数理計画法の復活と社会貢献への提案

成蹊大学　経済学部
新村秀一

1.　人生の目標

- 難しい統計学，数理計画法，数学を，使いやすく高機能なソフトで学習法を再編し，21世紀の新しい一般教養にしたい
  - 統計
    - SAS, SPSS, Statistica, JMP
    - 解説書とデモ版(学生が自宅で独学)
  - 数学
    - Speakeasy　（パソコンらくらく数学）
  - 数理計画法？(これまで，3冊の未発表原稿)
- 専門家教育とユーザー教育の分離

2.　数理計画法ソフト

2.1　LINDO社の製品

- LINDO:　LP,IP,QPが自然表記，大規模モデル作成に時間がかかる
- GINO　:　非線形最適化，販売中止
- VINO　:　表計算のアドイン，販売中止
- What'sBest! : Excelのアドイン，ABC分析，雛形モデル，セルのコピーで大規模モデルの作成が容易，可視性に優れ多くの人と結果を共有
- LINGO　: 2007年12月に啓示，集合モデルでモデルの変更に影響されない汎用モデル作成が可能
- LINDO　API　: Cのライブラリー

2.2　数理計画法の問題点

- なぜ，数理計画法の情報処理教育は，統計に比べ遅れたのか？
  - LINDOは，使いやすく分かり易いが
    - 企業の大規模問題にすぐに使えない
  - 数理計画法では，理論家の影響が強すぎる
    - 整数計画法が早いか遅いか
      - 解けますか

MIN = - 81 * X1 - 221 * X2 - 219 * X3 - 317 * X4 - 385 * X5 - 413 * X6; 12228 * X1 + 36679 * X2 + 36682 * X3 + 48908 * X4+ 61139 * X5 + 73365 * X6 = 89716837; @GIN(X1); ..., @GIN(X5); @GIN(X6);@BND(0, X1, 99999); ..., @BND(0, X6, 99999)

      - TSP
    - 初心者から専門家まで使いやすく，全ての機能が充実している
- なぜ，実社会における利用が進んでいないのか?
  - 問題解決志向でなかった
  - 数理計画法ソフトの未熟

2.3　数理計画法ソフトの現状

（1）　What'sBest!, LINGO, LINDO APIは，全て次の機能をもつ．

- 100万変数，10万制約式のような大規模LPでも，主単体，双対単体，バリアー法のいずれかで高速で解ける．
- 目的関数と制約式に2次項を含む大規模2次計画モデルを，バリア法で解くことができる．
- 任意の非線形計画法が解ける．
- 大域的ソルバーは，凸性が分からない非線形問題で，大域的最適解を保障できる．
- 他の制約式が線形，2次，非線形にかかわらず，整数変数が解ける．
- 大域的ソルバーは，整数変数を持つモデルでも稼動．

2.3 数理計画法ソフトの現状

（2） LINGO

- 入力データを前処理し，分析結果を後処理，任意の形式で印刷，という使いやすく強力な機能．
- ポートフォリオ分析で，複数のモデルを解いて効率フロンティアを描く．
- DEAが簡単に計算できる．
- 一つのモデルの出力を，次の最適化モデルに利用し，複数のモデルを自動的に解く強力なプログラミング機能．

2.3　数理計画法ソフトの現状

(3)　LINGOは,

- Excelやデータベースとのインターフェイスに優れ，データの入出力が容易.
- この機能を用い，全ての数理計画法モデルはモデルのサイズの変更から独立し，汎用化.
- 大学で教えたモデルが，そのまま企業の実システムで利用.

（4）　WBは，大規模で難しい線形，整数，非線形モデルを表計算で分析.

（5）　LINGOとWBは，Windows環境で稼動する，もっとも操作が容易なソフトの一つ.

- ◆ 数理計画法ソフトに使いやすさを求めてこなかったことが，統計ソフトに比べ大学教育や企業で普及してこなかった元凶.

(6)　LINGOとWBは，以上の機能を単独で実現し，費用も廉価.

- ◆ ユーザーは，線形，2次，整数，非線形を意識する必要がない.

2.4　数理計画法の教育元年

今年を，大学における数理計画法教育の元年に！

- 自然表記のスカラーモデルで教育し，汎用モデルで大型モデルの分析能力を教育
- 多くの雛形モデルの提供
- マニュアルをPDFで無料配布
- 価格を，統計ソフトより安く
- 自宅での学習は，デモ版を利用？
- 数理計画法は，数式で表される全ての計画問題，現象に対応できる，という強い信念
  - 自信を持って，統計も含む数学，物理，医学･･･，のより広い学問の基礎という位置づけ.

3.　成蹊大学での数理計画法

3.1　2007年度まで

- 学部の2年生でマネジメント・サイエンス
  - 前期：Speakeasyで数学モデル
    - 配列・行列・微積分の計算，近似式とライブラリー，乱数と積分，モンテカルロシミュレーション，行列とマルコフ過程，ローンの計算など
- 後期は，LINDOとWBで数理計画法の実習
  - 実行可能解・減少費用や双対価格，製品混合，配合(LINDOとWB)，PERT，輸送問題，人員配置，多期間在庫と財務，IP(分枝限定法を含む)とナップザック，ポートフォリオ分析(WB,LINGO)

3.2 2008年からの予定

2008年1月から2月で，テキストを作成

1章：　導入

2章　：非線形モデル，制約付きの関数の解

以下の章は，自然表記で説明後，汎用モデルの利用

3章：　製品混合で減少費用，双対価格，双対モデル

4章：　配合問題

5章：　TSP

6章：　PERT

7章：　ポートフォリオ分析と効率的フロンティア曲線

8章：　統計手法の一つ：SVM

9章：　DEA

４．非線形連立方程式

(1)製品の筐体(キャビネット)を，最小費用で設計したい．

MIN=0.1*(L*W+L*H)+0.2*H*W;
L*W+L*H+W*H>444;
W*L<656;
L*W*H>1516;
W/H<0.718;
W/H>0.518;

