

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

石井 隆稔[†] 赤倉 貴子[†] 植野 真臣^{††}

An Approximate Maximum Clique Algorithm Using Integer Programming
for Uniform Test Assembly

Takatoshi ISHII[†], Takako AKAKURA[†], and Maomi UENO^{††}

あらまし 本論文では、複数等質テスト構成の為に整数計画問題を用いた最大クリークの近似探索手法を提案する。複数等質テストとは、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。出題可能な項目のデータベース(アイテムバンク)から所望の性質を満たす出題項目の集合を計算機により大量に構成することを複数等質テスト自動構成と呼ぶ。この複数等質テスト構成の重要な課題の一つは、与えられたアイテムバンクから可能な限り多くのテストを構成することである。著者らは先行研究で、複数等質テスト構成を最大クリーク探索として扱うことで、従来の手法によるテスト構成数を大きく改善することに成功した。しかし、空間計算量が大きすぎるため、未だ理想的に必要なとされる数のテストを構成できない問題がある。本論文ではこの問題を緩和し、より多くのテストを構成する近似アルゴリズムを提案する。具体的には、探索に必要なグラフ構造を、整数計画問題を用いて適宜的に作成することで、探索の空間計算量を削減する。本論文ではシミュレーション及び実データを用いた実験により、本手法が他手法よりも同一の計算時間・計算機環境で多くのテスト群を構成できることを示した。

キーワード e テスティング, 項目反応理論, 複数等質テスト自動構成, 最大クリーク問題, 整数計画問題

1. ま え が き

e テスティングとは、情報技術を用いたテストによる受検者能力を測定する手法であり、様々な利点をもつ[1]。近年では、様々なテストがe テスティング上で行われつつあり、我が国でも情報処理技術者試験 IT パスポート [2] や医学部共用 CBT [3] 等様々な資格試験が e テスティングに移行しつつある。既に 2007 年、e テスティングの国際規格 ISO/IEC 23988:2007 [4] (国内規格 JIS X7221:2011 [5]) が策定され、今後益々 e テスティングの普及が見込まれる。

この規格の中で、資格試験や入学試験等のその結果が受検者に大きな影響を与えるテストでは“複数等質

テスト”を用いることが推奨されている。複数等質テストとは、それぞれのテストに含まれる項目は異なるが、受検者の推定得点の予測誤差が等しいテスト群である。例えば、資格試験などでは毎回の難易度が異なるように、テストの統計的な性質、得点分布や所要時間等が一定でなければならない。これまで複数等質テストはテスト管理者の勘と経験により構成されてきたが、近年、e テスティングの普及に伴い、テストの自動構成が可能となりつつある [1], [6], [7]。

一般に、e テスティングでは、テストの管理手法としてアイテムバンク方式が用いられる。アイテムバンクは出題する問題(以降、項目と呼ぶ)を管理するデータベースであり、項目ごとの出題分野や統計データ等を格納している。テストの自動構成とは、これらのデータを利用し、所望の性質をもつような項目の組み合わせ(つまり、テスト)を計算機により探索することである。一般に、この探索は数学的最適化問題として解かれる。例えば、Songmuang and Ueno は最適化問題の解探索手法の一つである Bees Algorithm を用いて

[†] 東京理科大学工学部情報工学科, 東京都
Department of Information and Computer Technology,
Tokyo University of Science, Tokyo, 125-8585 Japan

^{††} 電気通信大学大学院情報システム学研究科, 調布市
Graduate School of Information Systems, The University of
Electro-Communications, Chofu-shi, 182-8585 Japan
DOI:10.14923/transinfj.2016JDP7057

テスト構成を行う手法を提案・開発している [8]. 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 [1].

この複数等質テスト自動構成において最も重要な課題の一つが与えられたアイテムバンクからできるだけ多くのテストを構成することである. テスト構成数はテストシステムの信頼性に大きな影響を与える. 複数等質テストを採用する e テスティングシステムでは, 各受験者に可能な限り版の異なるテストを割り当て能力の測定を行う. そのため, テスト構成数が受験者数に対して小さい場合は, 同じテストが複数の受験者に出題され, 項目やテストが暴露・流出するリスクが高まる. したがって, 複数等質テストの構成手法の研究においては, 与えられたアイテムバンク・条件で可能な限り多くのテストを構成できることが求められている.

著者らは先行研究において, 与えられたアイテムバンク・構成条件で最大数のテストを構成する手法を提案した [9]. この手法はテスト構成を, グラフ上で定義される最適化問題である最大クリーク問題として行う. 具体的には与えられたアイテムバンク・テスト構成条件で構成可能な全てのテストを頂点, 二つのテストが等質かつ, 共通項目数が一定以下である場合に頂点 (テスト) 間に辺を引いたグラフから, どの 2 頂点も辺で結ばれているクリークと呼ばれる構造を探すことで複数等質テスト構成を行う. このクリークの中で最も多く頂点をもつものを探索・出力することで, 与えられたアイテムバンク・構成条件で理論的に最大数の複数等質テストを構成可能である [9].

しかし, 構成可能な全てのテストを頂点とするグラフは, 非常に大きな構造となるため, 最大クリークを探索することや, グラフ構造全てを主記憶上に保存することは困難である. そのため, 著者らは, このグラフからランダムに部分グラフを選択し, そこからの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複数等質テスト構成手法を提案した. 本手法により, それまでの既存研究より 10~100 倍以上多くのテストを構成できた [10].

ただし, この手法 [10] でも実用的には十分な数のテストを構成できない. 例えば, 複数等質テストによる e テスティングを実施している情報処理技術者試験「IT パスポート」は年間 8 万人以上 (H27) が受験している. また, テストは複数の出題領域をもち, 領域別のアイテムバンクはそれぞれ 500~2000 程度の項目をもつ.

このような試験では前の年に出題したテストを次の年に出題することは項目流出のリスクを増大させ, アイテムバンク中の項目の特性劣化の原因になることが知られている [11]. この問題を避けるために, 複数等質テスト構成には, 500~2000 程度のアイテムバンクから数十問の項目をもつテストを, 数十万程度構成することが求められる. しかし, 著者らの先行研究 [10] は世界で最も多くのテストを構成できる手法であるが, 現実的に必要とされる数のテストを実時間・実環境中で構成することはできない. この背景の下, より多くのテストを構成可能な手法の開発が急務である.

本論文では従来手法 [10] を改善し, 実時間・実環境中で従来手法 [10] より多くのテストを構成可能な手法を提案する. 従来手法は, テスト構成のためのグラフからランダムに部分グラフを作成し, その中から最大クリークを探索することを繰り返す手法である. したがって, 部分グラフサイズ (著者らの計算機環境では最大で 10 万) より大きなクリークを探索することはできない. また, 部分グラフ中にはクリークに含まれない頂点も多数保持しなければならない. そのため, 構成テスト数を $|C|$ とした場合, 少なくとも $O(|C|^2)$ の空間計算量を必要とする. 本論文ではこれらの問題を緩和し, 従来手法を改善した近似アルゴリズムを提案する. 具体的には, テスト構成のためのグラフの頂点を整数計画問題を解くことにより探索の必要に応じて構成することにより, 計算に必要な空間計算量を $O(|C|)$ へ減少させる. これにより, 提案手法は先行研究 [10] と比較し, 同一の計算量でより多くの複数等質テストを構成可能である.

最後に本論文では, この手法の有効性をシミュレーションデータ・実データを用いた実験で示す. 具体的には, 幾つかの現実的なテスト構成条件について従来手法とテスト構成数を比較し本手法の有効性を示す.

