社会システム分析のための統合化プログラム１０

－トレンドの検定・方程式ソルバー・非線形最小２乗法・多目的線形計画法・待ち行列シミュレータ－

福井正康，巴達仁貴*，細川光浩，奥田由紀恵

福山平成大学経営学部経営情報学科
*福山平成大学大学院経営学研究科経営情報学専攻

概要

我々は教育分野での利用を目的に社会システム分析に用いられる様々な手法を統合化したプログラム College Analysis を作成してきた。今回は新たに統計の分野でトレンドの検定と非線形最小２乗法、数学の分野で方程式ソルバー、ＯＲの分野で多目的線形計画法と待ち行列シミュレータを加えた。特に待ち行列シミュレータはこれまでのプログラムに平均到着数などの条件の時間変化に対応する機能を加えたもので、現実的な場面への適用が可能になった。

キーワード

College Analysis，社会システム分析，OR，統計，トレンドの検定，方程式ソルバー，非線形最小２乗法，多目的線形計画法，待ち行列シミュレータ

URL:　http://www.heisei-u.ac.jp/ba/fukui/

１．はじめに

我々が開発を続けている社会システム分析ソフトウェア College Analysis は Visual Basic.NET への変換後 [1]、ソフトウェアとしてだけでなく、教育手法まで含めた教育システムとして開発が進められ、基本統計と多変量解析については、社会人への講座が開ける程度に整備されてきた。その間新しい分析手法についてもいくつか追加し、プログラムの更新を行っている。この論文では新しく追加した、トレンドの検定、多目的線形計画法、方程式ソルバー、非線形最小２乗法、待ち行列シミュレータなどについて解説する。

トレンドの検定には、比率データの Mantel-extension 法、量的データの Jonkheere 検定と回帰分析による検定が含まれている [2]。方程式ソルバーについては今後改良が加えられると思われるが、非線形方程式をニュートン・ラフソン法 [3] を用いて解くものである。ニュートン・ラフソン法では解の初期値の与え方が問題となるが、実用的に使いやすくなるよう工夫を加えた。非線形最小２乗法でも基本的にニュートン・ラフソン法を利用したが、方程式ソルバーの場合に比べ、解の収束性が悪くなる傾向があり、乱数を利用して粗く解の初期値を与え、それを用いて精度の高い解を求めるという２段階で分析する方法を考えた。多目的線形計画法は複数の目的関数に対する線形計画法で、ここでは目的関数全体の充足率を高めるよう考えられたグローバル評価法と呼ばれる手法が用いられている [4]。待ち行列シミュレータはこれまでの待ち行列シミュレーションのプログラム [5], [1] に時間的に変化する平均到着数、平均サービス数、窓口数に対応する機能を追加したもので、これにより現実的なシミュレーションが可能となった。

２．トレンドの検定

トレンドの検定とはある順番に群を並べた場合に、その群のデータについての比率や平均値などの統計量が次第に大きくまたは小さくなってゆく傾向の有無を調べることである。まず、質的なデータに対する比率のトレンドの検定について説明する [2]。比率のトレンドの検定では Mantel-extension 法が利用されるが、これには以下のように表される統計量 Z または Z' が用いられる。

群 i （$i=1,2,3,\cdots m$）の個体数を n_i，反応した個体数を r_i として以下の量を考える。

$$O=\sum_{i=1}^{m} r_i X_i\ ,\quad E=\left(r\sum_{i=1}^{m} n_i X_i\right)\Big/N\ ,\quad V=\frac{r(N-r)}{N^2(N-1)}\left[N\left(\sum_{i=1}^{m} n_i X_i^2\right)-\left(\sum_{i=1}^{m} n_i X_i\right)^2\right]$$

ここに、$r=\sum_{i=1}^{m} r_i$，$N=\sum_{i=1}^{m} n_i$ である。また X_i については、最も簡単に $X_i=i$ とした。

これらを用いて漸近的に標準正規分布に従う統計量 Z を計算する。

$$Z=\frac{O-E}{\sqrt{V}}\xrightarrow[n_i\to\infty]{} N(0,1)$$

しかし実用上は以下のような Yates の連続補正項を加えた統計量 Z' を用いる場合が多い。

$$Z' = \frac{|O-E|-1/2}{\sqrt{V}} \xrightarrow[n_i\to\infty]{} N(0,1) \text{ の正の部分}$$

量的データに関する Jonckheere の順位和検定は分布によらない検定で、以下のように計算される統計量 Z または Z' を用いる。但し n_i と N についてはこれまでの定義と同じである。

群 i のデータ $x_{i\lambda}$ と群 j （$i<j$）のデータ $x_{j\mu}$ について、$x_{i\lambda}<x_{j\mu}$ なら w_{ij} を 1 増やし、$x_{i\lambda}=x_{j\mu}$ なら w_{ij} を $1/2$ 増やすという処理を群 i と群 j に含まれるすべてのデータについて行う。これは近似的な同順位の処理を行った Wilcoxon の順位和を計算することに等しい。この w_{ij} をすべての i, j （$i<j$）について合計し、以下の量を求める。

$$J=\sum_{i<j} w_{ij}\ ,\quad E=\left(N^2-\sum_{i=1}^{m} n_i^2\right)\Big/4\ ,\quad V=\left[N^2(2N+3)-\sum_{i=1}^{m} n_i^2(2n_i+3)\right]\Big/72$$

これらを用いて漸近的に標準正規分布する以下の統計量 Z を計算する。

$$Z=\frac{J-E}{\sqrt{V}} \xrightarrow[n_i\to\infty]{} N(0,1)$$

しかし実用上は上と同様に Yates の連続補正を加えた統計量 Z' を用いる場合が多い。

$$Z'=\frac{|J-E|-1/2}{\sqrt{V}} \xrightarrow[n_i\to\infty]{} N(0,1) \text{ の正の部分}$$