(2)　5000万円を，年金利4%の月次均等払いで，60万円以上の返済を考えた際，何ヶ月返済になるであろうか．

MIN=N;
P=A*(1-(1+i)^(-n))/i;
P=5000;　a<=60;　i=0.04/16;
@GIN(N);

5. 配合問題の汎用モデルの作成

	含有成分	Ｃｕ	Ｓｉ	Ｆｅ	Ｚｎ	Ｍｎ	Ｍｇ
	下限	1.8	10.8	0.88	1.6	0	0.34
	上限	2.2	11.2	0.9	1.8	0.3	0.35
原材料	単価（千円）						
X１	275	1.4	3.3	0.7	1.5	0.2	0.8
X２	275	2.5	8	0.8	4.5	0.2	0.3
X３	285	2.5	7.7	0.9	0.9	0.18	0.19
X４	285	2.5	9.5	0.9	0.9	0.18	0.09
X５	185	2.5	9.3	0.95	0.93	0.18	0.09
X６	235	2.3	8.4	0.8	3	0.21	1.4
X７	235	2.5	9	0.9	0	0	0
X８	260	0.2	0.2	0.5	0	0.5	0
X９	290	98	0	0	0	0	0
X10	340	0	97	0.5	0	0	0
X11	255	4	0.5	0.5	0.1	0.5	0.5

5.1 LINGOの自然表記モデル

```
MIN=275*X1+275*X2+285*X3+285*X4+185*X5+235*X6+235*X7+260*X8+290*X9+340*X10+255*X11;
[CU2] 1.4*X1+2.5*X2+2.5*X3+2.5*X4+2.5*X5+2.3*X6+2.5*X7+0.2*X8+98*X9+4*X11-CU=0;
[SI3] 3.3*X1+8*X2+7.7*X3+9.5*X4+9.3*X5+8.4*X6+9*X7+0.2*X8+97*X10+0.5*X11-SI=0;
[FE4] 0.7*X1+0.8*X2+0.9*X3+0.9*X4+0.95*X5+0.8*X6+0.9*X7+0.5*X8+0.5*X10+0.5*X11-FE=0;
[ZN5] 1.5*X1+4.5*X2+0.9*X3+0.9*X4+0.93*X5+3*X6+0.1*X11-ZN=0;
[MN6] 0.2*X1+0.2*X2+0.18*X3+0.18*X4+0.18*X5+0.21*X6+0.5*X10+0.5*X11-MN=0;
[MG7] 0.8*X1+0.3*X2+0.19*X3+0.09*X4+0.09*X5+1.4*X6+0.5*X11-MG=0;
[CU8] CU>1.8;
[CU9] CU<2.2;
[SI10] SI>10.8;
[SI11] SI<11.2;
[FE12] FE>0.88;
[FE13] FE<0.9;
[ZN14] ZN>1.6;
[ZN15] ZN<1.8;
[MN16] MN>0;
[MN17] MN<0.3;
[MG18] MG>0.34;
[MG19] MG<0.35;
[OTH1] X1+X2+X3+X4+X5+X6+X7+X8+X9+X10+X11=1;
[OTH2] X3=0.35;
END
```

- 実行可能解が無い:双対価格で解決
  - 新村秀一(1993)．数理計画法のパッケージ、中間言語、ライブラリー、オペレーションズ・リサーチ、38-3、112-130.
- 減少費用の説明
- **X3=0.35**, 親会社に幾ら請求すべきか?
- **WB**によるモデルの紹介

5.2　LINGOの集合モデル

```
SETS:
 CONTENT:  UL, LL;
 RAW:  PRICE, PRODUCT;
 MATRIX(RAW,CONTENT): SEIBUN;
ENDSETS
DATA:
 UL=2.2 11.2 0.9 1.8 0.3 0.35;
 LL=1.8 10.8 0.88 1.6 0 0.34;
 PRICE=275 275 285 285 185 235 235 260 290 340 255;
 SEIBUN=
1.4  3.3   0.7    1.5    0.2    0.8
2.5  8     0.8    4.5    0.2    0.3
2.5  7.7   0.9    0.9    0.18   0.19
2.5  9.5   0.9    0.9    0.18   0.09
2.5  9.3   0.95   0.93   0.18   0.09
2.3  8.4   0.8    3      0.21   1.4
2.5  9     0.9    0      0      0
0.2  0.2   0.5    0      0.5    0
98   0     0      0      0      0
0    97    0.5    0      0      0
4    0.5   0.5    0.1    0.5    0.5;
ENDDATA
MIN= @SUM( RAW(i):  PRICE(i)*PRODUCT(i));
@FOR( CONTENT(j):  @SUM(RAW(i): SEIBUN(i,j)*PRODUCT(i))<=UL(j));
@FOR( CONTENT(j):  @SUM(RAW(i): SEIBUN(i,j)*PRODUCT(i))>=LL(j));
@SUM(RAW(i):  PRODUCT(i))=1;
```

5.3　汎用モデル

	A	B	C	D	E	F	G	H	I	J
1		含有成分	Ｃｕ	Ｓｉ	Ｆｅ	Ｚｎ	Ｍｎ	Ｍｇ		
2		下限	1.8	10.8	0.88	1.6	0	0.34		1
3		上限	2.2	11.2	0.9	1.8	0.3	0.35		1
4	材料	単価（円）								0
5	×１	275	1.4	3.3	0.7	1.5	0.2	0.8		0
6	×２	275	2.5	8	0.8	4.5	0.2	0.3		0
7	×３	285	2.5	7.7	0.9	0.9	0.18	0.19		0
8	×４	285	2.5	9.5	0.9	0.9	0.18	0.09		0
9	×５	185	2.5	9.3	0.95	0.93	0.18	0.09		0
10	×６	235	2.3	8.4	0.8	3	0.21	1.4		0
11	×７	235	2.5	9	0.9	0	0	0		0
12	×８	260	0.2	0.2	0.5	0	0.5	0		0
13	×９	290	98	0	0	0	0	0		0
14	×１０	340	0	97	0.5	0	0	0		0
15	×１１	255	4	0.5	0.5	0.1	0.5	0.5		0
16			0	0	0	0	0	0		0
17			<=	<=	<=	<=	<=	<=		Not =
18			Not >=	Not >=	Not >=	Not >=	=>=	Not >=		
19										

