1D5-OS-11b-7

登録後コース時間割問題の基数制約を用いた制約モデルと SAT ソルバーを用いた解法

Solving Post-Enrollment Course Timetabling using Cardinality Constraint and SAT Solvers

佐古田 淳史[*1] 宋 剛秀[*2] 番原 睦則[*2] 田村 直之[*2]
Atsushi Sakoda Takehide Soh Mutsunori Banbara Naoyuki Tamura

[*1]神戸大学 大学院システム情報学研究科
Graduate School of System Informatics, Kobe University

[*2]神戸大学 情報基盤センター
Information Science and Technology Center, Kobe University

Timetabling is a problem of assigning a set of entities (e.g., tasks, events, people) to the limited number of resources over time, subject to a set of pre-defined constraints. International conferences and competitions of timetabling have been held and it has received increasing attention from both researchers and practitioners. In this paper, we consider an educational timetabling, called *Post-Enrollment Course TimeTabling* (PE-CTT). PE-CTT has been tackled by methods of Operations Research but we propose a method of using SAT solvers which have remarkable improvements in the last decade. We present a constraint model of using cardinality constraints for PE-CTT and several SAT encoding methods so that SAT solvers can handle them. Then, we show binary and linear search to reach optimal values with SAT solvers. In order to evaluate proposed methods, computational experiments have been performed on benchmark instances including ones of international timetabling competitions.

1. はじめに

時間割問題は，人，部屋，講義などの集合に対して，与えられた制約をできるだけ満たすように時間枠を割当てる組合せ最適化問題の一つである．時間割問題にはスポーツ時間割や教育時間割など幅広い実際的な応用があり，その重要性から国際時間割競技会 (**ITC**) や時間割国際会議 (**PATAT**) が開かれている．中でも**カリキュラムベースのコース時間割問題 (CB-CTT)** と**登録後コース時間割問題 (PE-CTT)** が活発に研究されている．本稿では ITC2002 と ITC2007 に使われた PE-CTT[*1] [McCollum 10] を研究の対象とする．

近年大規模な命題論理の充足可能性判定 (SAT) 問題を高速に解くことが可能な SAT ソルバーが実現され，プランニング，スケジューリング，制約充足問題 など，様々な分野への応用が進んでいる [井上 10, 鍋島 10]．更に，コース時間割問題の一種である CB-CTT では局所探索を用いて求められた既知の最良解を SAT 技術を用いた解法が更新するなど良い結果を示している [Banbara 13]．PE-CTT では SAT 技術を用いた従来研究はまだ多くないが，効果的な解法となる可能性がある．

本稿では，基数制約を用いた制約モデルと SAT ソルバーを用いた PE-CTT の解法の提案を行う．提案方法では，まず PE-CTT を多目的重み付き部分 Max-CSP として定式化を行う．この際，すべての制約は基数制約として記述される．次に基数制約を SAT ソルバーで扱うために SAT 符号化を行う．本稿では SAT 符号化法として，Sequential Counter, Totalizer，また組込み基数制約を用いた方法の計 3 つを比較する．上記 3 つの方法を予備実験で比較した結果，SAT 符号化法を用いる場合に比べ，組込み基数制約を用いることが有効であることが分かった．最終的にこの方法と，二分探索および線形探索を用いた最適化方法を PE-CTT に適用した結果，ベンチマーク 116 問中 39 問について，既知の最良解と一致する解を得ることに成功した．

連絡先: 佐古田淳史，神戸大学 大学院システム情報学研究科，兵庫県神戸市灘区六甲台町 1-1 神戸大学情報基盤センター田村研究室，078-803-5364，a-sakoda@stu.kobe-u.ac.jp

