

タイトル	解析的問題解決の教育におけるソルバーの利用に関する一考察：“解く”から“書く”ことに焦点を当てた教育に向けて
著者	上田，雅幸；Ueda, Masayuki
引用	北海学園大学経営論集，14(3)：31-42
発行日	2016-12-25

解析的問題解決の教育における ソルバーの利用に関する一考察

— “解く” から “書く” ことに焦点を当てた教育に向けて —

上 田 雅 幸

1. はじめに

数理モデルに基づく意思決定支援システムをマーケティングや医療などの分野に利用することの有効性を示す研究がいくつもあるにもかかわらず、そうしたシステムの導入率は低いままである (Lilien et al., 2004)。垣花 (1997) は、「(Operations Research/ Management Science 等の) 数理的手法が多くの人々に理解されて普及するためには、そのユーザを育てること、その考え方・問題の捉え方を教育する必要がある」と指摘している。

著者は、数理的手法を用いた意思決定 (以下、解析的問題解決) の教育において、Excel のソルバー機能 (以下、Excel ソルバー) の利用が有効であると考え¹⁾。今日、Excel のソルバー機能を活用することで、十分な数理的知識を持たない学生であっても解析的問題解決を学習できるように工夫されたテキストが多数出版されている (例えば、[5], [7], [8], [9], [12] 等を参照されたし)。Excel ソルバーの利用を想定したテキストの多くは、“数値計算にかかる負担の軽減” に焦点を当てている。例えば大野他 (2014) は、“Excel には、ソルバーというアドインソフトで、目的関数と制約条件を入力するとその解を計算してくれる強力な道具がある”と説明している。数値計算にかかる負担を軽減するだけであれば、最適化問題を解くための別のソフト

ウェア (最適化ソルバー) の利用も考えられる。近年コンピュータ及び数理最適化のアルゴリズムの性能は向上を続けており、最適化ソルバーを利用することも有効な手段の1つである。

扱う問題の規模に応じて使い分けるのであれば、小規模な問題には Excel ソルバー、大規模な問題には最適化ソルバーを用いることになる²⁾。これに対して、本研究では、扱える問題の規模以外に、“十分な数理的知識を持たない学生にとっての可読性”に着目する。本研究では、Excel ソルバーと最適化ソルバーを比較検討しながら、数理モデルを“解く”ことから問題状況を“書く”(整理すること)に焦点を当てた解析的問題解決の教育について考察する。

本論文は以下のように構成される。第2章では、Excel ソルバー及び最適化ソルバーの利用について整理を行う。第3章では、“数理モデルを解くこと”と“問題状況を整理すること”に焦点を当てながら、解析的問題解決の教育における Excel ソルバー及び最適化ソルバーの利用について考察する。第4章は結論である。

2. Excel ソルバー 及び最適化ソルバーの利用

①Excel ソルバー

今日、Excel ソルバーの利用を想定した解析的問題解決の教育向けのテキストが多数出版されている。また、多くの高等教育機関で Excel を操作できる環境を学生に与えているなど、Excel ソルバーを利用しやすい環境が整っている。

Excel ソルバーの利用を想定したテキストの多くは、まず「生産計画問題」のような問題が例題として与えられ、当該問題に対する数式モデルの作成、グラフやシンプレックス法を用いた解法が解説されている。その後で、Excel ソルバーによる解法の解説がなされ、練習問題を繰り返し解いていくかたちとなっている。

生産計画問題³⁾：

原材料 P, Q, R を用いて 2 種類の製品 A, B を生産している企業を考える。製品 A, B を 1 単位生産するのに原材料 P を 2 kg と 1 kg, Q を 1 kg と 1 kg, R を 1 kg と 3 kg 必要とする。原材料 P, Q, R の利用可能量はそれぞれ 1.6 トン, 1 トン, 2.4 トンとする。製品 A, B が 1 単位当たり 3 万円, 4 万円の利益をあげるとき、総利益を最大にする製品 A, B の生産量を求めよ。

Excel ソルバーを用いて問題を解くためには、当該問題に対して表計算モデルを作成する必要がある。表計算モデルに対してパラメータを設定して Excel ソルバーを実行すると、その解を自動的に求めることができる。図 2 は、「生産計画問題」に対して表計算モデルを作成し、Excel ソルバーを実行した結果である。表計算モデルに関する詳細な説明は省略するが、当該問題に対する数理モデル（図 1）と比較すると、その対応関係は理解しやすい。網掛けのセルは、決定変数及び目的関数を表すセルである。最適解が、製品 A を 300 単位、製品 B を 700 単位生産したときに総利益 3700 万円となることが分かる。

$$\begin{aligned} \max. & \quad 3X_A + 4X_B \\ \text{s.t.} & \quad 2X_A + X_B \leq 1600 \\ & \quad X_A + X_B \leq 1000 \\ & \quad X_A + 3X_B \leq 2400 \\ & \quad X_A, X_B \geq 0 \end{aligned}$$

