

項目露出を考慮した整数計画法による等質テスト構成

2021年1月28日

情報数理工学プログラム

学籍番号 1710086

植野 晶

指導教員 植野 真臣

令和2年度 情報数理工学プログラム卒業論文概要

平成 29 年度 入学	学籍番号 1710086
指導教員 植野 真臣	氏名 植野 晶
題目 項目露出を考慮した整数計画法による等質テスト構成	

概要

等質テスト構成のための整数計画法において項目露出を抑える手法を提案する。等質テストとは、各テストに含まれるテスト項目は異なるが、受検者の推定得点の予測誤差が等質なテスト群である。項目がテスト構成全体で出題される回数のことを露出数と呼ぶが、H. Wainer(2000)によると露出数が大きい項目は受験者間で共有されやすく、その項目の信頼性低下につながる事が知られている。石井ら(2015)は従来の手法により生成したテスト構成のうち最も露出率(=露出数/テスト構成数)の小さなテストを選択することで、従来の手法よりも項目露出を軽減することができた。しかし石井ら(2015)の手法ではテスト構成時には項目露出を考慮していないため、露出率に改善の余地がある。

提案手法ではテスト構成時に露出数が最大の項目をアイテムバンクから除外しテスト構成することで、従来手法よりも露出率の低い等質テスト群を構成することができた。

本論文では提案手法の有効性をシミュレーション及び実データを用いて示した。具体的には従来手法で露出率 0.164 ほどのテスト構成を提案手法では露出率 0.028 まで下げることができた。

目次

1	まえがき	2
2	項目反応理論	4
3	等質テストの自動構成アルゴリズム	5
3.1	等質テストのための最大クリーク問題	5
3.2	乱択法	7
3.3	整数計画法を用いた等質テスト構成	8
3.4	露出率を軽減するテスト構成手法	11
4	従来手法の問題点	13
5	提案手法	14
5.1	第1段階目のテスト構成手法	14
5.2	第2段階目のテスト構成手法	18
6	比較実験	20
6.1	第1段階目の比較実験	20
6.2	第2段階目の比較実験	21
7	むすび	26

1 まえがき

e テスティングとは，異なる問題で構成されるが，同一精度の測定を実現出来るコンピュータテストのことである．e テスティングを用いることで，同一能力の受験者が異なるテストを受験しても同一得点となる保証がある．そのために，受験者が同一精度で複数回の受験が可能となる．他にも様々な利点を持つことが知られている [1]．e テスティングでは”等質テスト”を用いることが推奨されている．”等質テスト”とはテストに含まれる項目は異なるが，出題項目数や得点の予測誤差がテスト間で等しいテスト群のことである．等質テストにより異なる項目からなるテストを同一能力の受験者が受験した場合同一得点になることが保証される．e テスティングの普及により，等質テストを自動構成する手法が数多く提案されている [2, 3, 4, 5]．

一般に e テスティングでは，アイテムバンクと呼ばれる出題項目を管理するデータベースが利用される．アイテムバンクには出題項目の統計データが格納されており，所望のテストの性質を満たす組み合わせを計算機で自動構成する．図 1 はテストの自動構成の概念図である．

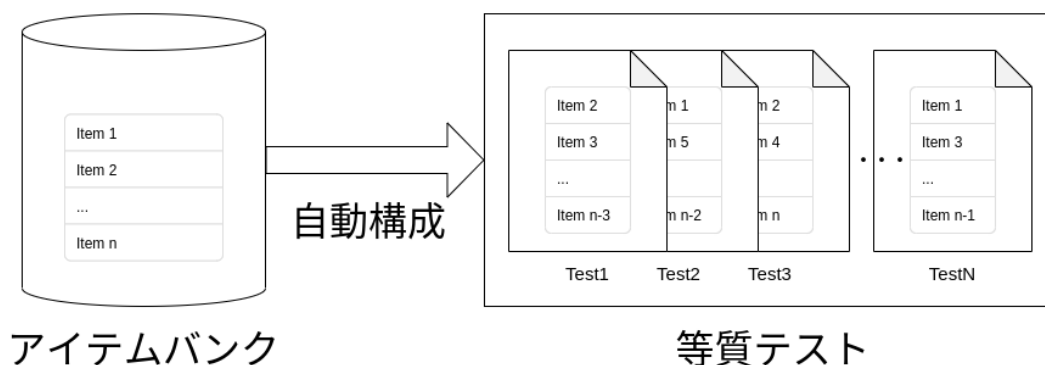


図 1 テストの自動構成の概念図

テストの自動構成は数理最適化問題として解かれる．例えば，Songmuang and Ueno (2010) は最適化問題の解探索手法の一つである Bees Algorithm を用いてテスト構成を提案した．この手法は情報処理技術者試験をはじめとして，我が国の国家試験で実際に使用されている [4]．

Ishii, Songmuang, and Ueno (2014) はグラフ上で定義される最大クリーク問題に帰着してテスト構成を行う手法を提案した．具体的には与えられたアイテムバンク・テスト構成条件を満たすテストを頂点，二つのテスト（頂点）が等質条件を満たす場合に辺を引いたグラフ構造から最大クリーク探索することで等質テストを構成する [6]．この手法は理論的に最大のテスト構成を保証するが，アイテムバンクの項目数に対して構成可能な頂点（テスト）数が組み合わせ爆発的に増加するため，最大クリーク探索が困難である．

Ishii, Songmuang, and Ueno (2014) はグラフからランダムに部分グラフを選択し、最大クリーク探索を繰り返すことによりグラフ全体の最大クリークを近似的に探索する等質テスト構成手法 (RndMCP 法) を提案した。本手法により、当時の既存研究よりも 10~100 倍以上多くのテストを構成できた [7]。最大クリーク探索はグラフの頂点集合を V とすると、最先端のクリーク探索手法 [8, 9] を用いても $O(|V|^2)$ の空間計算量を必要とするため、最大で 10 万程度のテスト構成が限界であるという問題があった。

RndMCP 法の空間計算量を緩和するために、Ishii and Ueno (2017) は第 1 段階で RndMCP 法を用いてメモリ限界まで大きな最大クリークを探索した後、第 2 段階目で第 1 段階目で求めたクリークの全頂点と隣接する頂点を整数計画法により逐次的に探索することで必要な計算量を $O(|V|)$ に削減させる手法を提案した [10, 11]。これにより 10 万を超える等質テストを構成することが可能になった。

しかし、先行研究 [7, 10, 11] では、ある項目がテスト構成全体で出題される回数 (以降、露出数と呼ぶ) に偏りが生じる問題がある。例えば露出数が大きい項目は受験者間で共有されやすく、その項目の信頼性低下につながる [12]。この偏りを軽減するために、Ishii and Ueno (2015) では RndMCP 法と整数計画法を用いてテスト構成し、その中から最も露出率 (= 露出数の最大値 / テスト構成数) が小さいテスト構成を選択する手法を提案した [13]。具体的には、探索した全てのクリークを等質テストの候補として保存しておき、最後にその候補で最も露出率が小さい等質テストを出力する。これによって、従来手法よりも露出率を軽減することができた。

Ishii and Ueno (2015) は RndMCP 法及び整数計画法でアイテムバンクの全項目から頂点を生成しているため、露出率の高い項目を含む頂点を生成する問題があった。本研究では RndMCP 法の部分グラフの頂点選択の際に、整数計画法を用いてグラフの頂点に含まれる回数が最大の項目以外から項目を選択し頂点を生成することで、クリーク探索後のテスト構成の露出率を RndMCP 法よりも抑える手法を提案する。

また、上記の手法で露出率を抑えることで、RndMCP 法よりも密な部分グラフを得ることができ、その結果より大きなクリークを見つけることができたので、生成した頂点をクリークに含まれる頂点と含まれない頂点に分けて、分析を行った。

さらに、第 2 段階目の整数計画法でも最大露出数の項目以外から項目を選択し、この整数計画法を逐次的に解くことで、項目露出を抑えつつテストを構成する手法を提案する。

これらの手法により従来手法 Ishii and Ueno (2015) と比較して露出率とテスト構成数を改善することができた。

実験ではシミュレーションデータと実データを用い、提案手法が従来手法と比較して露出率とテスト構成数を改善したことを示し、その要因について分析した。

2 項目反応理論

一般的に等質テストは，以下の構成条件を満たすテストの集合として定義する．（例えば [5, 6, 7]）

1. それぞれのテストでの受験者得点の予測誤差が等質である．
2. それぞれのテスト間の項目重複数が一定値以下である．（以降，項目重複数条件と呼ぶ）

受験者得点の予測誤差はテストの自動構成に関する研究（例えば [2, 4, 14, 15, 16]）において項目反応理論（Item Response Theory: IRT）[17, 18]におけるテスト情報量で評価されている．IRT とは受験者の項目への正答確率をモデル化したものである．これにより，異なる項目から構築されるテストを受けた受験者能力の同一尺度上での比較が可能となる．

