

〔研究ノート〕

# ExcelのソルバーとVBAの組合せによる分析

——ソルバーの利用例を紹介しながら——

青 木 博 明

## I はじめに

Excelソルバーは、Excelが持っている最大化・最小化問題を解く機能であり、各種の制約を設定することができる。また方程式を解くこともできる。他方Excel VBA (Visual Basic for Applications) もExcelに備わっているプログラミング言語である。本稿では、ExcelソルバーとExcel VBAを組み合わせることによって、経済と経営に関するより複雑な問題を解ける例を示す。“より複雑な”というのは、実際にはExcel VBAで問題の中のパラメータを連続的に変更しながらソルバーを実行して、最適化問題を解くことである。

Excelソルバーの利点は、数学的な解法を知らなくても、Excelのシート上で問題の定式化さえできれば、原則解を得ることができるところにある。実際の設定は後で説明するが簡単であり、少し習得すれば一般のビジネスパーソンでも利用できると言ってよい。制約として等式や不等式を設定でき、変数を整数やバイナリに設定することができる。それによって様々な最適化問題が解けるようになる。その例として、企業やビジネスの現場で直面する問題も多く挙げられる。本稿ではその例も紹介する。

ただ、ソルバーは数学的に解を得ているわけではなく、いくつかのヒューリスティックな逐次的計算によって解を得ているので、問題によっては、若干の誤差が出たり解が正しくなかったりするケースもある。よって、特に解が不自然と思われる場合は、変数の初期値や計算方法の設定を変えたりして、検算する必要がある。また得られた解が正しいとしても、それを鵜呑みにして、そのまま採用するのではなく、現状や他の方法との比較に利用するという観点も必要である。

すでにいくつかの文献が、Excelソルバーの利用方法を説明し利用例を紹介している。例えば、上田(2005)は資本予算計画やポートフォリオなど経済学的に重要な問題の解法を提示している。後藤(2012)は購読雑誌選定問題をナップザック問題の例として解くなどの興味深い問題を紹介している。本稿のExcelのソルバーとVBAを組み合わせるといったアイデアも後藤(2012)が一つのヒントになっている。藤澤他(2011)はORの説明を軸にしてソルバーの多くの利用例を紹介するとともに、Excelソルバーの設定方法、Excelソルバーと専用ソルバーとの比較、Excelのソルバーで使われている解法(アルゴリズム)および最適解を正確に得るための問題の定式化に関する注意事項など、Excelソルバーの実践的な利用のための包括的な説明をしている。

本稿の構成は以下のようになっている。IIソルバーの概要と利用方法、IIIソルバーの利用例、IVソルバーによる実際の解法の例、V ExcelソルバーとVBAの組合せによる問題の解法、むすびに、である。

## II ソルバーの概要と利用方法

ソルバーを利用するためには、Excelのシート上での最適化問題の設定、つまり決定変数(セルバーの

設定では変数セルと呼ぶ)と目的関数(同様に目的セルと呼ぶ)の指定と両者の関係の数式(代入や関数を含む)を使った直接・間接の記述, 変数間の等式・不等式の制約の設定および変数の種類の選択を行う必要がある。これには巧みな工夫や知識, アイデアが必要となることがある。その後, 解を得るための操作を行う。つまりソルバーの計算実行ボタンを押す。本稿ではExcelのバージョンをExcel 2016として設定の説明や計算を行っている。

## 2.1 ソルバーの設定と操作の方法<sup>1)</sup>

ソルバーの設定画面(以下, ソルバー画面と呼ぶ)での設定と実行は特に難しくない。ソルバー画面での設定には, 他にもオプション設定などがあるが, 基本的には以下に説明する「目的セル」「変数セル」「制約」に関する3つである。図1の実際のソルバーの設定画面を参照されたい。

### [目的セルの設定]

目的セルは目的とする, つまり最適化(最大化あるいは最小化)の対象となる変数に対応するセルである。目的セル変数は一つである。目標値として「最大値, 最小値, 値」があるのでどれかを選ぶ。ただし値は方程式を解くときに選ぶ。

### [変数セルの変更]

変数セルは, 目的変数を最適化させるために変化させる変数, つまり決定変数に対応するセルで, 一つもしくは複数のセルを指定する。“,”で区切ることによって離れた範囲を指定することができる。変数セルを変化させて目的セルを最大化させるので, 変数セルと目的セルを, 数式(代入や関数を含む)を使って, 直接・間接にシート上で関係づける必要がある。

### [制約条件の対象]

これは必ずしも必要ではないが, 現実的な問題になるほど各種の制約が必要となる。変数(セル)間を関係づける制約として等式と不等式(弱不等号のみで強不等号はない)があり, 変数の種類として整数と, 0または1のみをとるバイナリ変数を選ぶことができる。

変数の非負を指定するオプションもあるし, 不等式を利用して変数の非負条件の設定もできる。[制約条件]の[追加]をクリックすると, セル参照と制約条件が表示されるので, 必要項目を書き込む。左右の項目の間に[<=, =, >=, 区間, データ]が表示されるので, どれかを選択する。「区間」を選ぶと[制約条件]に[整数]が表示され, [データ]を選ぶと[バイナリ]が表示される。

### [解決]

ソルバー画面の設定が終われば, 解を計算するための実行ボタンである[解決]を押す。計算終了後, 変数セルに「(ソルバーが得た)解を記入する」か「元の値に戻す」かの選択画面が出るので選択する。前者を選べば得られた解が変数セルに書き込まれ, 後者を選べば変数セルの値は元のままである。計算には時間がかかることがある。解の探索に成功すれば“最適解が見つかりました。制約条件はすべて満たされました。”の, 失敗すれば“実行可能解が見つかりませんでした。”のメッセージが出る。ただし, 前者のメッセージが出てその解が正解とは限らない場合があるので注意が必要である。

### [レポート]

実行後「解答」「感度」「条件」のレポートを選択できるが, 本稿では説明しない。



※これは後出のソルバー問題（SM2）に対する実際のソルバー画面である。パイナリの設定の例示はない。

図1 ソルバーの設定画面

ソルバーの設定と操作の手順と方法をまとめると次になる。範囲の絶対参照は自動的に入る。

<操作手順> [データ] → [分析] → [ソルバー] → ソルバーの設定画面

<設定項目> [目的セル] [変数セル] [制約条件] (これは必須ではない)

## 2.2 Excelソルバーの解の探索方法と最適性の保証問題

### (1) Excelソルバーの解探索方法

Excelソルバーの解探索の方法には「GRG (Generalized Reduced Gradient: 一般化簡約勾配法) 非線形, シンプレックスLP (Linear Programming: 線形計画法), エボリューショナリー」の3つがあり, それらから選択できる。各探索方法については, 図1にもあるように『滑らかな非線形を示すソルバー関数にはGRG非線形エンジン, 線形を示すソルバー問題にはLPシンプレックスエンジン, 滑らかではない非線形を示すソルバー問題にはエボリューショナリーエンジンを選択してください。』の説明が表示される。エボリューショナリーはGA (Genetic Algorithm: 遺伝的アルゴリズム) のことである<sup>2)</sup>。

## (2) 解の最適性の保証問題

例えば3次4次関数に見られるように、限られた範囲のみの最大値(最小値)である局所的最適解が存在する場合(つまり目的変数のグラフに山や谷が複数存在する)、解の探索がそこで止まってしまう正しい解が得られないことがある。そのような場合、適宜初期値を変えて再計算することで正しい解が得られるときがある。その他、解の最適性の確実にするために望ましい点として、なるべく微分可能性が保持されるように定式化する、不連続関数を避ける、凸計画問題として定式化するなどが挙げられる<sup>3)</sup>。

## Ⅲ ソルバーの利用例<sup>4)</sup>

ソルバーの対象となる代表的な問題として挙げられるのが、目的関数と制約が一次式で示される線型計画問題(LP)であるが、他にも各種の非線形問題や変数が整数の値をとる整数計画、バイナリ(0 or 1)の値をとる0-1計画問題、その両方をとる0-1混合整数計画などがある。以下では、ソルバーによって解くことができる経済・経営の問題を拾い上げてみた。ただし明確な分類の基準があるわけではない。