※ X_A ：製品 A, X_B ：製品 B

図 1 「生産計画問題」に対する数理モデル

著者が調べた限りでは、[5], [7], [8], [9], [12] 等の多くのテキストにおいて、“表計算モデルは、代数的な数式モデルを表形式に書き換えたもの”と捉えられている。すなわち、表計算モデルは、数式モデルを作成した後に、Excel ソルバーを実行するために作成するものと認識されている。

	製品 A	製品 B	実際の利用量 (kg)	利用可能量 (kg)
原材料 P	2	1	1300	1600
原材料 Q	1	1	1000	1000
原材料 R	1	3	2400	2400
利益	3	4		
生産数 (単位)	300	700	総利益 (万円)	3700

図 2 「生産計画問題」に対する表計算モデル

②最適化ソルバー

今日, “コンピュータ及び数理最適化アルゴリズムの性能の向上”, “SCIP, GLPK, lp_solve といった非商用ソルバーの存在” など, 最適化ソルバーを利用しやすい環境が整ってきた⁴⁾。このような最適化ソルバーの一般的な利用方法としては, 解決すべき問題の入力ファイルを用意し, ソルバーを実行する方法が挙げられる。入力ファイルの形式としては, LP 形式, MPS 形式などがある。図 3, 図 4 は, それぞれ「生産計画問題」を LP 形式(正確には CPLEX-LP 形式)と MPS 形式により記述したものである⁵⁾。可読性の観点からすると, LP 形式の方が優れている。目的関数や制約条件の記述を見ると, 数理モデルを構成する目的関数や制約条件を(ほぼ)数式のまま記述していることが分かる⁶⁾。目的関数や制約条件を数式にして 1 つずつ記述しなければならないことを考慮すると, 大規模な問題を扱う場合には, モデリング言語を利用するなどの工夫が必要である。商用のモデリング言語としては, AMPL や GAMS などがある。図 5 は, 非商用のモデリング言語である ZIMPL により「生産計画問題」を記述したものである。

ソルバーの実行は, Windows のコマンドプロンプトを起動したうえで, コマンドをタイプして行う方法が一般的である⁷⁾。図 6 は, SCIP > read production.lp
SCIP > optimize
SCIP > write solution production.sol
とタイプし, SCIP を実行した結果である。最適化ソルバーを利用することにより, 複雑な計算を行うことなく「生産計画問題」に対する最適解が得られていることがわかる。

```
maximize ¥最大化問題
3 x1 + 4 x2 ¥目的関数
subject to ¥制約条件
Material_P: 2 x1 + 1 x2 <= 1600
Material_Q: 1 x1 + 1 x2 <= 1000
Material_R: 1 x1 + 3 x2 <= 2400
end ¥ファイルの終わり
```

※“¥”以降はコメントを表す。

図 3 「生産計画問題」に対する LP ファイル

```
NAME      production
ROWS
  N      Obj
  L      Ma_P
  L      Ma_Q
  L      Ma_R
COLUMNS
  x1      Obj      -3      Ma_P      2
  x1      Ma_Q      1      Ma_R      1
  x2      Obj      -4      Ma_P      1
  x2      Ma_Q      1      Ma_R      3
RHS
  RHS      Ma_P      1600      Ma_Q      1000
  RHS      Ma_R      2400
BOUNDS
  PL Bound      x1
  PL Bound      x2
ENDATA
```

図 4 「生産計画問題」に対する MPS ファイル

```

set I:={1..3}; #Material_P, Q, R
set J:={1..2}; #Product_A, Product_B
set A:=I+ {4}; # {Materials + Profit}
param a[A*J]:=
    |1,2|
    |1|2,1|
    |2|1,1|
    |3|1,3|
    |4|3,4;
#上限
param b[I]:=<1>1600,<2>1000,<3>2400;
var x[J] >=0; #決定変数
maximize obj: #目的関数
    sum<j> in J:a[4,j]*x[j];
subto Materials: forall <i> in I do #制約条件
    sum<j> in J:a[i,j]*x[j]<=b[i];

