

Short Report

小学校の時間割作成支援プログラム

Excel アドイン機能を利用した時間割作成

荒谷洋輔¹¹ 秋田県立大学 システム科学技術学部経営システム工学科

各学校の教務担当の方は、時間割作成に多大な労力を割いている。学校には、毎日に関わる授業時間作成だけでなく、試験時間の時間割作成や答案返却の時間割作成、範囲を広げれば試験監督割当作成もある。また、1年を通した通常授業時間割以外にも、季節に合わせた時間割なども必要となってくる。さらには、小学校高学年・中学校・高等学校になると、各科目の授業を専門の先生が担当するため、さらに時間割作成の難易度が増してくる。より具体的には、教員の同時限の授業重複を避けながら、期間内に必要な授業をクラスごとに割り当てる必要がある。また、教員の出講可能な日時を考慮する必要もある。本稿では、時間割作成の例として、小学6年生の時間割のモデルを提示することを目的とする。本稿で提案するモデルは、先行研究のモデルをより簡略化して、現場の人が扱いやすいものとした。具体的には、与えられた時間割の制約条件下で、割当問題をベースとする数理モデルを作成し、Numerical Optimizer による数値計算を行う。さらには、Excel アドイン機能を利用して時間割を Excel 表の形で出力する。

キーワード：時間割作成，割当問題，数理モデル，Numerical Optimizer

はじめに

研究背景

各学校の教務担当の方は、時間割作成に多大な労力を割いている。学校には、毎日に関わる授業時間作成だけでなく、試験時間の時間割作成や答案返却の時間割作成、範囲を広げれば試験監督割当作成もある。また、1年を通した通常授業時間割以外にも、季節に合わせた時間割（春の運動会、夏の水泳授業、修学旅行、秋の文化祭など）が必要となることも多い。さらには、小学校高学年・中学校・高等学校になると、各科目の授業を専門の先生が担当するため、さらに時間割作成の難易度が増してくる。より具体的には、教員の同時限の授業重複を避けながら、期間内に必要な授業をクラスごとに割り当てる必要がある。また、教員の出講可能な日時を考慮するだけでなく、利用可能な教室に割り当てることや、移動教室がある科目、連続した時限で行う科目、科目の順序に関する制約も守らなければならない。（以上、池上敦子、呉偉（2020）より一部文章を引用）

研究目的

本稿では、時間割作成の例として、小学6年生の時間割のモデルを提示することを目的とする。時間割作成の先行研究は様々なものがあるが、比較的最新の研究として池上敦子、呉偉（2020）がある。上述のモデルでは、クラス基本制約、教科制約、教員制約、教室制約などを満たすために、かなり複雑なものとなっている。

本稿で提案するモデルは、前述のモデルをより簡略化して、現場の人が扱いやすいものとした。具体的には、与えられた時間割の制約条件下で、割当問題をベースとする数理モデルを作成する。作成した数理モデルはSIMPLE 言語を用いてプログラミングし、Numerical Optimizer による数値計算を行う。さらには、Excel アドイン機能を利用して時間割を Excel 表の形で出力する。

本稿より、このモデルの基本的な考え方である「割当問題」と最適化専用ソフト Numerical Optimizer（最新版はNuorium Optimizer）の幅広い用途をより多くの読者に知って欲しいという筆者の願いも込められている。

時間割モデルの提案

また、割当問題はマス目を埋める問題に置き換えることができます。

研究対象

小学校の学習指導要領では、小学6年生における1年間の各科目の総授業時間数が以下のように定められている。それを1週間当たりに概算したものが、一番右の列の部分である。

表1: 各科目の授業時間数

科目	総授業時間数 (1年間)	1週間当たりの 授業時間数
国語	175	5
社会	105	3
算数	175	5
理科	105	3
音楽	50	1.5
図画工作	50	1.5
家庭	55	2
体育	90	3
道徳	35	1
特別活動	35	1
英語	70	2
総合	70	2

今回は最も標準的なものを考える。以上の一週間当たりの授業時間数制約の他に、研究背景・目的にあるさまざまな制約を満たすように時間割を作成すること（以下の表を埋めること）が今後の目標である。

表2: 小学校6年生の時間割表 (空欄)

	月	火	水	木	金
1					
2					
3					
4					
5					
6					

割当問題について

本研究のモデルを構築する上で重要となる考え方として「割当問題」がある。

=====

(Numerical Optimizer V22 チュートリアルからの引用による割当問題の紹介):

