Excelで始める数理最適化

後藤　順哉

本稿は，最も身近な最適化ソルバーと言える Excel ソルバーを用いて，「とりあえず最適化問題を解く」体験から数理最適化に親しむための教材を提供することを目的とする．具体的には，いくつかの例題を通して Excel ソルバーの使い方を天下り的に学びつつ，定式化とソルバーの結びつきや，Excel ソルバーの限界を概観し，本特集号の他の記事で紹介される数理最適化の発展的な利用のための足がかりを提供する．

キーワード：数理最適化，Microsoft Excel，ソルバー

1. はじめに

数理最適化（数理計画）は，OR の中核技術として理論的な成熟度を増す一方，高度な計算機環境の普及により，その実務的適用が期待されている．その求解を行うアルゴリズムを計算機上で実装したソフトウェアは一般に**ソルバー**と呼ばれ，非常に高額なものからインターネット経由で無償で入手可能なものまで，多数利用可能になっている．

その中で，表計算ソフトとして圧倒的な普及度を誇る Microsoft Excel（以下，Excel）の 1 機能として含まれるソルバー（以下，Excel ソルバー）は，数理最適化の入門教材としてふさわしい．Excel 自体が有償であるため，無償のソルバーに分類するわけにはいかないが，Excel が高等教育からビジネスにわたる広範な領域でデ・ファクト・スタンダード（事実上の標準）としての地位を確立し，数理最適化に馴染みがない人でもその基本的な使い方に通じている（or 習得しやすい）ことがその背景にある．筆者は 2007 年から勤務先にて Excel を使った OR の授業を行っているが，Excel ソルバーによる導入が学生に一定の満足度を，教員に一定の手応えをもたらすことは，その経験から実感するところである．また，Excel ソルバー自体は，**アドイン**と呼ばれる，ある意味おまけの機能として提供されているものの，線形計画（以下，LP），（混合）整数計画の他，非線形計画の（局所最適解の）求解機能まで含んでいる．

本稿では，Excel の基本操作については既にある程度馴染みがあり，LP が何なのかについて理解があることを前提に，ソルバーのエントリーモデルとして Excel ソルバーの紹介を行い[1]，本特集のテーマである整数計画の導入に代える．なお，Excel ソルバーの操作方法は Excel のヴァージョンに依存している部分もあり，本稿は Excel 2010 に基づいて記述されていることを予めご了解いただきたい．それ以前や以降のヴァージョンにおいても基本部分は変わらないと思われるが，その差分はインターネット検索などで埋めていただきたい．また，Excel と高い互換性のある無償表計算ソフト OpenOffice Calc でも，ある程度共通した操作でソルバー機能が利用可能であることを付け加えておこう．

2. Excel ソルバーを使うには

Excel ソルバーを初めて使う場合には，**ソルバーアドイン**の導入が必要である．

図 1　［データ］タブ内の［ソルバー］ボタンの有無を確認

Excel 2010 を起動した際，シート上部の［データ］タブの中に［ソルバー］ボタン（図 1）が表示されていれば，ソルバーを利用することができるが，そうでない場合は以下の手続きを行う．

(i) ［ファイル］タブの［オプション］をクリック．
(ii) 現れた「Excel のオプション」ダイアログボックス（以降，DB）最左列にある［アドイン］をクリック（図 2 左）．「アクティブでないアプリケーション アドイン」リストで［ソルバーアドイン］をクリックしてアクティブ（青くハイライトされ

ごとう　じゅんや
中央大学理工学部経営システム工学科
〒 112–8551 東京都文京区春日 1–13–27

[1] 紙面の都合上，LP，0-1 整数計画に絞る．その他の例題については拙著 [3] や類書（[1] など）を参照していただきたい．

る状態）にし，［設定］をクリック．現れた「アドイン」DB（図 2 右）の「有効なアドイン」欄の「ソルバーアドイン」にチェックを入れ，［OK］をクリック．

以上の操作の後，（インストールを経て，）図 1 のように，［データ］タブ内に［ソルバー］が現れれば，ソルバーが使えるようになる．

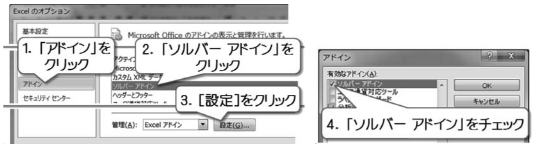


図 2　「Excel のオプション」DB と「アドイン」DB

3. 輸送問題を解いてみよう

それでは以下の例題をソルバーで解いてみよう[2]．

■例題 1（輸送問題）　あるメーカーの 3 カ所の工場 A，B，C の供給量，および，4 カ所の都市の需要量，そして各工場から各都市へ運ぶのにかかる 1 単位当たり輸送費用と該当経路上の輸送上限が，表のように与えられているとする．このとき，工場の供給上限，都市の需要，各経路の輸送上限を満たし，かつ，輸送費用の合計を最小にする輸送計画を求めよ．