```

※“#”以降はコメントを表す。

図 5 ZIMPL による「生産計画問題」の記述

```

solution status: optimal solution found
objective value:      3700
x1          300      (obj:3)
x2          700      (obj:4)

```

図 6 SCIP の実行結果

3. Excel ソルバー 及び 最適化ソルバーを利用した解析的問題解決の教育

前章で述べたように、Excel ソルバーや最適化ソルバーを利用することにより、複雑な計算を行うことなく解決すべき問題に対する最適解を求めることができる。数値計算にかかる負担を軽減できることは、特に文系学生のような十分な数理的知識を持たない学生の

解析的問題解決の学習へのモチベーションを維持させるのに有効である。この点に限れば、Excel ソルバーと最適化ソルバーのどちらを利用しても良いと思われる⁸⁾。

ここで問題となるのは、Excel ソルバーと最適化ソルバーどちらの利用においても、数理モデルが前もって作成されていると想定されていることである。十分な数理的知識を持っている学生であれば問題はない。しかしながら、解決すべき問題状況を代数的な数式で整理すること（すなわち、数理モデルの作成）は、十分な数理的知識を持たない学生にとって大きな負担となる。そのため、Excel ソルバーや最適化ソルバーを利用するだけでは、解析的問題解決の教育に対するモチベーションを維持することは難しい。（解析的問題解決の教育向けのテキストの多くがそうであるのだが、）「生産計画問題」のように数理モデルを構成する目的関数や制約条件式のパラメータとなる情報が整理されたかたちで提供されている問題であれば、前章のような手続により、Excel ソルバーや最適化ソルバーを実行できるであろう。これに対して、本章では、（少し）複雑な問題及び実際問題への取り組みを例に、解析的問題解決の教育における Excel ソルバー及び最適化ソルバーの利用について考察する。

Stevens et al. (2004) は、自身の講義における経験から、多くの学生が数理モデルの作成を間違えやすい問題として、以下の問題を挙げている。

①書店における問題：

書店 A は、配送センター B より、新しい小説のソフトカバー版 40 冊とハードカバー版 65 冊を受け取っている。しかしながら、その小説への予約状況から、書店 A は、明日の開店までに各タイプ（ソフトカバー版、ハードカバー版）について少なくとも 80 冊を必要としている。さらに、書店 A は、予約分の

160 冊分だけでなく、明日の販売に備えてできるだけ多くを入手したいと考えている。書店 A の“明日までにできるだけ多くの本を用意したい”という要求に対して、配送センター B は最善を尽くしたいと考えている。配送センター B には、(本の注文を箱単位で申し込む)「急送サービス」がある。急送サービスでは、“ソフトカバー箱”と“ハードカバー箱”の 2 種類が利用可能である。ソフトカバー箱には、6 冊のソフトカバー版の小説が入っている。ハードカバー箱には、5 冊のハードカバー版の小説と 2 冊のソフトカバー版の小説が入っている。急送サービスでは、ソフトカバー箱とハードカバー箱を合わせて 10 箱まで利用可能である。ただし、ハードカバー箱は 7 箱までしか利用できない。こうした状況のなかで、配送センター B が書店 A の要求にできるだけ応えるためには、どのように発送すればよいか。

$$\begin{aligned} \max. & 6X_S + 7X_H + 105 \\ \text{s.t.} & 6X_S + 2X_H + 40 \geq 80 \\ & 5X_H + 65 \geq 80 \\ & X_H \leq 7 \\ & X_S + X_H \leq 10 \\ & X_S, X_H \geq 0 \end{aligned}$$

※ X_S : ソフトカバー箱, X_H : ハードカバー箱

図 7 「書店における問題」に対する数理モデル

Stevens et al. (2004) は、文章問題から数理モデルを作成することの難しさに焦点を当てている。Stevens et al. (2004) は、問題状況を数字や変数を用いずに整理したものを明示的に作成したうえで、それをルールに従って数理モデルへと変換する方法を提案している。Stevens et al. (2004) によれば、当該方法を用いずに「書店における問題」に取り組んだ学生の多くが、1 本目の制約条件の左辺を“ $6X_S + 2X_S + 40$ ”のように間違えると指摘している。

図 8, 図 9 は、それぞれ「書店における問

```

maximize ¥最大化
6 x1 + 7 x2   ¥x1:SC_Box, x2:H_Box
subject to   ¥制約条件
SoftCover: 6 x1 + 2 x2 >= 40
HardCover: 5 x2 >= 15
HC_Box: 1 x1 <= 7 >= 0
SC_Box_and_HC_Box: 1 x1 + 1 x2 <= 10
SC_Box: 1 x2 >= 0
end
    
```