IRT の中で最もよく使われる 2-パラメータロジスティックモデル（2-Parameter Logistic Model: 2PLM）では能力者 j が項目 i に正答する確率を以下のようにモデル化する．

$$p_i(\theta_j) = \frac{1}{1 + \exp(-1.7a_i(\theta_j - b_i))} \quad (1)$$

ここで $\theta_j \in (-\infty, \infty)$ は受験者 j の能力パラメータ， $a_i \in [0, \infty]$, $b_i \in [0, \infty]$ はそれぞれ i 番目の項目の識別パラメータ，困難度パラメータと呼ばれる項目パラメータである．

IRT では項目 i において式 (1) を用いて計算したフィッシャー情報量を項目情報量 $I_i(\theta)$ と呼び，以下のように定義する．

$$I_i(\theta) = 1.7^2 a_i^2 p_i(\theta)(1 - p_i(\theta)) \quad (2)$$

また，テストに含まれる項目の項目情報量の総和をテスト情報量と呼び，以下のように表す．

$$I(\theta) = \sum_{i \in T} I_i(\theta) \quad (3)$$

ここで， T はテストに含まれる項目の集合である．このテスト情報量の逆数が受験者能力推定値の漸近分散に収束することが知られている [1]．

テストの自動構成手法（例えば，[2, 4, 14, 15, 16, 19]）ではテスト情報量における受験者の能力パラメータ θ_i を $\Theta = \{\theta_1, \dots, \theta_k, \dots, \theta_K\}$ のように幾つかの点でサンプリングし，離散的に扱っている．

表 1 テスト情報量への上下限制約

$\theta = -2.0$	$\theta = -1.0$	$\theta = 0.0$	$\theta = 1.0$	$\theta = 2.0$
0.0/0.2	0.1/0.3	0.1/0.3	0.1/0.3	0.0/0.2

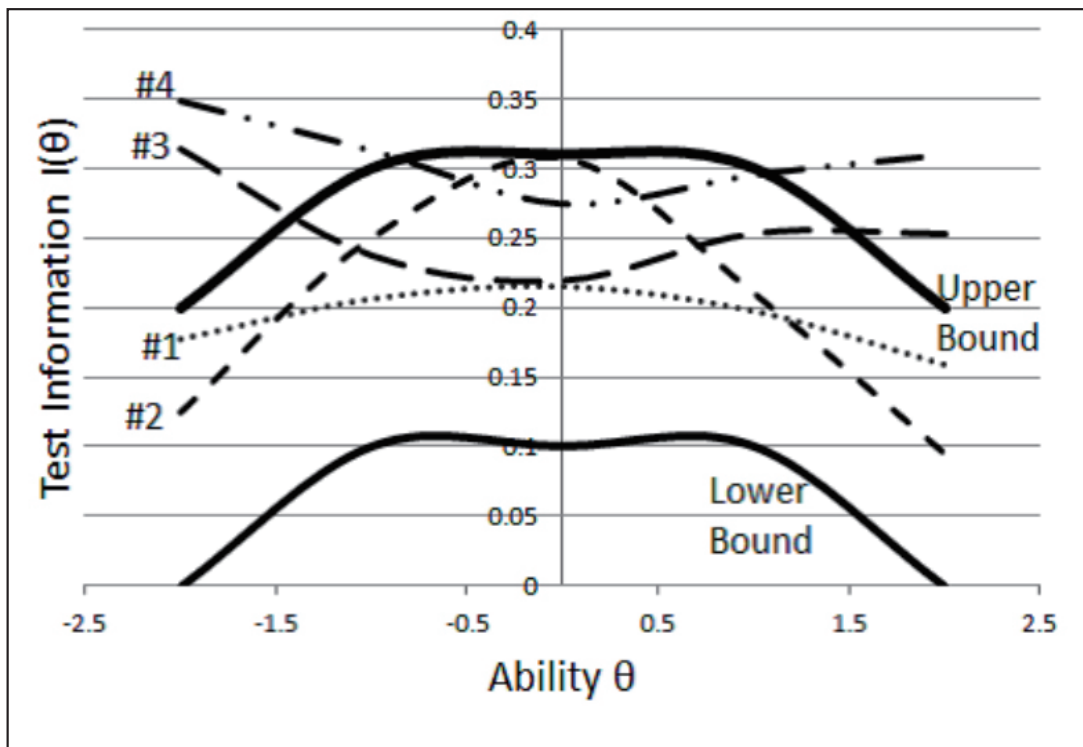


図 2 テスト情報量への上下限制約

3 等質テストの自動構成アルゴリズム

本節では提案手法と関連のある手法を紹介する。

3.1 等質テストのための最大クリーク問題

Ishii et al. (2014) は等質テスト構成をグラフ上で定義される最大クリーク問題に帰着させる手法 (RndMCP 法) を提案した [6]。ここでクリークはグラフの任意の 2 頂点が隣接する頂点集合である。

本手法では能力パラメータ $\Theta = \{\theta_1, \dots, \theta_k, \dots, \theta_K\}$ をサンプリングし, 各点ごとにテスト情報量 $I(\theta_k)$ の上下限制約 $(UB(\theta_k), LB(\theta_k))$ を計算し, すべての上下限制約を満たすテストは受験者得点の予測誤差が等質であるとする。

例えば図 2 は表 1 のテスト情報量への上下限制約を与えたときの概念図である。

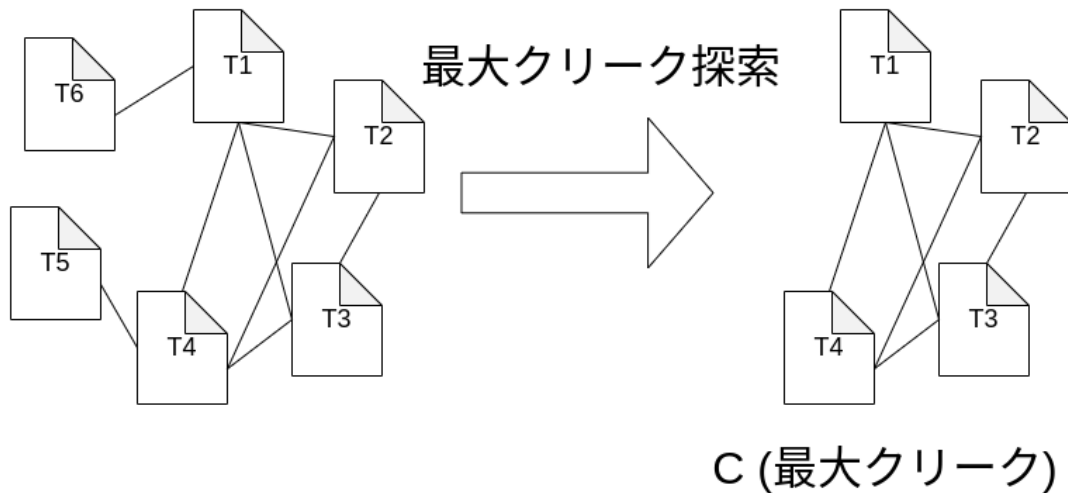


図3 等質テスト構成のためのクリーク探索の概念図

図2のテスト#1や#2はテスト情報量の上下制限約を満たすが、テスト#3や#4は上下制限約を満たさない。

生成されるテスト候補を以下のグラフ構造とみなし、グラフ構造の中からクリークを探索する。

1. 頂点: 項目重複数条件を除いたテスト構成条件を満たす、すべてのテストを頂点とする。
2. 辺: 二つの頂点に対応するテストが項目重複数条件を満たす場合、二つの頂点間に辺を引く。

このグラフが持つクリーク中の任意の二頂点は隣接しているため、テスト構成条件を満たす。したがってクリーク中の頂点に対応するテストはそれぞれ等質であり、頂点数が最大のクリークが最大の等質テストとなる(図3)。

等質テスト構成は頂点の集合を V 、辺の集合を E 、無向グラフを $G = (V, E)$ 、クリークを C として次のように定式化できる。

複数等質テスト構成のための最大クリーク問題

variables $C \subseteq V$

maximize $|C|$

subject to

$\forall v, \forall w \in C, \{v, w\} \in E$

*ここで $\{v, w\} \in E$ は頂点の組 v, w が次の条件で引かれた辺を意味する
 $(|v \cap w| \leq \text{重複項目数の上限値})$