表 1　輸送費用 c_{ij}

工場 ＼ 都市	1	2	3	4
A	4	1	3	3
B	9	2	7	10
C	8	1	10	9

表 2　上限 u_{ij}, 供給 a_i, 需要 b_j

工場 ＼ 都市	1	2	3	4	供給
A	28	25	21	28	60
B	17	15	15	25	45
C	22	17	19	10	35
需要	35	15	35	45	

この問題例は，工場 $i \in \{A, B, C\} =: I$ から都市 $j \in \{1, 2, 3, 4\} =: J$ への輸送量を（決定）**変数** x_{ij} と表して，以下のような LP として定式化できる[3]：

$$
(\mathrm{P1})\quad
\left|\;
\begin{array}{lll}
\underset{(x_{ij})}{\text{最小化}} & \displaystyle\sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij} & \cdots(1)\\
\text{条　件} & \displaystyle\sum_{j\in J} x_{ij} \le a_i, \quad i \in I & \cdots(2)\\
 & \displaystyle\sum_{i\in I} x_{ij} = b_j, \quad j \in J & \cdots(3)\\
 & 0 \le x_{ij} \le u_{ij},\ i \in I, j \in J & \cdots(4)
\end{array}
\right.
$$

ここで，c_{ij} は i から j への単位輸送費用（表 1），a_i は工場 i の供給量（表 2 最右列），b_j は都市 j の需要量（表 2 最下行），u_{ij} は i から j への輸送量上限（表 2 中央部）であり，いずれもデータとして与えられている．

Excel ソルバーに限らず，一般にソルバーを扱ううえで，このような定式化表現を明確に意識できることは必要条件と言ってよい．一方，それが最適化の実務的利用に対する壁となっている面もあり，特に x_{ij} のような二重添え字は，初学者にとって 1 つの壁となりうる．

しかしながら，この輸送問題について言えば，表計算ソフトである Excel のインタフェースのお蔭で，容易に問題の構造を理解することができる[4]．

■データをシートに入力する　例題 1 の LP をソルバーで解くにあたり，まずは与えられたデータを Excel のシートに入力する．ここでは図 3 のようにデータを与えたとして説明する[5]．具体的には，3×4 のセル範囲 `C4:F6` に費用 c_{ij} が，`C11:F13` に上限 u_{ij} が，3×1 の列範囲 `H18:H20` に供給 a_i が，1×4 の行範囲 `C22:F22` に需要 b_j が，それぞれ与えられている．

■目的関数や制約を表現する式を入力する　データの入力が完了したあと，ソルバーに渡す数式を入力する．今回の場合，目的関数式 (1)，供給制約 (2) の左辺式（3 本），需要制約 (3) の左辺式（4 本）を，それぞれセ

---

[2] 例題 1 は本特集のターゲットである整数計画ではないが，導入として，少し丁寧に Excel ソルバーの利用法を紹介する．

[3] 【初学者向けの補足】(P1) のような表現は**定式化**と呼ばれ，数理最適化問題としてどのような構造を持つのかを整頓するのに便利である．(1) 式は**目的関数**と呼ばれる．「最小化」とあるので，(1) 式が最も小さくなるような $x_{ij}, i \in I, j \in J$ を見つけるのが (P1) の目的である．ここでは，工場 i から都市 j に x_{ij} だけ輸送すると，$c_{ij}x_{ij}$ だけ費用がかかるので，(1) 式は「すべての工場からすべての都市への輸送にかかる費用の合計」を表している．一方，「条件」以下の (2)〜(4) 式は**制約条件**あるいは**制約式**と呼ばれ，$x_{ij}, i \in I, j \in J$ が満たさなければならない条件を記述している．(2) 式は，工場 i の 4 つの都市への出荷量合計 $\sum_{j\in J} x_{ij}$ が，供給可能量 a_i を超えられないという条件を，すべての工場 $i \in I$ について満たさなければならないことを意味する．(3) 式は，各都市 j に 3 つの工場から届く受取量 $\sum_{i\in I} x_{ij}$ が，需要量 b_j に一致するという条件を，すべての都市 $j \in J$ について満たさなければならないことを意味する．(4) 式は，工場 i から都市 j への輸送量 x_{ij} が非負（0 以上）で，かつ，上限量 u_{ij} を超えないことを意味する．制約条件を満たし，目的関数を最小（「最大化」の場合は最大）にする $x_{ij}, i \in I, j \in J$ を**最適解**，そのときの目的関数値を**最適値**という．

