールスマン問題(TSP)を改変したものを使用した. 前後比較試験に用いる事前・事後テストには, ナップサック問題の前提条件を変更可能とした問題を使用した. 回答欄は自由記述とし, 思考を十分にアウトプットできるように設計した.

前後比較試験に使用した教材は 3 章以降の分析のために, 以下の 3 ステップ, 合計 9 観点により評価を行った. ただし, ステップ 1 及び 2 はナップサック問題を純粋に解けているか. ステップ 3 は提案したハンドベルモデルによりプログラミング的思考が育成されているかを評価している.

ステップ 1: 問題を理解できているか

1.1. 問題の理解

出題した問の意図を理解して解けているか

ステップ 2: 問題が解けているか

2.1. 解の正解性

- 必要度がどれだけ正解に近づいているか
- 2.2. 条件への到達度
重量という条件に対し、問題を解けているか
- 2.3. 最適化の思考
複数の選択肢を考え、最適な組合せを選んでいるか
- ステップ 3: プログラミング的思考に到達しているか**
- 3.1. 解への実現性
能力に応じた実現可能な案が提示できているか
- 3.2. 選択の合理性
目的に沿った必要・不必要な選択ができるか
- 3.3. 前提条件の明確化
天気やレジャー等、目的や状況に応じた前提条件を設定できているか
- 3.4. 選択の適切性
分割による機能喪失や関連アイテムとの組合せ等、社会通念上相当な選択ができるか
- 3.5. 制約の自由性
重さの制限や手段、荷物の持ち方を設定することで、制約を適切に変更できているか
- 以上の 3 ステップ 9 項目の評価規準において、項目ごとに 3 レベル(0 点〜2 点)で評価を行った。

3. 分析

2.2 で得られたデータをもとに教材の効果について分析を実施した。本章ではまずはデータの特徴を述べ、その後 t 検定を実施し、有効性を確認する。

3.1 効果推定とデータ補正

表 1 は 2.2 の評価規準で評価した事前テストと事後テストの結果である。また図 2 は事前テストと事後テストの合計得点を単純に比較したものである。このグラフからハンドベルモデルによるプログラミング的思考の育成を実施した後は明らかに点数が増加している。理想的には対象実験としてハンドベルモデルによるプログラミング的思考の育成を実施しない対象群に対して、事前テストと事後テストと実施することが考えられるが、対照群の被験者に処置群と同様のナップサック問題だけを解かせることは現実的でない上に、事前事後という意味では有意な差が無いのが当然である。そこで本研究では次のように、見かけの効果となりうる成分に着目し、より正確な効果推定を行う。

評価規準のステップ 2 はナップサック問題という制限のもと問題を解くのにに対し、ステップ 3 は前提を自身で設定し問題を解くため、解答の自由度により、点数が偏る。すなわち、ステップ 2 と 3 はトレードオフの関係になっている。

またステップ 1 では、事前テストと事後テストの間で点数の増加がみられる。ハンドベルモデルの教材として、ナ

表 1 評価規準ごとの事前・事後テストの平均得点

ステップ	項目	事前テスト	事後テスト
1.問題自体が分かっているか	1.1 問いの理解	1.87	2.00
	2.1 解の正確性	1.30	1.07
2.問題が解けているか	2.2 条件への到達度	1.70	1.43
	2.3 最適性の思考	0.40	0.10
	3.1 解への実現性	0.13	1.00
3.プログラミング的思考	3.2 選択の合理性	0.23	0.70
	3.3 前提条件の明確化	0.00	0.47
	3.4 選択の適切性	0.00	1.53
	3.5 制限の自由性	0.00	0.83

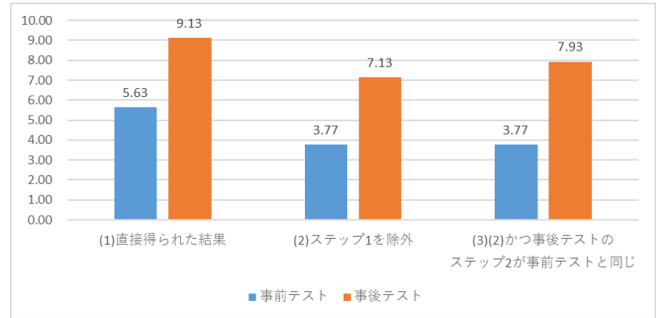


図 2 補正方法ごとの事前・事後テストの得点比較

ナップサック問題を扱っていないにも関わらず、事後テストの点数に増加があった原因として、事後テストではナップサック問題を解くのが 2 回目であったため、問題を解くことに慣れていたことが考えられる。

以上より、本手法の効果推定においては、(1)直接得られた結果データ、(2)評価規準のステップ 1 を除外したデータ、(3)(2)に対して、評価基準のステップ 2 は事前テストと同じ結果となとしたデータの補正方法が異なる 3 種に対して評価を行った。

3.2 T 検定

図 2 の補正方法(1),(2),(3)に対し、Welch の t 検定を実施した。以下の表 2 に補正方法ごとの検定結果を示す。ただし、危険率は $\alpha=0.05$ である。

表 2 補正方法ごとの t 検定

	(1)直接得られた結果		(2)ステップ1を除外		(3)(2)かつ事後テストのステップ2が事前テストと同じ	
	事前テスト	事後テスト	事前テスト	事後テスト	事前テスト	事後テスト
平均	0.6259259	1.0148148	0.4708333	0.8916667	0.470833333	0.991666667
分散	0.5963272	0.3358642	0.4341071	0.2278571	0.434107143	0.232301587
観測数	9	9	8	8	8	8
仮説平均との差異	0	0	0	0	0	0
自由度	15	15	13	13	13	13
t	-1.208354	-1.208354	-1.462978	-1.462978	-1.804568723	-1.804568723
P(T<=t) 片側	0.1228082	0.1228082	0.0836139	0.0836139	0.047174214	0.047174214
t 境界値 片側	1.7530504	1.7530504	1.7709334	1.7709334	1.770933396	1.770933396
P(T<=t) 両側	0.2456165	0.2456165	0.1672278	0.1672278	0.094348428	0.094348428
t 境界値 両側	2.1314495	2.1314495	2.1603687	2.1603687	2.160368656	2.160368656

3.1 で述べたように図 2 及び表 2 の(2),(3)は見かけの効果を補正している。表 2 において、t 検定の p 値は(2),(3)と減少しており、見かけの効果を補正すると、有意になる、すなわち、提案したハンドベルモデルには効果があるという結果になった。

4. おわりに

本研究では、情報処理学会 第 86 回全国大会で提案したプログラミング的思考の教材を定量的に検証し、教材がプログラミング的思考の育成に有意に働いていることを確認した。

しかしまだ、プログラミング的思考と環境や学力との関係は分かっていない。さらに長期的に見た思考力の育成も重要である。様々な観点でプログラミング的思考が思考力に与える影響も観測していきたい。

参考文献

- [1] 文部科学省, "小学校プログラミング教育の手引き(第三版)".
- [2] 田崎誠人, 三木良雄, "プログラミング教育におけるプログラミング的思考の教育方法と教材の提案", 情報処理学会第 86 回全国大会講演論文集, Vol.2024, No.1 (2024).