Algorithm 1 乱択法

Require: アイテムバンク, テスト構成条件

Ensure: 等質テスト群

```
1: procedure RndMCP( $L_1, L_2, CT$ )
2:    $C := \emptyset, C_{max} := \emptyset$ 
3:    $ST := current\ time$ 
4:   while ( $current\ time - ST$ ) <  $CT$  do
5:     /* Step1 */
6:      $V := L_1$ 個のテストをランダム生成      ▷ 項目重複数条件以外を満たす  $L_1$ 個のテスト
7:     /* Step2 */
8:      $G = (V, E)$  グラフ構築                ▷ 二頂点が重複項目数条件を満たす場合, 辺を引く
9:     /* Step3 */
10:     $C := MCP(G, L_2)$                        ▷  $G$ の最大クリークを時間  $L_2$ だけ探索
11:    if  $|C_{max}| < |C|$  then
12:       $C_{max} := C$ 
13:    end if
14:  end while
15:  return  $C_{max}$ 
16: end procedure
```

この手法により最大の等質テストを構築することが理論的に可能であるが, このアルゴリズムの実行に時間計算量 $O(2^{|V|})$, 空間計算量 $O(|V|^2)$ が必要であり, 頂点数 $|V|$ はテスト構成条件を満たすテストの総数となるが, その数はアイテムバンクのテスト項目数 n と比較して組み合わせ爆発的に増加する. ゆえに, 現在実施されているような数百~千以上のアイテムバンクから等質テストの構成を厳密に行うことは困難である.

3.2 乱択法

これらの計算コストの問題を緩和するため, Ishii, Songmuang, and Ueno (2014) はグラフ全体の最大クリークを近似的に探索する手法を提案した [7] (以降 RndMCP 法と呼ぶ). 3.1 で紹介した手法 [6] の問題点は等質テスト構成数が増加すると, グラフの探索空間が莫大となることである. そのため, RndMCP 法ではテスト構成候補グラフから部分グラフをランダムに抽出し, ここから最大クリーク探索を繰り返すことにより, グラフ全体の最大クリークを全体の最大クリークを近似的に探索する. 具体的には Algorithm 1 によりテスト構成を行う.

Step1~2 ではテスト構成条件の項目重複数条件以外を満たす L_1 個の頂点を持つテスト候補グラフをランダムに抽出する. L_1 はチューニングパラメータであり, メモリ上に保持できる頂点数の上限を計算機環境に合わせて設定する. Step3 では抽出した部分グラフの最大クリーク探索を計算時間 L_2 だけ行う. Step1~3 を計算時間 CT を過ぎるまで繰り返し, Step2 で得られた部分グラフの最大クリークのうち最大のものを出力する.

本手法は最大クリーク探索の時間・空間計算量をそれぞれ $O(L_2), O(L_1^2)$ に緩和する.

これらのパラメータは計算機環境に合わせて任意に設定できるため，3.1 で紹介した手法 [6] の時間・空間計算量 $O(2^{|V|})$, $O(|V|^2)$ に対して格段に扱いやすい．

この手法により一般的な規模（500～2000 項目程度）のアイテムバンクから最大で 10 万程度のテストを生成できた．

3.3 整数計画法を用いた等質テスト構成

RndMCP 法ではグラフの頂点数を $|V|$ としたとき，最先端の最大クリーク探索アルゴリズム [8, 9] を用いたとしても空間計算量を $O(|V|^2)$ 必要とするため 10 万個程度のテスト構成が上限であった．そこで，Ishii et al. (2017) は RndMCP 法と整数計画法を組み合わせた二段階探索手法（HybridPBR 法）を提案した [10, 11] ．

HybridPBR 法では現在探索中のクリーク C の全頂点と隣接する頂点を以下の整数計画法を用いて，逐次的に探索する．本手法では現在探索中のクリークに隣接する頂点のみを保存するため，グラフの頂点数を $|V|$ とすると，計算に必要な空間計算量は $O(|V|)$ に軽減される．ただしこの探索は $O(|V| \cdot 2^n)$ の時間計算量を必要とするため，RndMCP 法の最大クリーク探索の時間計算量 $O(L_2)$ に大幅に劣る．そこで RndMCP 法により計算機のメモリの限界の頂点数 L_1 を持つグラフで最大クリーク探索を行ってから，整数計画法を用いる手法に切り替えることで探索効率を改善する [10, 11] ．以下の整数計画法でクリークに隣接する頂点を探索する．

最大クリーク探索のための整数計画法

variables

$$x_i = \begin{cases} 1 & i \text{ 番目の項目がテストに含まれる} \\ 0 & \text{それ以外} \end{cases}$$

maximize

$$\sum_{i=1}^n \lambda_i x_i \quad (4)$$

subject to

$$\sum_{i=1}^n x_i = M(\text{テスト項目数}) \quad (5)$$

$$LB_{\theta_k} \leq \sum_{i=1}^n I_i(\theta_k) x_i \leq UB_{\theta_k} \quad (6)$$

$(k = 1, \dots, K)$

$$\sum_{i=1}^n X_{i,r} x_i \leq OC(\text{項目重複上限数}) \quad (7)$$

$(r = 1, \dots, |C|)$

$$X_{i,r} = \begin{cases} 1 & i \text{ 番目の項目が} \\ & C \text{ 中の } r \text{ 番目のテストに含まれる} \\ 0 & \text{それ以外} \end{cases}$$

制約条件はクリーク C の全頂点と隣接するための条件である。

目的関数に含まれる $\lambda_i (i = 1, 2, \dots, n)$ は互いに独立な $[0, 1)$ の連続一様分布であり、本問題が解かれるたびにリサンプリングされ毎回ランダムにテストが構成される。この定式化は Belov and Armstrong (2006) で用いられたランダムにテスト構成を行う整数計画問題への定式化を項目重複について一般化したものとなっている [5]。

具体的にはアルゴリズム 2 でテスト構成を行う。

Algorithm 2 整数計画問題を用いた最大クリーク探索

Require: アイテムバンク, テスト構成条件**Ensure:** 等質テスト群

```
1: procedure HybridRBP( $L_1, L_2, CT', \text{AddCnt}, \alpha, CT$ )
2:    $ST := \text{current time}$ 
3:   /* initialize */
4:    $\text{global } C := \text{RndMCP}(L_1, L_2, CT')$ 
5:    $\text{global } C_{max} := C$ 
6:   while ( $\text{current time} - ST < CT$ ) do
7:     /* add step */
8:      $\text{count} := 0$ 
9:     while  $\text{count} < \text{AddCnt}$  do
10:       $Sol := \text{IPsolve}(\text{itemBank}, C)$                                 ▷ 式(4) ~ 式(7)を解く
11:      if  $Sol \neq \emptyset$  then                                       ▷ IPが解けた場合
12:         $C := C \cup \{Sol\}$ 
13:         $\text{count} ++$ 
14:        if  $|C_{max}| < |C|$  then
15:           $C_{max} := C$ 
16:        end if
17:      else                                                               ▷ IPが解けない場合
18:         $\text{break}$ 
19:      end if
20:    end while
21:     $\text{DeleteStep}(\text{AddCnt}, \alpha)$ 
22:  end while
23:  return  $C_{max}$ 
24: end procedure
25: procedure  $\text{DeleteStep}(\text{AddCnt}, \alpha)$ 
26:   /* delete step */
27:    $\text{count} := 0$ 
28:   while  $\text{count} < (\text{AddCnt} \times \alpha)$  do
29:      $C := C \setminus \{c \in C\}$                                        ▷  $c$ はランダムに選択
30:      $\text{count} ++$ 
31:   end while
32: end procedure
```

本アルゴリズムは大きく”initialize”と”add step”と”delete step”に分かれている。

”initialize”では RndMCP 法によりメモリの限界まで頂点数 L_1 のグラフを生成し最大クリーク探索をしその解をクリーク C の初期値とする。

”add step”では現在探索中の等質テスト群であるクリーク C へ新しく整数計画問題により得られたテストを追加することでより大きなクリークを構成する。これを Addcnt 回繰り返すか整数計画法が解けなくなるまで行う。

”delete step”では現在探索中のクリーク C からランダムにテストを削除することで、

局所解（極大クリーク）へ収束することを回避している．したがって，本アルゴリズムは探索中のクリークへ頂点の追加・削除を繰り返すことで，より大きなクリークを探そうとする局所探索法（local search）となっている．