2. 複数等質テスト構成

一般に複数等質テストは, 以下の条件を満たすテストの集合であると定義する (例えば, [9], [10], [12]).

(1) それぞれのテストでの受検者の推定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が一定値以下である.

e テスティングでは, アイテムバンクという項目データベース (項目内容, 項目ごとの出題領域や統計データなどを格納している) を用いる. 複数等質テスト構

成とは、アイテムバンク中から、それぞれ推定得点の予測誤差の等しい項目の組み合わせ群を計算機により探索することであり、一般に組み合わせ最適化問題として解くことができる [8], [12]～[21].

図 1 は、複数等質テスト構成の概念図であり、アイテムバンクから、項目の組み合わせは異なるが互いに推定得点の予測誤差の等しいテスト群を選択している。

ここで推定得点の予測誤差は、先行研究 (例えば, [8]～[10], [12]～[27]) では、項目反応理論におけるテスト情報量で評価されている。

項目反応理論は、受検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、異なるテストを受けた受検者能力の同一尺度上での比較や、テストの能力測定精度の定量的な評価が可能となる。

項目反応理論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では、 j 番目の受検者が i 番目の項目に正答する確率を以下の式でモデル化する。

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

ただし、 $i = 1, \dots, n$, $j = 1, \dots, m$ であり、 n は項目数、 m は受検者数である。また、 $a_i \in [0, \infty)$ は i 番目の項目の識別力パラメータ、 $b_i \in (-\infty, \infty)$ は困難度パラメータと呼ばれる項目パラメータである。 $\theta_j \in (-\infty, \infty)$ は受検者 j の能力パラメータと呼ばれ、受検者の能力を表す潜在変数である。

等質テストでは、各テストでの受検者の項目反応理論での能力推定値の予測誤差を等しくするように構成する。ここで、能力推定値の予測誤差は、フィッシャー情報量の逆数に漸近的に一致することが知られている [1]. そのため、テスト構成では、テストごとの情報量を一定にすることで等質テストを構成することが一般的である [8], [13]～[27]. 式 (1) を用いて計算したフィッシャー情報量が“項目情報量”であり、以下のよ

うに表される。

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

また、テスト全体のテスト情報量は、以下のように定義される。

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

ここで T はテストに含まれる項目の集合である。

多くの先行研究では、このテスト情報量とテスト中の項目数をテストの構成条件として採用している。本研究でも先行研究と同様に、テスト情報量とテスト項目数をテスト構成条件として採用する。具体的には、同一の項目数であり、かつ、テスト情報量が設定された上限・下限制約を満たすテストを等質であるとする。

また、複数等質テスト中の任意の二つのテストは、互いに等質でありながらも、異なる項目の集合でなければならない。そのため、複数等質テストの構成条件には、これを担保するための制約も含まれる。具体的には、“任意の二つのテスト間の共通項目の数は一定値以下である”という条件を与える。本論文中ではこの条件を以後、項目重複数条件と呼ぶ。

テスト情報量への上下限制約の与え方は、能力パラメータ θ 上の幾つかの点をサンプリングし ($\Theta = \{\theta_1, \dots, \theta_l, \dots, \theta_L\}$), それらの点ごとに上下限 ($UB(\theta_l), LB(\theta_l)$) を設定する。

例えば表 1 では $\theta = \{-2, -1, 0, 1, 2\}$ の点をサンプリングし、 $\theta = -2$ の点には下限として $1.1 \leq I(\theta = -2)$ と上限として $I(\theta = -2) \leq 1.6$ を、 $\theta = -1$ の点には下限として $1.5 \leq I(\theta = -1)$ と上限として $I(\theta = -1) \leq 2.0$ を、 $\theta = 0$ の点には下限として $1.5 \leq I(\theta = 0)$ と上限として $I(\theta = 0) \leq 2.0$ を、設定している。

例えば、図 2 は、テスト情報量への上下限制約の概念図である。図中の実線はテスト情報量関数に与えられた上限と下限である。図中の # 1～# 4 は構成テストの情報量関数である。# 1, # 2 は共にこの制約の上限下限を満たしており、等質である。また、# 3,

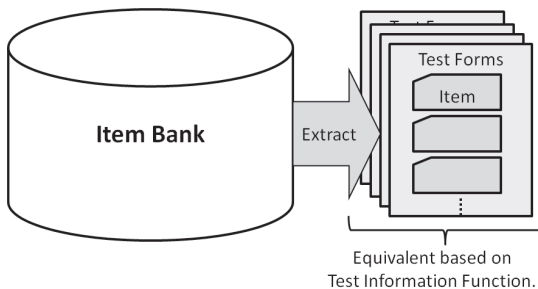


図 1 アイテムバンクからの複数等質テスト構成
Fig.1 Uniform test assembly from item bank.

表 1 情報量の上下限の例

Table 1 An example of upper and lower bound for test information.

テスト情報量 (下限/上限)				
$\theta = -2.0$	$\theta = -1.0$	$\theta = 0.0$	$\theta = 1.0$	$\theta = 2.0$
1.1/1.6	1.5/2.0	1.5/2.0	1.5/2.0	1.1/1.6

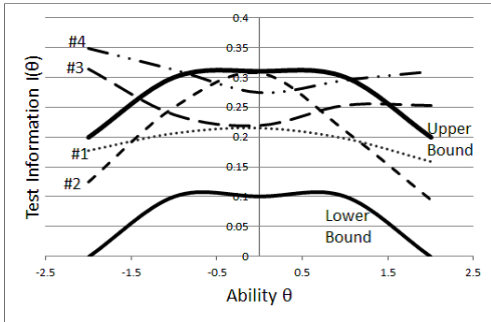


図 2 テスト情報量への上限下限の例
Fig. 2 An image of upper and lower bound for test information.

4 は上限制約を満たさないため、等質でない。

3. 複数等質テスト構成のための最大クリーク問題

複数等質テスト構成手法はこれまでも多数存在する [8]~[10], [12]~[27]. 本節では、提案手法と関係が深い石井ら [9] と Ishii et. al. [10] の手法を紹介する。

3.1 複数等質テスト構成のための最大クリーク問題

著者らの先行研究 [9], [10] では与えられたアイテムバンクから、条件を満たす最大数の複数等質テストを構成する手法を提案した。本手法は複数等質テスト構成を最大クリーク問題 (Maximum clique problem) として解く。具体的には、以下のグラフ中から最大クリークの探索・抽出を行い、複数等質テストを構成する。

頂点: 与えられたアイテムバンクから構成可能な、テストの構成条件を満たすテスト (以降, “可能テスト” と呼ぶ) 全てを頂点とする。

辺: 二つの可能テストが項目重複数条件を満たす場合その二つの頂点 (テスト) 間に辺を引く。

複数等質テスト構成は図 3 のように定式化できる。このグラフ中のクリークは複数等質テストである。クリークとは、完全グラフとも呼ばれ、その中の任意の 2 頂点が接続されている構造を指す。このグラフ中では、任意の頂点はテスト構成条件を満たす。また、クリーク中の任意の 2 頂点は接続されており、項目重複条件を満たしている。したがって、この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、このグラフ中の頂点数が最大のクリークは最大の複数等質テスト群となる。

```

variables
C
( $C \subseteq V$ :  $C$  is a subset of the set of all feasible tests  $V$ )
maximize | $C$ |
subject to
 $\forall v, \forall w \in C, \{v, w\} \in E$ 
* while,  $\{v, w\} \in E$  means the vertices pair of  $v, w$ 
has connected ( $|v \cap w| \geq$  (upper bound of the number of overlapping items)).
    
```

図 3 複数等質テスト構成のための最大クリーク問題
Fig. 3 Clique problem for uniform test construction.