[4] これは，学部低学年の LP の授業において，一般の LP に対する単体法の説明の前に，まず輸送問題に対する飛び石法を扱う意義に似ている（[2] など）．

[5] データの入力の仕方は任意であるが，経験を重ねていくと，ソルバーの操作（後述）に都合の良い入力法（自分の流儀のようなもの）を確立していくことが可能である．

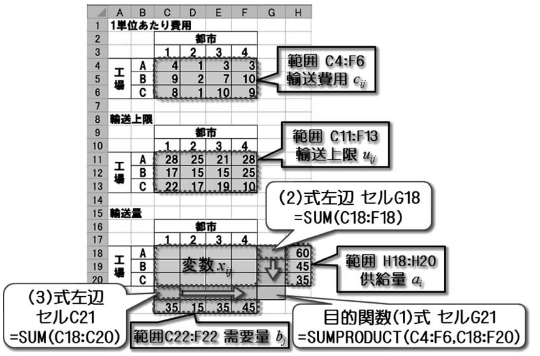


図 3　輸送問題を解くためのデータ入力例

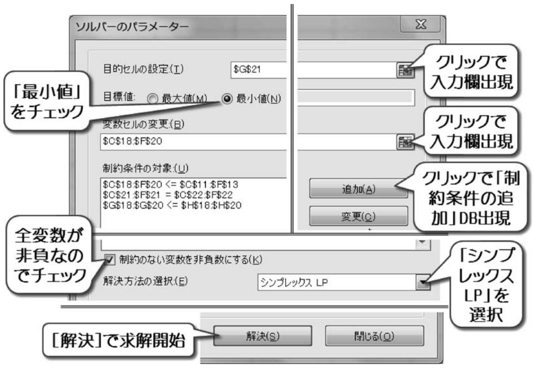


図 4　「ソルバーのパラメーター」DB

ルを設けて入力する必要がある．式は，c_{ij}, u_{ij}, a_i, b_j などのデータと変数 x_{ij} を結びつけて記述されるため，変数 x_{ij} のセル範囲を決めておく必要がある．そこで，図 3 のように，セル範囲 `C18:F20` を x_{ij} のセル範囲とする（と心の中で決めておく）．

それを踏まえたうえで，以下のように入力していく：

- セル `G21` に目的関数 $\sum_i \sum_j c_{ij}x_{ij}$ を記述するため，“`=SUMPRODUCT(C4:F6,C18:F20)`” と入力[6]．
- 制約式群 (2) の左辺式 $\sum_j x_{ij}$ をセル範囲 `G18:G20` に入力．具体的には，セル `G18` に “`=SUM(C18:F18)`” と入力したあと，下のセル `G19` と `G20` にコピー[7]．
- 同様に，制約式群 (3) の左辺式を `C21:F21` に入力．具体的には，セル `C21` に “`=SUM(C18:C20)`” と入力したあと，右隣のセル範囲 `D21:F21` にコピー[8]．

このように，等式制約にしろ不等式制約にしろ，制約式の入力には左辺式と右辺式を分けて入力する必要がある．

■ソルバーに目的関数や制約を表現する式を渡す　続いて，ソルバーに数理最適化問題としての情報を渡す．[データ] タブの [ソルバー]（図 1）をクリックすると，「ソルバーのパラメーター」DB が現れる（図 4）．

「ソルバーのパラメーター」DB では，以下のように設定を行う：

(i) 「目的セルの設定」欄に目的関数を格納しているセル位置 “`G21`” を入力する．これはキー入力しないで，欄の右端をクリックして細長い入力ボックスが現れたら，シートのセル `G21` をクリックするのがよい．この結果，「目的セルの設定」欄には “`$G$21`” と（絶対参照で）表示される．

(ii) 「目標値」欄には最適化のタイプを指定する．今回は最小化なので，「最小値」にチェックを入れる．

(iii) 「変数セルの変更」欄は変数 x_{ij} のセル位置を入力する．「目的セルの設定」と同様，右端のボタンをクリックして入力用のボックスが現れたら，セル範囲 `C18:F20` をドラッグすることで入力される．

(iv) 次に，「制約条件の対象」欄に制約条件 (2)〜(4) 式の情報を順番に追加する．

- まず，DB 右側にある［追加］ボタンをクリックする．現れた「制約条件の追加」DB の「セル参照」欄に，(2) 式左辺を格納したセル範囲 `G18:G20` をドラッグして位置を入力する．真ん中の関係子の欄はドロップダウンリストから “`<=`” を選択，左の「制約条件」欄には右辺式を格納した `H18:H20` をドラッグして指定する．「制約条件の追加」DB にて［追加］もしくは［OK］をクリックすると，制約式群 (2) が（3 本まとめて）ソルバーに登録される[9]．まだ登録する制約式 (3)，(4) があるので［追加］を選ぶ．
- 同様に，制約式群 (3) の 4 つの等式制約についても行う．「セル参照」欄にセル範囲 `C21:F21`，関係子を “`=`”，「制約条件」欄にセル範囲 `C22:F22` を指定する．制約式群 (4)

[6] 関数 `SUMPRODUCT(配列 1, 配列 2)` は配列 1 と配列 2 の内積を計算する関数である．

[7] `G18` を [Ctrl]+[C] でコピーし，`G19:G20` まで（ドラッグするなどして）アクティブにして [Ctrl]+[V] で貼り付けすれば OK．例えば，`G19` は “`=SUM(C19:F19)`” となる．

