

大学の時間割自動編成システムの研究

～ 数理情報学部を対象として～

2005MM039 光部翔太

指導教員：伏見正則

1 はじめに

現在南山大学の授業科目の時間割スケジュールリングは、職員の手によって膨大な時間をかけて行なわれている。しかし、そのような膨大な時間をかけてスケジュールリングされた時間割も不完全なものしかできないという現状がある。この状況を解決する為、今回の研究ではその時間割スケジュールリングをコンピューターに解かせることにより、短い時間で、より理想的な時間割のスケジュールリングを可能にすることを考える。

ただしこの問題を考える場合、大学や瀬戸キャンパス全体での授業科目のスケジュールリングは大規模な問題となるので、問題を解くことが困難になる。その為、今回は問題を縮小化し、数理情報学部の授業科目のみに着目してスケジュールリングを行なう。

また、参考文献 [1] [2] では高等学校、小学校の時間割自動編成システムの研究を行なっており、本研究はこれを参考にしている。

2 問題の説明

2.1 研究範囲

本研究では、南山大学数理情報学部の必修科目と選択必修科目について授業科目のスケジュールリングを行なう。そこで、数理情報学部の必修科目と選択必修科目である4つの科目（学部必修、学部選択必修、学科必修、学科選択必修）を変動させる科目（変動科目）とし、それ以外の授業科目（宗教、英語、体育、人間の尊厳、テーマ、教職、選択）を定置してある科目（定置科目）とする。

今回の問題では、定置科目はすでにスケジュールリングがなされていると考える。

2.2 研究方法

時間割編成問題とは、決められた時間帯・教員・科目・クラス・教室などを考慮し、すべての授業をそれぞれの時間帯および教室に割当て、与えられたすべての条件を満たすような時間割を編成する問題で、スケジュールリング問題の一種である。この問題はNP完全問題として知られ、厳密解を求めることが困難だといわれている。その為、厳密解にこだわらず、許容範囲に解（許容解）があればよいとされ、制約充足問題として定式化される。ここでいう許容解とは実行可能解の一部であり、制約条件によって許容範囲を定めて解を求めることを指す。

また、スケジュールリングは春学期と秋学期で行ない、それぞれの学期において時間割を編成する。

2.3 制約条件について

制約条件には、時間割編成を行なうにあたって必ず条件を満たさなければならない絶対条件と、時間帯や教員の都

合を考慮した考慮条件がある。

さらに、その二つの条件の下での制約を、環境要因・教員要因・授業要因の三つの観点から考えることとする。環境要因は教室に関する制約、教員要因は教員に関する制約、授業要因は科目に関する制約とする。

3 定式化

3.1 記号の定義

3.1.1 定数

S : 変動科目の添字集合

S_1^1 : 学部必修科目の添字集合 $S_1^1 \subset S$

S_2^1 : 学部選択必修科目の添字集合 $S_2^1 \subset S$

S_3^1 : 学科必修科目の添字集合 $S_3^1 \subset S$

S_4^1 : 学科選択必修科目の添字集合 $S_4^1 \subset S$

G : 学年の添字集合

S_g^2 : 学年 g に割り当てる科目の添字集合 $S_g^2 \subset S, g \in G$

S^3 : 二つの学年で同じ時間帯に割り当てる科目の添字集合 $S^3 \subset S$

S_n^4 : 同じ時間帯に割り当てる同一名称の複数科目の添字集合 $S_n^4 \subset S (n = 1, 2, \dots, 7)$

S_g^5 : 学年 g に割り当てる科目で他の変動科目と重複してはいけない科目の添字集合 $S_g^5 \subset S_g^1$

T : 時間帯の添字集合

K : 教員の添字集合

c_t : 時間帯 t に利用することのできる空き教室数

e_s : 科目 s に必要な教員数

$a_{sk} = \begin{cases} 1: \text{教員}k \text{ が科目}s \text{ の授業をする} \\ 0: \text{教員}k \text{ が科目}s \text{ の授業をしない} \end{cases}$

$b_{tk} = \begin{cases} 1: \text{教員}k \text{ が時間帯}t \text{ に都合が悪い} \\ 0: \text{教員}k \text{ の時間帯}t \text{ に都合が悪くない} \end{cases}$

$d_{tg} = \begin{cases} 1: \text{学年}g \text{ の時間帯}t \text{ に定置科目がある} \\ 0: \text{学年}g \text{ の時間帯}t \text{ に定置科目がない} \end{cases}$

$R = \{(s, k) | a_{sk} = 1, s \in S, k \in K\}$

3.1.2 変数

$x_{stg} = \begin{cases} 1: \text{学年}g \text{ の時間帯}t \text{ に科目}s \text{ の授業をする} \\ 0: \text{学年}g \text{ の時間帯}t \text{ に科目}s \text{ の授業をしない} \end{cases}$

$y_{stk} = \begin{cases} 1: \text{教員}k \text{ が時間帯}t \text{ に科目}s \text{ の授業をする} \\ 0: \text{教員}k \text{ が時間帯}t \text{ に科目}s \text{ の授業をしない} \end{cases}$