*1 http://satt.diegm.uniud.it/projects/pe-ctt/

2. 登録後コース時間割問題

2.1 PE-CTT の定義

登録後コース時間割問題 (Post-Enrollment Course Timetabling; PE-CTT) は，生徒，講義，教室，設備，時間枠に対する**絶対制約**と**強考慮制約**と**弱考慮制約**から構成される．すべての問題は 5 日 × 9 時限の 45 個の時間枠を持つ．ここで，時間割を講義に対する教室と時間枠の割当てとする．時間割が強考慮制約と弱考慮制約に違反した場合は**強ペナルティ**，**弱ペナルティ**と呼ばれる 2 種類のペナルティがそれぞれ課せられる．これらのペナルティに関して**強ペナルティの総和**，**弱ペナルティの総和**の 2 つの目的関数が与えられる．ただし強ペナルティ，弱ペナルティの順で優先順位が付けられる．絶対制約をすべて満たし，かつこれらのペナルティを最小化することが PE-CTT の目的となる．

2.2 問題の入力形式と例

PE-CTT の入力は以下のようなテキストファイルで与えられる．まず，1 行目は講義数，教室数，設備数，生徒数を表す．次に，各教室の収容可能人数が与えられる．

```
3 3 4 3
2 3 2
```

この場合，講義数は 3，教室数は 3，設備数は 4，生徒数は 3 であり，教室 1∼ 教室 3 はそれぞれ収容可能人数が 2 人，3 人，2 人である．

次に，各教室の持つ設備，各講義に必要な設備がそれぞれ 0-1 行列で与えられる．

	設備 1	設備 2	設備 3	設備 4
教室 1	o	o	-	o
教室 2	-	o	-	o
教室 3	o	-	o	-
講義 1	o	o	-	-
講義 2	-	o	-	o
講義 3	o	-	o	-

上部は教室の持つ設備，下部は講義に必要な設備を表す．この表から例えば教室 1 は設備 1，設備 2，設備 4 を持ち，講義 1 は設備 1，設備 2 を必要とすることが分かる．つまり講義 1 は教室 1 で開講可能である．

各生徒が履修している講義も 0-1 行列で与えられる．

	講義 1	講義 2	講義 3
生徒 1	-	o	-
生徒 2	o	o	-
生徒 3	-	o	o

例えば生徒 1 は講義 2 を履修することが分かる．

この他に，各講義の開講可能な時間枠，講義の順序関係も同様の 0-1 行列で与えられる．

2.3 PE-CTT の制約とベンチマーク

PE-CTT の制約は絶対制約 (H1∼H5)，強考慮制約 (M1)，弱考慮制約 (S1∼S3) からなる．それぞれの説明を以下に示す．

(H1) Conflicts: 生徒は同じ時間枠において複数の講義へ出席することはできない．

(H2) Compatibility: 講義は必要な設備を満たし，かつ履修人数以上の収容可能人数を持つ教室で開講されなければならない．

(H3) Occupancy: 同じ時間枠，同じ教室で開講できる講義は高々1 つである．

(H4) Availability: 講義は予め定められた時間枠で開講されなければならない．

(H5) Precedences: 講義は予め定められた順序で開講されなければならない．

(M1) Unscheduled Events: 講義は開講されなければならない．

(S1) Late Events: 各生徒は各日の最後の時間枠の講義へ出席するべきではない．

(S2) Consecutive Events: 各生徒が履修する任意の 3 つの講義は同じ日に連続して開講されるべきではない．

(S3) Isolated Events: 各生徒は 1 日に 1 つの講義にのみ出席するべきではない．

PE-CTT には幾つかのベンチマークセットが存在する．H1, H2, H3 は共通の制約であるが，他の制約はベンチマークセット毎に異なる．本研究で用いるベンチマークセットと制約の表を以下に示す．

Instances	H1	H2	H3	H4	H5	M1	S1	S2	S3
ITC 2007	o	o	o	o	o	o	o	o	o
Lewis&Paechter	o	o	o	-	-	o	-	-	-
ITC 2002	o	o	o	-	-	-	o	o	o
Meta. Network	o	o	o	-	-	-	o	o	o

PE-CTT はこれまで主にオペレーションズリサーチの分野で研究されており，上記のベンチマークセットは主に局所探索などのメタヒューリスティクスによって解かれてきた．これらの従来研究については [Ceschia 12] によくまとまっているためそちらを参照されたい．本研究では，従来研究とは異なり SAT ソルバーを用いた解法を提案する．次節から提案方法について説明を行う．