### (1) 解析的に記述される関数の最大・最小化

単純な例として挙げられるのは2次関数3次関数などの数式で記述される関数の最大化・最小化である。ただし不等式、変数の範囲・整数などの制約を設定できるし、例えば範囲によって異なる関数が対応するような場合にも応用できる。

### (2) ミクロ経済学の問題<sup>\* 5)</sup>

ミクロ経済学では、予算制約下での生産量や効用の最大化、生産での利潤の最大化、費用の最小化などの計算問題が出題されるが、これらをソルバーで解くことができる。筆算で解いた後の検算にも利用できる。ソルバーでは整数解を得ることもできる。

### (3) 線型計画法

多くの問題が線型計画法として記述できる。例えば、複数の材料を使って複数の商品を生産するときの生産額の合計を最大化する問題である。各材料の所与量と製品1個当たりの必要量、各商品の価格が与えられているときの生産額全体の最大化である。目的関数と制約が線型であり線型計画問題となる。生産額から材料費を引いたときの利潤を最大化する問題とすることもできる。次に2つの例を挙げる。

#### (3a) 商品の生産額の最大化<sup>\*</sup>

Ⅳ章で、ケーキの生産額を最大化する簡単な例を挙げる。小麦、玉子、チョコレートの材料から2種類のケーキを生産するときの生産額を最大化する問題である。もっと大掛かりな工場での複数の工業製品の生産計画にも応用できる。行程数に関する制約、在庫スペースの問題や生産ラインを錆びつかせないなどの理由による生産量の上限と下限の制約などを組み入れることができる<sup>6)</sup>。また農地面積や労働量、肥料の制約の下で、作物の全生産額を最大化するように各作物の生産量を選ぶ問題を考えることもできる<sup>7)</sup>。

#### (3b) 栄養摂取費用の最小化<sup>8)</sup>

複数の食品をとって各種の栄養をある値以上に摂取する際に、各食品に価格を設定して費用を最小化する問題が考えられる。

### (4) 輸送問題<sup>9)</sup>

商品を複数の場所(供給拠点)から複数の場所(需要拠点)に輸送するとき輸送費全体を最小化する問題である。各供給拠点から各需要拠点への輸送費の係数(表)が与えられており、それに輸送量かけた額が輸送費となる。

各供給拠点に供給量の上限がありそれ以上の供給が不可能で、各需要拠点にも需要量の下限があり、それ以上の量が必要である。輸送量の上限も指定でき、輸送が少しでも行われれば発生する固定費の設

定も可能である。

#### (5) 仕事割当て問題<sup>10)</sup>

複数のスタッフと複数の遂行すべき仕事があり、各仕事に必要な労働時間と、各スタッフが各仕事を担当したときにかかる時間またはコストが与えられている。その設定の下で、全ての仕事の遂行に必要な全時間またはコストを最小化する。「仕事をする=1しない=0」のバイナリ変数を使う。

各スタッフが提供できる労働時間の上限(下限も)を設定したり、コストの代わりに各スタッフの各仕事に対する能力係数を設定する問題にすることもできる。ここではスタッフつまり人間としたが、代わりに道具・設備や何らかの機能を持つか、役割を担うものとしてもよい。その機能や役割は、それが遂行すべき仕事とともにより抽象的なものにする 것도できる。

#### (6) グループ全体の希望度最大化問題

あるグループに属するメンバーにいくつかの商品を割当てる。予め各個人の各商品(1個当たり)に対する希望度を聞きとり表にしておく。同じ商品の割当てる個数の上限は1個または複数とすることもできる。最大化するのは、割当てることによって実現されるグループ全体の希望度・満足度の合計である。

メンバー間の不平等が大きくなるように、各個人が提示できる満足度の合計を同じにしたり、各個人の実現される満足度の差の上限を設定したりすることができる。その上限は実際にソルバーで計算をしながら、適切な値に設定することができる。

(5)と同様に商品を他のものに置き換えることができる。何らかの満足度を高めたり、その助けとなったりする人材や道具・設備である。この問題は最大化問題であるが、上の(5)の仕事割当て問題と似た構造を持っており、双対的な関係にあるといえる。(5)は遂行のレベルが特定されており、それに達するためのコストを最小化する問題だが、この問題はコストが決まっておりその制約の下でパフォーマンスを最大化する問題である。

#### (7) シフト(勤務日程表)問題\*

各営業日(曜日)にスタッフの出勤を割当てる。各営業日のスタッフの必要人数、各スタッフの希望最低出勤日、各スタッフの出勤可能日などを所与とする。全スタッフの給与の総額を最小化する。または、スタッフから各出勤日の希望度を聞きとり、スタッフ全体の希望度を最大化することもできる。上で述べたようなメンバー間の不平等が大きくなるような制約も設けられる。

#### (8) 資本予算計画<sup>11)</sup>

複数のプロジェクトから、利潤(正味現在価値)の合計を最大にするようなプロジェクトを採択する問題を考える。各プロジェクトには支出・必要項目として「初年度の支出」「次年度の支出」「技術者の必要時間」、目的変数として「正味現在価値」が対応している。各プロジェクト*i*の採択をバイナリ変数 $x_i$ (採択しない=0 採択する=1)で示す。 $x_i=1$ のとき、プロジェクト*i*に対応する支出・必要項目と「正味現在価値」が実現される。 $x_i=0$ のとき、実現されない。予算または制限時間に全体の所与量があり、それらの全使用量が所与量以下であるという制約がある。

また例えばプロジェクト1と2のどちらか一方のみを必ず実施するとか、プロジェクト1が実施されてはじめてプロジェクト2の実施が可能という制約も設定可能である。これらの制約を満たしながら、採択されたプロジェクトが全て実現された場合の利潤の合計を最大にする。

#### (9) 在庫問題\*

在庫問題では、仕入の方法・タイミングと仕入量によって売上量や粗利益の最大化を考えるが、これにもソルバーを利用できる。そのためには予め仕入量と粗利益を数式によって関係づける表を作っておく必要がある。その際、例えば過去の需要量を所与の需要量として計算することができる。仕入量を決定変数として粗利益を最大化する。在庫スペースの制限の下で、複数の商品の仕入量を決定変数として

同時にソルバーで解いて、粗利益全体を最大化することもできる。

#### (10) 価格その他の変数による経営戦略

価格と需要の関係を与えることで、価格と利潤を関係させ、価格を変数セル、利潤を目的セルとするソルバーの設定が可能である。さらには、価格以外の経営上の戦略変数を変数セルに対応させることもできる。

#### (11) 最短経路問題・巡回セールスマン問題

いくつかの地点があり、その内から始点と終点を選び、始点から終点に向かう移動距離・コストを最小化する問題である。移動距離をコストなど他の重み・評価基準にすることもできる。同様に移動距離を最小化する問題として巡回セールスマン問題があるが、これについては付録で説明を行う。

#### (12) ナップサック問題<sup>12)</sup>

ある容量のナップサックがあり、それに複数の種類の品物を入れるときに、ナップサックに入れる品物の価値を最大化する問題である。各品物には一つ当りの価値と容量が与えられている。ナップサックの容量を予算とすることもできる。同じ商品のナップサックに入れる個数の上限、品物の個数間の関係などの設定も可能である。ただし、品物の個数が増えると計算量が爆発的に増えるので注意が必要である。

上の(4)(5)(6)(7)のように、需要側と供給側があり、需要量に下限(それ以上必要である)、供給側に上限(それ以下しか提供できない)があり、供給量が需要量以上になることが条件であるという構造を持っている例がいくつか見られる。またバイナリ変数を使って「0と1」を「有と無」「可と不可」「採と不採」に巧みに対応させて問題を定式化している例も多い。

## IV ソルバーによる実際の解法の例

以下では、上のⅢで取り上げた問題からいくつかを選んで実際にソルバーを使って解いたものを示す。基本的な問題と少し応用的な問題がある。どの問題も変数・パラメータを非負とし、解法は全てGRG非線形を選んだ。得られた解の最適性は確認していない。

### (SM1) 予算制約下での労働と資本による生産量の最大化      ミクロ経済学の計算問題の例

労働と資本を使って財Aを生産しようとしている。生産に投入する労働量と資本量を $L$ と $K$ とし、労働と資本の価格を $w$ と $r$ とする。生産に使用可能な予算額を $B$ とする。また生産関数を $F(L, K)$ とする。予算制約の下で、生産額を最大化する問題は次のように定式化される。

$$\text{最大化 Max } F(L, K) \quad \text{財Aの生産量} \quad \text{s.t. } wL + rK \leq B \quad \text{予算制約}$$