本アルゴリズムの時間計算量は $O(CT)$ ，空間計算量は内部で使用する整数計画法の空間計算量が無視できるとすると， $O(|V|)$ となる．RndMCP 法と比較して空間計算量が $O(|V|^2)$ から $O(|V|)$ に減少しているため，構成可能なテストの上限は大きくなる．これにより，乱択法では 10 万個のテスト構成が上限であったが，本手法では 10 万を超えるテスト構成が可能になった．

3.4 露出率を軽減するテスト構成手法

先行研究 [7, 10, 11] では，ある項目がテスト構成全体で出題される回数（露出数）に偏りが生じる問題がある．露出数が大きい項目は受験者間で共有されやすく，その項目の信頼性が失われやすくなる [12]．項目 i の露出数は以下のように表される．

$$\sum_{r=1}^{|C|} X_{i,r} \quad (8)$$

ここで C はテスト構成（クリーク）で $|C|$ はその大きさで

$$X_{i,r} = \begin{cases} 1 & (\text{項目 } i \text{ がテスト } r \text{ に含まれる}) \\ 0 & (\text{項目 } i \text{ がテスト } r \text{ に含まれない}) \end{cases}$$

である．テスト構成 C における最大露出数を

$$E_C = \max_{i=1}^n \left(\sum_{r=1}^{|C|} X_{i,r} \right) \quad (9)$$

で表す．このときテスト構成 C における項目露出率は

$$\frac{E_C}{|C|} \quad (10)$$

である．

Ishii and Ueno (2015) はこの露出率を軽減するテスト構成を提案した [13]．Ishii and Ueno (2015) では RndMCP 法 (3.2 節) と整数計画問題を用いた最大クリーク探索 (3.3 節) を用いてテスト構成し，そのテスト構成の中で露出率が最小となるようなテスト構成を選択する．具体的にはアルゴリズム 3 のようにテストを構成する．

本アルゴリズムは大きく分けて”initialize”，”add step”，”delete step”，”output” から構成される．

”initialize”では RndMCP 法によりメモリ限界 L_1 の制限下で最大クリーク C を探索する．

”add step”では整数計画法により最大クリーク C に隣接する頂点を逐次的に探索し C

Algorithm 3 項目露出を軽減するテスト構成手法

Require: アイテムバンク, テスト構成条件**Ensure:** 等質テスト群

```
1: procedure lowItemExposure( $L_1, L_2, CT', \alpha, CT$ )
2:    $ST := \text{current time}$ 
3:   /* initialize */
4:    $global C := \text{RndMCP}(L_1, L_2, CT)$ 
5:    $global C_{candidate} := \emptyset$ 
6:   while ( $\text{current time} - ST < CT$ ) do
7:     /* add step */
8:      $Sol := \text{IPSolve}(\text{itemBank}, C)$  ▷ 式(4) ~ 式(7)を解く
9:     if  $Sol \neq \emptyset$  then ▷ IPが解けた場合
10:       $C := C \cup \{Sol\}$ 
11:       $C_{candidate} := C_{candidate} \cup \{C\}$ 
12:     else ▷ IPが解けない場合
13:        $\text{DeleteStep}(|C|, \alpha)$ 
14:     end if
15:   end while
16:   /* output */
17:    $C_{res} := C$ 
18:   for  $C_{cand}$  in  $C_{candidate}$  do
19:     if  $\frac{EC_{res}}{|C_{res}|} \leq \frac{EC_{cand}}{|C_{cand}|}$  then ▷ 露出率が最小の等質テストを選択する
20:        $C_{res} := C_{cand}$ 
21:     end if
22:   end for
23:   return  $C_{res}$ 
24: end procedure
25: procedure DeleteStep( $\text{cliqueSize}, \alpha$ )
26:   /* delete step */
27:    $count := 0$ 
28:   while  $count < (\text{cliqueSize} \times \alpha)$  do ▷ 最大露出項目を持つテストから削除する
29:      $C := C \setminus \{c \in C\}$ 
30:      $count ++$ 
31:   end while
32:    $C_{candidate} := C_{candidate} \cup \{C\}$ 
33: end procedure
```

に追加することで, より大きなテストを生成する. 生成したテスト C は $C_{candidate}$ に保存する.

”delete step”では”add step”で整数計画法が解けなかった場合, C から一定割合 (α) だけ頂点を削除する. この際, 最大露出項目を含むテストから削除することで露出数を減らしている. 削除によって変更されたクリーク C は $C_{candidate}$ に保存する.

”output”では生成したテストの集合 $C_{candidate}$ から露出率 $\frac{Ex_C}{|C|}$ が最も低いテスト C

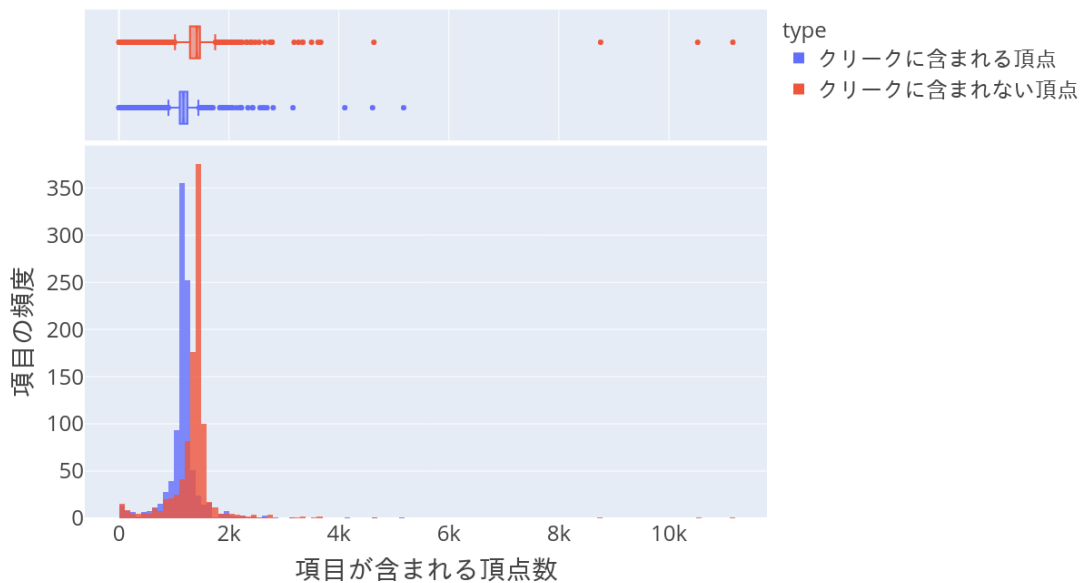


図4 項目が含まれる頂点数のヒストグラム (RndMCP)

を選択し出力する。

Ishii and Ueno (2015) によると, この手法は RndMCP 法等の従来手法 [2, 3, 7, 4] と比較していずれの手法よりも露出率が低くなった [13]。

4 従来手法の問題点

従来手法 Ishii and Ueno(2015) によりそれ以前の手法よりも露出率を改善することができたものの, テスト構成に 3.2 節の RndMCP 法や 3.3 節の整数計画法を利用しているため, 依然として露出数に偏りが生じる問題がある。

この問題を示すため, アイテムバンクサイズが $n = 978$ の実データに対して RndMCP 法でテスト構成し, 項目が含まれる頂点数について分析した。

RndMCP 法に与えたパラメータは, 生成する頂点数 $L_1 = 100000$, 最大クリーク探索の実行時間 $L_2 = 4hr$, 実験時間 $L_2 = 8hr$, 項目帳複数の上限 $OC = 5$ である。

生成した頂点がクリーク探索後のクリークに含まれるか, 否かで分け, 項目が含まれる頂点数のヒストグラムを作成した (図4). 図4の”type”の”in_clique”がクリークに含まれる頂点, ”out_clique”がクリークに含まれない頂点である.”in_clique”のヒストグラムは露出数のヒストグラムでもある. 図4より, クリークに含まれない頂点に含まれる項目のほうが多いことが分かる. 図4の”out_clique”より, ある2項目は生成した10000個以上の頂点に含まれるが, それらの頂点はクリークには含まれないことが分かる. 図4の”in_clique”では, ある3項目が生成した4000個以上の頂点に含まれ, それらの頂点はクリークに含まれるように, 露出数に偏りが生じていることが分かる。

表 2 クリークに含まれる頂点数と含まれない頂点数 (RndMCP)

