制約条件が厳しい時間割編成問題における制約条件の緩和とその評価

柿本 陽平 [1]　高橋 弘毅 [1]　島川 陽一 [2]

[1] 長岡技術科学大学 大学院 工学研究科 情報・経営システム工学専攻
[2] サレジオ工業高等専門学校 情報工学科

概要

本稿では，NP 完全問題の一つである時間割編成問題を制約充足問題として定式化し，単体法を用いて厳密解を求める．その結果から制約がどのように解に影響を与えるのか検証する．具体的には時間割編成において除外しても実行可能な解に影響のない制約条件を除き，計算時間や単体法の反復回数を基準として問題を難しくしている可能性のある制約を見つけ出す．得られた結果から 2 時限連続で行う科目や 1 日に同じ科目を行える回数を制限する制約といった科目の配置に影響を与える制約が問題を実行不可能にしていることがわかった．そこで，問題が実行不可能となるような制約を導入するために，制約の違反を許しその違反の量を目的関数により最小化する方法を提案する．

1　はじめに

NP 完全に属する問題は数多く存在し，代表的なものとして充足可能性問題 (SAT: satisfiability problem) や巡回セールスマン問題 (TSP: traveling salesman problem) 等が挙げられる．このような問題は厳密解を見つけ出すことは難しいが，近似解法を用いて効率的に解を得る手法の研究が盛んにおこなわれている．三神らの研究 [1] では，大規模な SAT に対して原問題の一部の制約を緩和した緩和問題を生成し，原問題の許容解を得るといった試みを行い計算速度向上を図っている．また，質の高い緩和問題を得るために 3 つの手法を用いてそれらを評価している．村野らの研究 [2] では，TSP に対して greedy 法，nearest neghbor 法，saving 法という 3 つの手法で初期解を構成し，得られた初期解に 2-opt 法，3-opt 法を適用し，解を改善させるという試みを行っている．また，規模が大きい問題に対して 2-opt 法，3-opt 法をそのまま適用すると膨大な計算量になってしまうため領域分割法を用いて計算量を削減している．

このように NP 完全問題に対する近似解法の適用事例は数多く存在するが，厳密解法の適用を検討している研究は少ない．そこで本稿では NP 完全問題の一つである時間割編成問題を例にとり，NP 完全問題に対する厳密解法の適用を提案し評価する．

大学や高等専門学校 (以下，高専) における授業の時間割は，それぞれの教育機関ごとに定める制約が多く存在する．ほとんどの教育機関ではそのような制約に違反することのない時間割を教職員が手作業で作成している．この問題を解決するために，時間割を計算機によって自動で作成するという試みがさまざまな手法で行われている．一般にこのような組み合わせ問題も NP 完全問題であり，厳密解を見つけ出すのは難しい．そこで様々なメタヒューリスティクス解法を応用し，このような組み合わせ最適化問題の解決が試みられている [3].

小中学校や高等学校における時間割編成問題では，大学や高専の場合と違い教室の収容人数がほぼ一律であったり，選択科目などが少ないため比較的短時間で複数の解を得ることができるが，一人の教員が複数の科目，コマを担当することがほとんどであり，隔週で行われる科目や複数の指導教員で行う科目なども存在するため解の自由度が低い [5]．一方，大学や高専の時間割においては授業ごとに教室が違う，教室ごとに収容人数が違う，科目・教員・教室が多い，学校ごとに違う制約が必要となるといった要因から，その計算量も膨大なものとなる．

時間割編成問題では，厳密解を見つけ出すのは難しいため近似解法の適用研究が盛んにおこなわれている．筒井らの研究 [4] では，近似解法としてニューロコンピューティングを用いている．さらにルールベースシス

テムとニューロコンピューティングの比較を行っており，ルールベースシステムの開発における知識獲得の困難さに対してニューロコンピューティングのアプローチのほうが有効であることを示している．また，スケジューリング問題においてはルールベースシステムのアプローチにニューロコンピューティングを組み合わせると有効であることも示唆している．

田中らの研究 [3] において時間割編成問題は教室 (部屋)，時間といった資源に授業などのイベントを割り当てる問題であるとし，適用できる手法について触れている．たとえば遺伝的操作やパラメータの設定などが設計者にゆだねられてしまうという弱点を持つ GA とある局所解に陥ってしまう可能性がある局所探索法を組み合わせる手法を紹介している．