具体的な価格・パラメータの値、生産関数を次のように設定する。この設定の下で生産関数を最大化する $L$ と $K$ の量を見つける。また最大化された生産量も計算する。

$$\text{具体的な値と関数} \quad w = 1,200 \quad r = 3,200 \quad B = 2,000,000 \quad F(L, K) = 4 L^{0.3} K^{0.7}$$

以下、解くべき問題をソルバーに適用し易い形に置き換えたものをソルバー・モデルと呼ぶ。次は上の問題のソルバー・モデルである。

<ソルバー・モデル SM1: 生産量の最大化>

[与与]      価格  $w = 1,200$   $r = 3,200$     予算額  $B = 2,000,000$     生産関数  $F(L, K) = 4 L^{0.3} K^{0.7}$

[変数セル]	$L$	$K$	投入する労働量と資本量
[制約]	$wL + rK \leq B$		予算制約
[目的セル]	最大化	$F(L, K)$	生産量

次のExcelの表は上の問題に対する実際のソルバー画面である。パラメータの値とそれに対する解(変数セル：二重線で囲む)と最適値(目的セル：太線で囲む)が示されている。以下で示す表も同様である。

表1 予算制約下での労働と資本による生産量の最大化 (SM1) のソルバー計算の表

	A	B	C	D	E	F	G	
1								
2								
3					$f(L, K) = 4L^{0.3}K^{0.7}$			
4		消費量	労働力 $L$	資本財 $K$	生産量 $f(x)$			
5		変数セル→	500.0	437.5	1821.5			
6								
7		制約	$w$	$r$	計		$B$	
8		価格	1200	3200	使用金額		予算	
9		使用金額	600,000	1,400,000	2,000,000	≤	2,000,000	
10			$wL$	$rK$	$=wL+rK$			
11			=C8*C5	=D8*D5	=C9+D9			
12								
13		二重線で囲んだのが変数セル (以下、同様)						
14								
15		太線で囲んだのが目的セル (以下、同様)						
16								

(SM2) ケーキの生産額の最大化 線型計画法の例

小麦, 玉子, チョコレートの材料があり, それらを使って 2 種類のケーキ 1 とケーキ 2 を生産し, その生産額の合計を最大化する。

小麦, 玉子, チョコレートを順に材料  $i$  ( $i=1, 2, 3$ ) とする。ケーキ  $j$  の価格を  $p_j$ , 生産量を  $x_j$  とする。ケーキ  $j$  を一個作るのに必要な材料  $i$  の量を  $a_{ij}$  (生産係数), 使用する材料  $i$  の所与量  $B_i$  とする。2 つのケーキの生産額の合計を  $R$  とする。各材料の所与量の制約の下で  $R$  を最大化する。

$$\text{最大化 } R = p_1 x_1 + p_2 x_2 \quad \text{生産額の合計} \quad \text{s.t. } \sum_j a_{ij} x_j \leq B_i \quad i=1, 2, 3 \quad (\text{材料の制約})$$

次は上の問題をソルバー・モデルに置き換えたものである。

<ソルバー・モデル SM2：生産額の最大化>

[所与]	$p_j$	$a_{ij}$	$B_i$	$i=1, 2, 3$	$j=1, 2$	価格	生産係数	材料所与量
[変数セル]	$x_1$	$x_2$						ケーキの生産量 (整数)
[制約]			$\sum_j a_{ij} x_j \leq B_i$	$i=1, 2, 3$				材料の制約
[目的セル]	最大化		$R = p_1 x_1 + p_2 x_2$					生産額

次のExcelの表は上の問題に対するソルバーの画面である。パラメータ  $p_j, a_{ij}, B_i$  の値とそれに対する解 (変数セル) と最適値 (目的セル) が表に示されている。

表2 ケーキの生産額の最大化 (SM2) のソルバー計算の表

	A	B	C	D	E	F	G	H	I	J
1										
2										
3			<生産量>			<価格>		生産価値額の計		
4			$x_1$ ケーキ1	$x_2$ ケーキ2		$p_1$	$p_2$	$R$		
5		変数セル→	6	15		500	350	8250	←目的セル	
6						=SUMPRODUCT(F4:G4,C4:D4) ↑				
7			<材料の制約>							
8			材料	材料係数	使用量			所与量		
9				$a_{i1}$	$a_{i2}$	$UsedB_i$	≤	$B_i$		
10		小麦粉	材料1	40	60	1140		1200		
11		玉子	材料2	6	4	96		300		
12		チョコレート	材料3	12	5	147		150		
13		※ 材料 $i$ の使用量を $UsedB_i$ とする				↑=SUMPRODUCT(D10:E10,\$C\$4:\$D\$4)				

(SM3) シフト (出勤割当) 問題 給与全体の最小化

各営業日にスタッフの出勤を割当てる。スタッフの数を  $n$ , 営業日数を  $m$  とする。各スタッフの希望最低出勤日, 各営業日に必要なスタッフの人数, 各スタッフの出勤可能日を所与とする。スタッフ  $i$  の営業日  $j$  の出勤の不可と可をバイナリ変数  $a_{ij}$  ( $0 =$  出勤不可  $1 =$  出勤可) で示す。 $x_{ij} \in \{0,1\}$  は計算用のバイナリ変数で,  $s_{ij} = x_{ij} \times a_{ij}$  として  $s_{ij} = 0$  ならば欠勤,  $s_{ij} = 1$  ならば出勤とする。全スタッフの給与の総額を最小化する。ただし, 全ての賃金率を定数  $c$  としているので, 全スタッフの出勤日の合計を最小化することと同じになる。表3内の表Aと表Bの積を表Cとして計算している。ここでは基本的な形を示すが, 後で示すように様々な修正・応用が可能である。

<ソルバー・モデル SM3> シフト問題 実際のパラメータ・係数・変数は表を参照 [所与]

スタッフの番号を  $i$ , 営業日の番号を  $j$  とする  $i=1, \dots, n \quad j=1, \dots, m$

$MD_i$ : スタッフ  $i$  の希望最低出勤日  $MS_j$ : 営業日  $j$  のスタッフの必要人数

$a_{ij} \in \{0,1\}$ : スタッフ  $i$  の営業日  $j$  の出勤の可と不可を示す係数 ( $0 =$  出勤不可  $1 =$  可能) (表A)

[変数セル]

$x_{ij} \in \{0,1\}$  ( $0 =$  欠勤  $1 =$  出勤) は計算用の変数で変数セル (表B)

$s_{ij} \in \{0,1\}$ : 各スタッフの各営業日の実際の出勤を示す ( $s_{ij}=0$  ならば欠勤  $s_{ij}=1$  ならば出勤) (表C)

$s_{ij} = x_{ij} \times a_{ij}$  つまり  $a_{ij}=1$  かつ  $x_{ij}=1$  のときのみ  $s_{ij}=1$  それ以外は  $s_{ij}=0$

(表による計算の記述: 表A ( $a_{ij}$ : 出勤可不可)  $\times$  表B ( $x_{ij}$ : 変数セル) = 表C ( $s_{ij}$ : 実際の出勤))