5.4　完全な汎用モデル

```
SETS:
 CONTENT: UL, LL;
 RAW: PRICE, PRODUCT;
 MATRIX(RAW,CONTENT): SEIBUN;
ENDSETS

DATA:
 UL=@OLE();
 LL= @OLE();
 PRICE=@OLE();
 SEIBUN=@OLE();
ENDDATA

MIN=@SUM( RAW(i):PRICE(i)*PRODUCT(i));
@FOR( CONTENT(j): @SUM(RAW(i): SEIBUN(i,j)*PRODUCT(i))<=UL(j));
@FOR( CONTENT(j): @SUM(RAW(i): SEIBUN(i,j)*PRODUCT(i))>=LL(j));
@SUM(RAW(i): PRODUCT(i))=1;
```

6.　DEA

- 雛形モデルを用い,
  - 2008年2月13日～14日に成蹊大学で開催されたDEAの国際会議に, 企業業績と平均年収の報告書提出.
  - 野球選手の年俸の評価
- 2008年より, 博士後期課程の研究指導.
- 2008年より, 学部生に, 統計手法と併用して教える.

6.1 DEAのCCRモデル

```
MODEL:
SETS:
  DMU/1..67/: !The decisionmaking units;
    SCORE,SCORE2; FACTOR/1..4/: TW,TW2;
DXF( DMU, FACTOR):  F, ! F( I, J) = Jth factor of DMU I;
        W, WW2;  ! Weights used to compute DMU I's score;
ENDSETS
 DATA:
  NINPUTS = 3;  F=@OLE( ); WGTMIN = .00004; BIGM = 999999;
ENDDATA
!------------------------------------------------------;
SUBMODEL DEA1:
MAX = TSCORE;
    TSCORE = @SUM( FACTOR(J)|J #GT# NINPUTS:  F(INOW, J)* TW( J));
   [SUM21] @SUM( FACTOR( J)|  J #LE# NINPUTS:     F( INOW, J)* TW( J)) = 1;
@FOR( DMU( K):
   [LE1]     @SUM( FACTOR( J)|  J #GT# NINPUTS: F( K, J) * TW( J))
         <= @SUM( FACTOR( J)|  J #LE# NINPUTS: F( K, J) * TW( J))       );
@FOR( FACTOR( J): @BND( WGTMIN, TW, BIGM));
ENDSUBMODEL
CALC:
@FOR( DMU( IU):
      INOW = IU;
      @SOLVE(DEA1);     SCORE( IU) = TSCORE;
      @FOR( FACTOR( J):
        W(IU,J) = TW( J)    );    );
ENDCALC
```

7．TSPの種々のモデル

7.1　単なる割り当て問題

```
MIN = @SUM( LINK(i,j): dist(i,j) * Y(i,j));
@FOR( city( K):
@SUM( city( I) | I #NE# K: Y( I, K))= 1;
@SUM( city( J) | J #NE# K: Y( K, J))= 1;
    Y( K, K) = 0; ) ;
@FOR(LINK(i,j): @BIN(y(i,j)); );
```