3. 制約モデル

PE-CTT を 2 種類のペナルティを持つ多目的重み付き部分 Max-CSP として定式化を行う．重み付き部分 Max-CSP は，制約充足問題 (CSP) の拡張問題であり，各変数のドメインは整数の有限部分集合である．絶対制約と考慮制約から構成され，各考慮制約には 1 以上のペナルティが割当てられている．絶対制約を全て満たし，違反する考慮制約のペナルティの総和を最小化する値割当てを求めることが目的となる．

3.1 変数の定義

多目的重み付き部分 Max-CSP として定式化するため，各講義 e，各時間枠 t に対して 0-1 変数 x_{et} を導入する．

$$x_{et} = 1 \iff \text{講義 } e \text{ が時間枠 } t \text{ に行われる．}$$

次に，各講義 e，時間枠 t，教室 r に対して 0-1 変数 x_{etr} を導入する．後述するように，x_{etr} は教室 r で開講可能な講義 e についてのみ定義する．

$$x_{etr} = 1 \iff \begin{array}{l}\text{講義 } e \text{ が時間枠 } t \text{ に}\\ \text{教室 } r \text{ で行われる．}\end{array}$$

変数 x_{et} と x_{etr} をチャネリングする制約は以下になる．このとき，T は全時間枠の集合，E は全講義の集合，R_e は講義 e を開講可能な教室の集合を表す．

$$\bigwedge_{t \in T} \bigwedge_{e \in E} \left(\sum_{r \in R_e} x_{etr} = x_{et} \right)$$

最後に各生徒 s，各時間枠 t について，0-1 変数 y_{st} を導入する．

$$y_{st} = 1 \iff \begin{array}{l}\text{生徒 } s \text{ が時間枠 } t \text{ に}\\ \text{何れかの講義を履修している．}\end{array}$$

変数 x_{et} と変数 y_{st} をチャネリングする制約は以下になる．このとき，S は全生徒の集合，E_s は生徒 s が履修する講義の集合を表す．

$$\bigwedge_{t \in T} \bigwedge_{s \in S} \left(\sum_{e \in E_s} x_{et} = y_{st} \right)$$

3.2 絶対制約の定式化

(H1) Conflicts: 各生徒 $s \in S$，各時間枠 $t \in T$ について，生徒 s が時間枠 t に履修できる講義 $e \in E$ は高々1 である．

$$\sum_{e \in E_s} x_{et} \leq 1$$

(H2) Compatibility: 各講義 $e \in E$，各教室 $r \in R$ について，講義 e は必要な設備を満たし，かつ履修人数以上の収容可能人数を持つ教室 r で開講されなければならない．変数 x_{etr} を，この制約を満たす講義 e と教室 r についてのみ定義する．

(H3) Occupancy: 各時間枠 $t \in T$，各教室 $r \in R$ について，時間枠 t，教室 r で開講できる講義は高々1 である．このとき，E_r は教室 r で開講可能な講義の集合を表す．

$$\sum_{e \in E_r} x_{etr} \leq 1$$

(H4) Availability: 各講義 $e \in E$ について，講義 e は開講可能と予め定められた時間枠の集合 T_e 以外の時間枠に開講されてはいけない．変数 x_{et} を，この制約を満たす講義 e，時間枠 t についてのみ定義する．

(H5) Precedences: 予め開講順序が定められた講義の対 $(e_1, e_2) \in P$，各時間枠 $t \in T$ について，講義 e_1 が時間枠 t に開講されるならば講義 e_2 は時間枠 t' $(t' \leq t)$ に開講できない．

$$x_{e_1 t} + \sum_{t'=0}^{t} x_{e_2 t'} \leq 1$$

3.3 強考慮制約の定式化

(M1) Unscheduled Events: 各講義 $e \in E$ は必ず 1 回開講されなければならない．これに違反した場合，講義毎にペナルティが 1 加算される．