[制約]

a)  $D_i$ : 各スタッフの実際の出勤日数  $\geq MD_i$ : 各スタッフの希望最低出勤日数

$$D_i \equiv \sum_j^m s_{ij} = \sum_j^m x_{ij} a_{ij} \geq MD_i$$



b)  $S_j$ : 各営業日のスタッフの実際の出勤人数  $\geq DS_j$ : 各営業日のスタッフの必要人数

$$S_j \equiv \sum_i^n s_{ij} = \sum_i^n x_{ij} a_{ij} \geq DS_j$$

[目的セル] 最小化

次に示す全スタッフの給与の総額  $TC$  を最小化する。

$$TC = c \sum_j s_j = c \sum_i \sum_j s_{ij} \quad \text{全スタッフの給与の総額}$$

表3 シフト(出勤割当)問題(SM3)のソルバー計算の表 単位コスト  $c=1$  として計算

表A 出勤可能日 $a_{ij}$ (0=出勤不可 1=可能)						
	1月1日	1月2日	1月3日	1月4日	1月5日	$MD_i$ :各スタッフの希望最低出勤日数
スタッフ1	1	1	1	0	1	2
スタッフ2	1	0	1	0	1	3
スタッフ3	1	1	1	1	0	3
スタッフ4	1	1	0	1	1	3
$DS_j$ : 各日の必要人数	4	3	3	2	2	
表B 計算用の変数 $x_{ij}$ (0=欠勤 1=出勤) (出勤の可否を考慮せずに変化)						
	1月1日	1月2日	1月3日	1月4日	1月5日	
スタッフ1	1	1	1	0	0	←変数セル
スタッフ2	1	0	1	0	1	
スタッフ3	1	1	1	1	0	
スタッフ4	1	1	0	1	1	
表C 実際の出勤の変数 $s_{ij} = a_{ij} \times x_{ij}$ (0=欠勤 1=出勤) 計						
	1月1日	1月2日	1月3日	1月4日	1月5日	$D_i$ :各スタッフの実際の出勤日数
スタッフ1	1	1	1	0	0	3
スタッフ2	1	0	1	0	1	3
スタッフ3	1	1	1	1	0	4
スタッフ4	1	1	0	1	1	4
計	$S_j$ :各日の実際の出勤人数	4	3	3	2	2
表C=表A×表B 目的セル: 最小化 全給与(出勤時間)→						14

上の問題については、各種の修正・応用が考えられる。それらを以下に述べる。

(1) 各スタッフの出勤日の希望度を設定

各スタッフに出勤日毎の希望の程度を希望度として提示させて、それと実際の出・欠(1 0)の積の合計を実現希望量とする。その際、各スタッフが提示する希望度の合計に上限を設けた方が、不公平がない。全スタッフの実現希望量の合計の最大化を目的とする。そうすると、上の例のように賃金の総額の

最小化を目的とすることはできないので、賃金の総額の上限を設定すべきかもしれない。その上限の値は何度かソルバー計算を行って算定すべきであろう。各スタッフの実現希望量の合計の差に上限を設けることで、不公平がないように設定することもできる。

#### (2) スタッフの実際の出勤日数と希望出勤日の差の最小化

各スタッフの希望出勤日数( $W_j$ )を設け「 $D_i$ :各スタッフの実際の出勤日数」と「 $W_j$ :各スタッフの希望出勤日数」の差の合計の最小化を目的とすることもできる。(1)よりも、こちらの方がスタッフから聞き出す情報量が少ない。この場合も、賃金の総額の最小化を目的にできないので、賃金の総額の上限を設定すべきかもしれない。

#### (3) スタッフ別の賃金率の設定と仕事係数

上のモデルでは全スタッフで賃金率を一定としたが、スタッフ毎の賃金率を設定することもできる。また、各スタッフの仕事の能力(能力係数)を設定し、労働時間に能力係数をかけて仕事量に変換することもできる。そして各営業日に必要なのは単に出勤するスタッフの数ではなく、スタッフが提供する仕事量の総計とする。

#### (4) 特定の日に特定のスタッフが必要

特定の日に特定のスタッフが必要という設定も可能である。例として1月5日は必ず山田の出勤が必要などである。その場合はある $i, j$ に対して「出勤変数: $x_{ij}=1$ 」という制約を設ける。

#### (5) 複数の勤務種別

日勤や早出、遅出のような複数の勤務種別の組合せを考えることも可能である<sup>13)</sup>。

### (SM4) 在庫問題 新聞タイプの商品の場合

ここでは新聞やケーキ、刺身、パンなどのように消費期限が短く、販売日(販売期間)を過ぎると価値がなくなる商品を考える。これを新聞タイプの商品と呼ぶ。売残りは毎日無料で廃棄されるものとする。与えられた毎期の需要量に対して売上量と粗利益を計算する。毎期の需要量として実際の過去の需要量を想定することができる。

商品の売値を $p$ 、仕入値を $c$ とする。販売日数を $N$ とし $t$ を期(販売日)とする。 $t$ 期の在庫量(期首の在庫量:仕入量に等しい)を $a_t$ 、売上量を $y_t$ 、需要量を $x_t$ 、粗利益を $R_t$ とする。全販売期間における粗利益 $R_t$ の合計を $R$ とする。売り切れた日数を $N_s$ とする。

在庫の計算では、需要量 $x_t$ が在庫量 $a_t$ 未満のときは「売上量 $y_t$ =需要量 $x_t$ 」(売残り)となり、 $x_t$ が $a_t$ 以上のときは「売上量 $y_t$ =在庫量 $a_t$ 」(売切れ)となる。よって粗利益の合計 $R$ について次が成り立つ。

$$R = \sum_{t=1}^{t=N} R_t = \sum (py_t - ca) \quad (y_t = x_t \text{ if } x_t < a, \quad y_t = a \text{ if } a \leq x_t) \quad (1)$$

$$= p \left( \sum_{x_t < a} y_t + \sum_{a \leq x_t} y_t \right) - Nca = p \left( \sum_{x_t < a} x_t + N_s a \right) - Nca$$

最後の式をExcelの関数で表現すると次のようになる。Dを全販売期間の需要量のシート上の入力範囲とする。右辺のカッコ内の左の項は売残りの場合の、右の項は売切れの場合の売上量である。

$$R = p \left( \text{SUMIF}(D, "<a") + \text{COUNTIF}(D, ">=a") \times a \right) - Nca \quad (2)$$

上で使ったExcelの関数の書式と機能は、「COUNTIF(範囲, 条件) …範囲内で条件を満たすセルの個数を返す」「SUMIF(範囲, 条件) …範囲内で条件を満たすセルの数値の合計を返す」である。

次に複数の商品の在庫問題を考える。商品数を  $M$  とする。したがって上の各変数に商品  $i$  の添え字が追加される。全商品の粗利益の合計を  $TR$  として、その最大化問題を考える。ただし在庫・陳列面積に限りがあると、店全体の在庫・陳列面積を  $TB$  とおく。各商品  $i$  の面積係数つまり商品一個当りに必要な在庫・陳列面積を  $\beta_i$  とおくと、 $\beta_i$  と  $a_i$  の積の合計が  $TB$  以下となる必要がある。商品  $i$  の全販売期間における粗利益を  $R_i$  とおき、その合計  $TR$  を最大化する。 $R_i$  は (1) (2) で示される  $R$  と同じである。よって次の制約付き最大化問題となる。

$$\text{最大化 全商品の粗利益の合計 } TR \equiv \sum_{i=1}^{i=M} R_i \quad \text{s.t.} \quad \sum_{i=1}^{i=M} \beta_i a_i \leq TB \quad \text{在庫面積の制約}$$

- <ソルバー・モデル SM4> 在庫の計算 実際のパラメータ・係数の値は表を参照
- [所与]  $x_{it}$ : 商品  $i$  の  $t$  期の需要量  $p_i$ : 商品  $i$  の売値  $c_i$ : 商品  $i$  の仕入値  $TB$ : 店全体の在庫面積  
 $N$ : 販売日数  $M$ : 商品数  $t=1, \dots, N$   $i=1, \dots, M$
- [変数セル]  $a_i$ : 商品  $i$  の在庫量 (仕入量)
- [制約]  $\sum_i \beta_i a_i \leq TB$  面積制約
- [目的セル] 最大化  $TR \equiv \sum R_i$  全商品の全販売期間における粗利益合計

次の表は上の在庫計算のシステムを表にし、ソルバーで解けるように設定したものである。数値例と解も表示されている。売切れ日数を  $NS_i$ 、売切れ率を  $SR_i (=NS_i/N)$  として、それらも表示している。