堀井らの研究 [7] では，資源制約付きプロジェクトスケジューリング問題 (RCPSP) を用いて

(1) 問題を 2007 年度と 2008 年度に実際に運用されたデータを使い編成する
(2) (1) に加え任意の科目が重複してもいい
(3) (1) に加え教員の勤務日を変える

という 3 つのパターンに分け，その有用性を示している．

近年では計算機の性能向上に伴い，問題を解く計算速度は短くなっている．高専や中規模な大学といった大規模でない問題に限られるが，厳密解法による求解に対する知見も有用であろう．しかし，今日では厳密解法を時間割編成問題へ適用するといった試みはあまりされていない．

本稿では，時間割編成問題を制約充足問題として定式化し，値を 0 と 1 に限定する 0-1 変数を緩和して線形計画問題として解き，0-1 にならない変数に対して分枝限定法を用いて最適解を求める．この時，制約条件の与え方により求解能力が変わることに着目しこの変化の程度を制約条件の難易度として評価する．

対象とする時間割はサレジオ工業高等専門学校の 2013 年度の実際の時間割を用いる．同校は 4 学科 5 学年で学生数は約 900 名，20 クラスで構成される．1 クラスの 1 週間のコマ数は 20 コマで学校全体で 400 コマである．ここに常勤と非常勤約 100 名の教員と科目を割り当てる．ここで本稿で対象としたサレジオ工業高等専門学校の時間割は，一般的な大学の時間割に比べ 1 日に実施するコマ数が午前，午後でそれぞれ 2 コマしかないことや教員数が少ないことで求解領域の自由度が低くなっているという点に注意されたい．

これを制約充足問題として時間割編成問題の定式化を行い，整数計画法を適用する．その時，各科目にかかる制約や制約条件に着目して，初期解の規模を変えていくことで科目の性質を調べ，また実行可能な解に影響を与えない制約条件を除外していき，その結果から評価指標を得る．ここで実行可能な解に影響を与えない制約条件とは，その制約条件を除外して線形計画問題を解いた解が実務上用いることが可能になる条件を指す．具体的な例として「同じ時間帯に 1 人の教員が持てる科目は 1 つまで」というような制約条件を除外すると 1 人の教員が同じ時間帯に異なる教室で授業を行わなければならなくなってしまい，実行不可能となるためこのような制約は実行可能な解に影響を与える制約条件となる．一方，「1 つの科目を 1 日 2 回以上行わない」といった制約条件は除外しても 1 日に同じ科目を行う回数が増えるだけであり，実務上実行自体は可能であるため，このような制約は実行可能な解に影響を与えない制約条件となる．

さらに，満たすことのできない制約条件に対して違反を許すことで緩和解を求める．その際，制約条件には違反を許すためのペナルティ項を付け加え，目的関数により違反量を最小化する．

2　時間割編成問題の定式化

2.1　変数の定義

定数と集合の定義を表 1，2 に示す．また決定変数を表 3 に定義する．x_{stwp} は教員の時間割，y_{sgdwp} はクラスの時間割，z_{srwp} は教室の時間割，k_{gdwp} は授業の実施を決定する変数である．なお，授業は 1 日 4 コマ (90 分 1 コマ) とする．定式化するにあたって高等学校と小学校の時間割を対象とした太田らの研究 [6] を参考にしている．

2.2　高専の時間割の特徴

本稿で扱う高専の時間割は小中高校の時間割とは違い，制約のかかる科目が多い．具体的には科目は一般科目，固定科目，連続科目の 3 種類にグループ化される．ここで固定科目はあらかじめ決められたコマに割り当てられる科目を指し，連続科目は 1，2 時限目もしくは 3，4 時限目に 2 時限連続で割り当てる科目を指す．一般科目は固定科目と連続科目以外の科目を指す．また，各クラスごとに使う教室が多いことや選択科目が多いことも

表 1　定数の定義

定数名	説明
a_{st}	1 : 教員 t が科目 s を受け持つ 0 : 教員 t が科目 s を受け持たない
h_{sr}	1 : 教室 r で科目 s を行う 0 : 教室 r で科目 s を行わない
e_{sgdwp}	1 : 曜日 w 時間 p に 学年 g 学科 d で科目 s を行う 0 : 曜日 w 時間 p に 学年 g 学科 d で科目 s を行わない
l_{gdwp}	1 : 曜日 w 時間 p に学年 g 学科 d で授業をする 0 : 曜日 w 時間 p に学年 g 学科 d で授業をしない
b_{sgd}	科目 s の学年 g 学科 d に必要な単位数
d_s	科目 s に必要な教員数
m_s	科目 s に必要な教室数