図 8 「書店における問題」に対する LP ファイル

```

solution status: optimal solution found
objective value:      65
x1          5      (obj:6)
x2          5      (obj:7)
    
```

図 9 SCIP の実行結果

題」を LP 形式により記述し、SCIP を実行した結果である⁹⁾。ソフトカバー箱とハードカバー箱をそれぞれ 5 箱ずつ発送することにより、既に届けられていた 105 冊に 65 冊(ソフトカバー版 30 冊とハードカバー版 35 冊)が加わり、合計 170 冊の小説が書店に届けられることがわかる。

最適化ソルバーを利用するためには、数理モデルを(ほぼ)そのまま記述すればよい。数理モデルが正しく作成された後であれば、問題なく解を導き出すことができる。LP ファイルが正しく数理モデルを表せているかを確認することはそれほど難しくはない。しかしながら、(数理モデルを理解するための工夫がなされているわけではないので,)“LP ファイルが、数理モデル化の対象となった解決すべき問題の状況を正しく表せているか”を確認することは容易ではない。すなわち、正しく数理モデルが作成されていなかった場

合、十分な数理的知識を持たない学生が LP ファイルを見て問題状況の整理の間違いに気付くことは難しい。このことから、最適化ソルバーを利用するだけでは、Stevens et al. (2004) が指摘する間違いを避けられない恐れがある。

これに対して、図 10 は、「書店における問題」を表計算モデルとして整理をし、Excel ソルバーを実行した結果である¹⁰⁾。最適化ソルバー同様、数理モデルが正しく作成された後であれば、問題なく解を導き出すことができる。表計算モデルが数理モデルを正しく表せているかを確認することも、それほど難しくはない。最適化ソルバーと異なるのは、表計算モデルの記述内容が学生にとって（比較的）理解しやすいということである。今日多くの高等教育機関において Excel を操作できる環境が提供されていることもあり、多くの学生は、物事を表形式に整理することに慣れ親しんでいる。表計算モデル（図 10）を確認することにより、“HC_Box (SC_Box) を（仮に）1 箱発送すると書店の HC 版 (SC 版) の小説は何冊になるか”などの確認は、（比較的）容易に行うことができる。Excel ソルバーを実行したときに意図しない結果となった場合、表計算モデル上でその原因を探することもできる。このことから、Excel ソルバーを利用した場合、Stevens et al. (2004) が指摘する間違いを避けられる可能性がある。

上記のような利点は、数理モデルを“解く”

ために数理モデルを表形式に書き換える作業としてではなく、解決すべき問題の状況を整理する（“書く”）手段として表計算モデルの作成を教育できることを示唆している。しかしながら、現状、表計算モデルの作成に関しては、Conway et al. (1997) において、信頼性、更新性に配慮した表計算モデル作成のガイドラインが考察されている程度である。

以下では、実際問題への取り組みを例に、“問題状況を整理する”ことを目的に表計算モデルの作成を図った状況を説明する。

②講義におけるグループ分け問題：

問題状況：H 大学経営学部において、「インターンシップ」を履修する学生は、実地研修前に「事前研究」と呼ばれる講義を履修しなければならない。当該講義のなかで、学生はグループに分けられ、割当てられた「分析企業」に関する業界・企業研究を行う。グループ分けの際には、さまざまな学生間での意見交換が図れるように、いくつかの「条件」を設けている。図 11 は、「事前研究」を履修している 30 名の学生に関して、「所属ゼミ」（12 ゼミ：ゼミ O からゼミ Z）、「実地研修先」（12 社：企業 A～企業 L）、「1 回目の分析企業」（10 社：企業 1～企業 10）を整理したものである。この状態から 2 回目のグループ分けを行う場合、グループ割け案を作成せよ。
条件：①学生が割当てられる「分析企業」は 1 つである、②「分析企業」には学生が 3 名

	SC Box	HC Box	入荷済	入手数(冊)	必要数(冊)
SC 版	6	2	40	80	80
HC 版	0	5	65	90	80
注文数(箱)	5	5		合計(冊)	170

注文数の上限	注文数	上限
HC Box	5	7
SC Box + HC Box	10	10

図 10 「書店における問題」に対する表計算モデル

学生	所属ゼミ	実地研修先	分析企業①
学生 1	T	C	5
学生 2	T	H	4
学生 3	Q	F	7
学生 4	Z	K	9
学生 5	P	K	6
学生 6	S	D	1
学生 7	R	C	1
...	...	...	...
学生 24	Y	I	2
学生 25	X	E	2
学生 26	R	H	7
学生 27	W	D	6
学生 28	W	A	7
学生 29	Z	C	4
学生 30	X	C	8