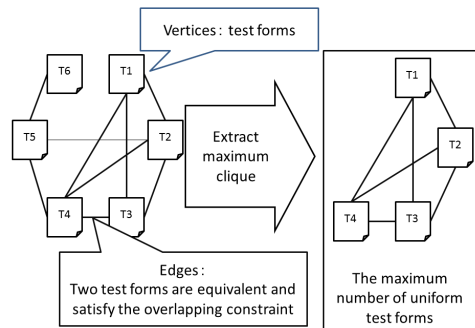


図 4 テスト構成のためのグラフ構造
Fig. 4 Graph structure for uniform test assembly.

結果として複数等質テスト構成は、最大クリーク問題として定式化でき、この最大クリーク問題を厳密に解くことにより、理論的に最大数を保証した複数等質テストを出力する。

例えば、図 4 は前述のように構成したグラフである。図 4 中には六つの頂点 (テスト) と項目重複数条件の満足を表す辺が 9 本ある。例えば、T5 と T6 はそれぞれテスト構成条件を満たす可能テストで、項目重複数条件を満たすため、辺で結ばれている。このグラフ中の最大クリークは {T1, T2, T3, T4} であり、これを抽出すると、与えられたアイテムバンクから構成できる最大数の複数等質テストとなる。

本手法は、e テスティングの黎明期に実施されていたような 80~200 項目中から 2~4 問程度のテストを構成するような条件において、従来手法 ([8], [13], [14] など) と比較し、10~100 倍以上のテストを構成できる。しかし、本最大クリーク問題を厳密に解くためには、頂点数が F の場合 $O(2^{0.1917F})$, $O(F^2)$ の時間・空間計算量を必要とする。これは、最大クリーク問題

を厳密に解くために必要な時間・空間計算量である。また複数等質テスト構成のためのグラフの頂点数 F は可能テスト (与えられたアイテムバンク・テスト構成条件から構成できる全てのテスト) 数であるため、アイテムバンクサイズ n に対して組み合わせ爆発的に増大する。そのため、現在実施されている数百～千以上のアイテムバンクから本手法を用いて複数等質テスト構成を厳密に行うことは困難であった。

3.2 乱択近似法

これらの計算コストの問題を緩和するため、著者らの先行研究 [10] では複数等質テスト構成のための最大クリーク探索を行う近似アルゴリズムを提案した。本手法 (以降、乱択近似法と呼称) は大規模アイテムバンクから従来手法 ([8], [13], [14] など) と比較し、10～1000 倍以上多くのテストを構成可能な手法である。

前節で紹介した手法 [9] (以降、厳密法と呼称) の時間・空間計算量は、頂点数 F のグラフからの最大クリーク探索問題に必要な時間計算量 $O(2^{0.1917F})$ とグラフ構造の保持に必要な空間計算量 $O(F^2)$ である [9]。この計算量の問題は複数等質テスト構成数が増加するとグラフの探索空間も莫大になってしまうことである。そこで乱択近似法では、具体的にはグラフ全体からランダムに部分グラフを抽出し、その部分グラフからの最大クリーク探索を繰り返すことで、グラフ全体の最大クリークを近似的に探索する。

具体的なアルゴリズムをアルゴリズム 1 に示す。本アルゴリズムの時間・空間計算量はそれぞれ、 $O(C_3)$ 、

$O(C_1^2)$ となる。ここで、 C_1 、 C_3 は計算コストに関するパラメータであり、計算機環境に合わせて任意に設定できる。これは、厳密法 [9] の時間計算量 $O(2^{0.1917F})$ と空間計算量 $O(F^2)$ に対して格段に扱いやすい。また、グラフ全域の最大クリークサイズ $|C_{max}|$ が計算コストパラメータ C_1 よりも小さい条件に限り、時間漸近的に数学的に保証可能な最大数のテストを構成できる。

この近似により、現在一般的な規模 (500～2000 項目程度) のアイテムバンクから 10 万程度のテストを構成を可能とした。本手法は、ランダムに作り出した頂点群からクリーク探索を繰り返すことによってより大きなクリークを探索する。しかし、ランダムに作り出された頂点群 (テスト群) は頂点間の連結 (つまりテスト間の項目重複条件の満足) を考慮せずに選択されるため、他の頂点との連結が少ない頂点 (同時に出题可能なテストの少ないテスト) 群も主記憶上に保持され、探索の対象となっていた。

理想的には図 5 のように全域グラフ (色が薄い頂点も含めた全てのグラフ構造) から部分グラフ (濃い色で示されたグラフ構造) が抜き出されれば、全域グラフ中での最大クリークを探索可能である。しかし、本手法は、図 6 のように他の頂点との連結が少ない、最大クリークとはなりにくい頂点を含んだ部分グラフを選び出してしまふことが多く、構成されるテスト数 $|C|$ に対し空間計算量は少なくとも $O(|C|^2)$ 以上となる問題や、選択した部分グラフ以上の大きさの複数等質テストを構成することができないという問題があった。そのため、本手法では現在、現実的に必要とされるテスト数を実環境・実時間で構成することは困難である。

アルゴリズム 1 乱択近似法 [10]

Require: アイテムバンク、テスト構成条件、 C_1 (Step1 で、構成するテスト数)、 C_2 (Step2 で最大クリーク探索にかかる時間)、 C_3 (アルゴリズム全体の計算時間)。

Ensure: 複数等質テスト

Step 1: (可能テストの構成)

与えられたテスト構成条件の項目重複条件以外を用い、それらを満たすテスト可能テストを C_1 個構成する。

Step 2: (複数等質テスト構成のためのグラフ構造の構成)

Step 1 で構成した C_1 個のテストをグラフの頂点とし、全ての 2 頂点 (全てのテストのペア) について項目重複数を数え、それが与えられた項目重複条件以下ならその頂点間に辺を引く。

Step 3: (最大クリーク探索)

Step 2 で構成したグラフより最大クリークを抽出する。ここでは最大クリークの探索を与えられた計算時間 C_2 だけ行う。(計算時間が C_2 を過ぎた場合、計算を打ち切る。) 得られた最大クリークと現状見つけている最大クリークとを比較し、より大きなものを現状見つけている最大クリークとして保存する。

Step 4: (これまで見つかった最大クリークとの比較)

計算開始からの経過時間が C_3 を上回っていれば計算を終了し、現状見つけている最大クリークを出力する。それ以外の場合、Step 1 へ戻る。

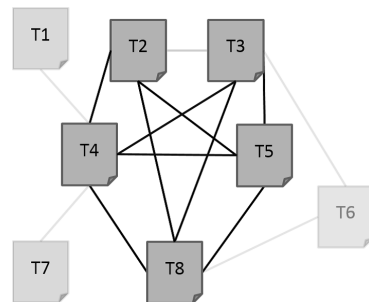


図 5 理想的な部分グラフ選択
Fig. 5 Ideal subgraph selection.

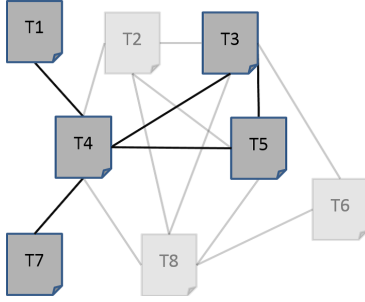


図 6 多くの場合の部分グラフ選択
Fig. 6 Frequent subgraph selection.