表 2　集合の定義

集合名	説明
Sub	科目 s の集合
Sub_1	連続科目の添字集合 ($\mathrm{Sub}_1 \subset \mathrm{Sub}$)
Sub_2	固定科目の添字集合 ($\mathrm{Sub}_2 \subset \mathrm{Sub}$)
Tch	教員 t の集合
Rm	教室 r の集合
Prd	時間 p の集合
Wk	曜日 w の集合
Dpm	学科 d の集合
Grd	学年 g の集合
R_1	教員 t が科目 s を受け持つような s, t の組み合わせ集合 $\{(s,t) \mid a_{st} = 1,\ s \in \mathrm{Sub}, t \in \mathrm{Tch}\}$
R_2	1 週間の単位数 b_{sgd} が 1 以上の時の s, g, d の組み合わせ集合 $\{(s,g,d) \mid b_{sgd} \geq 1,\ s \in \mathrm{Sub}, g \in \mathrm{Grd}, d \in \mathrm{Dpm}\}$
R_3	e_{sgdwp} が 1 となるときの s, g, d, w, p の組み合わせ集合 $\{(s,t,d,p,w) \mid e_{sgdwp} = 1,$ $s \in \mathrm{Sub}, g \in \mathrm{Grd}, d \in \mathrm{Dpm}, p \in \mathrm{Prd}, w \in \mathrm{Wk}\}$
R_4	教室 r で科目 s を行うような s, r の組み合わせ集合 $\{(s,r) \mid h_{sr} = 1,\ s \in \mathrm{Sub}, r \in \mathrm{Rm}\}$

小中高校と異なる点といえる．

2.3　制約条件の定式化

以下に示す CR_i とは i 番目の制約条件を表している．

CR_1　教員 t が受け持っている科目 s は曜日 w 時間 p に 2 以上存在しない :

表 3　決定変数の定義

変数名	説明
x_{stwp}	1 : 曜日 w 時間 p に 教員 t が科目 s を行う 0 : 曜日 w 時間 p に 教員 t が科目 s を行わない
y_{sgdwp}	1 : 曜日 w 時間 p に 学年 g 学科 d で科目 s を行う 0 : 曜日 w 時間 p に 学年 g 学科 d で科目 s を行わない
z_{srwp}	1 : 曜日 w 時間 p に 教室 r で科目 s を行う 0 : 曜日 w 時間 p に 教室 r で科目 s を行わない
k_{gdwp}	1 : 曜日 w 時間 p に 学年 g 学科 d で授業を行う 0 : 曜日 w 時間 p に 学年 g 学科 d で授業を行わない

$$\sum_{s \in \{s \mid (s,t) \in \mathrm{R}_1\}} x_{stwp} \leq 1,\ \forall t, \forall w, \forall p.$$

CR_2　教室 r で行われる科目 s は曜日 w 時間 p に 2 以上存在しない :

$$\sum_{s \in \{s \mid (s,r) \in \mathrm{R}_4\}} z_{srwp} \leq 1,\ \forall r, \forall w, \forall p.$$

CR_3　学年 g 学科 d で科目 s が 1 週間のうち行われる回数は，科目 s の必要単位数と等しい :

$$\sum_{w \in \mathrm{Wk}} \sum_{p \in \mathrm{Prd}} y_{sgdwp} = b_{sgd},\ (s,g,d) \in \mathrm{R}_2.$$

CR_4　曜日 w 時間 p に行われている科目 s は学年 g 学科 d でしか行われない :

$$\sum_{s \in \{s \mid (s,g,d) \in \mathrm{R}_2\}} y_{sgdwp} = k_{gdwp},\ \forall g, \forall d, \forall w, \forall p.$$

CR_5　各時間帯で授業を行っている教員数は，(各クラスの科目) × (その科目に必要な教員の総数) に等しい :

$$\sum_{t \in \{t \mid (s,t) \in \mathrm{R}_1\}} x_{stwp} = \sum_{\substack{d \in \{d \mid (s,g,d) \in \mathrm{R}_2\} \\ g \in \{g \mid (s,g,d) \in \mathrm{R}_2\}}} d_s y_{sgdwp},\ \forall s, \forall w, \forall p.$$