$$\sum_{t \in T} x_{et} = 1 \qquad \text{強ペナルティ} : 1$$

3.4 弱考慮制約の定式化

(S1) Late Events: 各生徒 $s \in S$ は各日の最後の時間枠 $t_l \in \{9, 18, 27, 36, 45\}$ に開講される講義へ出席するべきではない．これに違反した場合生徒 1 名毎に 1 ペナルティ．

$$y_{st_l} = 0 \qquad \text{弱ペナルティ} : 1$$

(S2) Consecutive Events: 各生徒 $s \in S$ が履修する任意の 3 つの講義は同じ日に連続して開講されるべきではない．これに違反した場合は生徒 1 名違反 1 回毎に 1 ペナルティ．ただし，$1 \leq (t \mod 9) \leq 7$ とする．

$$(\ y_{st} + y_{st+1} + y_{st+2} \neq 3\ ) \qquad \text{弱ペナルティ} : 1$$

(S3) Isolated Events: 各生徒 $s \in S$ は 1 日に 1 つの講義にのみ出席するべきではない．これに違反した場合は生徒 1 名毎に 1 ペナルティ．このとき，$1 \leq n \leq 5$ とする．

$$\sum_{t=9(n-1)+1}^{9n} y_{st} \neq 1 \qquad \text{弱ペナルティ} : 1$$

4. 基数制約の SAT 符号化

基数制約とは，0-1 変数の値が 1 になるものに上限または下限を設定する制約である．一般に以下の式で表される．

$$\sum_{i=1}^{n} z_i \ \rhd\ k \qquad \rhd \in \{\ \leq,\ \geq,\ =\ \}$$

今回の制約モデルではすべての制約で基数制約が現れる．

基数制約の SAT 符号化法として，現在までに Binomial, Totalizer [Bailleux 03]，Sequential Counter [Sinz 05] など様々な符号化法が提案されている．本研究では Totalizer，Sequential Counter を用いて基数制約の SAT 符号化を行い，比較を行う．

Sat4j [Le Berre 10] は充足解を求める際の制約として SAT 問題の節に加えて基数制約を直接扱うことができる．本稿では上記の SAT 符号化方法に加えてこの組込み基数制約も用いて比較を行う．組込み基数制約の動作の概要を $\rhd$ が $\geq$ の場合を例にとって説明する．入力は 0-1 変数の集合 $\{z_1, z_2, \ldots, z_n\}$ および整数定数 k であり，以下の式で表される．

$$\sum_{i=1}^{n} z_i \ \geq\ k$$

SAT ソルバーの探索において $\{z_1, z_2, \ldots, z_n\}$ のうち値が 0 に割当たる変数を監視する．値が 0 に割当たる変数の数が $n - k$ 個になれば残りの変数を 1 にするように値を割当てる．値が 0 に割当たる変数の数が $n - k$ 個より大きくなれば Conflict を返し，SAT ソルバーに学習節を生成させる．

5. 最適化

SAT ソルバーを用いて多目的重み付き部分 Max-CSP である PE-CTT を解くためには最適化が必要となる．強考慮制約，弱考慮制約に対しそれぞれ SAT 型の Max-SAT ソルバーに使われるブロック変数を導入する．ブロック変数に対して上限を設ける基数制約を記述し，繰り返し SAT ソルバーを起動する中で基数制約の上限を小さくすることで最適化を実現する．

提案方法では，まず変数 y および弱考慮制約を取り除き，強ペナルティの上限を 0 として SAT 符号化もしくは組込み基数制約を用いて SAT ソルバーによって解を求める．UNSAT であれば，上限を 1 増やす．SAT であれば，以下に示す 3 つの方法の何れかで弱ペナルティの最適化を行う．

二分探索 強考慮制約の最適化の際に得られた解の弱ペナルティの値を計算し，その値を上限，0 を下限として二分探索を行う．

線形探索 同様に得られた解の弱ペナルティの値を計算し，その値を上限，0 を下限として線形探索を行う．上限から値を減らす，または下限から値を増やすという 2 通りの探索方法がある．