4. 複数等質テスト構成のための整数計画問題を用いた最大クリーク問題近似アルゴリズム

そこで本研究では、前述の問題を緩和し、同一の計算機環境・計算時間でより多くのテストを構成可能な手法を提案する。従来手法では、ランダムに部分テストを構成し、その部分グラフからの最大クリーク探索を繰り返していたため、部分グラフ以上の大きさのクリークを探索できない問題や、部分グラフを保持するため構成テスト数を $|C|$ とすると少なくとも $O(|C|^2)$ の空間計算量を必要としていた。本手法では現在探索中のクリークに接続されている頂点のみを、整数計画問題を用いて、探索の必要に応じて構成することにより、部分グラフの構造を主記憶上に保持せずにグラフ全域からの最大クリーク探索を行い、必要とする空間計算量を $O(|C|)$ とする。これにより同一の計算機環境で従来手法よりも多くのテストを構成可能とする。

4.1 整数計画問題を用いた最大クリーク探索

具体的にはアルゴリズム 2 と以下の整数計画問題よりテスト構成を行う。

入力

現在探索中のクリーク C 、項目重複上限数、テスト項目数、テスト構成条件 (テスト情報量上限 $UB(\theta_l)$ ・下限 $LB(\theta_l)$)

変数

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

maximize

$$\sum_{i=1}^n \lambda_i x_i$$

アルゴリズム 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、**AddCount**(add step 中で追加する頂点数)、 α (delete step 中での削除頂点数にかかわるパラメータ)、**Time**(計算時間)。

Ensure: 複数等質テスト。

Step 1: (初期化) 現在探索中のクリーク C 、これまでに見つけた最大のクリーク C_{max} を空集合として初期化 ($C := \phi, C_{max} := \phi$) する。

Step 2: (add step) 頂点の追加回数 $count$ を 0 に初期化する。

Step 3: 可能テスト構成のための整数計画問題 (後述) を解く。解けた場合、解 \tilde{x} とする。解けない場合は Step 6 へ。

Step 4: 解 \tilde{x} で示されるテストを現在探索中のクリーク C へ追加する。このとき、 $|C| > |C_{max}|$ ならば、これまでに見つけた最大のクリーク C_{max} を C で更新する ($C_{max} := C$)。また、頂点の追加回数をインクリメントする ($count++$)。

Step 5: ここで追加回数 $count$ が与えられた **AddCount** よりも小さい場合、Step 3 へ、それ以外の場合 Step 6 へ。

Step 6: (delete step) 頂点の削除回数 $count$ を 0 で初期化する ($count = 0$)。

Step 7: 現在探索中のクリーク C の中からランダムにテスト x を選び、これを C 中から除去する。また削除回数 $count$ をインクリメントする ($count++$)。

Step 8: ここで削除回数 $count$ が与えられた **AddCount** $\times \alpha$ より小さい場合、Step 7 へ、そうでない場合、Step 9 へ。

Step 9: 計算開始からの経過時間が与えられた計算時間 **Time** 未満の場合 Step 2 へ、そうでない場合 Step 10 へ。

Step 10: 最大のクリーク C_{max} を出力する。

subject to

$$\sum_{i=1}^n x_i y_{k,i} \leq (\text{項目重複上限数}) \quad (2)$$

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

$$\sum_{i=1}^n x_i = (\text{テスト項目数}) \quad (3)$$

$$\sum_{i=1}^n I_i(\theta_l) x_i = I(\theta_l) \quad (4)$$

$$LB(\theta_l) \leq I(\theta_l) \leq UB(\theta_l) \quad (5)$$

$$(l = 1, \dots, L)$$

ただし、定数 $y_{k,i}$ は以下のとおりである。

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

* ここで λ_i ($i = 1, 2, \dots, n$) は互いに独立な $[0, 1)$ の連続一様分布からの乱数であり、本問題が解かれるたびにリサンプリングされるものとする。

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によって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。

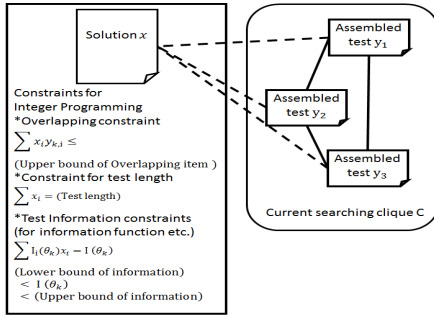


図 7 整数計画問題とグラフ構造の対応
 Fig. 7 Correspondence of the integer programming and the graph structure.

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

add step では現在探索中の等質テスト群：クリーク C へ新しく整数計画問題により構成されたテストを追加することにより、より大きなクリークを探す。

図 7 は本手法で用いられる整数計画問題を模式的に表したものである。この整数計画問題中では、クリーク C 中の全てのテスト：頂点と等質かつ項目重複数の少ない(辺で連結されている)テストを、整数計画問題を解くことにより探索している。整数計画問題中の制約式 (2) は C 中の k 番目のテストと構成中のテスト x との項目重複数を制限している。また、それ以外の制約式 (3)~(5) はいずれも等質条件であり、これらの条件を満たす構成テスト x は等質かつ構成テスト全てと項目重複数の少ないテストである。

また、目的関数 $\sum_{i=1}^n \lambda_i x_i$ は、毎回サンプリングされなおよすランダムな値 λ_i を用いて、毎回できる限り異なるランダムなテストを構成している。 λ_i は x_i を選び出した際の評価関数への重みであり、テスト x に項目 i を選び出す際の優先順位と捉えることができる。この値を毎回ランダムに与えることにより、前述した条件を満たすテストをランダムに構成している。この定式化は Belov [21] 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。

ここで、この整数計画問題を解く目的は、テスト構成条件と項目重複条件を満たすテストをランダムに構成することである。そのため、制約条件は厳密に満たしている必要があるが、目的関数が最大である必要はない。つまり、ここで構成されるテストは可能解であればよく、最適解である必要はない。そのため、IBM

社の CPLEX [28] 等の最適化問題ソルバーを用いる場合、例えば、線形緩和問題との解の上界下界ギャップを大きく設定し、厳密解を見つけなくても計算終了するよう設定しても、アルゴリズム全体の動きは変わらない。

最後に、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している。これにより、探索が局所解(極大クリーク)へ収束することを回避している。

本アルゴリズムの時間計算量は $O(\text{Time})$ 、空間計算量は、内部で使用する整数計画問題の空間計算量が無視できるとすると、 $O(|C|)$ となる。本アルゴリズムも乱択近似法と同様に計算量を任意に設定可能である。しかし、乱択近似法と比較して、提案手法は出力されるテスト数 $|C|$ に対する空間計算量を減じている。乱択近似法の空間計算量は $O(\mathbf{C}_1^2)$ である。この計算量パラメータ \mathbf{C}_1 は常に $|C| \leq \mathbf{C}_1$ が成立する。等号が成立する場合は、部分グラフ全体がクリークである場合であり、出力されるテスト数に対する空間計算量が最も小さくなる場合である。このときの乱択近似法の空間計算量は $O(|C|^2)$ となる。したがって、同数のテストを構成する場合、本アルゴリズムの空間計算量 $O(|C|)$ は乱択近似法の空間計算量 $O(|C|^2)$ よりも小さくなる。そのため、同じ計算機環境でテスト構成する場合、提案アルゴリズムのほうが構成可能なテスト数の上限が大きくなる。