群 i （$i=1,2,\cdots,m$）の数値 i を説明変数にして、データ $x_{i\lambda}$ を目的変数にする回帰分析もトレンドの検定として考えることができる。即ち、以下のような回帰モデルを考える。

$$x_{i\lambda}=a\cdot i+b+u_\lambda\ ,\quad u_\lambda \sim N(0,\sigma^2)\ ,$$

これを用いて $a\neq 0$ の検定を行い、群の並びでデータの値に傾向性が見られるか調べる。この回帰式の検定については参考文献 6) に詳しいのでここでは省略する。

ここからは具体的な画面を見ていこう。図 2.1 にトレンドの検定の分析メニュー画面を示す。

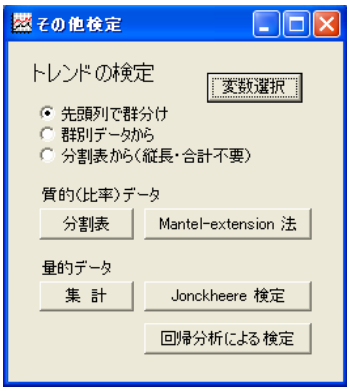


図 2.1 トレンドの検定分析画面

このメニューにはデータ形式の選択ボタンと「変数選択」ボタンがあるが、これらの使い方はこれまでの統計分析のものと同じである。質的データについての「分割表」と量的データについての「集計」も同様である。

図 2.2a のような分割表画面の質的データに対して、データ形式を「分割表から」として「Mantel-extension 法」ボタンをクリックすると図 2.2b のような結果表示画面が示される。

図 2.2a 分割表データ例

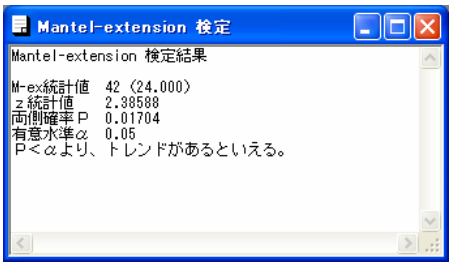


図 2.2b Mantel-extension 検定結果

量的データについては、図 2.3a のようなデータに対して、データ形式を「先頭列で群分け」として「Jonckheere 検定」ボタンをクリックすると、図 2.3b のような結果表示画面が得られる。また同じデータに対して、「回帰分析による検定」ボタンを押すと図 2.3c のような画面が示される。回帰分析による検定は図 2.3a のような先頭列で群に分けられたデータのみ利用可能である。

図 2.3a トレンドの検定量的データ例

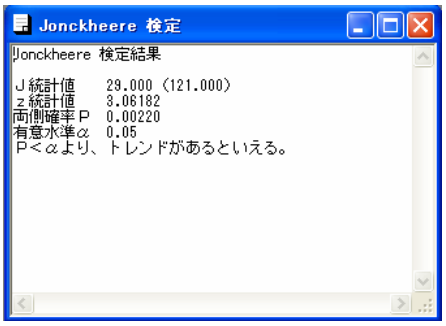


図 2.3b Jonckheere 検定結果

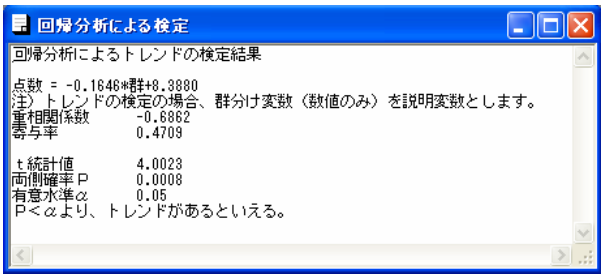


図 2.3c 回帰分析による検定結果

3．方程式ソルバーと非線形最小２乗法

3.1 方程式ソルバー

方程式ソルバーは非線形の連立方程式を解くツールであり、今後のプログラム発展のための基礎技術を得る問題でもある。

非線形連立方程式を以下の形に書こう。

$$f_i(x_1, x_2, \cdots, x_n) = 0 \qquad (i = 1, 2, \cdots, n) \tag{1}$$

ここでは上の方程式に対してニュートン・ラフソン法を適用して解を求める方法を示す[3)]。

今以下の関数を考える。

$$y = f_i(x_1, x_2, \cdots, x_n) \qquad (i = 1, 2, \cdots, n)$$

これらの関数に対して変数 x_i （$i = 1, 2, \cdots, n$）に $x_i = x_i^0$ の初期値を与える。この初期値における各関数の接平面（$n-1$ 次元平面）を考えるとその方程式は以下になる。

$$y = \sum_{j=1}^{n} f_{i,j}^0 (x_j - x_j^0) + f_i^0 \qquad (i = 1, 2, \cdots, n)$$

ここに、 $f_i^0 = f_i(x_1^0, x_2^0, \cdots, x_n^0)$， $f_{i,j}^0 = \left. \dfrac{\partial f_i(x_1, x_2, \cdots, x_n)}{\partial x_j} \right|_{x_1 = x_1^0, x_2 = x_2^0, \cdots, x_n = x_n^0}$

これらの接平面が、 $y = 0$ 平面と交わる点を求めると以下の方程式となる。

$$\sum_{j=1}^{n} f_{i,j}^0 (x_j - x_j^0) = -f_i^0 \qquad (i = 1, 2, \cdots, n)$$

これをクラメールの方法で解くと以下の解を得る。

$$x_j = x_j^0 + \frac{M_j}{D} \qquad (j = 1, 2, \cdots, n)$$

ここに、 $M_j = \begin{vmatrix} f_{1,1}^0 & \cdots & -f_1^0 & \cdots & f_{1,n}^0 \\ f_{2,1}^0 & \cdots & -f_2^0 & \cdots & f_{2,n}^0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ f_{n,1}^0 & \cdots & -f_n^0 & \cdots & f_{n,n}^0 \end{vmatrix}$ （j列）, $D = \begin{vmatrix} f_{1,1}^0 & \cdots & f_{1,j}^0 & \cdots & f_{1,n}^0 \\ f_{2,1}^0 & \cdots & f_{2,j}^0 & \cdots & f_{2,n}^0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ f_{n,1}^0 & \cdots & f_{n,j}^0 & \cdots & f_{n,n}^0 \end{vmatrix}$ （j列）