[8] 以上の操作で 8 つのセルに式が入力されていることを確認されたい．ここまではソルバーのアドインがなくても行うことができるが，以降の操作では，前節で確認したとおり，ソルバーの導入が完了していることが必要になる．

[9] 制約式の左辺と右辺がそれぞれ規則正しく並べられていると，このようにまとめて制約を登録することができる．

のうち，変数の上限制約 $x_{ij} \leq u_{ij}$ についても同様である．「セル参照」欄にセル範囲 C18:F20 を，関係子に “<=” を，「制約条件」欄にセル範囲 C11:F13 を，それぞれ指定する．

- 制約式群 (4) の非負制約については，すべての変数 x_{ij} が対象となるので，「制約のない変数を非負数にする」にチェックを入れるだけで登録できる．

(v)「解決方法の選択」は利用するアルゴリズムの選択であり，「シンプレックス LP」を選択する[10]．

これでソルバーの設定は終了である．あとは DB 下端の［解決］をクリックすると，ほどなく「ソルバーの結果」DB が現れ，「ソルバーによって解が見つかりました.」と表示される.「ソルバーの解の保持」にチェックを入れ，［OK］をクリックすると，シートのセル範囲 C18:F20 にはソルバーによって求められた解が保持され，セル G21 には最適値 669 が表示される．

このように，輸送問題は変数を表形式に並べることで，制約式に現れるその行和と列和を簡単にとることができ，Excel シートの上でその制約条件を認識することが容易な例となっている[11]．

4. ナップサック問題を解いてみよう

Excel ソルバーでは，一部または全部の変数の取りうる値を整数に限定した**整数線形計画**を解くこともできる．

■ 例題 2（購読雑誌選定問題） 某大学では年々増加する学術雑誌の購読料高騰に頭を悩ませている．来年度は予算もカットされることから，これまで購読してきた雑誌 a〜r のうちいくつかの雑誌の購読を中止し，数を絞って購読することが避けられない．そこで，所属教員にアンケートを募り，その希望度および価格や出版社をまとめたのが次の表である．予算が 100 であるとき，希望度合計を最大にするよう購読継続雑誌を選定せよ．

表 3 18 の雑誌の出版社と価格と購読希望度

雑誌 j	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r
出版社	B	B	B	E	E	E	E	E	E	E	E	S	S	S	S	S	S	S
価格 c_j	2	13	13	15	20	10	12	3	10	7	5	8	10	5	5	9	5	4
希望度 b_j	9	3	3	1	10	7	10	0	1	1	6	3	5	0	2	15	3	0

[10] 名前のとおり，シンプレックス法（単体法）を選択することになる．なお，Excel 2010 からは単体法の反復ごとに求解を中断し，基底解を確認するオプション機能が追加されている．

[11] より一般のネットワークフローについて同様に扱おうとすると，いろいろな困難が生じるが，Excel 関数 SUMIF を使うことで容易に扱えることを最後の節で指摘する．

各雑誌 j に対して，購読を継続する場合 1，中止する場合 0 をとるような**バイナリ変数（0-1 変数）**x_j を用意しよう．雑誌の集合を $J := \{\mathrm{a}, ..., \mathrm{r}\}$ とすると，この問題は次のように定式化される：

$$
(\mathrm{P2})\left|\begin{array}{lll}
\underset{(x_j)}{\text{最大化}} & \sum_{j \in J} b_j x_j & \cdots(1) \\
\text{条　件} & \sum_{j \in J} c_j x_j \leq 100 & \cdots(2) \\
 & x_j \in \{0,1\}, \quad j \in J & \cdots(3)
\end{array}\right.
$$

(P2) はナップサック問題として知られる，基本的な (0-1) 整数計画である．基本的に線形関数のみを用いて表現されているが，変数 $x_j, j \in J$ が 0 か 1 いずれかに限定されている点が (P1) と異なる．実際，この違いが求解の困難さを激増させる．(P2) を Excel ソルバーで解くために，図 5 のようにデータを入力しておこう．

図 5 ナップサック問題を解くためのデータ入力例

具体的には，セル範囲 B3:S3 に価格 c_j，B4:S4 に希望度 b_j，U3 に予算 “100” を入力する．セル範囲 B5:S5 に変数 x_j が出力されるとして，セル T3 に (2) 式の左辺が評価されるよう “=SUMPRODUCT(B3:S3,B$5:S$5)” と入力し，T4 にコピーする．セル T4 は目的関数が入力されたことになる．続いてソルバーを起動し，「制約条件の対象」以外を以下のように設定する：