7.2　星元成蹊大学教授のコンパクト解

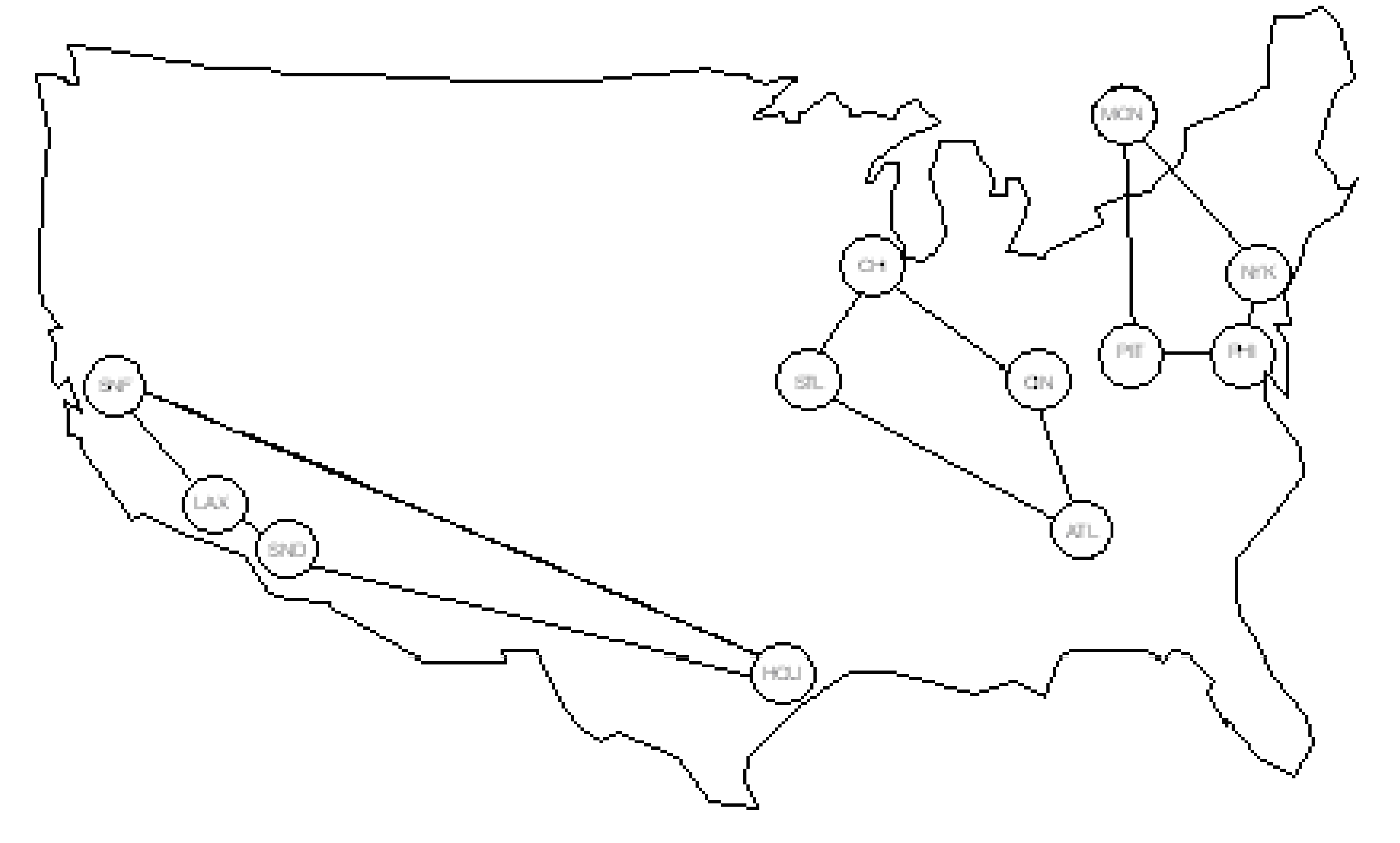
- [__5]　Ui－Uj＋n*Yij＜＝(n-1)　　　i=2，．．．，n，j=2，．．．，n

```
MODEL:
SETS:
 city:U;
 LINK( city, city):   dist,    Y;
ENDSETS
DATA:
 city = @OLE( );
 dist = @OLE( );
ENDDATA
 S=@size(CITY);!U(1)=s;
 MIN =  @SUM( LINK(i,j): dist(i,j) * Y(i,j));
 @FOR( city( K):
   @SUM( city( I) |  I #NE# K: Y( I, K))= 1;
   @SUM( city( J) |  J #NE# K: Y( K, J))= 1;
 ! Cannot go to yourself;    Y( K, K) = 0; );
 @FOR(LINK(i,j):@BIN(y(i,j)); );
@FOR( CITY(M) |  M #GE# 2: @FOR( CITY(N) | N #GE#2:
   @SUM( LINK( M,N): U(M)-U(N)+S*Y(M,N)) <= S-1;));
 !@SUM( LINK( M,N) | M #GE# 1 #AND# N#GE#1: U(M)-U(N)+S*Y(M,N)) <= S-1;
DATA:
@OLE( )=y;
ENDDATA
END
```

7.3 非対称距離(手動SUBTOUR CUT)

```
MODEL:
SETS:
CITY;
Y(CITY, CITY) | &1 #GT# &2:COST, Y;
ENDSETS
DATA:
 CITY=
  ATL  CHI  CIN HOU   LA  MON   NY  PHI  PIT  STL  SD  SF;
COST=
  702
  454  324
  842 1093 1137
 2396 2136 2180 1617
 1196  764  798 1857 2900
  864  845  664 1706 2844  396
  772  764  572 1614 2752  424   92
  714  459  284 1421 2464  514  386  305
  554  294  338  799 1842 1058 1002  910  622
 2363 2184 2228 1521   95 2948 2892 2800 2512 1890
 2679 2187 2463 2021  405 2951 3032 2951 2646 2125 500;
ENDDATA
MIN = @SUM( ROUTE: Y * COST);
@SUM( CITY( I) | I #GE# 2: Y(I, 1)) = 2;
@FOR( CITY( J) | J #GE# 2: @SUM(CITY(I) |  I #GT# J:
  Y(I, J)) + @SUM(CITY(K) | K #LT# J: Y(J, K))=2);
@FOR( ROUTE: Y <= 1);
END
```