この x_j（$j = 1, 2, \cdots, n$）を新しい初期値 x_j^1 として上の処理を繰り返す。我々のプログラムでは収束条件を、ある微小指定値 ε に対して以下とした。

$$\max_j \left| x_j^r - x_j^{r-1} \right| < \varepsilon$$

ニュートン・ラフソン法で問題になるのは、初期値の与え方である。我々はこれに対して少し時間がかかるが乱数で初期値を与えて、収束する解を集める方法を選択した。もちろんすべての解が求まる保証はないので、解の個数には十分注意する必要がある。

実際のプログラムの実行画面を図 3.1 に示す。

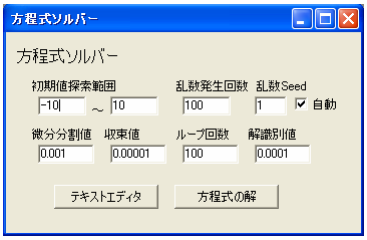


図 3.1 方程式ソルバーメニュー画面

メニューの「テキストエディタ」ボタンをクリックして表示されるテキストエディタに方程式を入力し、「方程式の解」ボタンをクリックして解を求める。その実行画面を図 3.2a と図 3.2b に示す。図 3.2b の検算の部分は誤差などによる不当な結果を排除するためのものである。正しい解であれば、「0」の値になる。

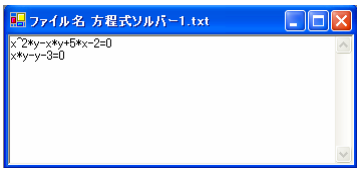


図 3.2a エディター画面

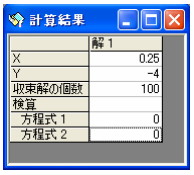


図 3.2b 計算結果画面

解の探索条件はメニューの「初期値探索範囲」と「乱数発生回数」や「乱数 seed」などで与えられる。「初期値探索範囲」は変数ごとではないので細かな設定方法ではないが、解が全く分からない最初の設定としては最も簡単である。またニュートン・ラフソン法に使われる微分を計算

するための「微分分割値」や解の収束を判断する「収束値」及び、逐次計算を行う最大回数である「ループ回数」や異なった解が計算された場合それを判定する「解識別値」も計算に利用される。それらのデフォルト値はメニューにある通りであるが、必要に応じて変更する。

3.2 非線形最小２乗法

非線形最小２乗法は、ある目的変数 y を m 個の説明変数 x_i （$i=1,2,\cdots,m$）と p 個のパラメータ a_k（$k=1,2,\cdots,p$）を含む非線形関数 $f(x_1,\cdots,x_m;a_1,\cdots,a_p)$ で予測することを目的とする。

$$y=f(x_1,\cdots,x_m;a_1,\cdots,a_p)+u$$

ここに u は誤差項である。

まず観測された N 個のデータについてパラメータ a_k を最適化する式を考える。今レコードを λ（$\lambda=1,2,\cdots,N$）として、以下の残差の２乗を表す統計量 Z を考える。

$$Z=\sum_{\lambda=1}^{N}\left[y_\lambda-f(x_{1\lambda,}\cdots,x_{n\lambda};a_1,\cdots,a_p)\right]^2 \qquad \text{最小化}$$

我々は Z の値を最小化するために、パラメータ a_k（$k=1,\cdots,p$）で微分する。

$$\frac{\partial Z}{\partial a_k}=-2\sum_{\lambda=1}^{N}\frac{\partial f(x,a)}{\partial a_k}\left[y_\lambda-f(x,a)\right]=0 \qquad (k=1,\cdots,p) \tag{2}$$

ここに書式の簡単化のため、$f(x,a)\equiv f(x_{1\lambda,}\cdots,x_{n\lambda};a_1,\cdots,a_p)$ としている。

式 (2) は y_λ と $x_{i\lambda}$ に実測値を使うことから、パラメータ a_k についての非線形連立方程式となる。よって前に述べた連立方程式 (1) の場合の解法が利用できる。即ち、$a_k=a_k^0$ を初期値として以下の値を考える。

$$a_k=a_k^0+M_k/D \qquad (k=1,\cdots,p)$$

$$M_k=\begin{vmatrix} Z_{11}^0 & \cdots & \overset{k\text{列}}{-Z_1^0} & \cdots & Z_{1p}^0 \\ Z_{21}^0 & \cdots & -Z_2^0 & \cdots & Z_{2p}^0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ Z_{p1}^0 & \cdots & -Z_p^0 & \cdots & Z_{pp}^0 \end{vmatrix},\quad D=\begin{vmatrix} Z_{11}^0 & \cdots & \overset{k\text{列}}{Z_{1k}^0} & \cdots & Z_{1p}^0 \\ Z_{21}^0 & \cdots & Z_{2k}^0 & \cdots & Z_{2p}^0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ Z_{p1}^0 & \cdots & Z_{pk}^0 & \cdots & Z_{pp}^0 \end{vmatrix}$$

ここに、$Z_{ij}^0=\partial^2 Z/\partial a_i\partial a_j\Big|_{a_1=a_1^0,a_2=a_2^0,\cdots,a_p=a_p^0}$ である。

これから先はここで求めた a_k を新たな初期値として同じ処理を繰り返す。これは前節で述べた逐次近似と同じ計算である。しかし、前節の場合は１回微分までであったが、今回は２回微分を使っている。数値計算の２回微分は一般に精度が悪くなり、この方法では解を見つけにくいことがある。ただ、１変数の場合は計算に問題がなかったことから、１つの変数について解を収束させ、次に変数を変えてまた収束させるという方法で、すべての変数について解が収束するまでこ