表4 在庫計算 (新聞タイプ) の (SM4) ソルバー計算の表

財の数	$N_{good}$	5	総所与在庫面積	$B$	110		
財番号	$i$	1	2	3	4		
売値	$p_i$	360	600	740	900		
仕入値	$c_i$	200	320	440	800		
資源係数	$\beta_i$	2.5	3.4	6	5.2		
日数	$N_i$	20	20	20	20		
利用在庫面積	$\beta_i a_i$	27.5	71.4	11.1	0.0		
		総利用在庫面積					
		110					
在庫量	$a_i$	11	21	2	0		
条件		← 変数セル					
if $x_i < a_i$	売残り時の全売上額	$R1_i$	7,560	28,800	0	0	← $p_i \text{SUMIF}(D, x_{it} < a_i)$
if $a \leq x_i$	売切日数	$NS_i$	18	17	20	20	← $\text{COUNTIF}(D, a_i \leq x_{it})$
	売切れ時の全売上額	$R2_i$	71,285	214,200	27,375	0	← $NS_i p_i a_i$
	全仕入額	$TC_i$	44,003	134,400	16,277	0	← $N_i C_i$
	総粗利益	$Ri = R1_i + R2_i - TC_i$	34,842	108,600	11,098	0	総粗利益の計 $TR$
	売切れ率	$SR_i$	0.90	0.85	1.00	1.00	154,540
		需要量					
需要量のデータ範囲: D →	$t$	$x_1$	$x_2$	$x_3$	$x_4$		← 以下、20期まで需要量のデータがあるが紙幅の関係で省略
	1	27	22	23	41		
	2	23	32	26	42		
	3	25	21	27	44		

新聞タイプの商品については「仕入値/売値= $c/p$ =売切れ率」となるように仕入量を決めれば、期待利潤が最大化する」という最適在庫量に関する定理(新聞売り子の定理)があり、 $p$ と $c$ 、確率分布が分かっていたら解析的に最適解が得られることになる。ここでは確率分布の代わりに需要量そのものを与えており、これを度数分布として捉え、さらに確率分布とみなすこともできるが、ソルバーを使えば、そのような定理を使わなくても、需要量が与えられている場合には上のように解くことができることになる。

ところで、ここでは与えられた需要量を過去の観測データとしてきたが、ある確率分布にしたがう乱数(整数)とすることもできる。さらにその確率分布は過去の需要データから推測されたものとすることもできる。その意味では、ここで行った過去の需要データをそのまま使う方法は、過去のデータから確率分布を推定し、その分布にしたがう乱数を発生し需要量とするという過程を飛ばした直接的な分析方法とも言える。

なお、新聞タイプ以外の消費期限が十分に長い商品についても表の計算システムを修正することで、上と同様な分析ができると考えられるが、より複雑になる。

上では述べたもの以外にも、変数の制約・条件・関係をソルバー上で設定する工夫・技術がある。それらを利用すると多様な設定が可能になる<sup>14)</sup>。

## V ExcelソルバーとVBAの組合せによる問題の解法

ExcelのソルバーとVBAを組み合わせることで、より複雑かつ実践的な問題を分析計算できることを示す。具体的には、金融ポートフォリオにおける効率的フロンティアの計算と作図およびタクシー運転手を例とした賃金の歩合とエージェントの行動の計算である。後者はエージェントの効用関数を前提としている。

### 5.1 株式の効率フロンティアの導出

ポートフォリオ選択では、複数の証券への投資を組合わせて、収益率のリターン(期待値)をできるだけ大きくし、収益率のリスク(標準偏差)をできるだけ小さくしようとする。金融の効率的フロンティアというのは、与えられた収益率のリターンの各値を維持しながら、複数の証券への投資配分によって最小化したリスクをグラフにしたものである。ここでは証券を株式とし、株式 $i$ の利益率のリターン $r_i$ と株式 $i$ と $j$ の共分散 $\sigma_{ij}$ を所与として、株式への投資の組合せがもたらす効率的フロンティアを計算する<sup>15)</sup>。

まずある値 $r_g$ 以上の収益率のリターンを維持しながら、リスク $\sigma_p$ を最小化するソルバー・モデルSVF( $r_g$ )を考える。投資全体の収益率のリターン $r_p$ が与えられると、リスク $\sigma_p$ の値は下記のモデルによって個々の株式の $r_i$ と $\sigma_{ij}$ と投資比率 $x_i$ から計算される<sup>16)</sup>。この $r_p$ がある値 $r_g$ 以上という制約の下で $\sigma_p$ を最小化する。

今このモデルのパラメータはリターンの値 $r_g$ のみとみなす。厳密に言えば、その他に株の種類や、データをとった期間もパラメータとみることができが、ここでは捨象する。収益率は月次で考える。空売りはないものとする。投資比率は0.001を単位とする。ただしソルバーでは $0 \leq v_i \leq 1000$ の範囲の整数 $v_i$ を変数セルとしてそれを1000で割っている。

<ソルバー・モデル：SVF( $r_g$ )> 収益率のリターンを維持してリスクを最小化する

[所与]  $r_i$ ：株式 $i$ の収益率の期待値  $\sigma_{ij}$ ：株式 $i$ と $j$ の共分散  $r_g$ ：総収益率のある値

[変数セル]  $x_i$ ：株式 $i$ への投資比率(単位は0.001)

整数  $v_i$  を  $0 \leq v_i \leq 1000, \sum v_i \leq 1000$  の変数セルとして  $x_i = v_i/1000$  とする

[制約]  $0 \leq x_i \leq 1 \quad \sum x_i \leq 1 \quad r_p = \sum r_i x_i \geq r_g$  (計算された総収益率が ある値  $r_g$  以上)

[目的セル] 最小化  $\sigma_p = \sqrt{\sigma_p^2} = \sqrt{\sum_i \sum_j \sigma_{ij} x_i x_j}$  投資全体の収益率のリスク

上のソルバー・モデル：SVF( $r_g$ )をVBAによって $r_g$ を変化させながら $x_i, r_p, \sigma_p$ を計算した結果が表5と図2である。図2の効率フロンティアは表5の計算結果から作図している。

表5 株式の効率フロンティアの導出 (VBAによるソルバー・モデル：SVF( $r_g$ )の連続実行)

	↓目的セル	↓ $r_g$ : 所与リターン (VBAによって変化させるパラメータ)	$r_g \leq r_p$							
$\sigma_p^2 =$	↓最小化(分散)	0.090		↓計			↓計			
$\Sigma_i (x_i \Sigma_j \sigma_{ij} x_j)$	0.020	$r_p = \Sigma x_i r_i$ 0.090		$\Sigma x_i \rightarrow$ 1.000	[ $\sigma_{ij}$ ] 5つの株価の共分散行列				1000	
$\Sigma_j x_i \sigma_{ij}$	0.021	$r_i \rightarrow$ 0.091		$x_i \rightarrow$ 0.972	0.021	0.000	0.003	-0.005	-0.002	972
	0.000	0.051		0.000	0.000	0.004	-0.001	0.002	0.000	0
	0.003	0.044		0.000	0.003	-0.001	0.001	-0.001	-0.001	0
	-0.005	0.041		0.008	-0.005	0.002	-0.001	0.006	0.001	8
	-0.002	0.041		0.020	-0.002	0.000	-0.001	0.001	0.001	20
	↑リスクの計算	↑株式iのリターン		↑ $v_i/1000 = x_i$					$v_i : 1 \sim 1000$ の整数(変数セル) ↑	

※最小化した分散の平方根を下の $\sigma_p$ としている。 ↑整数 $v_i$ を小数 $x_i : 0 \sim 1.000$ に変換

----- 以下  $r_g$ を変化させながらVBAによって計算( $\sigma_p$ と $r_p$ の組合せが効率フロンティア) -----

番号	標準偏 差:リスク	所与リター ン	計算リ ターン
	$\sigma_p$	$r_g$	$r_p$
1	0.015	0.042	0.042
2	0.016	0.045	0.045
3	0.020	0.048	0.048
4	0.025	0.051	0.051
5	0.032	0.054	0.054
6	0.038	0.057	0.057
7	0.046	0.060	0.060
8	0.055	0.063	0.063
9	0.064	0.066	0.066
10	0.073	0.069	0.069
11	0.083	0.072	0.072
12	0.093	0.075	0.075
13	0.103	0.078	0.078
14	0.113	0.081	0.081
15	0.122	0.084	0.084
16	0.132	0.087	0.087
17	0.142	0.090	0.090