SUBTOUR CUT



7.4 LOOPTSP(Ver.1)

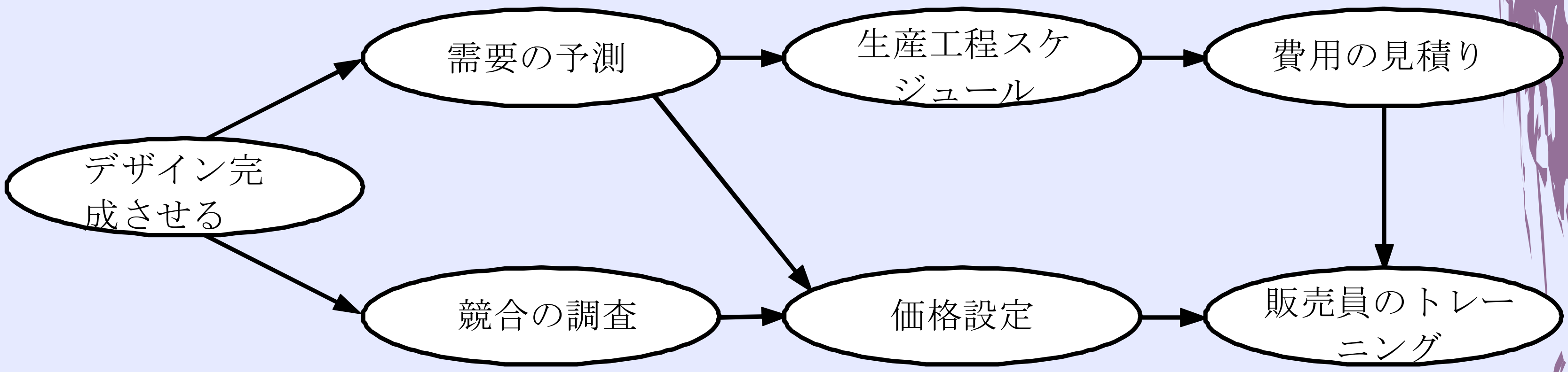
```
MODEL:
 ! (TSPCUT) Traveling Salesman Problem.  Find the shortest tour that
   visits each city exactly once. Subtour elimination method.
 ! Keywords: TSP, traveling sales person, routing, tour;
 SETS:
  CITY;
  LINK( CITY, CITY):
       DIST,  ! The distance matrix;
          Y;  ! Y( I, J) = 1 if link I, J is in tour;
  SUBTOUR: TOURSIZE;
  SXC(SUBTOUR,CITY): FLAG;
 ENDSETS
DATA:
  SUBTOUR = 1..20; ! Number subtour cuts we allow;
  CITY = @OLE( );
   ! Distance matrix for cities of National League of
      US baseball, circa 2004. It need not be symmetric;
  DIST =@OLE( );
ENDDATA
SUBMODEL TSP_CUT:
! Minimize total distance traveled;
 MIN = @SUM( LINK: DIST * Y);
! The Assignment constraints;
  @FOR( CITY( K):
  !  City K must be entered;
    @SUM( CITY( I) | I #NE# K: Y( I, K))= 1;
  !  City K must be departed;
    @SUM( CITY( J) | J #NE# K: Y( K, J))= 1;
  ! Cannot go to yourself;    Y( K, K) = 0;  );
! Subtour cuts;
  @FOR( SUBTOUR(t):
!  FLAG(t,i) = 1 if city i is in subtour t;
    @SUM( CITY(I)  |  FLAG(t,i) #EQ# 1:
      @SUM( CITY(J)  |  FLAG(t,j) #EQ# 1: Y(i,j))) <= TOURSIZE(t) - 1;     );
ENDSUBMODEL
CALC:
  @SET( 'TERSEO', 2);
  N = @SIZE( CITY);
  MXCUTS = @SIZE(SUBTOUR);
 ! Initially there are no subtour cuts;
 @FOR( SXC(t,i):
   FLAG(t,i) = 0;   );
   ICUT = 1;
! Loop over subtour cuts, ICUT;
 QUIT1 = 1;
 @WHILE( QUIT1 :
!  Solve current version;
   @SOLVE( TSP_CUT);
! Find subtour if any;
   TOURSIZE(ICUT) = 0;
! Loop over cities KURSTOP to find subtour starting at 1;
   KURSTOP = 1   QUIT2 = 1;
   @WHILE( QUIT2:
!    Loop over possible next cities j;
     @FOR(CITY(J):
      @IFC( Y(KURSTOP, J) #GT# .5:
        NEXT1 = J;      );););!    Next j;
      KURSTOP = NEXT1;
      TOURSIZE(ICUT) = TOURSIZE(ICUT) + 1;
      FLAG(ICUT,KURSTOP) = 1;
      !@WRITE( ' Next stop= ', CITY(KURSTOP),' Tour size=',TOURSIZE(ICUT),
         @NEWLINE( 1));
      @IFC( KURSTOP #EQ# 1:  ! Back home/Completed the subtour?;
        QUIT2 = 0;     );
          ); ! End loop over cities in subtour;
   ! If subtour is in fact a full tour, or out of space, get out;
    @IFC( TOURSIZE(ICUT) #EQ# N #OR# ICUT #GE# MXCUTS:
       !We are done;
       QUIT1 = 0;
       @FOR( LINK: Y = Y); !Fix Y;   );
   ! Get ready for next cut;
    ICUT = ICUT + 1;    );  !End loop over add cuts;
ENDCALC
SETS:
   LINKOPT( LINK)  |  Y( &1, &2) #GT# .5;
ENDSETS
CALC:
! Give simple report;
  @IFC( TOURSIZE(ICUT-1) #LT# N:
    @WRITE(' Sorry, optimum not found', @NEWLINE(1));
  @ELSE
    KURSTOP = 1;
    CUMDIST = 0;
    K = 1;
    @WRITE( 9*' ', 'Tour:    Distance:', @NEWLINE( 1));
    @WRITE( '     1:    ',CITY( KURSTOP),'  ', 8*' ','0',@NEWLINE(1));
    @FOR( CITY:
!     Get next city;
      @FOR( LINKOPT( I, J)  |  I #EQ# KURSTOP:
        NEXT = J;
        CUMDIST = CUMDIST + DIST( I, NEXT);
        K = K + 1;
        @WRITE( @FORMAT( K, '5.0f'),':    ',CITY( NEXT),
         '  ', @FORMAT( CUMDIST, '9.0f'),@NEWLINE(1));  );
      KURSTOP = NEXT;    );   );
ENDCALC
END
```

7.5 東京の不動産データ

- 50, 181, 592地点
  - LOOPTSP(Ver.1)では, 181地点で4時間でもとまらない
  - LOOPTSP(Ver.4)で, 230地点が3時間で求まる
  - 他社の製品は, 181地点を2週間で打ち切り
    - アルゴリズム, PCのクロック, チューニング
  - コンペの企画
- 300地点ぐらいは, 汎用ソルバーで, それ以上のモデルはLINDO　APIで開発?
- IPは, 輸送計画は容易だが, TSPや判別問題は困難

8. PERT

```
Max=10*Total+15*Detail+8*Appli+8*ApplTest+3*OSAppli+4*OSApHard+4*LastTest+3*Sales
+21*OSTest+5*OSHard+12*SysTest+11*HardTest+5*Market;
-Total=-1;
Total-Detail=0;
Detail-Appli-OSTest-HardTest=0;
Appli-ApplTest=0;
ApplTest+D1-OSAppli=0;
OSAppli+D2-OSApHard=0;
Sales=1;
HardTest-D4=0;
OSTest+D4-D1-OSHard=0;
OSHard-D2-SysTest-Market=0;
SysTest-D3=0;
OSApHard+D3-LastTest=0;
LastTest+Market-SalEs=0;
END
```