れを繰り返すという方法を取ってみた。この場合時間はかかるが、収束状況はかなり改善された。もう少し計算時間を短縮できる良い方法があると思われるが、分かり次第その部分のプログラムを書き直す予定である。

実際のプログラムのデータ画面、初期メニュー画をそれぞれ図 3.3a、図 3.3b に示す。まずメニュー画面の目的変数コンボボックスを設定し、計算式のテキストボックス内にパラメータを含む予測式を書く。これはロジスティック曲線と呼ばれる例 [7] である。我々のプログラムでは方程式ソルバーと同じく、初期値の設定は乱数で行う。但し、今回は計算に時間がかかることから、最初はメニュー画面 1 右上の「乱数から」ラジオボタンを選択し、微分分割値、収束値、ループ回数をゆるく設定し、乱数発生回数を増やしておく。図 3.3b のそれらの設定は初期値の設定を「乱数から」にした場合のデフォルト値である。初期設定が終わったら「最小２乗解」ボタンをクリックして計算を実行する。実行結果を図 3.3c に示す。

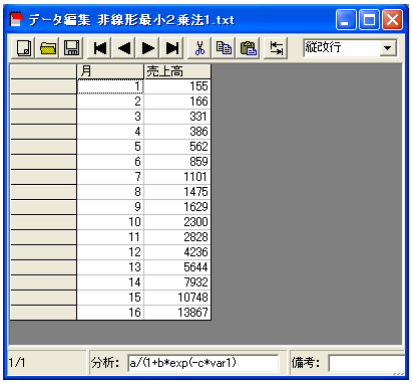


図 3.3a データ画面

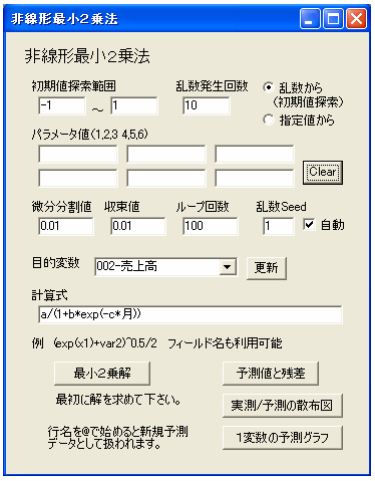


図 3.3b メニュー画面 1

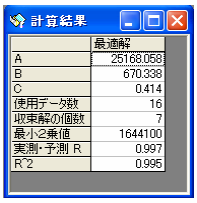


図 3.3c 解表示画面 1

この処理で一応の解が得られたならば、その解は図 3.4a メニュー画面２の「パラメータ値」のところに表示される。次にラジオボタンを「指定値から」に変更し、「最小２乗解」ボタンをクリックして分析を実行する。その際、初期値には先ほど自動的に設定されたパラメータの値を使用する。また計算に利用する「乱数発生回数」、「微分分割値」、「収束値」、「ループ回数」は精度を上げる設定になっている。実行結果を図 3.4b に示す。

解を求めた後、その解による実測値、予測値、残差は、「予測値と残差」ボタンをクリックすることで求まる。実行例を図 3.5 に示す。

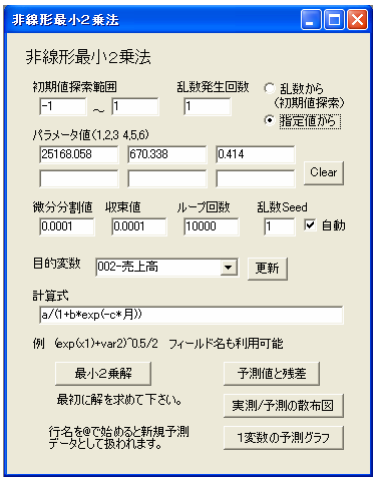
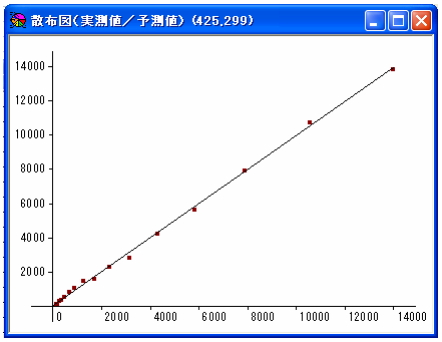


図 3.4a メニュー画面２　　図 3.4b 解表示画面２　図 3.5 予測値と残差画面

この実測値と予測値の関係を見る散布図は「実測／予測の散布図」ボタンで求まる。その例を図 3.6 に示す。また、説明変数が１変数の場合に限り、目的変数がその説明変数によってどのように予測されるかを「１変数の予測グラフ」をクリックすることで見ることができる。実行例を図 3.7 に示す。



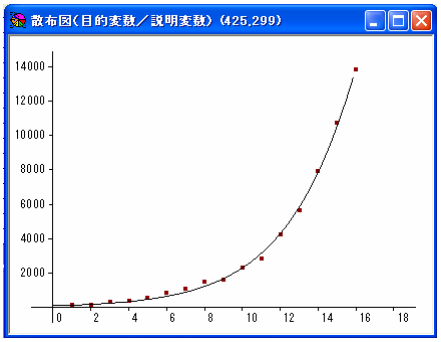


図 3.6 実測／予測の散布図　　図 3.7 １変数の予測グラフ画面

４．多目的線形計画法

線形計画法は与えられた線形の制約条件のもとで、線形の目的関数を最大化（最小化）する問題を解決するための手法であり、一般に以下の形で書くことができる [5]。

目的関数　　$z={}^t\mathbf{cx}$　最大化（最小化）

制約条件　　$\mathbf{A}_1\mathbf{x}\le\mathbf{b}_1,\ \mathbf{A}_2\mathbf{x}\ge\mathbf{b}_2,\ \mathbf{A}_3\mathbf{x}=\mathbf{b}_3,\ \mathbf{x}\ge\mathbf{0}$