株式 i への投資比率					
$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	計
0.005	0.143	0.439	0.000	0.371	0.958
0.028	0.161	0.427	0.003	0.381	1.000
0.094	0.154	0.316	0.042	0.393	1.000
0.161	0.147	0.206	0.081	0.405	1.000
0.228	0.140	0.095	0.120	0.417	1.000
0.292	0.144	0.000	0.140	0.425	1.000
0.354	0.128	0.000	0.120	0.398	1.000
0.417	0.110	0.000	0.098	0.374	1.000
0.479	0.098	0.000	0.083	0.340	1.000
0.541	0.083	0.000	0.065	0.311	1.000
0.604	0.066	0.000	0.044	0.286	1.000
0.666	0.052	0.000	0.026	0.256	1.000
0.728	0.035	0.000	0.005	0.231	1.000
0.793	0.009	0.000	0.003	0.196	1.000
0.854	0.000	0.000	0.019	0.128	1.000
0.913	0.000	0.000	0.011	0.076	1.000
0.972	0.000	0.000	0.008	0.020	1.000

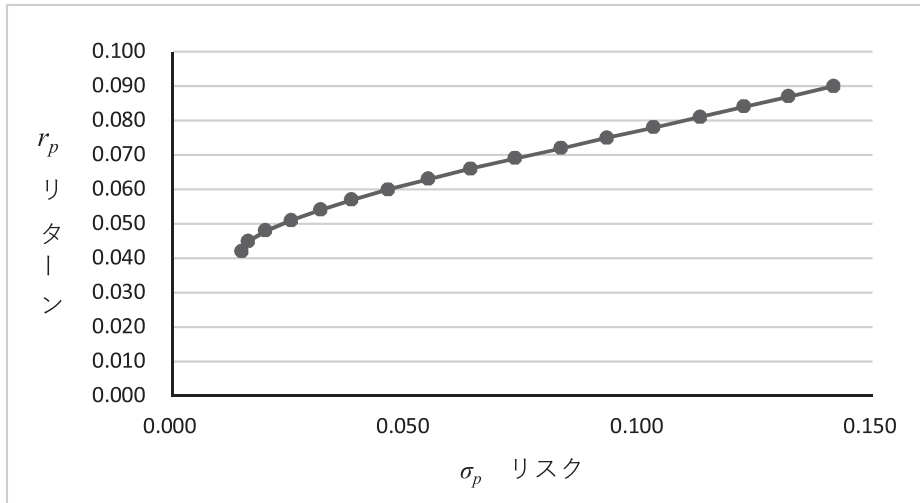


図2 株式の効率フロンティア (表5からの作図)

表5の計算つまりSVF ( $r_p$ )の連続実行を行なったVBAのコードの主要部分を次に示す。VBAの利用者にとっては特に難しいところはないと思われる。VBAのソルバーの利用に関する関数については後で説明している。

```

' ***** <株式の効率フロンティアの導出VBAのコード 主要部分> *****
' 株式の効率フロンティアの導出：ソルバー・モデル：SVF (rg)の連続計算
' 初期値と幅は計算範囲用のもので表5の外にある
NPF = 5: 初期値 = Cells (14, 16): 幅 = Cells (14, 17) 'NPF…株式の個数

For i = 1 To 17
  所与リターン = 初期値 + (i - 1) * 幅 : Cells (3, 5) = 所与リターン
  SolverOk SetCell:="$C$4", MaxMinVal:=2, ByChange:="$M$5:$M$9" 'ソルバー・モデルの基本設定
  SolverSolve UserFinish:=True 'ソルバーの計算を開始

  全分散 = Cells (4, 3): 計算リターン = Cells (4, 5)
  Cells (14 + i, 3) = Sqr (全分散)
  Cells (14 + i, 4) = 所与リターン : Cells (14 + i, 5) = 計算リターン

S = 0 'xi…投資比率 S…xiの合計
For j = 1 To NPF
  Cells (14 + i, 7 + j) = Cells (4 + j, 7): S = S + Cells (4 + j, 7)
Next j
  Cells (14 + i, 7 + NPF + 1) = S
Next i
'*****

```

< VBAでのソルバーの利用の設定とソルバー関連の関数 >

VBAでソルバーを利用する前にソルバー・アドインへの参照を設定する必要がある。Visual Basic Editor (VBAのエディター画面)で、[ツール]メニューの[参照設定]をクリックし[参照可能なライブラリ ファイル]の[Solver]チェック・ボックスをオンにする。

本稿においてVBAでソルバーを利用する命令を行なったときは、次の(3)(4)の関数を使った。他にもソルバーに関する関数があるが、ここでは説明しない。<sup>17)</sup>

ソルバーの基本的なモデル設定

SolverOk (SetCell, MaxMinVal, ValueOf, ByChange, Engine, EngineDesc) (3)

各引数の説明をする。引数は基本的にバリエーション型である。またいくつかの引数は省略可能である。[ ]はソルバー：パラメータ設定のダイアログ・ボックスまたはチェック・ボックスである。

SetCell … [目的セル]に相当。単一セルを指定。  
 MaxMinVal … [最大値][最小値][値]に相当。左から順に1, 2, 3で指定する。  
 ValueOf … 上でMaxMinValに3を指定した場合の目的セルの目標値である。  
 ByChange … [変数セル]に相当。セル範囲を指定する。  
 Engine … [解決方法の選択]の一覧に相当。問題の解決方法を選ぶ。Simplex LPの場合は1, GRG Nonlinearの場合は2, Evolutionaryの場合は3を指定する。  
 EngineDesc …上のEngineを文字列で指定する。

ソルバーの計算を開始

SolverSolve ( UserFinish , … ) (4)

UserFinish …Trueを指定すると、[ソルバー：探索結果]のダイアログ・ボックスを表示せずに結果を返す。Falseを指定するか省略すると、上のダイアログ・ボックスを表示して結果を返す。

他の引数の説明は省略する。

## 5.2 歩合率のエージェントの行動への影響

エイジェンシー関係とは、プリンシパル(依頼人・雇用者)がエージェント(代理人・被雇用者)を雇用しているが、単純に労働時間だけで売上額・生産額および給与が決まるのではなく、直接的な観測が難しいエージェントの仕事への努力・集中が売上げに大きく影響するような雇用関係をいう。プリンシパルがエージェントの行動に関して十分に観察ができず、または多大なモニタリング・コストを要するために、エージェントの利己的な行動によってプリンシパルの利益が大きく影響を受けることにもなる。ここでは、プリンシパルをタクシー会社、エージェントをタクシー運転手と想定して分析するが、その他のエイジェンシー関係にも適用できる。

エージェントである運転手(Tとする)の給与 $I$ は、固定給付きの歩合制であり、固定給 $I_0$ と売上額 $s$ の一定の割合( $w$ )を得るとする。エージェントの間接効用関数をCES (Constant Elasticity of Substitution) 関数とし、その変数を余暇 $L$ と給与 $I$ とする。<sup>18)</sup>売上額 $s$ は仕事にかかる努力量 $x$ の関数( $s=f(x)$ : 売上関数と呼ぶ)であるとする。努力量 $x$ は単に労働時間だけでは測れず、仕事への真剣さや集中力などを含むものとする。ただし、ここでは単純に余暇 $L$ と努力量 $x$ の合計を24(時間)とする( $L + x = 24$ )。運転手は、

与えられた固定給  $I_0$  と歩合  $w$  に対して効用関数  $U(L, I)$  を最大するように、エージェントは努力量  $x$  を決める。 $x$  が決まると売上額  $f(x)$  と余暇  $L$  と給与  $I = I_0 + wf(x)$ 、そして効用水準  $U(L, I)$  が決まり、また運転手の売上額から給与を差し引いた残りである会社の収益  $R = f(x) - (I_0 + wf(x))$  も決まる。ただし、上述の  $L + x = 24$ 、その他の設定にはいくつか問題があると思われる。それについては後で詳しく考察する。

以下にソルバー・モデルを記述する。 $f(x)$  の具体的な関数と各パラメータの値は次のモデルまたは表に示す。これらは詳細な検討によって決められたものでない。変数は非負とする。

<ソルバー・モデル：SVA( $w$ ) 歩合の変化とエージェントの行動

[所与]  $U(L, I) = A[\alpha L^{-\beta} + (1-\alpha)I^{-\beta}]^{-1/\beta}$  効用関数とパラメータ  $\alpha$  と  $\beta$   $A=1$   $I_0$ : 固定給

$f(x) = 6x^{0.6}$  売上関数

[変数セル]  $x$  努力量

[制約]  $L + x = 24$  余暇時間 + 努力 = 24  $I = I_0 + wf(x)$  給与 = 固定給 + 歩合給

[目的セル] 最大化  $U(L, I)$  効用水準

表6 歩合率の変化とエージェントの行動 (VBAによるソルバー・モデル：SVA( $w$ )の連続実行)