- 「目的セルの設定」：T4
- 「目標値」：「最大値」
- 「変数セルの変更」：B5:S5
- 「解決方法の選択」：「シンプレックス LP」

「制約条件の対象」については，まず例題 1 と同様にして (2) 式 “T3 <= U3” を入力する．注意すべきは，変数 x_j がバイナリ変数であることを指定する (3) 式の入力である．このために「制約条件の追加」DB において，「セル参照」欄にバイナリ変数を格納するセル範囲 B5:S5 を指定し，右隣の関係子を定義するドロップダウンリストから “bin” を選択する（図 6）．この選択操作だけで関係子欄に “=”，「制約条件」欄に “バイナリ” と表示される．なお，変数が 0 か 1 に制限されているので，「制約のない変数を非負数にする」はチェックを入れても入れなくてもよい．また，「解決方法の選

択」として「シンプレックス LP」を選択するが，整数変数が含まれ，かつ，目的関数や制約式がすべて線形である場合には，自動的に分枝限定法が実行される．

図 6　バイナリ変数の設定

あとは［解決］を選択すると，最近の PC であれば瞬時に最適解が求まる．シートのセル範囲 `B5:S5` で “1” となった雑誌を購読継続すればよい．ちなみに，最適値（希望度の和の最大値）は 73 となる．

5.　バイナリ変数の応用例

5.1　固定費付輸送問題

バイナリ変数の導入は数理最適化の記述力を大幅に高めてくれる．例題 1 の輸送問題を次のように改変した問題を考えよう．

■**例題 3**（固定費用の考慮）　各工場 i から都市 j に出荷する際，線形の変動費用の他に，表 4 のような固定費用 f_{ij} が生じる：

表 4　固定費用 f_{ij}

		都市 1	2	3	4
工	A	17	5	25	33
	B	18	5	20	12
場	C	15	20	5	15

例えば，工場 B から倉庫 3 へ少しでも輸送を行う場合，量 x_{B3} に比例した $c_{\mathrm{B3}}x_{\mathrm{B3}} = 7x_{\mathrm{B3}}$ の他に，$f_{\mathrm{B3}} = 20$ だけ追加して費用が発生する．

この修正を追加した問題の定式化は

$$
(\mathrm{P3})\left|\begin{array}{lll}
\underset{(x_{ij},z_{ij})}{\text{最小化}} & \displaystyle\sum_{i\in I}\sum_{j\in J}(c_{ij}x_{ij}+f_{ij}z_{ij}) & \cdots(1)\\
\text{条　件} & \displaystyle\sum_{j\in J}x_{ij}\le a_i, \quad i\in I & \cdots(2)\\
 & \displaystyle\sum_{i\in I}x_{ij}=b_j, \quad j\in J & \cdots(3)\\
 & 0\le x_{ij}\le u_{ij}z_{ij},\ i\in I, j\in J & \cdots(4)\\
 & z_{ij}\in\{0,1\}, \quad i\in I, j\in J & \cdots(5)
\end{array}\right.
$$

のような **0-1 混合整数計画**となる．各工場 i から各倉庫 j への輸送に対応してバイナリ変数 z_{ij} を導入し，(4) 式により「“$x_{ij} > 0$” となるのは $z_{ij} = 1$ のときのみ認められる」ようになっていることに注意しよう．

まず，例題 1 で利用したシートをコピー（または再利用）し，コピーしたシートのセル範囲 `L4:O6` に固定費用 f_{ij} のデータを追加し，追加された変数 z_{ij} を出力する範囲として `L11:O13` を確保しておこう（図 7）．そのうえで，以下の様に式を入力する：

- セル `G22` に “`=SUMPRODUCT(L4:O6,L11:O13)`”（(1) 式中の固定費用分 $\sum_i \sum_j f_{ij}z_{ij}$）
- セル `G23` に “`=SUM(G21:G22)`”（目的関数 (1) 式）
- セル範囲 `L18:O20` に，修正された x_{ij} の上限制約 $x_{ij} \le u_{ij}z_{ij}$ の右辺を設定しておく．具体的には，セル `L18` に “`=L11*C11`” と入力し，セル範囲 `L18:O20` にコピーすればよい．

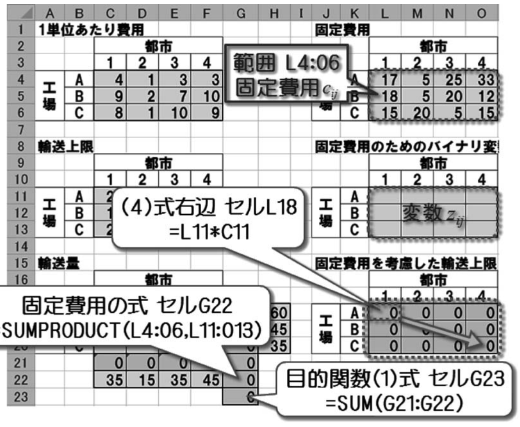


図 7　固定費用付輸送問題のデータ入力例

ソルバーを起動すると，例題 1 のソルバーの設定もコピーされている．そこで，次のとおり「ソルバーのパラメーター」DB の変更を行う．

(i)「目的セルの設定」欄を `G23` に変更．

(ii)「変数セルの変更」欄に範囲 `L11:O13` を追加[12]．

(iii)「制約条件の対象」欄のうち，“`C18:F20 <= C11:F13`” を選択したあと，［変更］をクリック．「制約条件の変更」DB 上の「制約条件」欄を “`L18:O20`” に変更．