クリークに含まれる頂点数	クリークに含まれない頂点数
45806	54194

表 2 は生成した頂点のうちクリークに含まれる頂点と含まれない頂点の数である。

この様に従来手法では頂点を生成した時点で頂点に含まれる項目に偏りが生じ、最大露出数が大きくなる問題に加えてクリークに含まれない頂点数が多いという問題がある。

5 提案手法

前節で述べたように従来手法 Ishii and Ueno (2015) [13] を用いたとしても RndMCP 法 (Algorithm3 の”initialize step”) や整数計画法 (Algorithm3 の”add step”) により露出数が偏り最大露出数が大きくなるという問題があった。本論文では RndMCP 法 (以降、第 1 段階目のテスト構成手法と呼ぶ) と整数計画法によるクリーク探索 (以降、第 2 段階目のテスト構成手法と呼ぶ) において、頂点生成の際に利用できるアイテムバンクの項目を制限することで、露出率及びテスト構成数を改善する手法を提案する。

5.1 第 1 段階目のテスト構成手法

前節で示したとおり、RndMCP 法において、アイテムバンクのすべての項目から選択して頂点を生成しているが、選択される項目に偏りが生じ、特定の項目の露出数が多くなるという問題がある。

提案手法では RndMCP 法の頂点選択 (Algorithm1 の step 1) において、項目ごとに既存の頂点に含まれる回数を計算し、その値が最大の項目をアイテムバンクから削除し、残りの項目から選択することで頂点を生成する。もし残りの項目から頂点生成できない場合は、RndMCP 法と同様にアイテムバンクのすべての項目から選択し頂点を生成する。(図 5) 本手法により項目が含まれる頂点数の最大値の増加を抑えつつ、頂点生成することができる。

クリークサイズは生成した頂点数以下となるので、項目の露出数はその項目が頂点に含まれる回数以下となる。(図 6)

したがって、提案手法により生成した頂点集合からグラフ構築し最大クリーク探索することで、最大露出数が小さいテストを構成することができる。

なお、本手法では残りの項目から頂点生成可能か確認する必要があるため、項目重複数条件式 (7) を除いた、式 (4) ~ 式 (6) の整数計画法を用いて頂点を生成している。

第 1 段階目の提案手法のアルゴリズムは Algorithm4 である。

step1 では部分グラフの頂点を L_1 個生成し頂点集合 V とする。itemCntV は各項目が V に含まれる回数を保持するための配列である。allowedItems は itemCntV が最大

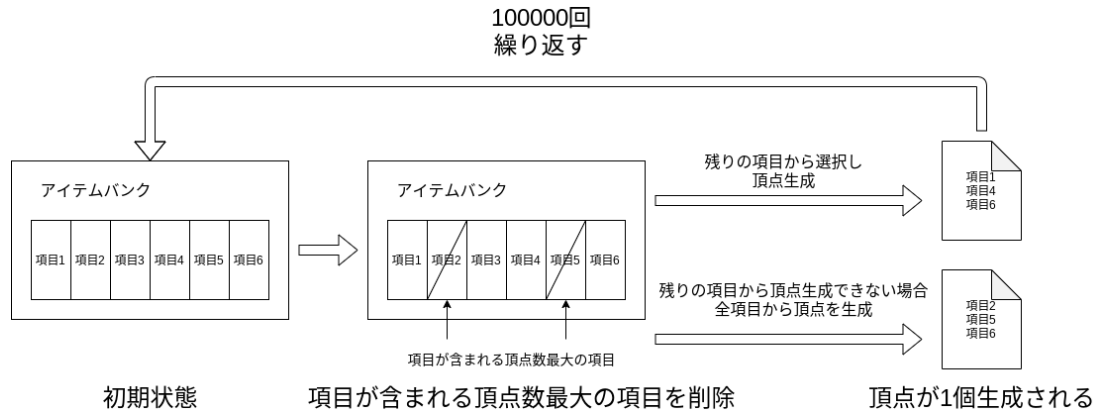


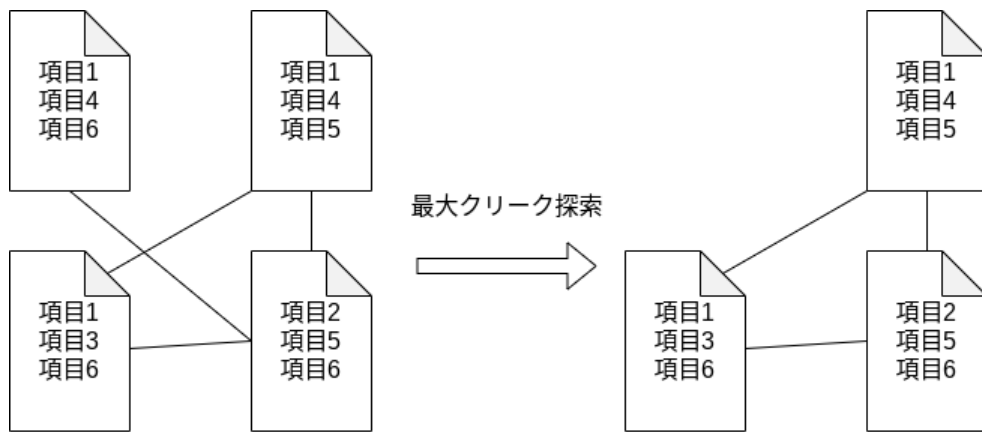
図5 第1段階目の提案手法の概念図

ではない項目の集合であり，これらの項目のみからなるアイテムバンクで，項目重複数条件（式（7））を除いた整数計画法（式（4）～式（6））を解き頂点（*Sol*）を生成する．*allowedItems* のみからでは解を求めることができない場合は，*RndMCP* 法と同様に与えられたアイテムバンクすべての項目からランダムに項目を選択し頂点を生成する．最後に頂点に含まれる項目について *itemCntV* をインクリメントする．

step2 では step1 で生成した頂点集合 *V* からグラフ *G* を構築する．具体的には項目重複数条件を満たす 2 頂点間に辺を引きグラフを構築する．

step3 では step2 で構築したグラフ *G* において最大クリーク探索をして，等質テスト構成をする．

以上 step1～step3 を時間 *CT* だけ繰り返し最大クリークを探索する．



頂点を生成しグラフを構築

最大クリーク

項目が含まれる頂点数

項目1	3
項目2	1
項目3	1
項目4	2
項目5	2
項目6	3

露出数

項目1	2
項目2	1
項目3	1
項目4	1
項目5	2
項目6	2

図 6 最大クリーク探索の例 (n=6, OC=1)

Algorithm 4 選択可能な項目を制限した RndMCP 法

Require: アイテムバンク, テスト構成条件**Ensure:** 等質テスト群

```
1: procedure LimitItemRndMCP( $L_1, L_2, CT$ )
2:    $C := \emptyset, C_{max} := \emptyset$ 
3:    $ST := current\ time$ 
4:   while ( $current\ time - ST < CT$ ) do
5:     /* Step1 */
6:      $V = \emptyset$ 
7:      $n := itemBank.size$ 
8:     for  $i \leftarrow 0$  to  $n - 1$  do
9:        $itemCntV[i] := 0$ 
10:    end for
11:    while  $|V| < L_1$  do
12:       $allowedItems = \emptyset$ 
13:      for  $i \leftarrow 0$  to  $n - 1$  do
14:        if  $itemCntV[i] \neq max(itemCntV)$  then
15:           $allowedItems = allowedItems \cup \{i\}$ 
16:        end if
17:      end for
18:       $Sol = IPsolve(allowedItems)$   ▷  $allowedItems$  のみからなるアイテムバンクで式
      (4) ~ 式 (6) の IP を解く
19:      if  $Sol = \emptyset$  then  ▷ IP が解けない場合
20:         $Sol =$  テスト 1 個をランダム生成
21:      end if
22:       $V = V \cup \{Sol\}$ 
23:      for all  $i \in Sol$  do
24:         $itemCntV[i] ++$ 
25:      end for
26:    end while
27:    /* Step2 */
28:     $G = (V, E)$  グラフ構築  ▷ 二頂点が重複項目数条件を満たす場合, 辺を引く
29:    /* Step3 */
30:     $C := MCP(G, L_2)$   ▷  $G$  の最大クリークを時間  $L_2$  だけ探索
31:    if  $|C_{max}| < |C|$  then
32:       $C_{max} := C$ 
33:    end if
34:  end while
35:  return  $C_{max}$ 
36: end procedure
```