図 11 「グループ分け問題」の問題状況

ずつ割当てられる, ③「分析企業」には「実地研修先」が異なる学生が割当てられる, ④「分析企業」には「所属ゼミ」の異なる学生が割当てられる, ⑤ 2 回目の「分析企業」は 1 回目とは異なる学生により構成される, ⑥ 学生は 1 回目と 2 回目とで異なる「分析企業」が割り当てられる。

図 12 は, 「グループ分け問題」に対して作成した表計算モデル (の一部) である。縦軸には学生を並べ, 「ゼミ」, 「実地研修先」, 「分析企業①」(1 回目の分析企業) がまとめられている。網掛けのセルは, “各学生に関して, 当該分析企業を割当てる (1) / 割当てない (0) を表す決定変数である。表計算モデルでは, 後続作業において Excel ソルバーへのパラメータの設定作業を効率的に行えるように, “学生が割当てられた分析企業の数”, “分析

学生	分析企業②										ゼミ	実地研修先	分析企業①	行和	上限
	1	2	3	4	5	6	7	8	9	10					
学生 1	0	0	0	0	0	0	0	0	0	0	T	C	5	0	1
学生 2	0	0	0	0	0	0	0	0	0	0	T	H	4	0	1
学生 3	0	0	0	0	0	0	0	0	0	0	Q	F	7	0	1
学生 4	0	0	0	0	0	0	0	0	0	0	Z	K	9	0	1
学生 5	0	0	0	0	0	0	0	0	0	0	P	K	6	0	1
学生 6	0	0	0	0	0	0	0	0	0	0	S	D	1	0	1
学生 7	0	0	0	0	0	0	0	0	0	0	R	C	1	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
学生 24	0	0	0	0	0	0	0	0	0	0	Y	I	2	0	1
学生 25	0	0	0	0	0	0	0	0	0	0	X	E	2	0	1
学生 26	0	0	0	0	0	0	0	0	0	0	R	H	7	0	1
学生 27	0	0	0	0	0	0	0	0	0	0	W	D	6	0	1
学生 28	0	0	0	0	0	0	0	0	0	0	W	A	7	0	1
学生 29	0	0	0	0	0	0	0	0	0	0	Z	C	4	0	1
学生 30	0	0	0	0	0	0	0	0	0	0	X	C	8	0	1
列和	0	0	0	0	0	0	0	0	0	0					
上限	3	3	3	3	3	3	3	3	3	3					

図 12 「グループ分け問題」の表計算モデル①

企業に割当てられた学生の数”をカウントするためのセル（“行和”，“列和”）が用意されている。また，条件⑤の設定を効率的に行うために，図 13 のような表があらかじめ作成された。当該表は，1 回目に割当てられた分析企業ごとに，学生が 2 回目にどの分析企業を割当てられるかをカウントするものである。図 13 から，例えば分析企業①において企業 1 を割当てられた学生は，分析企業②では企業 3，企業 4，企業 7 を割当てられていること（すなわち，1 回目に分析企業 1 を割当てられた学生は，2 回目の割当てにおいて異なる分析企業が割当てられていること）を確認できる。条件③，④に関しても，「所属ゼミ」，「実地研修先」に着目した同様の表が作成された。

著者は，担当するゼミのなかで，「グループ分け問題」に対する表計算モデルの作成を課題として課したことがある。結論からいうと，この取組みは失敗した。テキストの問題を通じて割当て問題などを解いたことのある学生は，表の上に“決定変数”，“行和”，“列和”，“上限”を追加した。また，Excel の抽出機能

を活用しながら，「所属ゼミ」，「実地研修先」，「分析企業①」ごとに学生を分けるなど，当該問題の制約条件を記述するための準備を進めることができた。しかしながら，「グループ分け問題」は，決定変数が Excel ソルバーで扱える変数の上限を超えてしまう。Excel ソルバーを実行することができないため，1 つ 1 つの条件を正しく表の上に記述できているかを確認できないまま，表計算モデルの作成を進めなければならなかった。その結果，完成された表計算モデルは，いくつかの条件が見落とされたものとなっていた。“Excel ソルバーを実行し，その結果を確認できない状況”では，問題状況を表形式に整理する作業がスムーズに行えない恐れがある。