(iv)［追加］をクリック．「セル参照」欄にセル範囲 “`L11:O13`” を，関係子に “`bin`” を選択し，z_{ij} がバイナリ変数であるという制約（(5) 式）を追加．

「ソルバーのパラメーター」DB で［解決］をクリックすると，しかる後ソルバーの求解が終了する[13]．結

[12] 具体的には，“`C18:F20`” の直後，カンマ “`,`” で区切ったあと，範囲 `L11:O13` をドラッグして “`C18:F20,L11:O13`”（実際の表示は “`$`” を含む絶対参照形式）となるように指定する．

[13] ここで「ソルバーによって公差内で整数解が見つかりました．すべての制約条件を満たしています.」のようなメッセージが出力される場合は最適性の保証が弱いので，（「ソ

果，最小費用は 816（うち変動費用は 674）という輸送計画が得られる．

5.2　割引付購読雑誌選定問題

例題 2 について，以下のような修正を考えよう．

> **■例題 4（出版社 E の提案）**　例題 2 の状況において，出版社 E から「5 つ以上の雑誌を購読するならば，（E 社の雑誌購入額を）3 割引きにする」という申し出があったとする．このとき，例題 2 で求めた希望度合計 73 を維持したまま，購入金額をどこまで小さくすることができるだろうか?

まず，すべての雑誌集合を J，出版社 E の雑誌集合を E としよう（つまり $E := \{\mathrm{d}, ..., \mathrm{k}\} \subset \{\mathrm{a}, ..., \mathrm{r}\} =: J$）．$j \in E$ に対しては 5 冊以上購読の場合 3 割引きになるので，この情報をとり入れるため，「E の雑誌が 5 冊以上継続になったときのみ 1，そうでなければ 0」となるバイナリ変数 $z_j \in \{0, 1\}$，$j \in E$ を導入しよう．

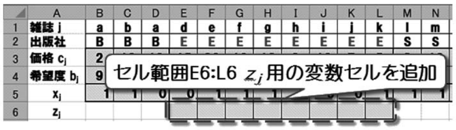
これを導入することで，当該問題は以下のように定式化できる：

$$
(\mathrm{P4})\quad
\left|\begin{array}{llll}
\underset{(x_j, z_j)}{\text{最小化}} & \displaystyle\sum_{j \in J} c_j x_j - 0.3 \sum_{j \in E} c_j z_j & & \cdots (1) \\
\text{条　件} & \displaystyle\sum_{j \in J} b_j x_j \geq 73 & & \cdots (2) \\
& z_j \leq x_j, & j \in E & \cdots (3) \\
& z_j \leq \frac{1}{5} \displaystyle\sum_{i \in E} x_i, & j \in E & \cdots (4) \\
& x_j \in \{0, 1\}, & j \in J & \cdots (5) \\
& z_j \in \{0, 1\}, & j \in E & \cdots (6)
\end{array}\right.
$$

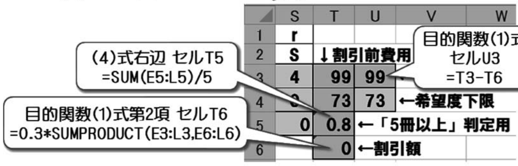
詳細な説明は省くが，ざっと言えば，「(6) 式にあるように z_j は 0 または 1 の値しかとりえず，支払額 (1) 式を小さくするためには $z_j = 1$ となる j を増やしたほうがよい．しかし，(4) 式から，$x_i = 1$ となる i が 5 つ以上ないと $z_j = 1$ にはできない．」という仕組みである[14]．(P4) を解くために，図 8 のように例題 2 のシートを改変する．まず，$z_j, j \in E$ の変数セルとしてセル範囲 `E6:L6` を用いることとする（図 8(a)）．そのうえで

- セル `T5` に “`=SUM(E5:L5)/5`”（(4) 式右辺）
- セル `T6` に “`=0.3*SUMPRODUCT(E3:L3,E6:L6)`”（割引額：目的関数 (1) 式第 2 項）
- セル `U3` に “`=T3-T6`”（目的関数 (1) 式）