4.2 初期解探索を改善したアルゴリズム

4.1 で紹介したアルゴリズムは、現在探索中のクリークにノードの追加・削除を繰り返し、現在探索中のクリークに似たクリークを次々に探索していく局所探索である。アルゴリズム 2 では初期解として空集合を与え探索を開始しているが、ある程度大きな複数等質テスト群を初期解として与えることで、探索の効率を改善可能である。

アルゴリズム 3 は初期解の探索に乱択近似法 [10] を用いて解探索の効率を向上させたアルゴリズムである。乱択近似法は比較的短時間で多くのテストを含む複数等質テストを構成可能である [10]。これを用いることで素早くある程度良い複数等質テストの初期解を構成できる。

本アルゴリズムは、まず、初期解の探索に乱択近似法を用いて素早くある程度大きな複数等質テストを探索し、その探索された複数等質テスト群を整数計画問題を用いたアルゴリズムにより、より多くのテストを

アルゴリズム 3 初期解探索を改善した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, **AddCount**(add step 中で追加する頂点数), α (delete step 中での削除頂点数にかかわるパラメータ), **Time**(計算時間).

Ensure: 複数等質テスト.

Step 1: (初期化) 現在探索中のクリーク C , これまでに見つけた最大のクリーク C_{max} へ, 乱択近似法 [10] を用いて構成した複数等質テスト群をセットする.

Step 2~10: アルゴリズム 2 と同様.

含む複数等質テストへ改善するアルゴリズムである. これにより, 乱択近似法と比較し, より多くのテストを構成可能である.

本アルゴリズムの時間計算量は $O(\text{Time})$, 空間計算量は, $O(C_2^2)$ となる. これは初期解の探索に乱択近似法を使用しているためである. 初期解探索後の計算量は前節でのアルゴリズムと同様に $O(|C|)$ となる. したがって, 構成されたテスト数に対する本アルゴリズムの空間計算量 $O(|C|)$ は乱択近似法の空間計算量 $O(|C|^2)$ よりも小さくなる. そのため, 同じ計算機環境でテスト構成する場合, 提案アルゴリズムのほうが構成可能なテスト数の上限が大きくなる.

4.3 初期解探索の改善と探索効率

解の探索効率が改善されたかどうかを検証するため, 整数計画問題を用いた最大クリーク探索 (アルゴリズム 2, 以下 RBP 法) と初期解探索を改善した整数計画問題を用いた最大クリーク探索 (アルゴリズム 3, 以下 HybridRBP 法) を比較した. 具体的には, 各手法を用いて同一計算機環境, アイテムバンク, テスト構成条件でテストを構成し, 構成テスト数が計算時間によりどのように変化するかを調べた.

RBP 法・HybridRBP 法で使用するパラメータは, **AddCount** = 1000, α = 10%とした. また HybridRBP 法中の乱択近似法 [10] で用いるパラメータは, $C_1 = 100000$, $C_2 = 3\text{hr}$, $C_3 = 3\text{hr}$ とした.

使用したアイテムバンクは, 実データを模した 500 のシミュレーション項目を含むアイテムバンクであり, 其々の項目は識別力パラメータ a は $\log_2 a \sim N(0, 1^2)$, 困難度パラメータ b は $b \sim N(0, 1^2)$ として発生させた. このアイテムバンクから, 表 2 の情報量条件をもつ 25 項目のテストを構成することをテスト構成条件とした. 本条件は実際に運用された e テスティングにおけるテスト構成条件であり, 現在実施される規模での部分テスト構成を模している. また, それぞれの

表 2 大規模テスト構成のための情報量条件 (下限/上限)
Table 2 Constraint for large scale test assemble.

テスト情報量 (下限/上限)				
$\theta = -2.0$	$\theta = -1.0$	$\theta = 0$	$\theta = 1.0$	$\theta = 2.0$
2/2.4	3.2/3.6	3.2/3.6	3.2/3.6	3.2/3.6

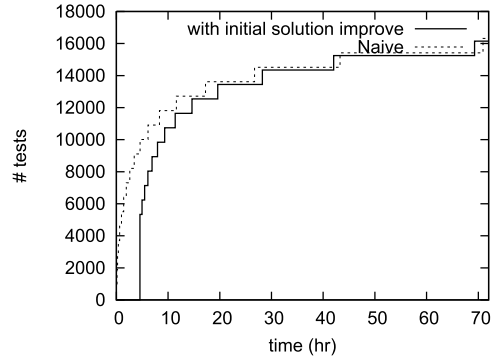


図 8 初期解探索を改善した場合とそれを行わない場合の時間とテスト構成数の関係 (項目重複率 20%)

Fig. 8 Time and number of tests assembled by RBP and HybridRBP (overlap = 20%).

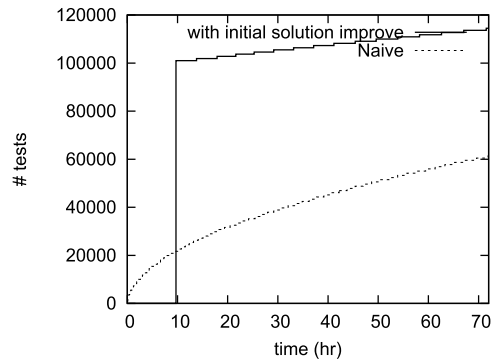


図 9 初期解探索を改善した場合とそれを行わない場合の時間とテスト構成数の関係 (項目重複率 40%)

Fig. 9 Time and number of tests assembled by RBP and HybridRBP (overlap = 40%).

テストで許される項目重複数の上限は 5 項目 (20%), 10 項目 (40%) までとした. また, 両手法で用いられる整数計画問題の解探索には CPLEX [28] を用いた. CPLEX 中では整数計画問題として, デフォルトのオプション (LP 緩和問題との解ギャップ 1%以下で計算打ち切り) で解探索を行った.

結果が図 8, 9 である. 図中の破線が RBP 法によるテスト構成数で実線が HybridRBP 法でのテスト構成数である.

図 8, 図 9 より, RBP 法のテスト構成数は対数的に増加することが分かる. これは, 内部で使用してい

る整数計画問題の規模がテスト構成数に比例して増加している為であり、これにより、この整数計画問題を解くために必要な時間が指数的に増加するためである。

図 9 からは、HybridRBP 法が RBP 法よりも短時間で多くのテストを構成していることが分かる。これは初期解探索を、乱択近似法を用いて計算を行っているため、ある程度大きなテスト集合から探索を開始できているためである。

一方で、図 8 では、HybridRBP 法も RBP 法も最終的に構成できるテスト数が同程度である。つまり、このような条件では乱択近似法 [10] による初期解探索は時間あたりの探索効率を改善しないことが分かる。この違いは、探索空間の大きさが影響していると考えられる。構成できるテスト数が多い条件 (項目重複条件が 40%) では初期解探索により、ある程度大きなテスト集合から探索を行うことが有効であるが、構成できるテスト数が少ない条件 (項目重複条件が 20%) では、初期解探索で得られる初期解の構成テスト数が大きくないため、初期解を空集合から探索した場合でも構成できるテスト数に大きな違いがないと考えられる。

つまり、構成できるテスト数が少ない場合は、初期解探索を行う HybridRBP 法は行わない RBP 法と同等のテスト数を構成可能であり、構成できるテスト数が多い場合、HybridRBP 法は RBP 法より多くのテストを構成可能であると考えられる。