3.2 目的関数

この問題では制約条件を満たす解を求めればよいので、目的関数は設定しない。

3.3 制約条件

3.3.1 絶対条件

環境要因

(A1) 各時間帯で教室数を超える授業数を割当ててはできない

$$\sum_{g \in G} \sum_{s \in S} x_{stg} \leq c_t, \forall t$$

教員要因

(B1) 教員は決められた科目を授業する

$$\sum_{t \in T} y_{stk} = a_{sk}, \forall s, \forall k$$

(B2) 教員は一つの時間帯に、一つの授業しかできない

$$\sum_{s \in \{s | (s,k) \in R\}} y_{stk} \leq 1, \forall t, \forall k$$

(B3) 教員の都合の悪い時間帯に授業はできない

$$\sum_{s \in \{s | (s,k) \in R\}} y_{stk} b_{tk} = 0, \forall t, \forall k$$

(B4) 各時間帯の各科目における教員数は、全ての学年の教員数と同じになる

$$\sum_{g \in G} x_{stg} e_s = \sum_{k \in K} y_{stk}, \forall t, \forall s \in S \setminus S^3$$

授業要因

(C1) 科目は一週間の時間帯のどこかで必ず割り当てる

$$\sum_{g \in G} \sum_{t \in T} x_{stg} = 1, \forall s \in S \setminus S^3$$

(C2) 学年 g に割り当てる科目集合 S_g^2 に含まれる科目はその学年に割り当てる

$$\sum_{t \in T} x_{stg} = 1, \forall s \in S_g^2, \forall g$$

(C3) 二つの学年で同じ時間帯に割り当てる科目集合 S^3 に含まれる科目を同じ時間帯に割り当てる

この式は (B4), (C1) の場合分けの式により

$$\sum_{g \in G} x_{stg} e_s = 2 \sum_{k \in K} y_{stk}, \forall t, \forall s \in S^3$$

$$\sum_{g \in G} \sum_{t \in T} x_{stg} = 2, \forall s \in S^3$$

(C4) 変動科目と定置科目の重複はしてはいけない

$$x_{stg} + d_{tg} \leq 1, \forall s, \forall t, \forall g$$

(C5) 同じ時間帯に割り当てる同一名称の複数科目集合 S_n^4 に含まれる科目 s_1 と s_2 を同じ時間に割り当てる

$$x_{s_1 t g} = x_{s_2 t g}, \forall s_1 \in S_n^4, \forall s_2 \in S_n^4, \forall t, \forall g$$

(C6) 学年 g に割り当てる科目で他の変動科目と重複してはいけない科目の添字集合 S_g^5 に含まれる科目は他の変動科目と重複してはいけない

$$\sum_{s \in S_g^5} x_{stg} \leq 1, \forall t, \forall g$$

4 実行結果

最適化ソフトウェアの CPLEX により問題を解き、学年 g が 1~4 の場合の実行結果 x_{stg} を得たので、その例を表 1 に示す。変数は x_{stg} が 9,288 個、 y_{stk} が 85,914 個、制約式は全体で 26,541 個、実行時間は約 7 秒であった。(CPU は Intel Pentium4 2.66GHz, メモリ 512MB)

表 1 第 1 学年 ($g = 1$) の場合の実行結果 x_{stg}

科目	時間帯																	
	mon1	mon2	mon3	mon4	tue1	tue2	tue3	tue4	wen1	wen2	thu1	thu2	thu3	thu4	fri1	fri2	fri3	fri4
S1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
S2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S3	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
S4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
S5	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S6	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S7	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S11	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
S12	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S13	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
S15	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
S17	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
S19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S25	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
S26	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
S27	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
S28	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
S29	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
S30	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
S31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

∴ S129 まで全て 0 である

表 1 では 1-0 の表になっているので、とても見にくい表となってしまっている。そのため、CPLEX によって得た 1-0 の解を Excel に書き込み、その 1-0 の表を Excel の VBA (Visual Basic for Applications) によって見やすい表にすることを試みた。その結果として、表 2 を得た。

表 2 第 1 学年 ($g = 1$) の場合の実行結果 x_{stg} の略表

mon1	S12					
mon2	S2					
mon3	S5	S6	S7			
mon4	S13	S15	S17			
tue1	S1					
tue2	S3					
tue3	S11					
tue4						
wen1						
wen2						
thu1	S25	S26	S27	S28	S29	S30
thu2						
thu3						
thu4						
fri1	S4					
fri2	S14	S16	S18			
fri3						
fri4						

5 おわりに

現段階では考慮条件、教職・再履修に関する制約を考慮していない必要最低限の制約条件のために、実用的なスケジュールはできていない。また、今回のシミュレーションは 2008 年度の科目・教員のデータを元に、集まらないデータは任意に定めた模擬のデータである。

その為、今後の課題としてさらに現実に基づいたデータに対する制約を考え、計算結果の表も見やすくするなど改善をしていく必要がある。

参考文献

- [1] 太田 正和, 鈴木敦夫: 時間割自動編成システムの研究, 日本 OR 学会 2006 年秋季研究発表アブストラクト集 (2006), pp. 58-59.
- [2] 太田 正和: 時間割自動編成システムの研究, 南山大学大学院数情報研究科修士論文 (2007).