デザイン完成させる
需要の予測
競合の調査
生産工程スケジュール
価格設定
費用の見積り
販売員のトレーニング

	A	B	C	D	E	F
1						
2						
3						
4		DESIGN	FORECAST		DESIGN	10
5		DESIGN	SURVEY		FORCAST	14
6		FORECAST	PRICE		SURVEY	3
7		FORECAST	SCHEDULE		PRICE	3
8		SURVEY	PRICE		SCHEDULE	7
9		SCHEDULE	COSTOUT		COSTOUT	4
10		PRICE	TRAIN		TRAIN	10
11		COSTOUT	TRAIN			
12						

汎用PERTモデル

```
SETS:
   TASKS / DESIGN, FORECAST, SURVEY, PRICE,
    SCHEDULE, COSTOUT, TRAIN/: TIME, ES, LS, SLACK;
   PRED(TASKS, TASKS) ;
ENDSETS
DATA:
   TASKS=@OLE( );
   PRED= @OLE( );
   TIME = @OLE( );
ENDDATA
@FOR(TASKS(J) | J #GT# 1:
 ES(J) = @MAX(PRED(I, J): ES(I) + TIME(I))
);
@FOR(TASKS(I) | I #LT# LTASK:
 LS(I) = @MIN(PRED(I, J): LS(J) - TIME(I));
);
@FOR(TASKS(I): SLACK(I) = LS(I) - ES(I));
ES(1) = 0;
LTASK = @SIZE(TASKS);
LS(LTASK) = ES(LTASK);
```

9. 投資分析

	A	B	C	D	E
1					
2					
3		X	Y	Z	期待利益
4	利益	1.3	1.2	1.08	1.2
5	投資比率	0	1	0	リスク
6	分散共分散	3	1	−0.5	2
7		1	2	−0.4	
8		−0.5	−0.4	1	
9					

9.2　自然表記

MIN=3*X*X+2*Y*Y+Z*Z+2*X*Y-X*Z-0.8*Y*Z;

X+Y+Z=1;

1.3*X+1.2*Y+1.08*Z>1.15;

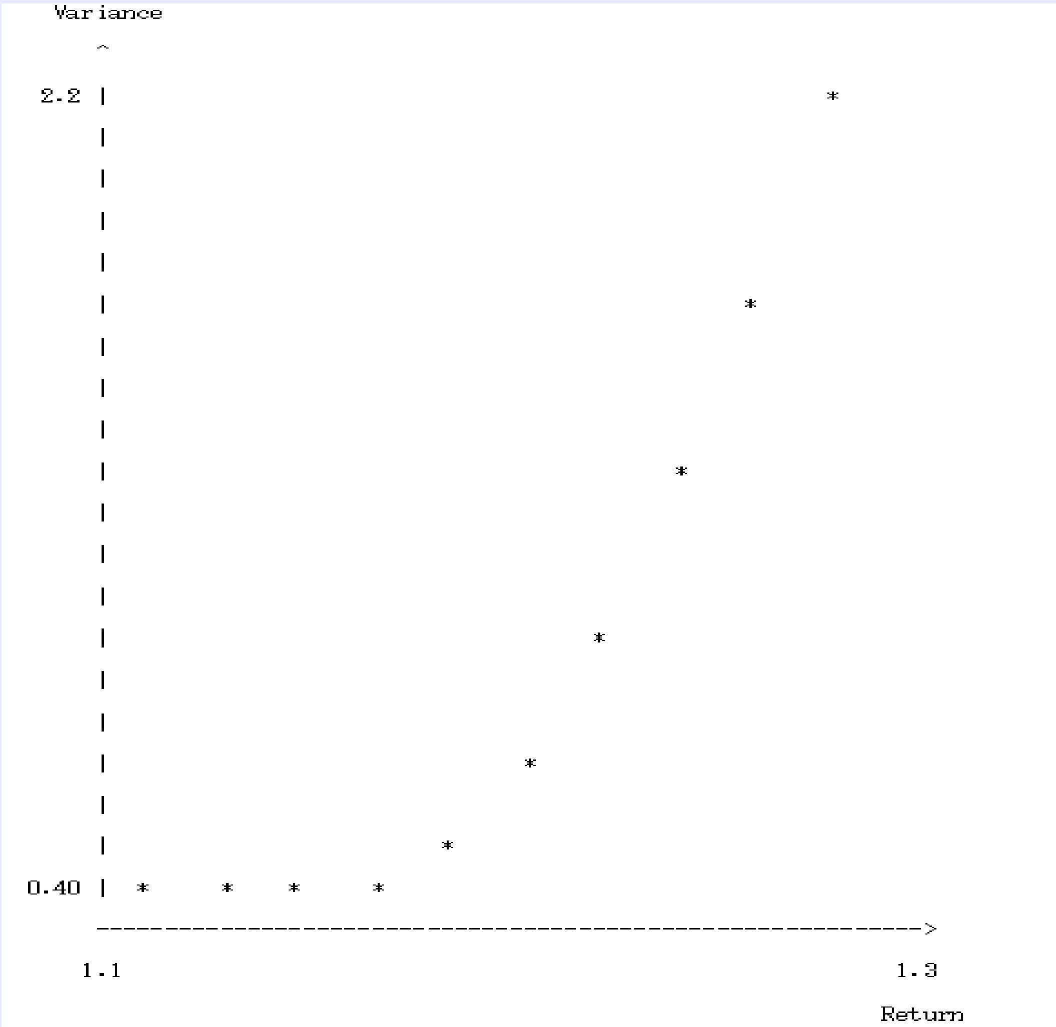
9.3 汎用PortFolioモデル

```
MODEL:
SETS:
   ASSET: RATE, UB, INVEST;
   COVMAT( ASSET, ASSET): V;
ENDSETS
DATA:
  ASSET = GOOGLE, YAHOO, CISCO;
  RATE = @OLE( );   UB   = @OLE( );   V    = @OLE( );
ENDDATA
SUBMODEL SUB1:
   MAX = @SUM( COVMAT( I, J): V( I, J) * INVEST( I) * INVEST( J));
   RETURN = @SUM( ASSET: RATE * INVEST);
   ! Must be fully invested;
   @SUM( ASSET: INVEST) = 1;
   ! Upper bounds on each;
   @FOR( ASSET: @BND( 0, INVEST, UB));
   ! Must achieve target return;
   RETURN >= RET_LIM;
ENDSUBMODEL
CALC:
   !@SET( 'DEFAULT');
   !@SET( 'TERSEO', 2);
   @SET( 'STAWIN', 0);
   @SOLVE(SUB1);
ENDCALC
END
```