5. 評価実験

最後に提案手法 (HybridRBP) の有効性を示すため二つの実験を行った。具体的には、提案手法 (HybridRBP) と、従来手法 (BST 法 [13], GA 法 [14], BA 法 [8], ExMCP 法 [9], RndMCP 法 [10]) のテスト構成数を様々なテスト構成条件、アイテムバンクで比較した。

5.1 小規模アイテムバンクでのテスト構成数比較

まず、小規模なアイテムバンクからのテスト構成数を比較した。これは、比較を行った ExMCP 法がこれまで行われてきた小規模なアイテムバンクからのテスト構成のみに対応しているためである。

本比較には、三つのシミュレーションアイテムバンクと二つのテスト構成条件を用いた。シミュレーションアイテムバンクは実データを模して発生させた 80, 100, 120 の項目セットであり、それぞれの項目は、識別力パラメータ a は $\log_2 a \sim N(0, 1^2)$, 困難度パラメータ b は $b \sim N(0, 1^2)$ として発生された。

これらのアイテムバンクから、表 4 の条件をもつ 4 項目のテストを構成することをテスト構成条件とした。本条件は実際に使用された、これまで行われていた構成されるテストを小規模な部分テスト構成に分割してテスト構成を行う条件である。小さな部分テストでは、その領域だけで情報量を担保することが難しい。この理由で、テスト全体で情報量を担保する条件でテスト構成を行うことが多い。本条件はこのような想定でのテスト構成を模したものであり、情報量条件の下限が 0.0 等、低い値からも許容されている。条件 1 と比較し、条件 2 は、情報量への上下制限約が緩和されており、構成可能なテスト数が増加する条件である。また、それぞれのテストで許される項目重複数の上限は 0 項目 (0%), 1 項目 (25%), 2 項目 (50%) までとした。

BST 法, GA 法, BA 法へは、目標となるテスト情報量関数を以下の式で与えた。

$$(\text{目標情報量関数}) = \frac{(\text{情報量下限}) + (\text{情報量上限})}{2}. \quad (6)$$

BST 法, GA 法, BA 法, RndMCP 法には計算時間を 6 時間, RndMCP には $C_1 = 100000$, $C_2 = 900s$, $C_3 = 6hr$ を与えた。HybridRBP 法には計算時間を **Time** = 6hr, **AddCount** = 1000, $\alpha = 10\%$, 初期解探索のための乱択近似法 [10] で用いるパラメータは、 $C_1 = 100000$, $C_2 = 900s$, $C_3 = 900s$ とした。BST 法, HybridRBP 法中の整数計画問題の解法手段には CPLEX [28] を採用した。CPLEX 中での解探索はデフォルトのオプション (LP 緩和問題との解ギャップ 1%以下で計算打ち切り) を用いて解探索を行った。

表 3 はそれぞれの手法・条件での構成テスト数をまとめたものである。本実験では、ExMCP 法は、条件 1 では与えた計算時間 6 時間で計算を終了することができなかった。そのため、6 時間の時点で計算を打ち切り、その時点で見つかった最大数のテスト数を表 3 に示した (†)。また、条件 2 では空間計算量の問題で本実験環境では計算が不可能であった (‡)。

BST 法, GA 法, BA 法はテスト構成数を最大化する手法ではないため、テスト構成数を最大化する ExMCP 法, RndMCP 法, 提案手法に比較し、構成テスト数が少なかった。

ExMCP 法は理論的には厳密解を保証するが、厳密な計算には膨大な時間計算量 $O(2^{0.19171F})$ を必要とするため、実用的には計算を打ち切って用いる。ExMCP 法は (条件, アイテムバンクサイズ, OC) = (1, 80, 1) や (1, 100, 0), (1, 100, 1) で他手法よりもテ

表 3 小規模アイテムバンクでの各手法のテスト構成数
Table 3 The numbers for each methods in small item banks.

Item Bank	OC	Constraint 1						Constraint 2					
		BST	GA	BA	ExMCP	RndMCP	HybridRBP	BST	GA	BA	ExMCP	RndMCP	HybridRBP
80	0	7	8	8	9 [†]	9	9	7	8	8	0 [‡]	9	10
	1	20	40	64	100 [†]	101	103	20	74	78	0 [‡]	136	144
	2	20	82	242	1462 [†]	1455	1474	20	347	301	0 [‡]	3036	3135
100	0	8	7	8	10 [†]	10	11	9	9	9	0 [‡]	11	11
	1	25	36	76	131 [†]	132	134	25	88	87	0 [‡]	182	184
	2	25	80	292	2325 [†]	2304	2260	25	312	346	0 [‡]	4839	5053
120	0	9	6	9	11 [†]	11	11	10	10	11	0 [‡]	12	13
	1	30	29	82	152 [†]	150	152	30	92	102	0 [‡]	243	250
	2	30	68	336	2913 [†]	2855	2767	30	269	407	0 [‡]	7190	7578

†: The maximum number of uniform test forms detected in 6 hr.

‡: A memory insufficiency problem interrupted the test construction.

表 4 小規模テスト構成のための情報量条件 (下限/上限)

Table 4 Constraints for small scale test assemble.

ID	Information Function (Lower/Upper Bound)				
	$\theta = -2.0$	$\theta = -1.0$	$\theta = 0$	$\theta = 1.0$	$\theta = 2.0$
1	0.0/0.2	0.1/0.3	0.3/0.5	0.1/0.3	0.0/0.2
2	0.0/0.4	0.1/0.5	0.3/0.7	0.1/0.5	0.0/0.4

ト構成数が少なかった。

RndMCP 法は ExMCP 法を少ない計算コストで動作可能とした手法であり、今回の実験でも、(1, 80, 1) と (1, 100, 1) 以外の条件では ExMCP 法と同じかそれよりも少ないテストを構成した。(1, 80, 1) と (1, 100, 1) の結果では、乱択により、部分グラフを構成しているため、作成された部分グラフがクリーク探索を行いやすい (例えば、ある程度大きなクリークが最大クリークしかないような) 部分グラフであったため、ExMCP 法よりも大きなクリークを発見できたと考えられる。

提案手法は (1, 100, 2) や (1, 120, 2) を除く条件で、従来法と同じかそれ以上のテストを構成できた。条件 1 はテスト構成のためのグラフ構造が比較的小さいため、グラフ全域の接続状態から最大クリークを探索する ExMCP 法や、グラフ全域の大部分を含む部分グラフの接続状態から最大クリークを探索する RndMCP 法が、現在探索中のクリークの近傍の頂点のみを探索する提案手法よりも、同一の計算時間でより多くのテストを構成できることが分かる。

一方、条件 2 では、テスト構成のためのグラフ構造が使用した計算機環境では保持できないほど大きくなったため、ExMCP 法ではメモリ不足により計算が行えず、RndMCP 法でもグラフ全域の非常に小さな部分グラフからしか探索を行えなかったため、提案手

表 5 実アイテムバンクの詳細

Table 5 Details of actual item bank.