ここに、$\mathbf{x}(n\times1)$ は各要素が非負に制限された変数行列であり、$\mathbf{c}(n\times1)$, $\mathbf{b}_i(m_i\times1)\ge\mathbf{0}$,

$\mathbf{A}_i(m_i \times n)$ （$i = 1,2,3$）は係数行列である。

この問題は生産計画や輸送計画など様々な分野で利用されているが、現実には目的関数が唯一であるようなケースはむしろ珍しく、いくつかの目的関数をある程度最適化するような解を求めることが多い。このような問題に答えるのが多目的線形計画法である。多目的線形計画法では線形計画法の制約条件はそのままで、目的関数の数を２つ以上考える。ここではこの問題に対するグローバル評価法と呼ばれる解法を説明する[4)]。

今我々は以下の多目的線形計画問題を考える。

目的関数　　$z^a = {}^t\mathbf{c}^a\mathbf{x}$ （$a = 1,2,\cdots,p$）　最大化（最小化）

制約条件　　$\mathbf{A}_1\mathbf{x} \le \mathbf{b}_1$，$\mathbf{A}_2\mathbf{x} \ge \mathbf{b}_2$，$\mathbf{A}_3\mathbf{x} = \mathbf{b}_3$，$\mathbf{x} \ge \mathbf{0}$

最初にまず目的関数aの最適解を考えて、その解を用いたz^aの最適値をz^{a*}とする。目的関数aについて、ある解$\mathbf{x}$の充足率r^aを以下のように定義する。

$$r^a = 1 - \left| (z^{a*} - {}^t\mathbf{c}^a\mathbf{x}) / z^{a*} \right|$$

ここで最適解の目的関数の値が0に近い場合、充足率が負になる場合があるので気を付ける。目的関数の値は目的関数に定数項を加えれば自由に変更できるが、これによって充足率の値も変わってくるのでこの方法にも問題がある。今後充足率の定義についても検討しなければならない。

我々は各目的関数の１－充足率の値の合計を新しい目的関数にして、上の制約条件で線形計画問題を考える。

目的関数　　$z = \sum_{a=1}^{p} (1 - r^a)$　　最小化

制約条件　　$\mathbf{A}_1\mathbf{x} \le \mathbf{b}_1$，$\mathbf{A}_2\mathbf{x} \ge \mathbf{b}_2$，$\mathbf{A}_3\mathbf{x} = \mathbf{b}_3$，$\mathbf{x} \ge \mathbf{0}$

またこの目的関数にウェイトw^a （$a = 1,2,\cdots,p$）を付けてもよい。

目的関数　　$z = \sum_{a=1}^{p} w^a (1 - r^a)$　　最小化

制約条件　　$\mathbf{A}_1\mathbf{x} \le \mathbf{b}_1$，$\mathbf{A}_2\mathbf{x} \ge \mathbf{b}_2$，$\mathbf{A}_3\mathbf{x} = \mathbf{b}_3$，$\mathbf{x} \ge \mathbf{0}$

これらの線形計画問題を最良化問題と呼び、その最適解$\mathbf{x}^*$を最良化妥協解と呼ぶ。

次に具体的な実行例について説明する。図 4.1 に多目的線形計画法のメニュー画面を示す。線形計画問題の式を表すデータは基本的に表形式で入力するが、必要に応じてテキスト形式でも入力できる。メニューの「テキストエディタ」ボタンでテキスト入力用のエディターが表示され、式をそこに書き込むことができる。「テキスト入力」ボタンはテキスト形式を表形式に変換し、「テキスト出力」ボタンは表形式をテキスト形式に変換する役割を持つ。表形式のデータの例を図 4.2 に示す。また、テキスト形式のデータの例を図 4.3a、図 4.3b に示すが、どちらの形式でも同じ表形式のデータを作り出す。テキスト出力のデフォルトは図 4.3a の形式である。

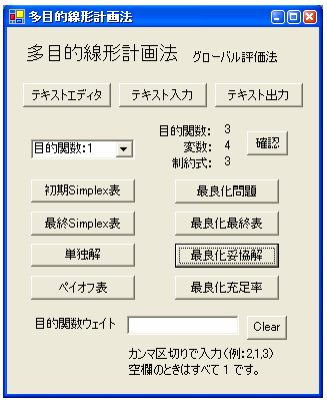


図 4.1 多目的線形計画法メニュー画面

図 4.2 表形式データ画面

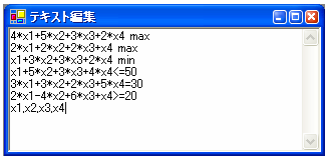


図 4.3a テキスト形式データ画面 1

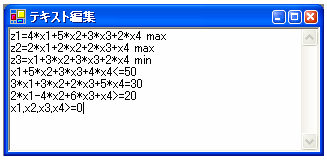
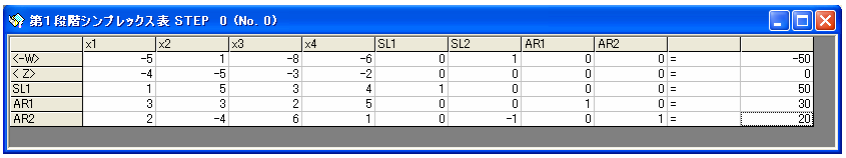