CR_6　各時間帯で授業を行っている教室数は，(各クラスの科目) × (その科目に必要な教室の総数) に等しい :

$$\sum_{r \in \{r \mid (s,r) \in \mathrm{R}_4\}} z_{srwp} = \sum_{\substack{d \in \{d \mid (s,g,d) \in \mathrm{R}_2\} \\ g \in \{g \mid (s,g,d) \in \mathrm{R}_2\}}} m_s y_{sgdwp},\ \forall s, \forall w, \forall p.$$

CR_7　固定科目は決められた時間に割り当てる：

$$y_{sgdwp} = e_{sgdwp},\ s \in \mathrm{Sub}_2.$$

CR_8　授業を行ってはいけない時限 p に授業を入れてはならない：

$$k_{gdwp} = l_{gdwp}.$$

CR_9　連続科目は 2 時間連続で授業を行う：

$$y_{sgdw(2p-1)} = y_{sgdw(2p)},$$
$$p \leq 2, s \in \mathrm{Sub}_1, \forall g, \forall d, \forall w.$$

CR_{10}　ある授業間で空きができないように 2，3 時限目には科目を割り当てる：

$$k_{gdwp} = 1,\ p \in (2, 3),$$
$$(g, d, w, p) \in \{(g, d, w, p) \mid (s, g, d, w, p) \notin \mathrm{R}_3\}.$$

CR_{11}　連続科目と固定科目を除く科目は，1 日 2 回以上行わない：

$$\sum_{p \in \mathrm{Prd}} y_{sgdwp} \leq 1,\ s \notin \mathrm{Sub}_1 \cup \mathrm{Sub}_2,\ \forall g, \forall d, \forall w.$$

CR_{12}　連続科目は午後に割り当てる：

$$y_{sgdw(2p-1)} + y_{sgdw(2p)} = 2,$$
$$p = 2, s \in \mathrm{Sub}_1, \forall g, \forall d, \forall w.$$

3　数値実験

今回数値実験に用いた高専の時間割編成問題の概要を表 4 に示す．ここで教員における常勤と非常勤講師の区別には特に意味がなく，例外とは各学科 5 年生の卒業研究に充てる教員となっている．例外とする理由として卒業研究には各学科の教員が全員割り当てられてしまうため，その時間に他学年において専門科目が行えなくなることを防ぐことがある．クラスは学年と学科の組み合わせで 1 クラスとし，コマは曜日，時限の組み合わせで 1 コマとする．

対象とした高専では 4 学科 5 学年，教員 100 人であり月曜から金曜までの全 400 コマに授業を割り当てる問題となる．決定変数が 124, 820，制約式が 46, 489 となる．本稿では時間割編成問題に単体法を適用する．実行環境を以下に示す．

- OS：Microsoft Windows 7 Professional SP1 64bit
- CPU：Intel®Core™i5 2.67GHz
- メモリ：6GB

なお，本実験では株式会社 NTT データ数理システムの数理計画法パッケージ Numerical Optimizer V15 を用いた．

表 4　数値計算に用いた時間割編成問題の概要

要素	要素の内訳	要素数	合計
科目	一般科目	224	270
	固定科目	26	
	連続科目	20	
教員	常勤	53	100
	非常勤講師	43	
	例外	4	
クラス	学年	5	20
	学科	4	
コマ	曜日	5	20
	時限	4	
教室	教室	60	60

3.1　数値実験の方法

数値実験をするにあたって以下の 3 つの方法を用いる．

方法 1

方法 1 では CR_1 から CR_{12} の下で実行可能な時間割を編成することを目的とする．制約を満たす解が得られないとき，その原因と考えられる制約条件を除外する．なお，除外することでどのような教育機関でも実行不可能となってしまう制約条件 CR_1 から CR_6 は除外対象としない．

方法 2

方法 2 では全コマに対して固定科目が占めるコマの数 (初期解) が変化することで計算時間がどのように変化していくのか検証する．本実験で用いた時間割の空きコマ数は 400 コマであり，すべての科目を固定すると 343 コマを占める．固定する科目を本来の固定科目である科目と連続科目を埋めた状態 (400 コマ中 74 コマ) から 100 コマ，125 コマ， 150 コマ， 175 コマ， 200 コマというように増やしていき計算時間を比較する．このとき，方法 1 で除外された制約条件は含めず計算する．