Item Bank Size	Parameter a			Parameter b		
	Range	Mean	SD	Range	Mean	SD
978	0.12~3.08	0.43	0.20	-4~4.55	-0.22	1.16

法のほうがより多くのテストを構成できたと考えられる。本実験は 80~100 項目から 4 項目のテストを構成するという小規模なテスト構成を想定しており、現在、実施されているテスト構成では、本実験と比較しテスト構成のためのグラフ構造は遥かに大きくなる。したがって、一般的なテスト構成においては、本手法が最も多くのテストを構成すると考えられる。

5.2 大規模アイテムバンクでのテスト構成数比較

最後に、大規模なアイテムバンクでのテスト構成数の比較を行った。比較を行った手法は、BST 法 [13]、GA 法 [14]、BA 法 [8]、RndMCP 法 [10] である。

本比較には、三つのシミュレーションアイテムバンクと実アイテムバンクを使用した。シミュレーションアイテムバンクは実データを模した項目を 500, 1000, 2000 もち、それぞれの項目は、識別力パラメータ a は $\log_2 a \sim N(0, 1^2)$ 、困難度パラメータ b は $b \sim N(0, 1^2)$ として発生された。実データは実際に運用されていたアイテムバンクであり、実アイテムバンクの詳細は表 5 のとおりである。

これらのアイテムバンクから、表 2 の情報量条件をもつ 25 項目のテストを構成することをテスト構成条件とした。また、それぞれのテストで許される項目重複数の上限は 0 項目 (0%)、5 項目 (20%)、10 項目 (40%) までとした。それぞれの手法には計算時間を 24 時間、RndMCP 法には $C_1 = 100000$ 、 $C_2 = 60\text{min}$ 、 $C_3 = 24\text{hr}$ を与えた。HybridRBP 法中のパラメー

表 6 大規模アイテムバンクでの各種法のテスト構成数の比較

Table 6 The numbers for each methods in large scale item banks.

itembank size	OC	BST	GA	BA	RndMCP	HybridRBP
500	0	12	3	5	10	17
	5	20	23	96	4380	14000
	10	20	21	107	99983	104525
1000	0	21	4	6	17	34
	5	40	17	104	46305	55837
	10	40	19	105	100000	105504
2000	0	53	8	12	32	70
	5	80	22	104	96876	101223
	10	80	23	103	100000	104604
978 (actual)	0	24	9	9	16	35
	5	39	283	371	40814	53080
	10	39	286	381	100000	104604

タは, **AddCount** = 1000, α = 10%, また初期解探索のための乱択近似法 [10] で用いるパラメータは, $C_1 = 100000$, $C_2 = 3hr$, $C_3 = 3hr$ とした. BST 法, GA 法, BA 法へは, 目標となるテスト情報量関数を式 (6) で与えた. BST 法, HybridRBP 法中の整数計画問題の解法手段には CPLEX [28] を採用した. CPLEX 中での解探索はデフォルトのオプション (LP 緩和問題との解ギャップ 1%以下で計算打ち切り) を用いて解探索を行った.

表 6 にそれぞれの手法の各条件での構成テスト数を示した. 項目重複のある条件 (OC = 5, 10) では前節での結果と同様に, BST 法, GA 法, BA 法はテスト構成数を最大化する手法ではないため, テスト構成数を最大化する RndMCP 法, 提案手法に比較し, 構成テスト数が少なかった. 従来手法と比較し, RndMCP 法, 提案手法は数百倍から数千倍のテストを構成できている.

項目重複のない条件 (OC = 0) では BST 法のほうが RndMCP 法よりも多くのテストを構成できている. これは RndMCP 法で探索に用いられる部分グラフがグラフ全体と比較し小さく, 効率的な探索が行えなかったことが原因と考えられる. RndMCP 法が最大クリークを発見するためには, 部分グラフの選択時に最大クリークを含むグラフを部分グラフとして選択しなければならないが, 探索を行うグラフの頂点数が非常に多い場合はそのように部分グラフを選択できる可能性が下がるため, グラフ全体の最大クリークよりも小さなクリークしか見つけることができなかつたと考えられる.

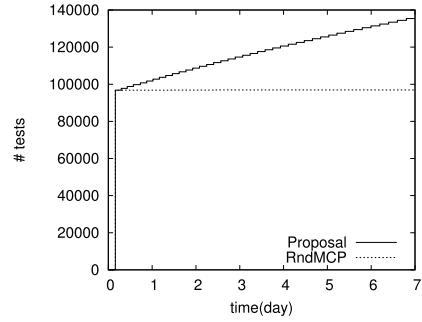


図 10 計算時間 7 日までの提案手法と RndMCP 法のテスト構成数

Fig. 10 The number of assembled tests by RndMCP and Proposal in 7 day.

一方で, 提案手法は探索中のクリークに接続された頂点をグラフ全域から探索できるため, RndMCP 法と比較し, 効率良く探索を行えたと考えられる. その結果, RndMCP 法より多くのテストを同一の計算機環境で構成し, 項目重複のない条件でも BST 法よりも多くのテストを構成したと考えられる.

ただし, 項目重複条件 OC = 10 では RndMCP 法と比較し提案手法のテスト構成数はあまり増加していない. これは, 4.3 での結果からも分かる通り, 構成テスト数が増える条件では, 整数計画問題の制約条件も増加し, これにより整数計画問題によるテスト構成に要する時間が指数的に増えるためである. 4.3 での結果より, より長く計算時間を与えることで提案手法の構成テスト数は徐々に増える. 一方, RndMCP 法は使用した空間計算量パラメータ $C_2 = 100000$ 以上にテストを構成することができないため, 実験で使用した計算機環境ではこれ以上テストを構成することはできない. そのため, RndMCP でのテスト構成数が 10 万程度となる (アイテムバンクサイズ, 項目重複条件) = (1000, 5) や (2000, 5), (2000, 10) では提案手法に計算時間を更に与えることで, RndMCP 法とのテスト構成数の差は大きくなる.

図 10 は (アイテムバンクサイズ, 項目重複条件) = (2000, 5) の条件で計算時間を 168 時間 (7 日間) まで延長したときの RndMCP 法と提案手法の構成テスト数をプロットしたものである. 図中の破線が RndMCP 法, 実線が提案手法を表している. 最終的には, RndMCP 法は 96949 版のテストを, 提案手法は 136403 版のテストを構成した. 図 10 からわかる通り, RndMCP 法はテスト構成数が空間計算量の

問題で 10 万を超えることができないため、計算時間を長くしても構成テスト数を増加することが難しいことが分かる。一方、提案手法は計算時間を延ばすことでテスト構成数を増加することが可能であり、7 日目の時点で 13403 版のテストを構成し、RndMCP 法に比べ 1.4 倍程度のテストを増加できていることが分かる。本研究では実験の都合上、計算時間を延長した結果については一つの条件で示したが、他の RndMCP 法のテスト構成数が 10 万に近い条件についても提案手法に計算時間を更に与えることで、RndMCP 法とのテスト構成数の差は大きくなると考えられる。

6. む す び

本論文では、e テスティングにおいて、複数等質テスト構成のための最大クリーク探索を行う近似アルゴリズムを提案した。本手法はテスト構成のための最大クリーク探索において整数計画問題を用いて探索を行うべきグラフの頂点を逐次的に構成することにより、同一の計算機環境で従来手法よりも多くのテストを構成可能にした。

本手法の有効性を示すため、本論文ではシミュレーション及び実データを用いた実験を行い、これまでの手法中で最も多くのテストを構成可能であることを示した。

ただし、本手法でも理想的に必要とされる数のテストは構成できていない。そのため、今後の課題としてはより多くのテストを構成可能な手法の開発が挙げられる。また、より実用的な手法とするため、現実にくめられている条件を取り込んだ手法を提案していく。例えば、本手法での構成テスト群には、項目の露出 (exposure, 出題回数) に偏りが生じ得る。例えば、ある問題は、構成されたテストの 15% に出題されているが、別の問題は構成されたテストのたった 2% にしか出題されない状況が生じ得る。一般にテスト運用上、このような項目使用回数・露出率の偏りは好ましくない。なぜなら、露出の多い項目は受検者間で共有されやすく、経年による運用において、その項目の信頼性が失われやすくなるためである [11]。そのため、それぞれの項目の露出回数をコントロール可能な手法が必要となる。今後は、この議論についても取り組んでいきたい。

