

テスト・問題集編成における数理最適化技術の応用

05001379	Classi 株式会社	*廣田正之	HIROTA Masayuki
05001388	Classi 株式会社	石井康貴	ISHII Yasutaka
01012660	東京理科大学	池辺淑子	IKEBE Yoshiko
02203940	東京理科大学	鮎川矩義	SUKEGAWA Noriyoshi

1. はじめに

Classi 株式会社は、日本の高等学校向けに教育プラットフォームサービス Classi を提供しており、機能の1つに Web テストがある。この機能では、教員が問題を組合せてテストを編成し、生徒たちに配信できる。テスト以外の場面にも、宿題や演習など問題集として使われることもある。

教員は単元、難易度、解答時間、問題数などの観点で、約 10 万問 (2021/04 時点) の中から所望の問題を組合せてテスト・問題集を編成する。しかし、問題数の多さ故に生成可能なテスト・問題集のパターンは膨大となるため、所望のテスト・問題集を編成するには多大な時間がかかる。また、近年では教員の勤務時間は重大課題とされており、教材の準備は効率的に行えることが望ましい。Classi では、事前に問題を組合せたテストも用意しているが、Classi の学習コンテンツ開発者も教員と同様に問題の選定に時間を要している課題がある。

上記のような課題を解消する方法として、数理最適化技術を応用したテスト・問題集の自動編成というアプローチが考えられる。テストの自動編成については先行研究が複数存在する。例えば、等質なテストを最大数構成する研究 [1] や、生徒の能力評価を目的としたテストを編成する研究 [2] がある。しかし、これらの研究の目的意識と Classi における活用用途には異なる部分があり、結果を直に適用できない。

そこで今回は、2 パターンのテスト・問題集の編成を検討した。1 つ目は、Classi 学習コンテンツ開発担当者の支援を目的とした、バリエーションのあるテストを編成すること、2 つ目は、さらなる発展として、近年教育現場で求められている個別最適な学びの実現を目的とした、生徒一人一人の学習到達度に応じた個別最適な問題集を編成することである。本発表では、検討結果と数値実験結果について報告する。

2. バリエーションのあるテスト編成

Classi では、学校の教員にとって使いやすいテストの企画開発を行っている。例えば、朝学習のように短時間で取り組むケースを想定し、単一の単元で基礎的な問題に絞ったテストを提供している。他にも模擬試験対策を想定し、複数の単元が絡む応用問題を集めたテストも提供している。

実際にテストを作成する際のフローは概ね次の通りである: (i) 企画意図に沿って、出題単元・問題数・難易度・解答時間などの要素を決定 (ii) Classi で提供中の問題の中から問題を選び出し、上記制約を満たす組合せパターンを比較検討しテストを構成 (iii) 現状利用可能な問題で所望のテストが構成できない場合は、新たに問題を作成。

今回は (ii) に焦点をあてる。実際のテスト作成の現場では (i) の制約を満たす組合せパターンは膨大であり、人力で比較検討するのは困難である。そこでテスト案の編成を最適化問題として定式化することを考える。(i) の制約をベースとし、追加の制約や目的関数を何通りか定義し、様々なバリエーションのテスト案を構成することで、担当者の問題組合せ検討の労力を軽減することを目指す。

バリエーションを出すための追加制約としては、問題形式や出典問題集、過去の Classi 提供テストでの利用有無などが考えられる。目的関数としては、教員自作テストでの問題採用回数の合計最大化や、単元より細かい学習項目のカバレッジ最大化などが考えられる。

ここでは例として、“(i) の制約+教員自作テストでの採用回数の合計が最大”となるテストの構成を、整数計画問題として定式化する。まず変数 x_i を、問題 i を採用するなら 1、非採用なら 0 となる 2 値変数とする ($i \in I$)。 S_i を問題 i の教員自作テスト採用回数とすると、目的関数は $\sum_{i \in I} S_i x_i$ となる。単元、難易度ごとの出題数の制約は $\sum_{i \in I_{ud}} x_i = N_{ud}$ となる。ただし、 I_{ud} は単元 u 、難易度 d の問題の

インデックス集合, N_{ud} はテストに含まれる単元 u , 難易度 d の問題数である ($u \in U, d \in D$). 解答時間の制約は $|\sum_{i \in I} T_i x_i - T| \leq \Delta$ となる. ただし, T_i は問題 i の解答目安時間, T はテスト解答目安時間, Δ はテスト解答目安時間で許容するずれである.