方法 3

方法 3 では各制約条件が解に与える影響を調べるため制約条件を一つずつ除外し，そのときの計算時間と単体法の反復回数を比較する．このとき，方法 2 と同様に方法 1 で除外した制約条件は含めず計算し，時間割が常に実行可能になるように制約条件を除外していく．

3.1.1　方法 1-結果

方法 1 で示した条件で実際に計算してみたところすべての制約条件の下で時間割を編成することはできなかった．しかし，CR_{12} を除外することで実行可能な解を得ることができた．CR_{12} を除外したのは，通常の科目の配置の組み合わせは 20 通りであるが，連続科目は 2 時限連続で行うため半分の 10 通りになり，午後に割り当てるとさらにその半分の 5 通りになってしまうという点が他の制約条件より問題を難しくしていると考えたためである．計算時間は 212.20 秒であった．

完成した時間割の一例を表 5 に示す．表 5 の各コマは上から科目名，教員名，教室名を表している．

表 5　時間割出力結果の一例

	月曜日	火曜日	水曜日	木曜日	金曜日
1		データ... 教員 D 307		体育実... 教員 G... 校庭	
2	離散数... 教員 A 307			国語 教員 H... 307	情報ネ... 教員 J 307
3	統計解... 教員 B 122,307	数値計... 教員 D 122,307	オペレ... 教員 A 307	情報工...	技術文... 教員 K 307,410
4	プログ... 教員 C, 他 122,307	解析学... 教員 E 307	計算機... 教員 F 307	教員 I, 他 307, 他	解析学... 教員 E 307

表 5 において満たされている制約条件について記述する．まず CR_3 の単位数に関する制約は，このクラスにおける解析学の授業を $b_{sgd} = 2$ と設定し，火曜日 4 時限目と金曜日 4 時限目に科目が配置されているため満たされている．CR_8 のあるコマにおける科目配置を禁止する制約では，このクラスの月曜日 1 時限目に授業を入れないよう設定し，表 5 はこれにしたがっているため満たされている．なお，金曜日 1 時限目に空き時間ができているのは CR_8 に従った結果ではなく，他の制約を満たすように科目を配置した結果である．CR_9 の連続科目に関する制約では，2 時限連続で行うよう設定した情報工学実験 V が木曜日 3，4 時限目に行われているため，これも満たされていることがわかる．CR_{10} の空き時間に関する制約では，火曜日 2 時限目と水曜日 2 時限目で違反しているように見えるが，これは CR_8 にしたがった結果であり，その他のコマにおいて違反していないため満たされている．CR_{11} の 1 日に同科目を 2 回以上配置しないという制約は，必要単位数が 2 と設定されている解析学が各々違う曜日に配置されているため満たされている．

その他の制約は全クラス，教員，教室の時間割を確認したところすべては CR_1 から CR_{11} に従った結果であることがわかった．

3.1.2　方法 2-結果

ここで方法 1 で除外された CR_{12} は計算に含めない．方法 2 で示した条件で実験を行った結果，得られた計算時間のグラフを図 1 に示す．

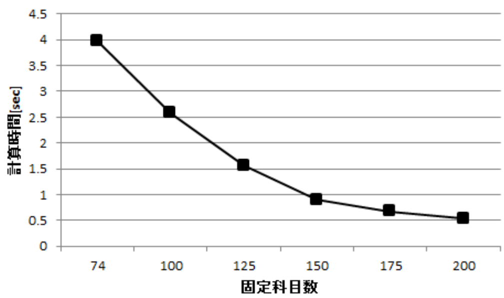

図 1　固定科目数と計算時間の関係

図 1 より固定科目数が増えることで計算時間が短くなることがわかる．また，連続科目を固定せず計算した場合の計算時間は 212.20 秒であり，連続科目を固定した場合，3.98 秒と大幅に計算時間が短くなっている．ここから，連続科目と固定科目を除く科目の中には，連続科目ほど影響を与える科目がなかったことを示唆している．固定科目数 100 から 250 の間では計算時間が緩やかに減少していっていることから固定科目数を増やすことで問題の規模は小さくなることがわかる．