謝辞 本研究の一部は科学研究費補助金 (若手研究 (B), 代表: 石井隆稔) 「e テスティングの信頼性向上の為の項目暴露を考慮した複数等質テスト構成手法の

開発」(15K16260) 及び、(基盤研究 (A), 代表: 植野真臣) 「大規模型 e テスティング・システムとその運営モデルの開発」(15H01772) の助成を受けた。

文 献

- [1] 植野真臣, 永岡慶三, e テスティング, 培風館, 2009.
- [2] 谷澤明紀, 本多康弘, “情報処理技術者試験における e テスティング,” 日本テスト学会 第 12 回大会 発表論文抄録集, pp.54-57, 2014.
- [3] 仁田善雄, 齋藤宣彦, 後藤英司, 高木 康, 石田達樹, 江藤一洋, “医療系大学間共用試験における e テスティング,” 日本テスト学会 第 12 回大会 発表論文抄録集, pp.58-59, 2014.
- [4] ISO, “Iso/iec 23988:2007, information technology – a code of practice for the use of information technology (it) in the delivery of assessments,” 2007.
- [5] JIS, “Jis x7221:2011, jis x7221, アセスメント提供における情報技術 (IT) 利用の規範,” 2011.
- [6] 植野真臣, “e テスティング: 最先端テスト技術,” 信学誌, vol.92, no.12, pp.1017-1021, Dec. 2009.
- [7] 植野真臣, “e テスティング: 先端理論と技術,” 情報システム学会誌, vol.26, no.2, pp.204-217, 2009.
- [8] P. Songmuang and M. Ueno, “Bees algorithm for construction of multiple test forms in e-testing,” IEEE Trans. Learning Technologies, vol.4, pp.209-221, 2011.
- [9] 石井隆稔, ソムアン・ボクボン, 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,” 信学論 (D), vol.J97-D, no.2, pp.270-280, Feb. 2014.
- [10] T. Ishii, P. Songmuang, and M. Ueno, “Maximum clique algorithm and its approximation for uniform test form assembly,” IEEE Trans. Learning Technologies, vol.7, no.1, pp.83-95, 2014.
- [11] H. Wainer and Educational Testing Service, “CATS: Whither and Whence,” Educational Testing Service, vol.2000, no.2, pp.i-15, 2000.
- [12] D.I. Belov and R.D. Armstrong, “A constraint programming approach to extract the maximum number of non-overlapping test forms,” Computational Optimization and Applications, vol.33, pp.319-332, 2006.
- [13] W.J. van der Linden, Liner Models for Optimal Test Design, Springer, 2005.
- [14] K.-T. Sun, Y.-J. Chen, S.-Y. Tsai, and C.-F. Cheng, “Creating irt-based parallel test forms using the genetic algorithm method,” Applied Measurement in Education, vol.2, no.21, pp.141-161, 2008.
- [15] E. Boekkooi-Timminga, “Simultaneous test construction by zero-one programming,” Methodika, vol.1, pp.101-112, 1987.
- [16] F.B. Baker, A.S. Cohen, and B.R. Barmish, “Item characteristics of tests constructed by linear programming,” Applied Psychological Measurement, vol.12, no.2, pp.189-199, 1988.
- [17] E. Boekkooi-Timminga, “The construction of parallel tests from irt-based item banks,” J. Educational

Statistics, vol.15, pp.129–145, 1990.

- [18] J.J. Adema, “Methods and models for the construction of weakly parallel tests,” *Applied Psychological Measurement*, vol.16, no.1, pp.53–63, 1992.
- [19] W.J. van der Linden and J.J. Adema, “Simultaneous assembly of multiple test forms,” *J. Educational Measurement*, vol.35, no.3, pp.185–198, Sept. 1998.
- [20] D.I. Belov and R.D. Armstrong, “Monte carlo test assembly for item pool analysis and extension,” *Applied Psychological Measurement*, vol.29, pp.239–261, 2005.
- [21] D.I. Belov, “Uniform test assembly,” *Psychometrika*, vol.73, no.1, pp.21–38, 2008.
- [22] J.J. Adema, E. Boekkooi-Timminga, and W.J. van der Linden, “Achievement test construction using 0-1 linear programming,” *European Journal of Operational Research*, vol.55, no.1, pp.103–111, 1991.
- [23] J.J. Adema, “Implementations of the branch-and-bound method for test construction problems,” *Methodika*, vol.6, pp.99–117, 1992.
- [24] L. Swanson and M.L. Stocking, “A model and heuristic for solving very large item selection problems,” *Applied Psychological Measurement*, vol.17, no.2, pp.151–166, 1993.
- [25] R.D. Armstrong, D.H. Jones, and C.S. Kuncze, “IRT test assembly using network-flow programming,” *Applied Psychological Measurement*, vol.22, no.3, pp.237–247, 1998.
- [26] R.M. Luecht, “Computer-assisted test assembly using optimization heuristics,” *Applied Psychological Measurement*, vol.22, no.3, pp.224–236, 1998.
- [27] G.-J. Hwang, P.-Y. Yin, and S.-H. Yeh, “A tabu search approach to generating test sheets for multiple assessment criteria,” *IEEE Trans. Education*, vol.49, no.1, pp.88–97, 2006.
- [28] ILOG, “Ilog cplex user’s manual 11.0,” 2007.

(平成 28 年 4 月 8 日受付, 8 月 2 日再受付,
9 月 13 日早期公開)



石井 隆稔

2008 年電気通信大学・情報通信工卒。2011 年電気通信大学大学院情報システム学研究科博士前期課程修了。2014 年同大大大学院情報システム学研究科博士後期課程修了。博士(工学)。2014 年首都大東京システムデザイン学部特任助教。2016 年東京理科大学工学部助教, 現在に至る。e テスティングの研究・開発に従事。



赤倉 貴子 (正員: シニア会員)

神戸大・文化学・社会文化(行動学習)博士退。阪大(人間科学・教育システム工学)・博(人間科学)。神戸大・法卒, 同博士前期・後期了, 法博。四條畷学園短大, 芦屋大を経て 2001 年より東京理科大学工学部経営工学科勤務。2005 年教授。2016 年東京理科大学工学部情報工学科教授, 現在に至る。教育工学, 知財工学に関する研究に従事。本会教育工学研究専門委員会・前委員長。日本教育工学会・前副会長, 現在, IEEE Education Society Japan Chapter Chair, 日本教育工学会評議員, 編集委員, 日本知的資産経営学会理事, 編集委員。1993 年日本教育工学会論文賞, 1996 年日本教育工学会研究奨励賞, 2007 年 ICCE 2007 Best Poster Presentation Award。



植野 真臣 (正員)

1992 年神戸大学大学院教育学研究科修了, 1994 年東京工業大学大学院総合理工学研究科修了。博士(工学), 東京工業大学, 千葉大学, 長岡技術科学大学を経て 2006 年より電気通信大学勤務, 2013 年教授。2016 年電気通信大学大学院情報理工学研究科教授, 現在に至る。電気通信大学大学院情報システム学研究科教授。人工知能, e テスティング, e ラーニング, ベイズ統計の研究に従事。2008 年 The 20th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2008) Best Paper Award。