割当問題とは、集合 A の要素を集合 B の要素のどれに割り当てるかを決定する問題です。

(例) $A = \{a_1, a_2, a_3, a_4\}$, $B = \{b_1, b_2, b_3, b_4\}$

$$a_1 \rightarrow b_1, \quad a_2 \rightarrow b_4, \quad a_3 \rightarrow b_2, \quad a_4 \rightarrow b_3$$

表3: 割当問題のマス目を埋める問題への変換

	a_1	a_2	a_3	a_4
b_1	○			
b_2			○	
b_3				○
b_4		○		

例題: 4×4 のマス目があります。このマスに石を置きたいのですが、全ての縦横に関して石の数が2つになるように配置します。さてどのように置けば良いでしょうか。

	1	2	3	4
1	x_{11}	x_{12}	x_{13}	x_{14}
2	x_{21}	x_{22}	x_{23}	x_{24}
3	x_{31}	x_{32}	x_{33}	x_{34}
4	x_{41}	x_{42}	x_{43}	x_{44}

(解) この問題を定式化するためには、0-1 変数を用いる必要があります。この問題の場合、上図の各マスに対して0-1 変数を対応させます。

変数: $x_{11}, x_{12}, x_{13}, x_{14}, \dots, x_{44} \in \{0, 1\}$

目的関数: なし

制約条件:

$$x_{11} + x_{12} + x_{13} + x_{14} = 2$$

(1行目を横に見たときに石を2つ置く)

$$x_{21} + x_{22} + x_{23} + x_{24} = 2$$

$$x_{31} + x_{32} + x_{33} + x_{34} = 2$$

$$x_{41} + x_{42} + x_{43} + x_{44} = 2$$

$$x_{11} + x_{21} + x_{31} + x_{41} = 2$$

(1列目を横に見たときに石を2つ置く)

$$x_{12} + x_{22} + x_{32} + x_{42} = 2$$

$$x_{13} + x_{23} + x_{33} + x_{43} = 2$$

$$x_{14} + x_{24} + x_{34} + x_{44} = 2$$

このモデルを Numerical Optimizer で実行すると、最後に以下の表示がされ、この例題の答を確認できます。

0	1	0	1
0	0	1	1
1	1	0	0
1	0	1	0

どの縦横にも二箇所ずつ石が置いてある(1と表示されている)のがわかります。(割当問題の引用紹介ここまで)

=====

モデルの定式化と計算結果

モデルの定式化

前述の割当問題の考え方を応用して、新たな時間割モデルを提案する。数理モデルはSIMPLE言語を用いてプログラミングし、Numerical Optimizerによる数値計算を行う。計算環境はIntel^(R)Core^(TM)i7-8750H CPU@4.21GHz、メモリは8.00GBを使用した。

```
//添え字と集合

//科目の集合
Set Subjects="国語 算数 社会 理科 音楽 図工 家庭 体育 道徳 英語 総合 特別";

//時間の集合
Set Jikan="1 .. 6";
Element i(set=Jikan);

//午前の時間の集合
Set Gozen(superSet = Jikan);
Gozen ="1 .. 4";
Element k(set = Gozen);

//午後の時間の集合
Set Gogo(superSet = Jikan);
Gogo ="5 6";
Element l(set = Gogo);

//曜日の集合
Set Day="1 .. 5";
Element j(set=Day);

//変数
DiscreteVariable x(dom=Subjects, index=(i,j));

//目的関数：今回はなし

//一週間の授業時間制約
sum(Boolean(x[k,j] == "国語"), (k,j)) == 5;
sum(Boolean(x[k,j] == "算数"), (k,j)) == 5;
sum(Boolean(x[i,j] == "社会"), (i,j)) == 3;
sum(Boolean(x[i,j] == "理科"), (i,j)) == 3;
sum(Boolean(x[i,j] == "音楽"), (i,j)) == 1;
sum(Boolean(x[i,j] == "図工"), (i,j)) == 2;
sum(Boolean(x[i,j] == "家庭"), (i,j)) == 2;
sum(Boolean(x[i,j] == "体育"), (i,j)) == 3;
sum(Boolean(x[i,j] == "道徳"), (i,j)) == 1;
sum(Boolean(x[i,j] == "英語"), (i,j)) == 2;
sum(Boolean(x[i,j] == "総合"), (i,j)) == 2;
sum(Boolean(x[i,j] == "特別"), (i,j)) == 1;
```

```
//国語と算数は午前中に授業をする制約
sum(Boolean(x[k,j] == "国語"), k) == 1;
sum(Boolean(x[k,j] == "算数"), k) == 1;

//社会、理科、体育は各曜日に1時間までの制約
sum(Boolean(x[i,j] == "社会"), i) <= 1;
sum(Boolean(x[i,j] == "理科"), i) <= 1;
sum(Boolean(x[i,j] == "体育"), i) <= 1;

//各授業の曜日&時間&科目の指定
x["1,1"] == "国語";
x["2,3"] == "体育";
x["6,5"] == "英語";

//求解
solve();

//結果出力
x.val.print();
```