上記のような場合，Excel ソルバーで扱える規模の類似問題の解決を図り，条件の設定の仕方が正しかったことを確認する方法もある。しかしながら，解決すべき問題をそのまま扱えたほうが，学生のモチベーションを維持させるのには有効であると思われる。この点からすると，“扱える問題の規模”も重要であることがわかる。「グループ分け問題」は，最適化ソルバーを用いて解くことができる。図 14，図 15 は，それぞれ「グループ分け問題」を ZIMPL により記述し，SCIP を実行した結果である。

表計算モデルから「グループ分け問題」の解を求める方法の 1 つとして，OpenSolver を用いる方法がある（Mason et al., 2010）。OpenSolver は，Excel ソルバー同様，Excel 向けのアドインソフトである。非線形計画問題を扱うことはできないが，扱える問題の規模に制限はない。図 16 は，「グループ分け問題」に対して，（Excel ソルバーによる「グループ分け問題」への取組みで解説したように）問題状況を表形式に整理（すなわち，表計算モデルの作成）した後に OpenSolver を実行した結果である。「所属ゼミ」，「実地研修先」，「分析企業」に関する全ての条件を満

1 回目分析企業より

	分析企業②									
分析企業①	1	2	3	4	5	6	7	8	9	10
1	0	0	1	1	0	0	1	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0

※「分析企業①」の 1 行目には，説明のために仮の値を入力している。

図 13 「グループ分け問題」の表計算モデル②

```

set I:={1..30}; #学生
set J:={1..10}; #割当て企業
var x[I*J] binary; #0-1 変数
maximize obj: #目的関数
    x[1,1]+x[1,2]+x[1,3]+x[1,4]+x[1,5]+x[1,6]
    +x[1,7]+x[1,8]+x[1,9]+x[1,10];
#制約条件
#条件①
subto con1: forall <i> in I do
    sum<j> in J:x[i,j]=1;
#条件②
subto con2: forall <j> in J do
    sum<i> in I:x[i,j]=3;
#条件③
#実地研修が A の学生に関して
set Ken_A:={10,15,19,20,22,28};
subto con3: forall <j> in J do
    sum<i> in Ken_A:x[i,j]<=1;
set Ken_B:={12,14};
subto con4: forall <j> in J do
    sum<i> in Ken_B:x[i,j]<=1;
...
set Ken_J:={16,21};
subto con10: forall <j> in J do
    sum<i> in Ken_J:x[i,j]<=1;
set Ken_K:={4,5};
subto con11: forall <j> in J do
    sum<i> in Ken_K:x[i,j]<=1;

#条件④
#所属ゼミが P である学生に関して

```

図 14 「グループ分け問題」の記述 (ZIMPL)

たしたグループ分け案が得られている (図 17 参照)。OpenSolver の利用方法は、表計算モデルの作成も含めて、Excel ソルバーと (ほとんど) 変わらない。問題状況を表形式

```

set Zemi_P:={5,8,17};
subto con12: forall <j> in J do
    sum<i> in Zemi_P:x[i,j]<=1;
set Zemi_Q:={3,16};
subto con13: forall <j> in J do
    sum<i> in Zemi_Q:x[i,j]<=1;
...
set Zemi_Y:={15,19,24};
subto con19: forall <j> in J do
    sum<i> in Zemi_Y:x[i,j]<=1;
set Zemi_Z:={4,20,29};
subto con20: forall <j> in J do
    sum<i> in Zemi_Z:x[i,j]<=1;

#条件⑤
#分析企業①が企業 1 の学生に関して
set First_1:={6,7,23};
subto con21: forall <j> in J do
    sum<i> in First_1:x[i,j]<=1;
set First_2:={14,24,25};
subto con22: forall <j> in J do
    sum<i> in First_2:x[i,j]<=1;
...
set First_9:={4,9,19};
subto con29: forall <j> in J do
    sum<i> in First_9:x[i,j]<=1;
set First_10:={17,21,22};
subto con30: forall <j> in J do
    sum<i> in First_10:x[i,j]<=1;

#条件⑥
subto con31: sum<i> in First_1:x[i,1]=0;
subto con32: sum<i> in First_2:x[i,2]=0;
...
subto con39: sum<i> in First_9:x[i,9]=0;
subto con40: sum<i> in First_10:x[i,10]=0;