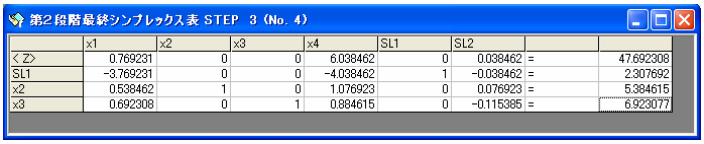
図 4.3b テキスト形式データ画面 2

上のデータから各目的関数についての単独解を求めるには、まず左上のコンボボックスで目的関数を選択し、「初期 Simplex 表」、「最終 Simplex 表」、「単独解」ボタンを利用する。これらの表示画面をそれぞれ図 4.4、図 4.5、図 4.6 に示す。これらは線形計画法における出力形式に等しい。



	x1	x2	x3	x4	SL1	SL2	AR1	AR2		
<-W>	-5	1	-8	-6	0	1	0	0	=	-50
< Z>	-4	-5	-3	-2	0	0	0	0	=	0
SL1	1	5	3	4	1	0	0	0	=	50
AR1	3	3	2	5	0	0	1	0	=	30
AR2	2	-4	6	1	0	-1	0	1	=	20

図 4.4 目的関数 1 の初期 Simplex 表画面



	x1	x2	x3	x4	SL1	SL2		
< Z>	0.769231	0	0	6.038462	0	0.038462	=	47.692308
SL1	-3.769231	0	0	-4.038462	1	-0.038462	=	2.307692
x2	0.538462	1	0	1.076923	0	0.076923	=	5.384615
x3	0.692308	0	1	0.884615	0	-0.115385	=	6.923077

図 4.5 目的関数 1 の最終 Simplex 表画面

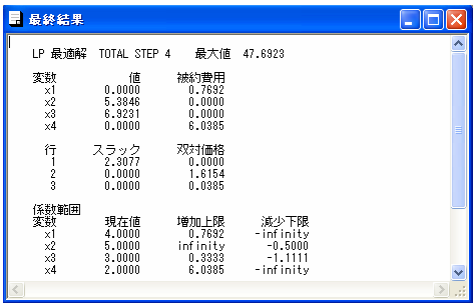
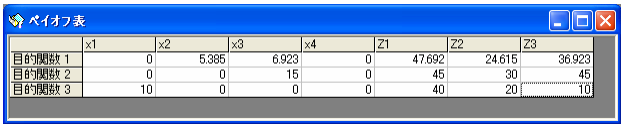


図 4.6 目的関数 1 の単独解画面

それぞれの単独解で他の目的関数がどのような値になるのかを表すものがペイオフ表である。メニューの「ペイオフ表」ボタンをクリックすると図 4.7 のようなペイオフ表画面が表示される。

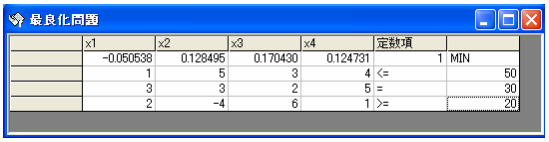


	x1	x2	x3	x4	Z1	Z2	Z3
目的関数 1	0	5.385	6.923	0	47.692	24.615	36.923
目的関数 2	0	0	15	0	45	30	45
目的関数 3	10	0	0	0	40	20	10

図 4.7 ペイオフ表画面

例えばここで行名が目的関数 1 の行は目的関数 1 の単独解を使った 3 つの目的関数の値を表しており、逆に列名が Z1 の列は 3 つの目的関数の最適解を使った目的関数 1 の値を表している。

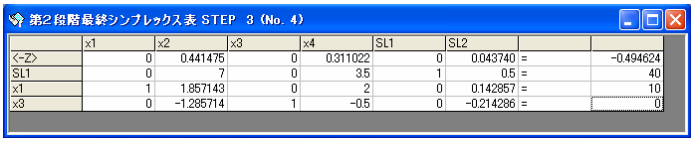
これらの単独解を使った最良化問題のデータは「最良化問題」ボタンをクリックして図 4.8 のように得られる。



	x1	x2	x3	x4	定数項	
	-0.050538	0.128495	0.170430	0.124731	1	MIN
	1	5	3	4	<=	50
	3	3	2	5	=	30
	2	-4	6	1	>=	20

図 4.8 最良化問題画面

この最良化問題の解は、「最良化最終表」ボタンと「最良化妥協解」ボタンによって得られる。それらを図 4.9 と図 4.10 に示す。この最良化妥協解による各目的関数の充足率はメニューの「最良化充足率」ボタンで求めることができる。その例を図 4.11 に示す。



	x1	x2	x3	x4	SL1	SL2		
<-Z>	0	0.441475	0	0.311022	0	0.043740	=	-0.494624
SL1	0	7	0	3.5	1	0.5	=	40
x1	1	1.857143	0	2	0	0.142857	=	10
x3	0	-1.285714	1	-0.5	0	-0.214286	=	0

図 4.9 最良化問題最終シンプレックス表画面

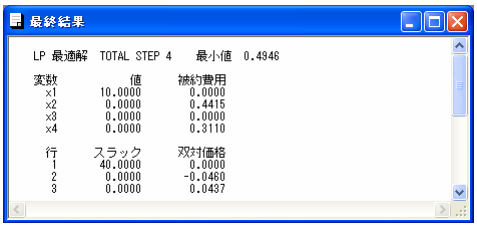


図 4.10 最良化妥協解画面

図 4.11 最良化妥協解と充足率画面

5. 待ち行列シミュレータとその他の変更点

5.1 待ち行列シミュレータ

参考文献 1) で示した待ち行列シミュレーションプログラムは様々な条件の下に定常解を求めるシミュレータであった。我々はこれに機能を追加し、単位時間当たりの平均到着数（到着数と略す）、単位時間・窓口数当たりの平均サービス数（サービス数と略す）、窓口数の時間的な変化に対応できるようにした。メニュー画面を図 5.1 に示す。