5.2 第 2 段階目のテスト構成手法

第 1 段階目では各項目が頂点に含まれる回数を計算し，それが最大の項目を除いて頂点生成することで等質テスト構成（クリーク）における項目の露出数を抑えたが，第 2 段階目でも同様に各項目の露出数を計算し，それが最大の項目を除いて，最大クリークのための整数計画問題（式（4）～式（7））を解きクリークの隣接頂点を逐次的に探索することで，項目の露出数を抑えつつ，テストを構成する手法を提案する．

具体的には Algorithm5 で第 2 段階目のテストを構成する．

initialize では第 1 段階目の提案手法で初期解を求め，初期解における項目露出数を計算する．

add step では時間 CT だけ最大クリーク C に隣接する頂点を逐次的に探索する．本手法では第 1 段階目の手法と同様に最大露出数ではない項目からなる項目集合 `allowedItems` に対して，最大クリーク探索のための整数計画問題（式（4）～式（7））を解く．もし `allowedItems` から解を得ることができない場合は，従来手法と同じように与えられたアイテムバンクすべての項目に対して整数計画問題を解く．いずれの場合でも解を得ることができない場合は，局所解（極大クリーク）に陥っているので，delete step で頂点を削除する．解を得ることができた場合は最大クリーク C にその解を追加し，解に含まれる項目の露出数をインクリメントする．

delete step ではクリーク C から一定割合（ α ）だけ頂点をランダムに削除する．削除した頂点に含まれる項目の露出数はデクリメントする．

以上の add step が第 2 段階目のテスト構成の提案手法である．

Algorithm 5 選択可能な項目を制限し整数計画問題を用いた最大クリーク探索

Require: アイテムバンク, テスト構成条件**Ensure:** 等質テスト群

```
1: procedure LimitItemHybridRBP( $L_1, L_2, CT', \alpha, CT$ )
2:    $ST := current\ time$ 
3:   /* initialize */
4:    $global\ C := LimitItemRndMCP(L_1, L_2, CT')$            ▷ 第1段階目の提案手法を利用
5:    $n := itemBank.size$ 
6:   for  $i \leftarrow 0$  to  $n - 1$  do
7:      $global\ exposure[i] := 0$ 
8:   end for
9:   for all  $v \in C$  do
10:    for all  $i \in v$  do
11:       $exposure[i] ++$            ▷ 露出数の初期化
12:    end for
13:  end for
14:  while ( $current\ time - ST$ ) <  $CT$  do
15:    /* add step */
16:     $allowedItems := \emptyset$ 
17:    for  $i \leftarrow 0$  to  $n - 1$  do
18:      if  $exposure[i] \neq max(exposure)$  then
19:         $allowedItems := allowedItems \cup \{i\}$ 
20:      end if
21:    end for
22:     $Sol := IPSolve(allowedItems, C)$            ▷ 式(4)~式(7)を解く
23:    if  $Sol = \emptyset$  then           ▷ 提案手法のIPが解けない場合
24:       $Sol := IPSolve(itemBank, C)$ 
25:      if  $Sol = \emptyset$  then           ▷ 従来手法のIPも解けなかった場合
26:         $DeleteStep(|C|, \alpha)$ 
27:        continue
28:      end if
29:    end if
30:     $C := C \cup \{Sol\}$ 
31:    for all  $i \in Sol$  do
32:       $exposure[i] ++$ 
33:    end for
34:  end while
35:  return  $C$ 
36: end procedure
37: procedure DeleteStep( $cliqueSize, \alpha$ )
38:   /* delete step */
39:    $count := 0$ 
40:   while  $count < (cliqueSize \times \alpha)$  do
41:      $C := C \setminus \{c \in C\}$            ▷  $c$  はランダムに選択
42:     for all  $i \in c$  do
43:        $exposure[i] --$ 
44:     end for
45:      $count ++$ 
46:   end while
47: end procedure
```

表 3 実アイテムバンクの詳細

Pool Size	Parameter a			Parameter b		
	Range	Mean	SD	Range	Mean	SD
978	0.12 ~ 3.08	0.43	0.2	-4 ~ 4.55	-0.22	1.16

表 4 テスト情報量の条件

$\theta = -2.0$	$\theta = -1.0$	$\theta = -0.0$	$\theta = 1.0$	$\theta = 2.0$
2.0/2.4	3.2/3.6	3.2/3.6	3.2/3.6	2.0/2.4

6 比較実験

提案手法の有効性を示すために、第 1 段階目と第 2 段階目のテスト構成において従来手法 Ishii and Ueno (2015) [13] との比較実験を行った。実行環境は Ubuntu18.04 を OS とする計算機 (CPU: Intel Core i9-9900X 3.50GHz, RAM 128GB) である。

本実験ではシミュレーションによるアイテムバンクと、実データによるアイテムバンクを用いた。シミュレーションによるアイテムバンクは 500, 1000, 2000 項目を持ち、項目の識別力パラメータ a を $\log_2 a \sim N(0, 1^2)$, 困難度パラメータ b を $b \sim N(0, 1^2)$ として発生させた。実データによるアイテムバンクは 978 項目をもち、識別力パラメータ a , 困難度パラメータの詳細 b は表 3 の通りであった。

テスト情報量の条件は情報量の和が、表 4 の 5 つの推定能力値 θ で下限値/上限値の中に収まることである。

テストの構成数はいずれも 25 項目である。項目帳複数条件 (OC) は従来手法 [13] にならない OC=0, 5, 10 の 3 通りの条件で評価した。

従来手法 [13] と提案手法の整数計画法の求解には CPLEX12.9[20] を使い、整数計画緩和の解との相対ギャップが 10^{-4} 以下で打ち切るようにした (デフォルトのオプション)。またテスト構成の整数計画法では厳密解を求める必要はなく、制約条件を満たす許容解を見つけることができれば等質テストを構成できる。そこで CPLEX に 60 秒で求解を打ち止めるよう設定した。

6.1 第 1 段階目の比較実験

$L_1 = 100000$, $L_2 = 4hr$, $CT = 4hr$ とし、従来手法 [13] の RndMCP と第 1 段階目の提案手法で比較実験をした。ただし $n = 978$ の場合は $CT = 4hr$ では頂点を L_1 個集めることができなかったため、 $CT = 8hr$ として実験した。

重複項目数の条件の上限値を OC, テスト構成数を $|C|$, 最大露出数を E_C , 露出率を $\frac{E_C}{|C|}$ として結果を表 5 に示す。

表 5 第 1 段階目の実験結果

Pool Size	OC	従来手法 (RndMCP)			提案手法		
		$ C $	E_C	$\frac{E_C}{ C }$	$ C $	E_C	$\frac{E_C}{ C }$
500	0	10	1	0.1	10	1	0.1
	5	4339	354	0.08159	4989	365	0.07316
	10	99976	13009	0.13012	99970	5372	0.05374
1000	0	18	1	0.05556	18	1	0.05556
	5	46392	3364	0.07251	50778	1626	0.03202
	10	100000	8740	0.08740	100000	2847	0.02847
2000	0	32	1	0.03125	32	1	0.03125
	5	96732	3761	0.03888	97422	1398	0.01435
	10	100000	4035	0.04035	100000	1418	0.01418
978	0	18	1	0.05556	19	1	0.05263
	5	45806	5177	0.11302	54794	1726	0.03150
	10	100000	16495	0.16495	100000	2755	0.02755

表 6 クリークに含まれる頂点数と含まれない頂点数 (提案手法)

クリークに含まれる頂点数	クリークに含まれない頂点数
54794	45206

すべての場合において提案手法の露出率が従来手法の露出率以下となった。頂点に含まれる回数が最も多い項目以外から頂点生成する提案手法が露出率を抑えるのに有効であることが示された。

また $OC = 5$ の場合においてテスト構成数が提案手法のほうが従来手法よりも大きい。そこでテスト構成数が大幅に増加した $n = 978, OC = 5$ のクリーク探索する前のグラフにおいて、頂点が最大クリーク探索後のクリークに含まれるか、否かで分け、項目が含まれる頂点数のヒストグラムを作成した (図 7)。

図 7 より提案手法では、従来手法の図 4 と比較して、項目ごとに見るとクリークに含まれる頂点に含まれる項目のほうが多いことが分かる。クリークに含まれる頂点・含まれない頂点共に項目が偏りなく含まれ、結果として最大露出数が 1726 と小さいことが分かる。