6. 性能評価

提案方法は SAT 型制約プログラミングシステム Scarab [Soh 13] を用いて実装を行った．実装したシステムの評価について以下に述べる．

6.1 予備実験: SAT 符号化法の比較

基数制約の SAT 符号化方法 2 つと，組込み基数制約の性能比較を行う予備実験を行った．ベンチマークセットは絶対制約 H1，H2，H3，および強考慮制約 M1 のみで構成される Lewis&Paechter 全 60 問を用いる．ただし，M1 を絶対制約に変更する，すなわち強ペナルティ0 の決定問題となる．比較に用いる方法は以下の 7 通りの組合せになる．

SAT 符号化を用いる方法 (6 通り)
{ Totalizer，Sequential Counter } ×
{ Sat4j(glucose)，Sat4j(SAT)，glucose 2.2 }

組込み基数制約を用いる方法 (1 通り)
Sat4j(SAT) × 組込み基数制約

実験環境は CPU: Intel Xeon 3.16 Ghz，メモリ: 16GB である．制限時間は 600 秒とした．各方法の解けた問題数と計算時間の関係を表したカクタスプロットを図 1 に示す．カクタスプロットにおいては，グラフは右にあるほど多くの問題を解き，下にあるほど高速に求解していることを表す．図 1 の Native Encoder は 組込み基数制約を表す．

予備実験の結果，Sat4j(SAT) × 組込み基数制約が他の方法に比べ高速に多くの問題を解くことが分かった．

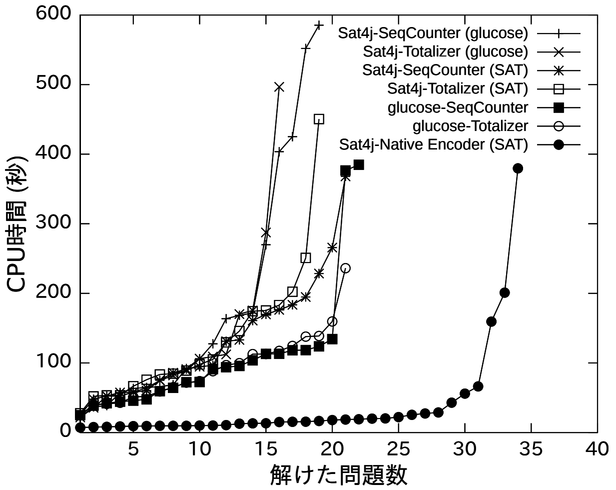

図 1: SAT 符号化法と組込み基数制約の比較

表 1: 下限から開始される線形探索の結果

	問題数	#Valid	#Best
ITC2007	24	17	0
Lewis&Peachter	60	34	34
ITC2002	20	20	0
Meta. Network	12	12	5
計	116	83	39

6.2 本実験: PE-CTT の最適値探索

2.3 節で説明した PE-CTT の 4 つのベンチマークセットに対して，提案方法の評価を行う．予備実験で良い性能を示した Sat4j(SAT) × 組込み基数制約 の方法を用いる．最適化方法は 5 節で説明した二分探索，線形探索 (下限からスタート，上限からスタート) の 3 つである．実験環境は予備実験と同じである．制限時間は各求解あたり 600 秒とした．

今回の実験では 3 つの最適化方法のうち，下限から開始される線形探索が他の 2 つに比べてより高速にペナルティの小さい解を見つけた．下限から開始される線形探索の結果を表 1 に示す．表 1 の 1 列目はベンチマークセットの名前を表す．2 列目は各ベンチマークセットの問題数を表す．3 列目は強ペナルティが 0 の解が求まった問題数を表す．4 列目は既知の最良解に到達した問題数を表す．

強ペナルティが 0 の解は，116 問中 83 問について求めることができた．また，その 83 問中 39 問については既知の最良解を求めることに成功した．ベンチマークセット毎に結果を見ると，ITC2002，ITC2007 については 1 問も既知の最良解を導くことができなかった．この原因としては 3 つ考えられる．