変数・制約式の意味：

1. 科目の集合：小学校6年生に学習する12科目を定義する。
2. 時間の集合：各曜日は6時間目までであるので、そのように定義する。同時に、午前の時間（Gozen, 4時間目まで）と午後の時間（Gogo, 5・6時間目）は、それぞれ時間の集合（Jikan）の部分集合という形で定義する。
3. 曜日の集合：月曜日から金曜日まで5列分定義している。
4. 変数：取りうる値は1, 0とする。該当する科目が入るとき1, そうでないとき0をとることを表す。
5. 目的関数：今回はモデルの簡略化のため「なし」とする。
6. 一週間の授業時間制約：前述の設定した値を制約式とする。ただし、今回の場合は端数のある科目について「音楽を1, 図画工作を2」と設定した。
7. 国語と算数は午前中に授業をする制約：
これは、教育現場の声から、児童にとって負荷が大きい科目は午前中に終わらせるための制約である。
8. 社会、理科、体育は各曜日に1時間までの制約：これは、各曜日の科目のバランスを考えた制約である。
9. 各授業の曜日&時間&科目の指定：
「x[” , ”] == ” ”」で「時間」と「曜日」と「科目」を指定する。

今回は例として、「月曜日1時間目は国語」「水曜日2時間目は体育」「金曜日6時間目は英語」を入力している。必要に応じて適宜入力可能である。

この所は、外部からの非常勤講師・体育・家庭科など、使用時間・場所に制限があるものについて、先に設定しておくのと良いと筆者は考えている。

モデルの計算結果

前節で作成したプログラムを Numerical Optimizer で実装した結果は以下の通りである。

```
[Result]
STATUS    NORMAL
TERMINATE REASON    All constraints satisfied
ITERATION COUNT    17
PENALTY    0
RANDOM SEED    1
ELAPSED TIME(sec.)    0.00
x[1,1] = "国語"
x[1,2] = "国語"
x[1,3] = "算数"
x[1,4] = "音楽"
x[1,5] = "社会"
x[2,1] = "特別"
x[2,2] = "体育"
x[2,3] = "体育"
x[2,4] = "理科"
x[2,5] = "算数"
x[3,1] = "算数"
x[3,2] = "算数"
x[3,3] = "国語"
x[3,4] = "算数"
x[3,5] = "体育"
x[4,1] = "家庭"
x[4,2] = "家庭"
x[4,3] = "理科"
x[4,4] = "国語"
x[4,5] = "国語"
x[5,1] = "理科"
x[5,2] = "図工"
x[5,3] = "道徳"
x[5,4] = "社会"
x[5,5] = "総合"
x[6,1] = "社会"
x[6,2] = "英語"
x[6,3] = "総合"
x[6,4] = "図工"
x[6,5] = "英語"
```

結果の出力時間は 0.00 秒である。よって、このモデルと同規模の問題は実用上全く問題なく時間割作成可能であることが分かる。

計算結果の Excel 出力

最後に、Numerical Optimizer で計算した結果を Excel 出力する。出力には Numerical Optimizer Excel アドイン機能を用いる。詳しくは参考文献 Numerical Optimizer V22 Excel アドインマニュアルを参照。

- (1) まずは、空欄の時間割表を Excel で作っておく。
- (2) リボンの「アドイン」→「MSI Numerical Optimizer」→「モデル」またはシートの右クリックメニュー→「MSI Numerical Optimizer」→「モデル選択」で表示されるファイル選択ダイアログボックスで elementary __ timetable.smp (Numerical Optimizer ランタイム版の場合は elementary __ timetable.exe) を選択する。

最適化モデルが選択されると定数一覧、変数一覧、目的関数といったモデル情報が読み込まれ、ブックの左側に下図のような設定画面が表示される。尚、設定画面のメニューはリボンと同等の内容である。