と，それぞれ入力する（図 8(b)）．



(a) セル範囲 `E6:L6` に z_j，$j \in E$ のセルを追加



(b) 割引を表現する式を追加

図 8　例題 2 のシートを改変する

次にソルバーを起動し，以下のように設定する：

- 「目的セルの設定」：`U3`
- 「目標値」：「最小値」
- 「変数セルの変更」：`B5:S5,E6:L6`
- 「制約条件の対象」：
  - (i) `T4 >= 73`（(2) 式）[15]
  - (ii) `E6:L6 <= E5:L5`（(3) 式）
  - (iii) `E6:L6 <= T5`（(4) 式）
  - (iv) `B5:S5 = バイナリ`（(5) 式）
  - (v) `E6:L6 = バイナリ`（(6) 式）
- 「解決方法の選択」：「シンプレックス LP」

［解決］をクリックするとすぐに最適解が見つかり，希望度 73 を維持したまま費用を 87 まで減少させることができることがわかる[16]．

6.　Excel ソルバーを使う場合の注意

■整数制約　例題 2〜4 ではバイナリ変数を用いた定式化と，Excel ソルバーの設定方法を見てきた．変数の値を 0 と 1 に限らず，一般の整数に広げることも容易にできる．具体的には，「制約条件の追加」の際，関係子として “`bin`” でなく，“`int`” を選ぶと，対象となる変数を整数に限定できる．他の関数が線形関数のみであれば，「解決方法の選択」として「シンプレックス LP」を選択すると，例題 2〜4 の **0-1**（混合）整数計

ルバーパラメーターのダイアログボックスに戻る」にチェックを入れ，［OK］をクリックして，）「ソルバーのパラメーター」DB に戻り，［オプション］をクリックして現れる「オプション」DB の「すべての方法」タブ内の「整数の最適性 (%)」欄を “0” としてから，再度ソルバーを実行する．

[14] 整数計画の定式化に関するきちんとした説明については，本特集の藤江哲也氏の記事を参照されたい．

[15] または，図 8(b) のようにセル `U4` に “73” を入れておき，`T4 >= U4` とする．

[16] 現実の組織では，「ついた予算はなるべく使い切る」という発想が普通かもしれない．割引の設定において，予算 100 以下で希望度が最大になる雑誌を選定する問題は演習問題としよう．

画のときと同様に，分枝限定法による求解が行われる．

■非線形計画　Excel ソルバーでは LP や混合整数 LP の他，非線形計画も扱うことができる．その基本となるのは，局所解を求める「非線形 GRG」と，Excel 2010 から標準ソルバーに搭載された「エボリューショナリー」という 2 つのアルゴリズムである[17]．さらに前者については，Excel 2010 から「マルチスタート」という，ランダム生成した複数の初期点から出発し，最もよい解を出力するオプション機能も追加されている．しかし，Excel に標準で備わるソルバーにおいて，整数制約付きの非線形計画に対して最適性を保証するアルゴリズムは，現時点で提供されていない．Excel ソルバーでは，表計算で便利な `IF` や `MAX`，`MIN` などの非線形関数の使用を受容し，解らしきものを出力することも多い．しかしながら，それらには最適性の保証がないことは注意しなくてはならない．

■変数や制約条件の制限　Excel の標準ソルバーでは，利用できる変数（変数セル）や制約条件の最大数がそれぞれ 200，100 と制限されている[18]．このサイズは，現実的な問題を標準ソルバーで解こうとするうえで大きな障害となる[19]．

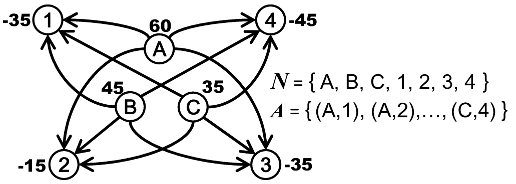


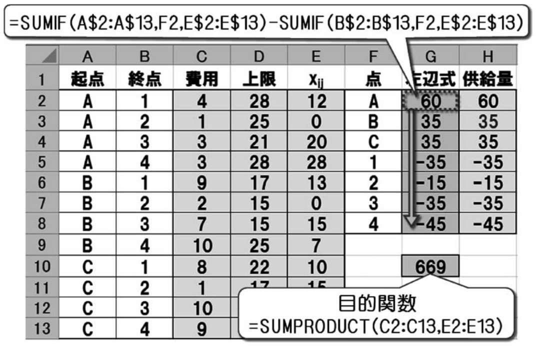
図 9　最小費用流問題としての輸送問題

問題によっては，無駄な変数を定義しないことで乗り越えられることもある．ネットワーク・フローの代表的問題である最小費用流問題を考えてみよう．この問題は，例題 1 で取り上げた輸送問題を，より一般的なネットワーク上に拡張したもので，最短路問題を含む多くの基本的問題を含む．例えば，例題 1 は，工場も都市も等しく節点 $v \in N := I \cup J$ で表し，需要を「$-1\times$ 供給」ととらえ，$a_j := -b_j, j \in J$ としたうえで，$A := \{(i,j) : i \in I, j \in J\}$ なるグラフ (N, A) 上で費用最小の $(x_{ij})_{(i,j)\in A}$ を求める最小費用流問題とみなせる（図 9）．総供給 $\sum_i a_i$ と総需要 $\sum_j b_j$ が等しいとき，最小費用流問題は次のような定式化で与えられる：

$$
(\mathrm{P5})\left|\begin{array}{lll}
\underset{(x_{ij})}{\text{最小化}} & \displaystyle\sum_{(i,j)\in A} c_{ij}x_{ij} & \cdots(1) \\
\text{条　件} & \displaystyle\sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ij} = a_i,\ i \in N & \cdots(2) \\
 & 0 \le x_{ij} \le u_{ij}, \qquad (i,j) \in A & \cdots(3)
\end{array}\right.
$$