表 6 は生成した頂点のうちクリークに含まれる頂点と含まれない頂点の数である。

以上より提案手法は露出数の偏りが抑えられて、クリークサイズも大きくなることが分かった。

6.2 第 2 段階目の比較実験

第 2 段階目の提案手法の有効性を確認するために、第 1 段階目の提案手法により求めたクリークを初期解として、第 2 段階目で従来手法 Ishii and Ueno (2015) (Algorithm 3)

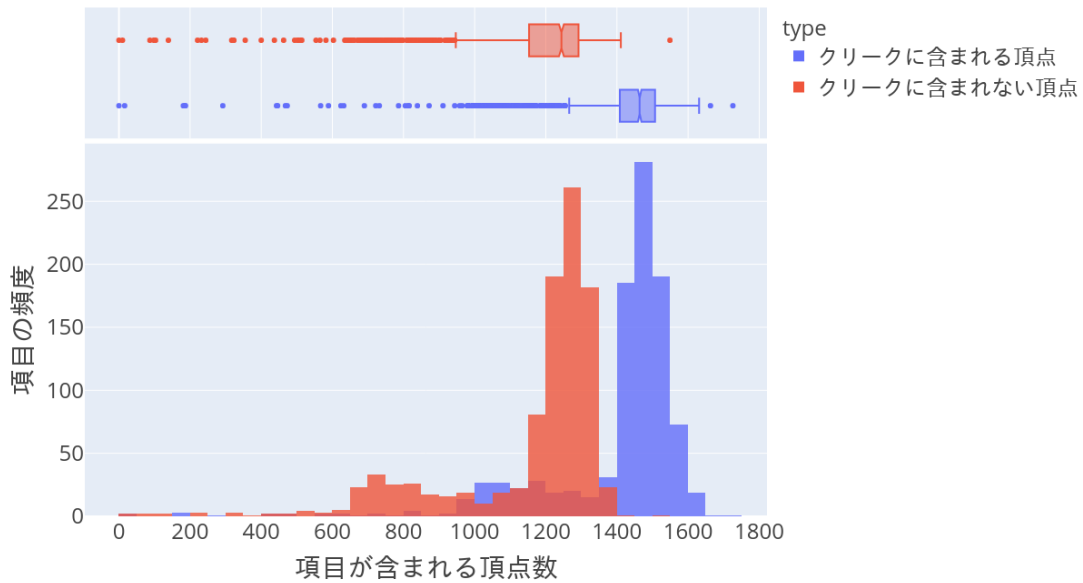


図 7 項目が含まれる頂点数のヒストグラム (提案手法)

表 7 第 2 段階目の実験結果

PoolSize	OC	従来手法			提案手法		
		$ C $	E_C	$\frac{E_C}{ C }$	$ C $	E_C	$\frac{E_C}{ C }$
500	0	18	1	0.05556	18	1	0.05556
	5	10533	756	0.07177	10271	562	0.05472
	10	(*)99970	5372	0.05374	101433	5448	0.05371
1000	0	35	1	0.02857	33	1	0.03030
	5	50789	1626	0.03201	53206	1626	0.03056
	10	(*)100000	2847	0.02847	101352	2886	0.02848
2000	0	70	1	0.01429	65	1	0.01538
	5	97528	1399	0.01434	98379	1398	0.01421
	10	(*)100000	1418	0.01418	101453	1439	0.01418
978	0	36	1	0.02778	33	1	0.03030
	5	54894	1728	0.03148	56636	1726	0.03048
	10	(*)100000	2755	0.02755	100875	2779	0.02755

と提案手法 (Algorithm5) の等質テスト構成のための整数計画法を実行する実験をした。パラメータは従来手法, 提案手法ともに L_1, L_2, CT' は第 1 段階目の実験の通りで, $\alpha = 0.1, CT = 6hr$ として実験を行った。結果を表 7 に示す。

表中の (*) は初期解が選ばれたことを示している。OC = 0 以外では第 2 段階目の提案手法は従来手法に比べ, テスト構成数が大きく, 露出率も OC = 5 では抑えられている。n = 1000, 2000 の OC = 10 では従来手法のほうが露出率がわずかに小さいが, これは従

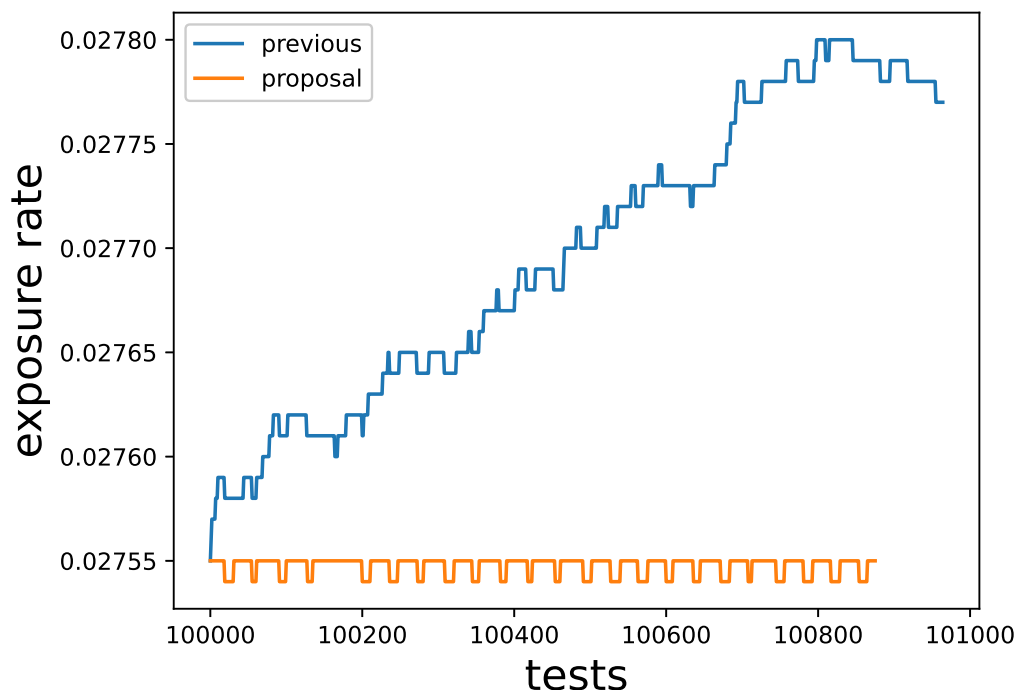


図 8 第 1 段階目が提案手法の場合の $n = 978, OC = 10$ のテスト構成数と露出率の遷移

来手法が露出率が最も小さい初期解を選択しているからである。

第 2 段階目の従来手法は構成した等質テスト構成のうち、露出率が最も小さいテスト構成を選択するアルゴリズムであるため、初期解として露出率が小さいクリークを与えると、それを最終的なテスト構成として選択してしまうという問題がある。図 8 は $n = 978, OC = 10$ のテスト構成数と露出率の遷移をプロットしたものである。

図 8 より実験を初めて 6 時間後では従来手法は提案手法よりもテスト構成数は大きいですが、露出率が初期解よりも大きくなってしまったため、最終的な解としてはクリークサイズ 100000 の初期解を選択してしまうことが分かる。

第 1 段階目が従来手法 (RndMCP 法)、第 2 段階目も従来手法、すなわち従来手法 Ishii and Ueno (2015) の場合は図 9 のようなテスト構成数と露出率の遷移になる。初期解の露出率が大きいため、テスト構成数の増加とともに露出率は減少していくことが分かる。この場合は露出率最小のテスト構成を選択しても、初期解が選択されることはないが露出率は提案手法と比較して大幅に大きくなってしまふ。

$OC = 0$ ではいずれの場合においても従来手法のほうが提案手法よりも露出率が低くなる。従来手法は今まで構成したテスト構成のうち最も露出率が低いものを選択しテスト構成とするが、提案手法は実験終了時のクリークがテスト構成になるという違いがその要因として考えられる。 $OC = 0$ の場合は、もし露出数が 2 以上ならば重複項目数が 1 以上と