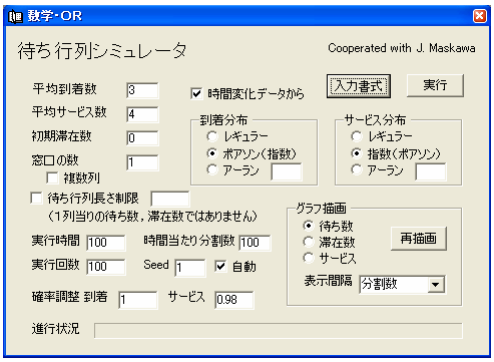


図 5.1 待ち行列シミュレータメニュー画面

このメニュー画面は以前のものに「時間変化データから」チェックボックスと「入力書式」ボタンの機能を追加したものである。時間変化を与えるデータの例を図 5.2 に示す。

図 5.2 時間変化データ

これは、シミュレーションの時間 $0 \le t < 50$ （単位時間）の間、到着数 3、サービス数 4、窓口数 1、$50 \le t < 55$ の間、到着数 6、サービス数 4、窓口数 1、$55 \le t < 100$ の間、到着数 6、サービス数 4、窓口数 2 とするものである。さらに細かな設定は時刻の分割を多くするだけである。この状況は最初定常状態だった待ち行列が、到着数が増えたことで行列が伸び、すぐに窓口数を増やして対応したということになる。

メニューの「時間変化データから」チェックボックスをチェックして、「実行」ボタンをクリックすると図 5.3a と図 5.3b の実行結果が得られる。

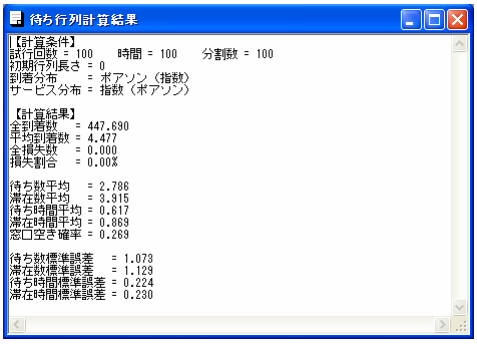


図 5.3a 待ち行列結果表示画面

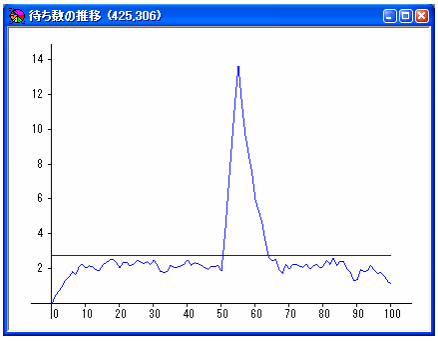


図 5.3b 待ち行列待ち数グラフ画面

このシミュレーションに関して、到着数とサービス数については条件の変化するところですぐさま変化させるが、窓口数については以下のように取り扱っている。窓口を開ける場合、条件の変化するところでサービスの空いた窓口を開ける。客は待ち数の少ない窓口に行く設定になっているので、その窓口の客の数は自然に増えてゆく。窓口を閉める場合、条件の変化するところで窓口への客の到着を止め、サービスを待っている客はそのままサービスを待つ。この場合、窓口が完全に終了するのは、指定時間より遅れることになる。

5.2 ヒストグラムの変更

ヒストグラムを描くとき、以前は初期値と分割幅を指定することになっていたが、初心者はこれに手間取るようなので、自動で描かせることにした。必要があれば細かい設定もできる。また、ヒストグラムから正規性を確認する際、基準がないので確認しにくかったが、新しくその分布の平均と分散から得られる正規分布曲線を加えることができるようにした。曲線の表示・非表示はグラフのメニュー［設定－回帰直線（近似曲線）[On/Off]］で切り替わる。正規曲線を加えたヒストグラムの表示画面を図 5.4 に示す。

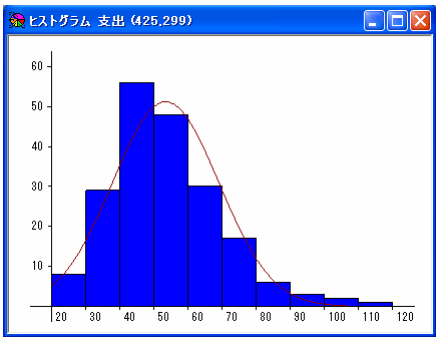


図 5.4 ヒストグラム

5.3 実験計画法のメニュー画面の変更

実験計画法のメニュー画面は検定の順をたどれるようになっていたが、今回はこれを量的データの検定メニューに準拠する形に書き換えた。具体的なメニュー画面を図 5.5 に示す。

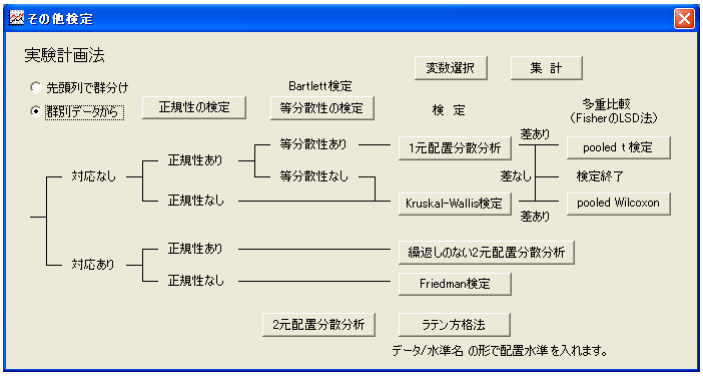


図 5.5 実験計画法メニュー画面

ここでは元々の２元配置分散分析を繰返しのないものとあるものに分け、データ間の対応の有無に関連させている。

5.3 データの予測について

重回帰分析や数量化 I 類など予測手法に使われる分析では、新しいデータに関する予測値を求めることが多い。そこでデータに予測を行うための行を含めることを考えた。エディターのメニュー［挿入/削除－予測行追加］で、エディターの最終行に行名が「@」の空行が追加される。行名の先頭文字が「@」の場合、その行は予測行とみなされ、係数などを求める計算には使用されないが、予測値と残差を計算する際には使用することとする。実際の予測行の画面を図 5.6 に示す。