一般の (N, A) に対し，もし例題 1 をまねして $(i,j) \in A$ の起点 i を行，終点 j を列にした矩形のセル範囲に変数セルを用意すると，変数セルの数が必要以上に多くなってしまう．むしろ，(P5) のように，A に含まれる (i,j) の分だけ変数を定義し，各節点 $i \in N$ での流量の帳尻を合わせる制約式 (2) を簡単に記述できればよい．すなわち，(2) 式左辺の $\sum_{j:(i,j)\in A} x_{ij}$ や $\sum_{j:(j,i)\in A} x_{ij}$ を簡単に評価できれば，より大きな最小費用流問題を Excel ソルバーで解く可能性が多少広がる．

特定の (i,j) に関する足し合わせを実現するには関数 `SUMIF` が役に立つ[20]．図 10 は，例題 1 で（供給合計 $\sum_i a_i$ と需要合計 $\sum_j b_j$ が一致するように）工場 B の供給量 a_{B} を 35 に変更した場合に対して，最小費用流問題を関数 `SUMIF` を使って Excel ソルバーで解かせるシート例を示したものである． 一般の場合にも，最小限の変数の導入ですむことが想像できるであろう（計算は演習問題としよう．例題 1 と最適値は一致する）．このように，同じ問題でも，入力方法を変えることで制限を緩和することができる場合がある．



	A	B	C	D	E	F	G	H
1	起点	終点	費用	上限	x_{ij}	点	左辺式	供給量
2	A	1	4	28	12	A	60	60
3	A	2	1	25	0	B	35	35
4	A	3	3	21	20	C	35	35
5	A	4	3	28	28	1	-35	-35
6	B	1	9	17	13	2	-15	-15
7	B	2	2	15	0	3	-35	-35
8	B	3	7	15	15	4	-45	-45
9	B	4	10	25	7			
10	C	1	8	22	10		669	
11	C	2	1					
12	C	3	10					
13	C	4	9					

図 10　関数 `SUMIF` を使って疎性を利用

■ VBA の活用　Excel には VBA と呼ばれるプログ

[17] 前者は一般化簡約勾配法，後者は遺伝的アルゴリズムである．

[18] Excel にソルバーを提供している Frontline Systems 社からその拡張版を買えば，そういった制限は緩和される（詳しくは同社 HP：`www.solver.com/` などを参照されたい）．

[19] 例えば，輸送問題の特殊ケースとして知られるクラス編成問題 [2] を解こうとすると，学生 20 人・10 クラスの問題が限界となる．

[20] 関数 `SUMIF`（和をとる行位置を検索する列範囲，検索のキー，和をとる対象列範囲）

ラミング機能も備わっており，これを利用したプログラミングを行い，そこからソルバーに関する関数を呼び出すことで利用価値が高まることもある．例えば，DEA（データ包絡分析法）では，ある程度実用性の高いモデルであっても変数の数が 200 以下に抑えられる一方，繰り返し LP を解く手間が問題になってくる．そのような場合，VBA で for ループなどの繰り返し操作をプログラムし，そのサブルーチンでソルバーを呼び出すようにすると，格段に使い勝手が向上する．

■あくまで入門用　このように，Excel に標準的に備わる機能のみであっても，組合せによってはかなりの程度身近に数理最適化を実践できる．しかしながら，やはり変数や制約の数の制限は，他の商用ソルバーなどと比べても非常に大きな制限となるのは否めない．特に，実用上慎重に解きたい問題や，データマイニングに代表される大規模データに対し数理最適化を適用したい場合などは，Excel ソルバーの使用は早々に諦めて，本特集の宮代隆平氏の記事を参考に，他のパワフルなソルバーを利用することを考えたほうが賢明である．

とはいえ，Excel ソルバーは最も身近なソルバーであり，本稿で見てきたような（実用的には小さいが，手で解くにはしんどい）問題が瞬時に解けてしまう．初学者にとって数理最適化の導入としてふさわしい教材であると言う所以はここにある．

参考文献

[1] 大野勝久，伊藤崇博，田村隆善，『Excel によるシステム最適化』，コロナ社，2001.
[2] 今野浩，『線形計画法』，日科技連，1987.
[3] 藤澤克樹，後藤順哉，安井雄一郎，『Excel で学ぶ OR』，オーム社，2011.