(1) まず弱考慮制約の数が多いことが挙げられる．今回の制約モデルでは S1 が 生徒数 ×5 個，S2 が 生徒数 ×35 個，S3 が 生徒数 ×5 個，つまり弱考慮制約全体で 生徒数 ×45 個の基数制約を生成する．これは生徒数が最も少ない問題で 4500 個，生徒数が最も多い問題で 45000 個になる．またブロック変数もこれと同数用意する必要があり，弱考慮制約のブロック変数に上限を与える基数制約は最大で 45000 個の変数を持つ．

(2) 次に変数 y_{st} の数が多いことも考えられる．変数 y_{st} は 生徒数 × 時間枠数 個生成され，変数 x_{et} とのチャネリング制約も同数必要になる．これも生徒数が最も少ない問題で 4500 個，生徒数が最も多い問題で 45000 個になる．つまり変数 y_{st} および弱考慮制約を加えて問題を解く際は，そうでない場合に加え 9000 ∼ 90000 個の変数および制約，そして 4500 ∼ 45000 個の変数を持つ基数制約が 1 つ増えることになる．これが問題を解くことが難しくなる理由と考えられる．

(3) また，今回は計算時間として 600 秒と比較的短い時間しか与えていないことも原因の一つとして考えられる．

7. まとめ

本稿では，登録後コース時間割問題に対して基数制約を用いた制約モデルと SAT ソルバーを用いた解法を提案した．2 つの SAT 符号化法と，組込み基数制約を用いた 7 つの組合せの比較を行ったところ，Sat4j(SAT) × 組込み基数制約が他の方法に比べ高速に多くの問題を解くことが分かった．この方法と，下限から開始される線形探索を用いた最適化方法を PE-CTT に適用した結果，ベンチマーク 116 問中 39 問について，既知の最良解と一致する解を得ることに成功した．しかし，現在の制約モデルでは弱考慮制約に関する基数制約が非常に多く生成されることも分かった．今後の課題として，弱考慮制約の制約モデルの改良とより長い制限時間での性能評価を行う予定である．

参考文献

[Bailleux 03] Bailleux, O. and Boufkhad, Y.: Efficient CNF Encoding of Boolean Cardinality Constraints, in *Proceedings of the 9th International Joint Conference on Principles and Practice of Constraint Programming (CP 2003), LNCS 2833*, pp. 108–122 (2003)

[Banbara 13] Banbara, M., Soh, T., Tamura, N., Inoue, K., and Schaub, T.: Answer set programming as a modeling language for course timetabling, *Theory and Practice of Logic Programming*, Vol. 13, pp. 783–798 (2013)

[Ceschia 12] Ceschia, S., Gaspero, L. D., and Schaerf, A.: Design, engineering, and experimental analysis of a simulated annealing approach to the post-enrolment course timetabling problem, *Computers & Operations Research*, Vol. 39, No. 7, pp. 1615 – 1624 (2012)

[井上 10] 井上 克巳, 田村 直之: SAT ソルバーの基礎, 人工知能学会誌, Vol. 25, No. 1 (2010).

[Le Berre 10] Le Berre, D. and Parrain, A.: The Sat4j library, release 2.2, *Journal on Satisfiability, Boolean Modeling and Computation*, Vol. 7, pp. 59–64 (2010), system description

[McCollum 10] McCollum, B., Schaerf, A., Paechter, B., McMullan, P., Lewis, R., Parkes, A. J., Gaspero, L. D., Qu, R., and Burke, E. K.: Setting the Research Agenda in Automated Timetabling: The Second International Timetabling Competition, *INFORMS Journal on Computing*, Vol. 22, No. 1, pp. 120–130 (2010)

[鍋島 10] 鍋島 英知: SAT によるプランニングとスケジューリング, 人工知能学会誌, Vol. 25, No. 1 (2010).

[Sinz 05] Sinz, C.: Towards an Optimal CNF Encoding of Boolean Cardinality Constraints, in *Proceedings of the 11th International Joint Conference on Principles and Practice of Constraint Programming (CP 2005), LNCS 3709*, pp. 827–831 (2005)

[Soh 13] Soh, T., Tamura, N., and Banbara, M.: Scarab: A Rapid Prototyping Tool for SAT-Based Constraint Programming Systems, in *SAT*, pp. 429–436 (2013)