$A$	$\alpha$	$\beta$	$U(L, I) = A[\alpha L^{-\beta} + (1-\alpha)I^{-\beta}]^{-1/\beta}$
1	0.4	0.9	
$L$ 余暇時間	$I$ 所得	$U$ 効用関数	←目的セル
16.1	24.1	20.1	
$x$ 努力量	$s = f(x)$ 売上額	$R$ 会社収益	(パラメータ) $w$ 歩合の変化
7.9	20.8	-3.3	
固定給 $I_0$	$w$ 歩合	$w_0$ 初期値	$\Delta w$ 幅
5	0.92	0.35	0.03

以下 VBAによるソルバー・モデルの連続計算の結果

$w$	歩合	$x$ 努力量	$s = f(x)$ 売上額	$I$ 所得	$R$ 会社収益	$U$ 効用関数
0.35		9.21	22.73	12.96	9.78	13.64
0.38		9.15	22.64	13.60	9.04	14.08
0.41		9.08	22.55	14.24	8.30	14.51
0.44		9.01	22.44	14.88	7.57	14.92
0.47		8.94	22.34	15.50	6.84	15.32
0.50		8.87	22.23	16.12	6.12	15.71
0.53		8.80	22.12	16.72	5.40	16.08
0.56		8.73	22.01	17.33	4.69	16.45
0.59		8.66	21.91	17.92	3.98	16.80
0.62		8.59	21.80	18.52	3.28	17.14
0.65		8.52	21.69	19.10	2.59	17.48
0.68		8.45	21.59	19.68	1.91	17.80
0.71		8.38	21.48	20.25	1.23	18.12
0.74		8.31	21.38	20.82	0.56	18.43
0.77		8.25	21.28	21.39	-0.11	18.73
0.80		8.18	21.18	21.94	-0.76	19.02
0.83		8.12	21.08	22.50	-1.42	19.31
0.86		8.06	20.99	23.05	-2.06	19.59
0.89		8.00	20.89	23.60	-2.70	19.87
0.92		7.94	20.80	24.14	-3.34	20.14



表6の計算結果が示すように、歩合率 $w$ が増えるにつれて努力量 $x$ が減っている。つまり余暇 $L$ は増えている。よって売上額 $s$ が減り、Tの所得も減り同時に会社の収益 $R$ も減っている。ただ、ここでは示さないが、効用関数のパラメータの値によっては $w$ が増えるにしたがい $x$ が増える場合がある。例えば $\beta = -0.4$ のときは $w$ が増えるにしたがい $x$ が増える。これ以上の分析は行わないが、各パラメータと努力量の動きの関係を分析することも興味深い。ただし、効用関数が解析的に記述される場合は、解析的な分析によってもこれらのことが計算できるであろう。

上の計算より、会社は $w$ の変化にしたがい $x$ の変化をみて、売上額 $s$ そして会社の収益 $R$ が大きくなるように $w$ の値を調整できる。ただし運転手Tが会社を辞めないように、Tの効用をあるレベル以上に保つ、という参加制約を守る必要がある。またパレート改善可能な、つまりエイジェントであるTの効用を高め、かつプリンシパルであるタクシー会社の収益も増加する $w$ の範囲を見つけることが可能になるかもしれない。

今回は行わないがVBAによって歩合率 $w$ と同時に固定費 $I_0$ も変化させて、努力量 $x$ や売上額 $f(x)$ 、会社の収益 $R$ の変化を計算することもできる。これによって、パレート改善可能な $w$ と $I_0$ の範囲が見つかる可能性が高まるかもしれない。

上のモデルの設定上の問題点と課題を考察する。

- (1) 売上が単に労働時間の関数ではなく、仕事に注ぐ努力量の関数とすれば「24=余暇(時間) + (仕事への)努力量」のような単純な関係ではなくなるであろう。一般的な人間の心理から言えば、現時点での努力量が大きいほど、それ以上の努力量の増加に伴う代償として減少する余暇(または精神的余裕)は大きくなると思われる。
- (2) 同じ努力量に対する売上額は確率的とする方が一般的と考えられるが、今回はそれがモデル化されていない。固定給はその変動に対してエイジェントの給与の安定をもたらす装置としての役割を持つ。また仕事に注ぐ努力量も、エイジェントの調子や気分の波があり確率的かもしれない。
- (3) ここでは、売上関数 $f(x)$ や効用関数 $U(L, I)$  またそのパラメータ $\alpha$ と $\beta$ の値を仮に設定したが、現実的に応用するとすれば、より詳細な検討による特定が必要である。理論的考察とともに、観測データやアンケート、インタビューからそれを特定しなければいけない。
- (4) 当然個人差があり、それが効用関数や売上関数またそのパラメータの値の違いとして反映される。各賃金体系の下で異なるパラメータを持つ複数のエイジェントがどのような行動をとるかを、VBAとソルバーでシミュレーションし、それを集計して会社の収益がどのように変化するか見て、どの賃金体系がプリンシパルにとって有利かを分析することができる。売上額のレベルの違いによる歩合率や固定給の変更を考えることもできる。また複数の賃金体系を設定し、そこからエイジェントが自ら望む賃金体系を選んで行動するシミュレーションを行うことも興味深い。
- (5) 効用関数を用いた分析が適切なものか否かは議論の余地がある。効用関数を仮定しないエイジェント・モデルも検討すべきである。

以上の点からも、今回の分析は実践的な点では未完成な部分が多く、より実践的な分析・モデルのための一つの方法・形を示したに過ぎないと言える。ただ、VBAによってソルバー・モデルのパラメータを変化させながら、簡単に最適解を得ることは、エイジェンシー関係に関する分析の有力な手法となり得る。ここでの分析はその基礎を与えるものと言える。

## むすびに

Excelは普及率が高く、誰もが簡単に利用環境を整えることができる表計算ソフトであり、初心者にも分かり易い。つまり入口が広い。しかし、同時に奥が深く、一歩踏み込めば専門的な分析もできるソフトである。専門的な分析としては、専門的な各種の関数、データ分析機能、シート上での計算処理が可能なプログラミング言語であるExcel VBAなどが挙げられよう。そしてそれに加えて、本稿で取り上げたソルバーが挙げられる。ソルバーは、問題を定式化するだけで解くことができ、Excel VBAに較べて利用に要する知識は多くないと言える。ただその分、Excelソルバーの利用能力において他のユーザーと差を付けるには、ソルバーによって諸種の問題を解く力が必要であり、そのためには応用のための様々な知識や技術を獲得しなければいけない。それにはより様々なタイプの問題をソルバーで解く必要がある。

本稿では経済・経営に関する問題を中心にExcelソルバーの利用例、またExcel VBAと組合せることでより複雑な問題が解ける例を示したが、その有用性を考えれば、今後さらにビジネス・パーソンや経営戦略を担う立場のユーザーが利用すべきであると考え。よってビジネス・データを扱う大学の科目やコースにおいても積極的に取り入れられるべきだと思う。実際、筆者は担当している経済学部のビジネスデータ分析の講義でExcelのソルバーを教えてきた。今後も、多くの利用例を示しつつその有用性を伝えていきたいと考えている。

以 上

### 付録 巡回セールスマン問題についての考察<sup>19)</sup>

巡回セールスマン問題について考察する。この問題についてソルバーで定式化を行って計算したが、正しい計算が得られないか、または解が不安定であったので、ここで定式化のみを次に簡潔に提示する。解が上手く得られないのは、問題が不連続または凸計画でないためと思われる。その点での改善が必要になる。

巡回セールスマン問題とは、複数の都市を巡回するときの総移動費用を最小化する問題である。指定された全ての都市を一巡し、出発地に戻る。各2都市間の移動コストは与えられている。各都市を $\{C_1, C_2, \dots, C_n\}$ とすると、それらを並び変えることによって作り出される順列の順で移動した場合の移動コストを計算して、その合計が最小になるような順列を探し出すことがこの問題を解くことになる。