9.4 効率的フロンティアの汎用モデル

```
MODEL:
! Solves the generic Markowitz portfolio
  model in a loop to generate the points
  on the efficient frontier;
SETS:
   ASSET: RATE, UB, INVEST;
   COVMAT( ASSET, ASSET): V;
   POINTS: XRET, YVAR;
ENDSETS
DATA:
! Number of points on the
   efficient frontier graph;
  NPOINTS = 10;
  POINTS = 1..NPOINTS;
! The stocks;
  ASSET = GOOGLE, YAHOO, CISCO;
! Expected growth rate of each asset;
  RATE = @OLE( );
! Upper bound on investment in each;
  UB   = @OLE( );
! Covariance matrix;
  V    = @OLE( );
ENDDATA
! Below are the three objectives we'll use;
SUBMODEL SUB_RET_MAX:
   [OBJ_RET_MAX] MAX = RETURN;
ENDSUBMODEL
SUBMODEL SUB_RET_MIN:
   [OBJ_RET_MIN] MIN = RETURN;
ENDSUBMODEL
SUBMODEL SUB_MIN_VAR:
   [OBJ_MIN_VAR] MIN =
    @SUM( COVMAT( I, J): V( I, J) * INVEST( I) * INVEST( J));
ENDSUBMODEL
!and the constraints;
SUBMODEL SUB_CONSTRAINTS:
   ! Compute return;
   RETURN = @SUM( ASSET: RATE * INVEST);
   ! Must be fully invested;
   @SUM( ASSET: INVEST) = 1;
   ! Upper bounds on each;
   @FOR( ASSET: @BND( 0, INVEST, UB));
   ! Must achieve target return;
   RETURN >= RET_LIM;
ENDSUBMODEL
CALC:
! Set some parameters;
   ! Reset all params;
   @SET( 'DEFAULT');
   ! Output error messages only;
   @SET( 'TERSEO', 2);
   ! Suppress status window;
   @SET( 'STAWIN', 0);
! Solve to get maximum return;
   RET_LIM = 0;
   @SOLVE( SUB_RET_MAX, SUB_CONSTRAINTS);
! Save maximum return;
   RET_MAX = OBJ_RET_MAX;
! Solve to get minimum return;
   @SOLVE( SUB_RET_MIN, SUB_CONSTRAINTS);
! Save minimum return;
   RET_MIN = OBJ_RET_MIN;
! Interval between return points;
   INTERVAL =
    ( RET_MAX - RET_MIN) / ( NPOINTS-1);
! Loop over range of possible returns,
  minimizing variance;
   RET_LIM = RET_MIN;
   @FOR( POINTS( I):
      @SOLVE( SUB_MIN_VAR, SUB_CONSTRAINTS);
      XRET( I) = RET_LIM;
      YVAR( I) = OBJ_MIN_VAR;
      RET_LIM = RET_LIM + INTERVAL;   );
! Display the results;
   @WRITE( '     Return      Variance', @NEWLINE( 1));
   @FOR( POINTS: @WRITE( @FORMAT( XRET, '#12.6G'),
    @FORMAT( YVAR, '#12.6G'), @NEWLINE( 1))   );
ENDCALC
CALC:
! The remainder of the model graphs the efficient frontier;
   NHASHY = 20;
   NHASHX = 60;
   SCALE = 10;
   V0 = @FLOOR( YVAR( 1) * SCALE);
   V0 = V0 / SCALE;
   V1 = @FLOOR( YVAR( NPOINTS) * SCALE + .5);
   V1 = V1 / SCALE;
   R0 = @FLOOR( RET_MIN * SCALE);
   R0 = R0 / SCALE;
   R1 = @FLOOR( RET_MAX * SCALE + .5);
   R1 = R1 / SCALE;
ENDCALC
```