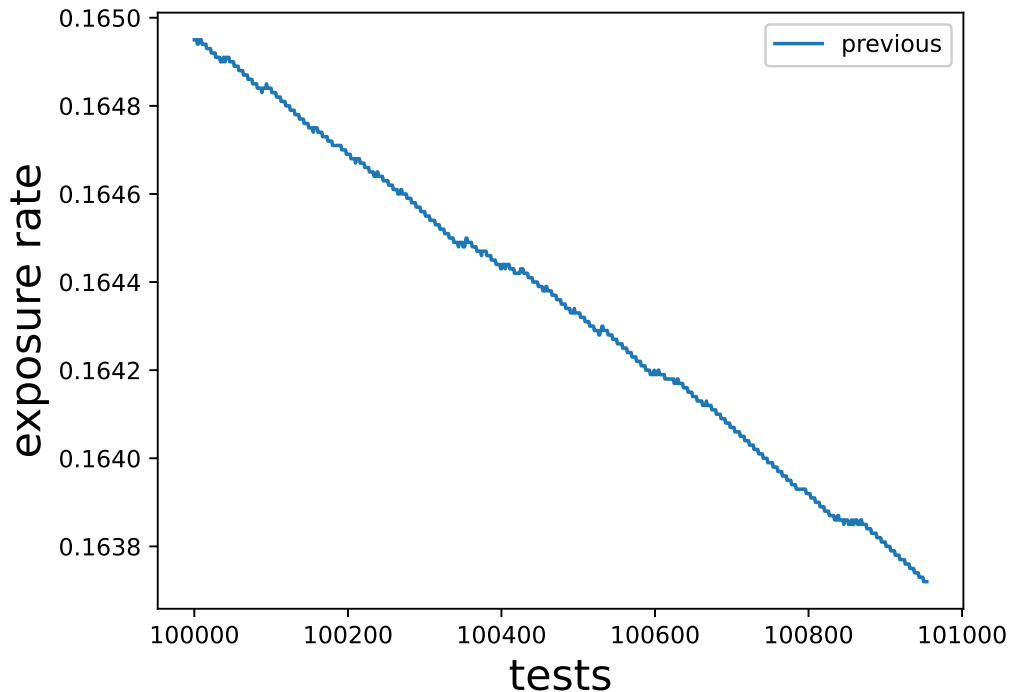


図9 第1段階目が従来手法の場合の $n = 978$, $OC = 10$ のテスト構成数と露出率の遷移

なり項目重複数条件に違反するので、露出数は必ず 1 になる。したがって露出率が最小のテスト構成はテスト構成数が最大のものである。また $OC = 0$ では構成可能なテスト構成が小さく、局所解に陥ることがある。局所解に陥った場合 delete step で $\alpha = 0.1$ だけ削除するが、提案手法で実験終了直前に削除してしまった場合は、テスト構成数が小さくなってしまう。一方で従来手法は今までに構成したテスト構成のうち露出率が最小のものを選択するため、実験終了直前に削除したとしても、今までに構成した中で露出率が最小、すなわちテスト構成数最大のテスト構成を選択するので、“delete step”によるテスト構成数減少の影響を受けない。したがって $OC = 0$ においては従来手法のほうが露出率が低くなると考えられる。

表8に第1段階目と第2段階目が従来手法の Ishii and Ueno (2015) の実験結果と、第1段階目と第2段階目が提案手法の実験結果をまとめた。

$OC=0$ 以外では提案手法が従来手法よりも露出率が小さいことが確認できる。テスト構成数については、初期解からの増加数は $n = 500, 978$ については従来手法のほうが大きい。特に $n = 1000, 2000$ の特に $OC = 10$ については提案手法のほうが大きいことが表5と表8から分かる。

表 8 従来手法と提案手法の実験結果

PoolSize	OC	従来手法 (Ishii and Ueno (2015))			提案手法		
		$ C $	E_C	$\frac{E_C}{ C }$	$ C $	E_C	$\frac{E_C}{ C }$
500	0	18	1	0.05556	18	1	0.0556
	5	11448	876	0.07652	10271	562	0.05472
	10	101399	13115	0.12934	101433	5448	0.05371
1000	0	35	1	0.02857	33	1	0.003030
	5	49392	3472	0.07029	53206	1626	0.03056
	10	101264	8783	0.08673	101352	2886	0.02848
2000	0	70	1	0.01429	65	1	0.01538
	5	97675	3777	0.03867	98379	1398	0.01421
	10	100818	4048	0.04015	101453	1439	0.01418
978	0	36	1	0.02778	33	1	0.0303
	5	48258	5269	0.10918	56636	1726	0.03048
	10	100954	16528	0.16372	100875	2779	0.02755

7 むすび

本論文では項目露出を抑制しつつ等質テスト構成する手法を提案した．本手法はクリーク探索における頂点生成の際に選択可能な項目を露出数が最大のもの以外に制限することで，テスト構成の露出率を改善した．シミュレーションデータと実データの実験により，第1段階目では露出率，テスト構成数ともに改善し，第2段階目では露出率を抑制しつつ初期解からテスト構成数を増やすことができ，提案手法の有効性が確認できた．今後は第1段階目で項目が含まれる頂点数の偏りを抑制したことで，なぜテスト構成数が増加したのかを分析することを考えている．また，本手法では第2段階目の整数計画法の求解が提案手法と比較して時間がかかる．Fuchimoto and Ueno (2020) [21] が提案する並列アルゴリズムに本手法を応用し，露出率を抑制しつつ更に大きな等質テストを構成する手法を開発することを考えている．

参考文献

- [1] 植野真臣 and 永岡慶三. *e テスティング*. 培風館, 2009.
- [2] Wim J van der Linden. *Linear Models for Optimal Test Design*. Springer, 2005.
- [3] Koun Tem Sun, Yu Jen Chen, Shu Yen Tsai, and Chien Fen Cheng. Creating irt-based parallel test forms using the genetic algorithm method. *Applied measurement in education*, 21(2):141–161, 2008.
- [4] Pokpong Songmuang and Maomi Ueno. Bees algorithm for construction of multiple test forms in e-testing. *IEEE Transactions on Learning Technologies*, 4(3):209–221, 2010.
- [5] Dmitry I Belov and Ronald D Armstrong. A constraint programming approach to extract the maximum number of non-overlapping test forms. *Computational Optimization and Applications*, 33(2-3):319–332, 2006.
- [6] 石井隆稔, 植野真臣, et al. 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*, 97(2):270–280, 2014.
- [7] Takatoshi Ishii, Pokpong Songmuang, and Maomi Ueno. Maximum clique algorithm and its approximation for uniform test form assembly. *IEEE Transactions on Learning Technologies*, 7(1):83–95, 2014.
- [8] Etsuji Tomita, Sora Matsuzaki, Atsuki Nagao, Hiro Ito, and Mitsuo Wakatsuki. A much faster algorithm for finding a maximum clique with computational experiments. *Journal of Information Processing*, 25:667–677, 2017.
- [9] Chu Min Li, Hua Jiang, and Felip Manyà. On minimization of the number of branches in branch-and-bound algorithms for the maximum clique problem. *Computers & Operations Research*, 84:1–15, 2017.
- [10] Takatoshi Ishii and Maomi Ueno. Algorithm for uniform test assembly using a maximum clique problem and integer programming. In *International Conference on Artificial Intelligence in Education*, pages 102–112. Springer, 2017.
- [11] 石井隆稔, 赤倉貴子, and 植野真臣. 複数等質テスト構成における整数計画問題を用いた最大クリーク探索の近似法. 電子情報通信学会論文誌 *D*, 100(1):47–59, 2017.
- [12] Howard Wainer. Cats: Whither and whence. *Psicologica*, 21(1):121–133, 2000.
- [13] Takatoshi Ishii and Maomi Ueno. Clique algorithm to minimize item exposure for uniform test forms assembly. In *International Conference on Artificial Intelligence in Education*, pages 638–641. Springer, 2015.
- [14] Ellen Boekkooi-Timminga. The construction of parallel tests from irt-based item banks. *Journal of Educational Statistics*, 15(2):129–145, 1990.

- [15] Ronald D Armstrong, Douglas H Jones, and Zhaobo Wang. Automated parallel test construction using classical test theory. *Journal of Educational Statistics*, 19(1):73–90, 1994.
- [16] Wim J van der Linden and Jos J Adema. Simultaneous assembly of multiple test forms. *Journal of educational measurement*, 35(3):185–198, 1998.
- [17] Frederic M Lord and Melvin R Novick. *Statistical theories of mental test scores*. IAP, 2008.
- [18] Frank B Baker and Seock Ho Kim. *Item response theory: Parameter estimation techniques*. CRC Press, 2004.
- [19] Ronald D Armstrong, Douglas H Jones, and Charles S Kuncze. Irt test assembly using network-flow programming. *Applied Psychological Measurement*, 22(3):237–247, 1998.
- [20] IBM. Ilog cplex optimization studio cplex user’s manual 12.9, 2019.
- [21] 淵本 壱真 and 植野 真臣. 等質テスト構成における整数計画法を用いた最大クリーク探索の並列化. 電子情報通信学会論文誌 *D*, 2020.