```

図 14 「グループ分け問題」の記述 (ZIMPL) (続き)

solution status: optimal solution found

objective value: 1

x#1#1 1 (obj:1)

x#2#2 1 (obj:0)

x#3#1 1 (obj:0)

x#4#1 1 (obj:0)

x#5#2 1 (obj:0)

x#6#2 1 (obj:0)

x#7#3 1 (obj:0)

...

x#24#8 1 (obj:0)

x#25#10 1 (obj:0)

x#26#8 1 (obj:0)

x#27#9 1 (obj:0)

x#28#10 1 (obj:0)

x#29#10 1 (obj:0)

x#30#9 1 (obj:0)

※解のなかで値が 0 となった変数は表示されない。

図 15 SCIP の実行結果

に整理することの教育は、どちらを利用する場合においても有効である。

1 回目分析企業より

分析企業①	分析企業②									
	1	2	3	4	5	6	7	8	9	10
1	0	1	0	0	0	0	0	1	0	1
2	0	0	0	0	1	1	0	0	0	1
3	0	0	0	0	0	1	1	1	0	0
4	0	0	1	0	0	0	1	0	1	0
5	1	0	0	1	0	0	0	1	0	0
6	0	1	0	0	1	0	0	0	0	1
7	1	0	1	0	0	0	0	0	1	0
8	0	0	0	1	0	1	1	0	0	0
9	1	0	0	1	1	0	0	0	0	0
10	0	1	1	0	0	0	0	0	1	0

図 17 OpenSolver の実行結果（条件⑤）

学生	分析企業②										ゼミ	実地研修先	分析企業①	行和	上限
	1	2	3	4	5	6	7	8	9	10					
学生 1	1	0	0	0	0	0	0	0	0	0	T	C	10	1	1
学生 2	0	0	0	0	0	0	1	0	0	0	T	H	1	1	1
学生 3	0	0	1	0	0	0	0	0	0	0	Q	F	2	1	1
学生 4	0	0	0	0	1	0	0	0	0	0	Z	K	2	1	1
学生 5	0	0	0	0	0	0	0	0	0	1	P	K	7	1	1
学生 6	0	1	0	0	0	0	0	0	0	0	S	D	6	1	1
学生 7	0	0	0	0	0	0	0	1	0	0	R	C	7	1	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
学生 24	0	0	0	0	0	0	0	0	0	1	Y	I	S	1	1
学生 25	0	0	0	0	0	1	0	0	0	0	X	E	R	1	1
学生 26	0	0	0	0	0	0	0	0	1	0	R	H	...	1	1
学生 27	0	0	0	0	1	0	0	0	0	0	W	D	Y	1	1
学生 28	1	0	0	0	0	0	0	0	0	0	W	A	X	1	1
学生 29	0	0	1	0	0	0	0	0	0	0	Z	C	R	1	1
学生 30	0	0	0	1	0	0	0	0	0	0	X	C	W	1	1
列和	3	3	3	3	3	3	3	3	3	3					
上限	3	3	3	3	3	3	3	3	3	3					

図 16 OpenSolver の実行結果

4. ま と め

本研究では、解析的問題解決における Excel ソルバー及び最適化ソルバーの利用に関する考察を行った。Excel ソルバーと最適化ソルバーのどちらの利用においても、数理モデルが前もって作成されていると想定されている。しかしながら、(特に文系学生のような)十分な数理的知識を持たない学生にとって、数式モデルは分かりづらいものである。Excel ソルバーや最適化ソルバーを数値計算にかかる負担の軽減を目的に利用するだけでは、学生の解析的問題解決の教育へのモチベーションを維持させることは難しい。

今日、多くの学生が物事を表形式に整理することに慣れ親しんでいる。本研究では、数理モデルを“解く”ためではなく、問題状況を整理する(“書く”)手段としての表計算モデルの作成に着目した。表計算モデルの作成に関して、Excel ソルバーの利用を想定したテキストは、完成された表計算モデルの解説を行うことはあるものの、どのようにその表計算モデルが作成されたのかを解説すること(ほとんど)ない。本研究では、実際問題への取り組みを例に、“学生が、スプレッドシート上に必要な項目を追加するなどしながら、問題状況の整理を(ある程度)進められること”を示した。また、(特に複雑な問題を扱う場合には,)“ソルバーを実行して、その都度問題状況の整理の仕方が正しいことを確認しながら表計算モデルの作成を進められる環境が重要であること”を示した。