9.5　効率的フロンティア



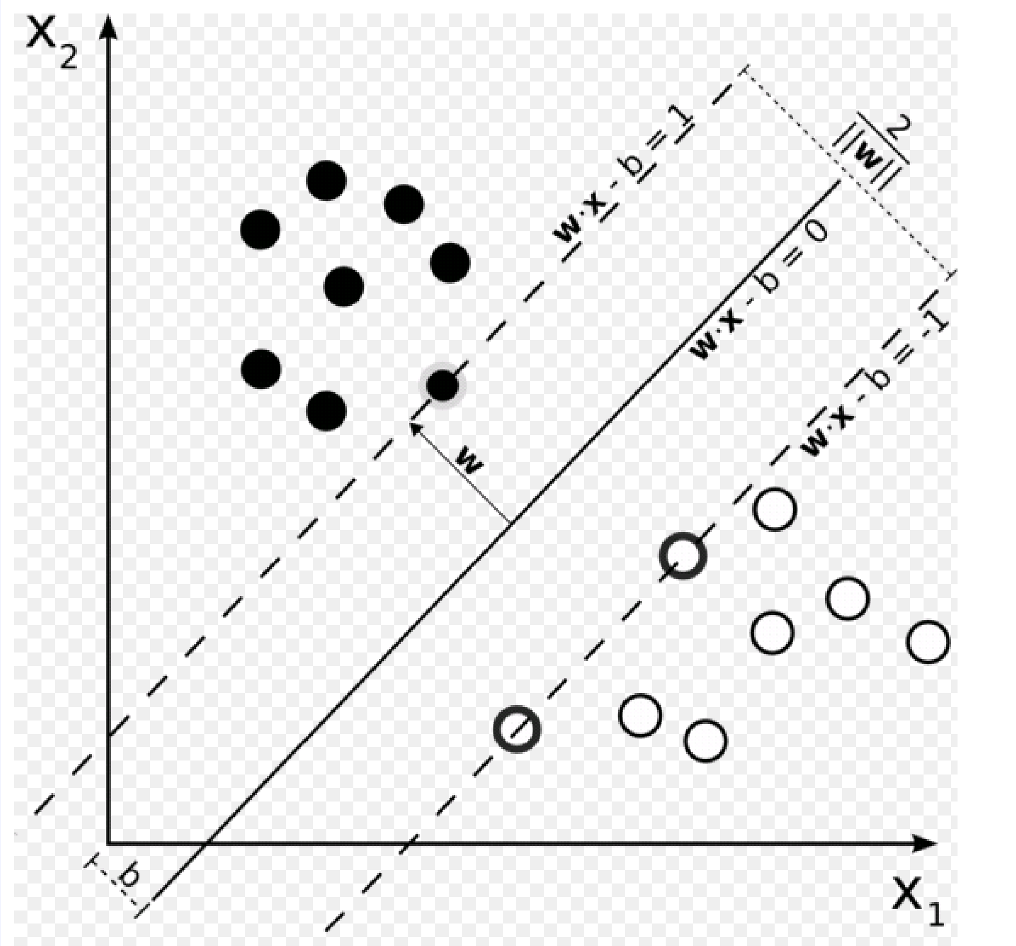
10. 統計とSVM

10.1 統計

統計は，数理計画法の広大な応用領域(年賀状)

- 回帰分析
  - L1, L2, Lpノルム
- 判別分析
  - 1970年代から数多くの研究
- SVM(サポート・ベクター・マシン)
- コンジョイント分析

マージン概念



X_2
X_1
$\mathbf{w}\cdot\mathbf{x} - b = 1$
$\mathbf{w}\cdot\mathbf{x} - b = 0$
$\mathbf{w}\cdot\mathbf{x} - b = -1$
$\frac{2}{\|\mathbf{w}\|}$
$\mathbf{w}$
$\frac{b}{\|\mathbf{w}\|}$

10.3 SVMの汎用モデル

```
MODEL:
SETS:
 P/X1..X3/: VAR;
 N/1..40/:E,SCORE;
 D(N,P):IS;
ENDSETS
DATA:
 IS=@OLE( );
ENDDATA
SUBMODEL sub1:
 MIN=OBJ ;
  OBJ = SVM1/2+c*SVM2;
  SVM1=@SUM(P(j) | j #NE# pn : VAR(j)^2) ;
  SVM2= @SUM(N(i):E(i));
 @FOR(N(i): @SUM(P(j):IS(i,j)*VAR(j)) > 1-E(i));
 @FOR(P(j):@FREE(VAR(j)));
ENDSUBMODEL
CALC:
 !@SET('DEFAULT');@SET('TERSEO',2);
 !C=0.1;
 !pn=@size(p);
 !MNI=0;
 @solve(sub1);
  !@FOR(N(i): SCORE(i)=@SUM(P(j):IS(i,j)*VAR(j)));
  !@FOR(N(i):  @IFC(SCORE(i) #LT# 0 : MNI=MNI+1));
ENDCALC
DATA:
 !@OLE( )=MNI;
 !@OLE( )=SVM1;
 !@OLE( )=SVM2;
 @OLE( )=VAR;
ENDDATA
END
```

11. 整数計画法

- 2008年3月18日，ある外資系コンサルタント企業からの評価依頼
  - 6,959 GIN，3798 制約，54,774 非ゼロ
  - 外資系ソフトでは解が簡単に求まっている(約10分)ので，LINGOと比較したい.
  - (有)マイス 市川均は，オプションを指定しないで， 1時間50分で解けた
  - 新村の場合 （ LINGO10 対 LINGO11)
    - 実は，LPで 13秒 ： 9秒 で解が出る
    - 整数計画法で解く場合は，オプションを指定する
      - Hurdle option at 577.1 で，1分16秒 ： 5分51秒
      - Relative option at 0.01 で，1分45秒 ： 3分18秒
      - Absolute option at 2 で， 38分48秒 ： 5分36秒
- 整数計画法の対応
  - 必ず，LPで解く
  - 64bit PC で64 bit Ver.を用いる
  - Memory を32MB から変更
  - ソルバーのオプション指定
  - TSPではモデルの作り方で100倍は異なる

12. まとめ

12.1 大学教育

- 数学，統計などあらゆる分野を飲み込んだ情報教育として復権
- 教えたことが，社会に出ても，大規模モデルが開発できなければ，教育の効果は無い
  - 入力に手間のかかるものは，WBで教育
    - 雛形モデルとABCステップ
  - LINGOで
    - 自然表記なモデルで説明
    - 汎用モデルの使い方

12.2 スローガン

- 大学や研究
  - 数理計画法ソルバーの著しい改良
    - 制約条件付きの関数の解を求める
    - LP,IP,QP, NLPを意識しないで解ける
    - 汎用モデルの作成
    - 大域的最適解
    - 全ての学生の，21世紀の一般教養
- 原料高の今こそ，日本の企業の隅々まで，数理計画法を普及させる好機
  - 卒業生の働く場を作る責務