3.1.3　方法 3-結果

ここで除外する制約条件は CR_7 から CR_{11} とし，方法 1 で除外された CR_{12} は計算に含めない．除外した制約条件と単体法の反復回数の関係を示したグラフを図 2 に示す．縦軸は対数目盛で示している．反復回数は小さいほどその制約が強いことを示している．CR_7 は固定科目に関する制約であり，この制約条件を外すことで反復回数が大きくなっていることから固定科目は問題を簡単にしていることがわかる．

図 3 は図 2 において CR_8 から CR_{11} に着目し，描き直した図である．なお，縦軸は対数目盛から線形目盛に変更している．図 3 より CR_8 から CR_{11} の中でも特に CR_{11} を除外したときの反復回数が少ないことから問題

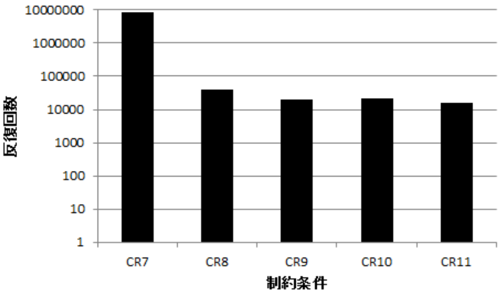

図 2　除外した制約条件と反復回数の関係

に影響を与えやすい制約であることを示唆しており，逆に CR_8 は CR_9，CR_{10}，CR_{11} を除外したときに比べ反復回数が多いことから弱い制約であるといえる．CR_9 と CR_{10} を除外したときの反復回数に違いはあまりなく，この二つの制約は同程度の強さであることが考えられる．除外した 5 個の制約の中で特に反復回数が少なくなった CR_{11} は 1 日に同科目を 2 回以上入れてはならないという制約であった．

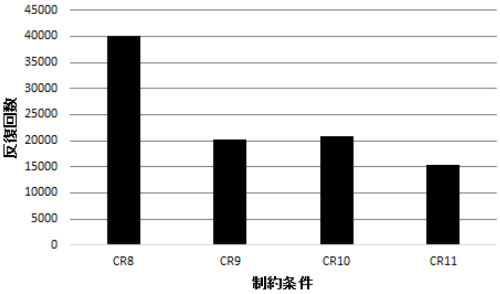

図 3　CR_8 から CR_{11} を除外した場合の反復回数との関係

3.2　制約式の緩和と目的関数の導入

事例として用いた高専の時間割では，実験・実習は実施時間が延長することが多く，多くの担当者は午後に割り当てを希望する．CR_{12} のように科目の配置を強く制限するような制約は実行可能な解を見つける際大きな障害となり，連続科目をすべて午後に割り当てる時間割の編成は不可能であった．そのため前項では特に制約が強い CR_{12} を除くことで CR_1 から CR_{11} を満たす解を得た．そこでこのような制約条件が存在する場合に以下の 2 つの方法を検討する．

1 つ目の方法では，以下のように CR_{12} に整数変数であるペナルティ項 α_{sgdpw} を追加し，問題を緩和する：

$$y_{sgdw(2p-1)} + y_{sgdw(2p)} = 2 + \alpha_{sgdwp},$$
$$-\frac{1}{2}\sum_{s\in \mathrm{Sub}_1}\sum_{g\in \mathrm{Grd}}\sum_{d\in \mathrm{Dpm}}\sum_{w\in \mathrm{Wk}} \alpha_{sgdwp} \leq N,$$
$$p = 2, s \in \mathrm{Sub}_1, \forall g, \forall d, \forall w,$$

ここで $N = |\mathrm{Sub}_1| \times |\mathrm{Grd}| \times |\mathrm{Dpm}| \times |\mathrm{Wk}| - n$ とし n は制約違反の許容量である．n を外部から与えることでどの程度午後に割り当てられない科目を午前に割り当てるかをはじめに与えることにする．

制約条件 CR_{12} は連続科目を「なるべく」午後に割り当てたいという考えから目的関数として扱うことも可能である．そこで 2 つ目の方法では CR_{12} を以下の目的関数で表す：

$$\min.\ F = -\sum_{s\in \mathrm{Sub}_1}\sum_{g\in \mathrm{Grd}}\sum_{d\in \mathrm{Dpm}}\sum_{w\in \mathrm{Wk}} (y_{sgdw(2p-1)} + y_{sgdw(2p)}),$$
$$p = 2, (s, g, d) \in \mathrm{R}_2.$$