注

- 1) Excel は Microsoft 社の登録商標である。
- 2) 扱える変数は 200 個まで等、Excel ソルバーには扱える問題の規模に制限がある。
- 3) 奥田 (2001) p.7 から引用
- 4) SCIP, GLPK, lp_solve は、混合整数計画 (MIP: Mixed Integer Programming) 問題を解くことがで

```
max: /*objective function*/
3 x1 + 4 x2;
/*subject to*/
Material_P: 2x1 + 1x2 <= 1600;
Material_Q: 1x1 + 1x2 <= 1000;
Material_R: 1x1 + 3x2 <= 2400;
```

※“/*”と“*/”に挟まれた部分はコメントを表す。

図 18 「生産計画問題」に対する LP ファイル

きる最適化ソルバーである。当然、整数条件のみからなる問題、整数条件を含まない問題でも解くことができる。SCIP は、アカデミックな目的であれば、無償で利用することができる。

- 5) lp_solve (LPSolveIDE) の利用においては、lp_solve 固有の形式に従ったものを利用した(図 18 参照)。
- 6) LP 形式において、デフォルトでは全ての変数は非負変数として扱われる。そのため、図 3 では非負条件を省略している。
- 7) コマンドプロンプトにより利用する方法以外にも、“問題の記述、ソルバー実行、解の出力”の一連の作業を支援する統合開発環境 (IDE: Integrated Development Environment) を利用する方法もある。GLPK の統合開発環境としては GUSEK, lp_solve の統合開発環境としては LPSolveIDE が利用可能である。
- 8) Excel ソルバーにより扱える問題の規模には制限があることを考えると、最適化ソルバーの方が適しているといえる。
- 9) (CPLEX-) LP 形式において、目的関数に含まれる定数項は、あらかじめ取り除いておく必要がある。また、制約条件の定数項は、あらかじめ 1 つにまとめたものを右辺に置いておく必要がある。そのため、数理モデル(図 7)と LP 形式の記述(図 8)では定数項の値が異なっている。
- 10) Excel ソルバーの実行の際には目的関数に含まれる定数項をあらかじめ取り除く作業を行っていないため、SCIP の実行結果とは目的関数の値が異なっている。
- 11) 「分析企業①」の 1 行目には、説明のために仮の値を入力している。

謝 辞

本研究は、平成 27 年度北海学園大学学術研究助成（研究代表者：大平義隆）を受けて行われた。

参考文献

- [1] Beliën J, Colpaert J, De Boeck L, Eyckmans J, Leirens W, “Teaching integer programming starting from an energy supply game”, *INFORMS Transaction on Education*, Vol.13, No.3, pp.129-137, 2013
 - [2] Conway, D. D., Ragsdale, C. T., “Modeling Optimization Problems in the Unstructured World of Spreadsheets”, *Omega, International Journal of Management Science*, Vol.25, No.3, pp.313-322
 - [3] Mason, A. J., Dunning, I., “Opensolver: open source optimization for excel”, *Proceedings of the 45th Annual Conference of the ORSNZ*, pp.181-190, 2010
 - [4] 垣花京子, アンケートにみる文科系の OR 教育, オペレーションズ・リサーチ: 経営の科学 Vol.42, No.6, pp.400-405, 1997
 - [5] 荻田正雄, 上田太一郎, 中西元子, 『Excel でできる最適化の実践らくらく読本』, 同友館, 2009
 - [6] Lilien, G. L., Van Bruggen, G. H., Starke, K., “DSS Effectiveness in Marketing Resource Allocation Decision: Reality vs. Perception”, *Information System Research*, Vol.15, No.3, pp.216-235, 2004
 - [7] 村井直志, 『企画・戦略スタッフのための「入門」科学的的意思決定』, 秀和システム, 2011
 - [8] Nagraj Balakrishnan, Barry Render, Ralph M. Stair, “Managerial Decision Modeling with Spreadsheets”, Prentice Hall, 2012
 - [9] 大野勝久, 逆瀬川浩孝, 中出康一, 『Excel ソルバーで学ぶオペレーションズリサーチ』, 近代科学社, 2014
 - [10] 奥田和重, 『経営科学入門』, ムイスリ出版, 2001
 - [11] Stevens, S. P., Palocsay, S. W., “A Translation Approach to Teaching Linear Program Formulation”, *INFORMS Transaction on Education*, Vol.4, No.3, pp.38-54, 2004
 - [12] 高井英造・真鍋龍太郎, 『問題解決のためのオペレーションズ・リサーチ入門—Excel の活用と実務的例題—』, 日本評論社, 2000
- GLPK:<http://www.gnu.org/software/glpk>
lp_solve:<http://lpsolve.sourceforge.net/>
SCIP:<http://scip.zib.de>