したがってまずExcelで順列を作ることになるが、その方法として、まず $1 \sim n$ までの整数の任意の同じ並びを、基点(一つ前のセル)を同じセルとする行(表頭)と列(表側)に入力し、その差を計算する表をその内側に作る。その差は全部で $n \times n$ 個あるが、その整数の並びが順列ならば、その内0となる差の個数は $n$ 個となる。なぜならば、順列ならば同じ数値は一度しか出てこず、表の対角にしか0が生じないからである。Excelのシート上ではCONUTIF関数を使って表の0を数え、それが $n$ に等しいことを制約とする。

作られた順列にしたがって移動するコストは、各都市間の移動コストの表を予め作成しておいて、先に作った順列と、任意のセルの値を相対的な位置で返すOFFSET関数を利用して計算する。OFFSET( $P, X1, X2$ )の $P$ に移動コストの表の基点(固定)、順列にしたがって、 $X1$ に順列のある番号(移動前の都市番号)、 $X2$ に順列の次の番号(移動先の都市番号)を入れていき、最後に最初の都市に戻る。したがって $n$ 回OFFSET関数を利用する。そしてその合計が最小化すべき全移動コストとなる。次にそのソルバー・モデルを示す。

<巡回セールスマン問題のソルバー・モデル>

都市に1から $n$ の番号をつけ、順列そのものに番号 $t$ をつける ( $t = 1, \dots, n!$ )。  $p(i,t)$ を順列 $t$ にしたがって都市を並べ変えたときの $i$ 番目の都市の番号とする。  $TC(i,j)$ を都市 $i, j$ 間の移動費用、  $TTC$ を総移動費用とする。

[所与]  $TC(i,j) \quad i, j = 1, \dots, n$       都市 $i, j$ 間の移動費用

[変数セル]  $p(i,t) \quad i = 1, \dots, n \quad t = 1, \dots, n!$       都市を並べる順列

[制約]  $(p(i,t)$ が順列であること)

[目的セル] 最小化      都市を一巡した時の総移動費用 (次式の最後の項は最初の都市に戻る移動費用)

$$\text{総移動費用} \quad TTC = \sum_{i=1}^{n-1} TC(p(i,t), p(i+1,t)) + TC(p(n,t), p(1,t))$$

注

- 1) ソルバーの利用の前に、Excelのオプションでソルバーのアドイン設定を確認しておく必要がある。その他、利用のための環境設定については藤澤他(2011)が詳しい。
- 2) 藤澤他(2011)のpp50-82に解探索のアルゴリズムについて詳しい説明がある。
- 3) 藤澤他(2011)のpp80-82、後藤(2012)のp181を参照。本稿では最適性の保証の問題については、これ以上論じない。
- 4) 以下の例は、筆者が実際にソルバーで定式化し問題を解いて、例題として授業で受講生に示したものの、あるいはその準備ができていたものである。
- 5) 以下で\*をつけたものは、IVでソルバーの定式化を示したものである。ミクロ経済学の問題の例としては、予算制約下での生産量の最大化を示している。
- 6) 日花(2005)のpp177-184に分析の例がある。
- 7) Wikipediaの「線型計画法」を参考にした。
- 8) 牧野(2007)p9で紹介されている。
- 9) 上田他(2005)のpp105-109を参考にした。また後藤(2012)のpp179-180では輸送問題において固定費が発生する場合の定式化を示している。
- 10) 上田他(2005)のpp110-113を参考にした。
- 11) 上田他(2005)のpp114-117を参考にした。
- 12) 後藤(2012)のp180では、ナップザック問題を購読雑誌選定問題に応用している。
- 13) インターネットのサイト [html : http://mt-soft.sakura.ne.jp/web\\_dl/template/shift/shift\\_9/pg\\_2.html](http://mt-soft.sakura.ne.jp/web_dl/template/shift/shift_9/pg_2.html) に詳しく述べられている。詳細は述べないが、勤務種別を変数化し(日勤→1 早出→2 遅出→3 0→休み)とし、また勤務種別のCOUNTIF関数を利用している。その際、目的変数を、過不足(各日の勤務の必要人数と割当て人数の差の合計)として、それを最小にすることができる(可能な限り0に近づける)。このサイトのページでは、1週間の内5日間勤務して2日連休という勤務パターンを設定して、各曜日の必要人数を満たす勤務シフトや連続勤務不可条件の設定など、実践的なシフトのためのソルバーの設定が多く紹介されている。
- 14) 例として、以下のような変数間の制約に関する工夫がある。 $x, y, z$ はバイナリ変数とする。
  - ①  $[x \leq y]$ とおくと、 $y=1$ が $x=1$ の必要条件となる。つまり $x=1$ に対応する事柄が「可・採」となるためには $y=1$ に対応する事柄が「不可・不採」となる必要がある。
  - ②  $C$ を固定費と変動費の合計、 $M$ を $x$ が取り得る上限、 $a$ を固定費とする。 $[t \leq Mz, C = az + ct]$ とおくと、 $0 < t$ の必要条件は $z=1$ であり、このとき固定費用 $a$ が費用 $C$ に加わる。
  - ③  $x + y = 1$ の $x$ と $y$ のどちらか一方だけが1である。 $x + y + z = 1$ も同様。
- ②は後藤(2012)のp179を参照した。
- 15) 効率フロンティアに関する詳しいモデル・説明は枇々木(2016)、枇々木他(2005)などを参照。
- 16) ここで計算に用いた株式の収益率のリターン $r_i$ と共分散 $\sigma_{ij}$ は、名前は示さないが実在する5企業の2016/10から2017/10の月次の株価データからExcelを用いて実際に計算した推定値である。表5では計算に用いた $r_i$ と $\sigma_{ij}$ の値の四捨五入したものを示している。企業の選択は、企業の収益率の平均値と互いの相関係数(共分散)の値を考慮し

て行った。リターンがある程度高く、収益率が主要な動きとは逆の動きをする、つまり相関係数が負となる企業の株が有力な候補となると言える。株の選び方によっては、効率的フロンティアを描く際、選択された株の多くが購入量0という結果になる。この問題をクリアするためには、まずより多くの株で計算した後、この基準で株を絞り込むという2段階の選択が有効だと考える。

- 17) Microsoft社のデベロッパーセンターのサイトの「ソルバー関連のVBA関数を使用する」のページを参照した。このサイトにはより詳しい説明がある。
- 18) CES関数を選んだ理由は、効用関数として標準的であり各タイプの関数を柔軟に内包するからであるが、検討が必要であろう。
- 19) 巡回セールスマン問題については、藤澤他(2011)のpp153-159にもソルバーによる解法の説明があるが、本稿とは変数セル・制約などが異なる。

### 参考文献

- 上田和明, 菊田正男, 淵上美喜, 高橋玲子, 古谷都紀子 (2005) 『Excelで学ぶデータマイニング入門』オーム社。  
 後藤順哉 (2012) 「Excelで始める数理最適化」オペレーションズ・リサーチ, 57巻, 4号, pp175-182。  
 日花弘子 (2005) 『仕事に役立つExcelビジネスデータ分析』ソフトバンククリエイティブ。  
 枇々木則雄 (2016) 「ポートフォリオ最適化入門」オペレーションズ・リサーチ, 61巻, 6号, pp335-340。  
 枇々木則雄, 田辺隆人 (2005) 『ポートフォリオ最適化と数理計画法』朝倉書店。  
 藤澤克樹, 後藤順哉, 安井雄一郎 (2011) 『Excelで学ぶOR』オーム社。  
 牧野真也 (2010) 「Excelによる経済・経営分野の情報処理V—線形計画問題—」和歌山大学経済学部Working Paper Series, ISSN 1343-3997。

### 参考ホームページ

- 1) Wikipediaの「線形計画法」(採録日: 2017年9月16日)
- 2) サイト『ダウンロードセンター』の「勤務シフト表の自動作成」のページ  
[http://mt-soft.sakura.ne.jp/web\\_dl/template/shift/index-jump.html?shift\\_9/main.html](http://mt-soft.sakura.ne.jp/web_dl/template/shift/index-jump.html?shift_9/main.html) (採録日: 2017年9月5日)
- 3) サイト『Microsoft社のデベロッパーセンター』の「ソルバー関連のVBA関数を使用する」ページ  
<https://msdn.microsoft.com/ja-jp/vba/excel-vba/articles/using-the-solver-vba-functions> (採録日: 2017年9月15日)

(2017年11月24日掲載決定)