以上のことを整理すると, 解くべき整数計画問題は次のようになる:

$$\begin{aligned} & \text{Maximize}_{(x_i)_{i \in I}} \sum_{i \in I} S_i x_i \\ & \text{s.t.} \quad \sum_{i \in I_{ud}} x_i = N_{ud} (\forall u \in U, d \in D) \\ & \quad \left| \sum_{i \in I} T_i x_i - T \right| \leq \Delta \quad (1) \\ & \quad x_i \in \{0, 1\} (\forall i \in I). \quad (2) \end{aligned}$$

当日は, 上記で紹介したものも含め検討した定式化と得られた結果, 及び学習コンテンツ開発担当者から得られたフィードバック内容を報告する.

3. 生徒個別最適な問題集編成

近年, 生徒一人一人の特性や学習進度等に応じた, 個別最適な学びが求められている [3]. 一方, 実際の学校現場では, 教員は一人で数十人もしくは百人を超える生徒を担当しており, 生徒一人一人にあった問題の選択, およびその問題集を使った個別指導が困難なため, クラスで同じ問題集を使用するケースが多くなっている. この課題を解決することを目的として, 生徒個別最適な問題集を編成することを検討する. 今回は, Classi に蓄積された解答データを基に, ある正答率以上の問題は正解できている生徒に, より難度の高い正答率の問題を解けるようになってもらえるよう, 問題集を編成することを考える.

問題集を編成するにあたり, 生徒ごとの学習到達度に応じた問題候補の集合を生成する. まず, 問題 i の全解答者の正答率を α_i とする. 次に, 対象の生徒 s の解答データから, その生徒が正解できている問題の正答率 α_{Cs} と目標とする正答率 α_{Ts} を計算する. 最後に, 正答率が $[\alpha_{Ts}, \alpha_{Cs}]$ の範囲の問題のインデックス集合を I_s とする.

問題集を編成する問題の選定方法は, α_{Cs} 付近の難度の低い問題を中心に問題集を編成する, α_{Ts} 付近の難度の高い問題を中心に問題集を編成する

等, 様々であるが, 今回は, 生徒が段階を踏んで学習できるよう, α_i が一様なばらつきをもつ問題集を編成する. 例えば, $\alpha_T=0.5, \alpha_C=0.7$ で問題集が 5 問の場合, 各問題の正答率が 0.5, 0.55, 0.6, 0.65, 0.7 となるように問題集を編成する.

$[\alpha_{Ts}, \alpha_{Cs}]$ の範囲で α_i が一様なばらつきをもつ問題集を編成するために, 解くべき混合整数計画問題は, 十分大きな値 M を用いて, 次のようになる:

$$\begin{aligned} & \text{Maximize}_{\delta, (x_i)_{i \in I_s}} \delta \\ & \text{s.t.} \quad (1), (2), \\ & \quad M(2 - x_i - x_j) + |\alpha_i - \alpha_j| \geq \delta \\ & \quad (\forall i, j \in I_s | i < j). \quad (3) \end{aligned}$$

以下の実験環境, および汎用ソルバー CBC を用いて, 上記の定式化の数値実験を行った.

- OS : macOS Catalina 10.15.7
- CPU : Intel Core i7 / メモリ : 16 GB

結果として, 編成された問題集は, 想定通り α_i が一様なばらつきをもっていた. しかし, 候補問題数 $|I_s|$ に応じて, 計算時間に影響が見られた. $|I_s|$ が 54 問のとき 1 分, 75 問のとき 13.5 分, 100 問を超えると数時間経っても結果が返ってこなかった. これは制約式 (3) を i, j について展開すると, $O(|I_s|^2)$ 本の式ができてしまい, 制約式の本数が多いことが原因であると考えている. 現在, α_i と α_j の差がある一定値以下の場合にのみ有効となるよう, 改善を試みている. 当日は, 詳細な実験結果と改善状況について報告する.

参考文献

- [1] 石井隆稔, ソンムァンポクポン, 植野 真臣: 最大クリーク問題を用いた複数等質テスト自動構成手法. 電子情報通信学会論文誌 D Vol.J97-D No.2 pp.270-280
- [2] O. Geiger and S. Markovitch: Algorithmic Exam Generation. *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, pp.1149-1155.
- [3] 文部科学省. 「個別最適な学び」と「協働的な学び」の一体的な充実. https://www.mext.go.jp/a_menu/shotou/new-cs/senseiyouen/mext_01317.html