図 4 に数値実験の結果を示す．ペナルティ項の導入により，午後に割り当てられた連続科目数は 12 となった．一方，目的関数を導入した場合 17 まで割り当てられた．

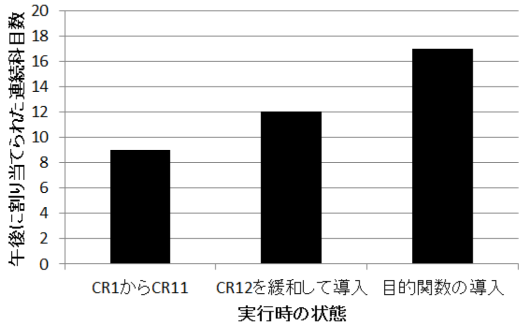

図 4　午後に割り当てられた連続科目数

目的関数の導入により解が改善されることは明白であるが，CR_{12} にペナルティ項を導入したときになぜ解が改善されたのかを考察する必要がある．CR_{12} の違反量を最小化しないということは CR_{12} はいくら違反しても良いということになり，結果としてこの制約条件は無視されると考えていたが，図 4 より CR_{12} の緩和は少なくともこの問題において改善の要因となっていることがわかる．ここからペナルティ項を導入しても元の制約条件に従う解が発見された時点でその方向に解探索が進んでいくものであると考える．

また，n を午後に連続科目が 3 科目以上配置されないよう設定したところ実行不可能になった．これより，連続科目の配置を制限する制約の導入は実行可能な時間割を編成するにあたって課題になるといえる．

4　まとめ

本稿では，時間割編成問題を制約充足問題として定式化し，厳密解法を適用した．その結果，一般の大学に比

べて求解領域の自由度が低い時間割を題材として 11 個の制約の下で厳密解を得ることができた．また，制約がどのように解に影響を与えるのか検証して問題をより簡単にするのは固定科目に関する制約であることがわかった．逆に問題を難しくしている制約は CR_9，CR_{10}，CR_{11}，CR_{12} のような科目の配置を制限する制約であることもわかった．この点に関しては 1 つの制約ではなく複数の制約がお互いの性質を変えている可能性が考えられる．そこで今後の研究では複数の制約を同時に除外していくことでどのような制約が干渉するのか，制約同士の関係性を示していく必要がある．

通常では導入できない 12 個目の制約式を緩和するために違反を許す項を制約条件に加え，目的関数を導入することで最小の違反を許容する解を得ることができた．これを利用することで，教員の要望といったあいまいな制約条件に対応することが可能になった．また教育機関ごとにある完全に満たすことはできないが，可能な限り満たしてほしい制約条件は，目的関数として導入することで実現が可能になった．

今回提案した手法を用いることで「制約条件をなるべく満たす」という表現が可能になるため，時間割編成問題以外の NP 完全問題にも応用が期待できる．

参考文献

[1] 三神直彬，鍋島英知：“大規模 SAT 問題の求解のための緩和法の検討と提案”，人工知能学会全国大会論文集 (CD − ROM)， Vol. 28， pp. ROMBUNNO. 1D5-OS-11B-4IN (2014)

[2] 村野剛教，松本直樹：“巡回セールスマン問題の近似解法の比較 ”，電子情報通信学会技術研究報告， Vol. 102， No. 625， pp. 1-6 (2003)

[3] 田中雅博：“メタヒューリスティック手法による時間割編成の自動化”，システム制御情報学会誌，Vol. 45，No. 12，pp. 725-732 (2001)

[4] 筒井茂義：“クラススケジューリング問題へのニューロコンピューティング適用について”，日本経営工学会誌，Vol. 44，No. 5，pp. 408-415 (1993)

[5] 大坪正和，倉重賢治，亀山嘉正：“中学校における時間割編成への取り組み”，日本経営工学会論文誌，Vol. 57，No. 3，pp. 231-242 (2006)

[6] 太田正和，鈴木敦夫：“時間割自動編成システムの研究”，日本オペレーションズ・リサーチ学会秋季研究発表会アブストラクト集 2006，pp. 58-59 (2006)

[7] Horio, M. and Suzuki, A. : “Application to a University Course Timetabling Problem by a General Project Scheduler”, Lecture Notes in Operations Research, Vol. 8, pp. 266-273 (2008)