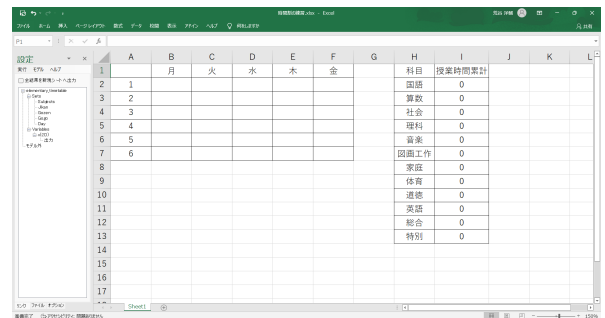


図1 : 時間割表 (空欄)

- (3) 「x(2D)」の表全体を選択した状態で設定画面のモデルツリーの「x(2D)」の右クリックメニュー→「選択範囲を出力」をクリックする。
「x(2D)」に出力データとして選択範囲が登録され、モデルツリーの「x(2D)」は太字になる。
- (4) 実行ボタンをクリックして、計算結果を Excel に出力する。

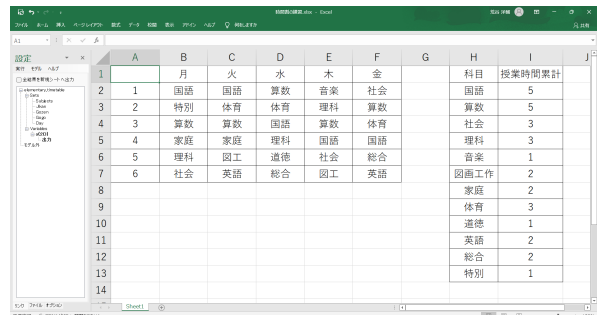


図2 : 時間割表 (計算結果を出力したもの)

補足として、Excel 内にある countif 関数を利用すると、授業時間数の合計が正しいかどうかを確認できる。

おわりに

本稿では、小学校における時間割作成問題に取り組んだ。基本的な考え方は、池上敦子 (2018) のナース・スケジューリング問題やバイト先でのシフト・スケジューリング問題と同じ「割当問題」である。今回は一週間分の表を Numerical Optimizer で計算して、その結果を Excel 出力することに成功した。計算時間は 1 秒もかかっていないので、これと同様の規模の問題ならば実用上全く問題ないと言える。

今後に向けて、現場の声を反映させて、もっと実用に即した制約を入れることも検討課題である。また、今回は簡略化のため目的関数を「なし」と設定したが、適切な目的関数の導入もこれからの研究課題である。

参考文献

池上敦子 (2018). 『ナース・スケジューリング—問題把握とモデリング—』. 近代科学社.

Numerical Optimizer V22 マニュアル

https://www.msi.co.jp/nuopt/download/manual/module/manual_NumericalOptimizer_V22.zip.

池上敦子, 呉偉 (2020). 「学校時間割作成」『オペレーションズ・リサーチ：経営の科学』65(3), 148–156.

〔 令和 4 年 6 月 27 日受付
令和 4 年 8 月 22 日受理 〕

Class schedule Making Support Program of Elementary School Class schedule making utilizing add-in features

Yousuke Araya ¹

¹ *Department of Management Science and Engineering, Faculty of Systems Science and Technology, Akita Prefectural University*

Abstract: Persons in charge of school affairs of each school spend much time on class schedule making. A school has to work not only on the school hour planning for every day but also test schedule making and time schedule making for answer sheet return, and moreover they need to work on proctor allocation planning. Moreover, class schedules for each season are required in addition to normal class schedule for the year. Moreover, since specialized teacher are in charge of each subject in upper grades in elementary schools, junior high schools and high schools, class schedule making become even more difficult. More specifically, it becomes necessary to assign required lessons to every class within the specific period of time while avoiding class duplication for the same time periods of the instructors. Moreover, it is necessary to consider the date and time when the instructors are available for the classes. In this paper, the authors aim at presenting a model of class schedules for sixth graders as an example of class schedule making. The model proposed in this paper was obtained simplifying the model of the past study and making it easy to treat. Specifically, the model creates a mathematical model based on allocation problems and performs numerical computation by Numerical Optimizer under the constraint of the given class schedule. Moreover, it outputs class schedules as an Excel list using Excel add-in features.

Keywords: Class schedule making, allocation problem, mathematical model, Numerical Optimizer