図 5.6 予測行の挿入画面

この予測機能を追加した分析には、図 5.7 に示すように、メニューの「予測値と残差」ボタンに近いところに、「行名を@で…」というメッセージを入れる。「予測値と残差」ボタンをクリックすると、図 5.8 のような予測値を含んだ結果が表示される。

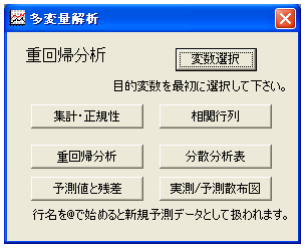


図 5.7 改定後の分析画面

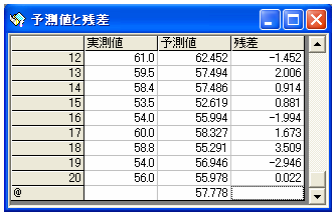


図 5.8 予測値と残差

６．おわりに

我々はプログラム開発と同時にこのソフトを用いた教育手法の開発を行ってきた。対象をこれから統計処理を学ぼうとされる方に設定し、教材はまず少人数のゼミで試し、次は３０～４０人程度の授業で、最終的に一般の方を対象とした講座で有効性を検討している。統計学については、もう少し手直しをする必要もあるが、ほぼ完成に近づいている。今後は講座のテキストという形でなく、独習用の教科書にまとめて行く必要がある。また、ＯＲや意思決定論など経営科学の分野では、少人数のゼミから人数の多い授業に移そうとしているところである。これはまだ演習が充実しておらず十分とはいえない状態である。

今回追加された分析で、方程式ソルバーと非線形最小２乗法については、まだ完成された印象がない。今後、計算手法や結果の表示方法について再検討が必要であろう。特に非線形最小２乗法の計算は、Excel のソルバーに比べて時間がかかる。改良すべき課題であろう。また結果表示についても、小さな表が出力されるだけなので、もう少しいろいろな角度から解を眺められるようにしたいと思う。

多目的線形計画法については現在グローバル評価法と呼ばれる方法だけが取り上げられている。

この方法では各目的関数の充足率を　１－｜（最適値－妥協値）÷最適値｜で与え、充足率の合計が最大になるように最良化妥協解をとる。しかし、最適値が 0 の近辺、例えば最大化問題で最適値 1、妥協値 -1 の場合など、充足率の意味をなさない場合もある。このような問題をどのように扱うか今後他の評価法についても調べて行かなければならない。

待ち行列シミュレータについては理論で表せないシミュレーションの解が本当に正しいのか疑問が残る。現実の待ち行列と比較検討を行う必要性を感じる。ヒストグラムの自動化と正規分布曲線の追加は、授業で使用する中で特に必要性を感じていた。実験計画法についても授業の中での説明とメニュー画面の不統一が気になっていた。また著者の理解不足から、繰返しのない２元配置分散分析や Friedman 検定をデータ間の対応と結びつけていなかった。今回の変更で分かり易くなったものと思う。

この論文では述べていないが、我々は現在時系列分析についてプログラムを開発中である。その際、モデルを示すだけでなく予測値の表示は必要不可欠である。特に重回帰分析や数量化 I 類を使ったモデルもあり、これまで作られた分析にも予測値を表示する機能を追加する必要があった。また授業でも予測値を Excel で計算する問題があり、学生にとっては面倒な課題でもあった。これらの理由から、各分析に予測値を計算する機能を加えることにした。手始めは時系列分析への応用を考えて、重回帰分析と数量化 I 類に機能を追加した。今後他の分析へも適用して行く。

参考文献

1) 福井正康・光平直嗣・細川光浩, 社会システム分析のための統合化プログラム９　－Dematel 法・社会的意思決定手法・その他の機能強化－, 福山平成大学経営研究, 3 号, (2007) 87-107.
2) 古川俊之監修・丹後俊郎, 新版 医学への統計学, 朝倉書店, 1993.
3) 篠崎壽夫・松下祐輔, 工学のための応用数値計算法入門（上）, コロナ社, 1976.
4) 木下栄蔵, わかりやすい意思決定論入門, 近代科学社, 1996.
5) 福井正康・増川純一, 社会システム分析のための統合化プログラム３　－線形計画法・待ち行列シミュレーション－, 福山平成大学経営情報研究, 4 号, 99-115, 1999.
6) 福井正康, 社会システム分析のための統合化プログラム４　－基本統計－, 福山平成大学経営情報研究, 5 号, 89-100, 2000.
7) 上田太一郎監修・高橋玲子・村田真樹・淵上美喜・藤川貴司・近藤宏・上田和明, Excel で学ぶ時系列分析と予測, オーム社, 2006.

Multi-purpose Program for Social System Analysis 10

- Trend Tests, Equation Solver, Nonlinear Least Squares Method, Multi-purpose Linear Programming, Queuing Simulator -

Masayasu FUKUI, Badarengui*, Mitsuhiro HOSOKAWA and Yukie OKUDA

Department of Management Information, Faculty of Management,
Fukuyama Heisei University
* Department of Management Information, Graduate School of Management,
Fukuyama Heisei University

Abstract

We have been constructing a unified program on the social system analysis for the purpose of education. This time we create new programs of trend tests and nonlinear squares method in the field of statistics, equation solver in the field of mathematics, and multi-purpose linear programming and queuing simulator in the field of OR. Especially, the queuing simulator which is added a new function corresponding to the temporal alteration of conditions can be used in the practical situations.

Keywords

College Analysis, social system analysis, OR, statistics, trend tests, equation solver, nonlinear least squares method, multi-purpose linear programming, queuing simulator

URL: http://www.heisei-u.ac.jp/ba/fukui/