シリーズ　3：　シカゴ大学・ビジネススクールのL.Schrage教授の英知に学ぶ問題解決学(中巻)

これまでの問題解決学との違い！

・使いやすく最高水準の機能を持つLINGO,ExcelのアドインソルバーのWhat's Best! (WB!),開発用の最適化ライブラリーのLINDO/APIという問題解決の最適化ソフトウエアの開発．統計ソフトが，データで表される問題解決学（データの科学）の便利な道具であり，最適化技術は数式で表される全ての現象を分析する「モデルの科学」の便利な道具である．

・評価版を無償で提供し，本書の内容を深く理解できる．ただし，大きな雛型モデルは実行できません．

・147の文献や書籍を調査し，LINGOとWB!で解ける雛型モデルを公開．もし，自分に合った問題を解決したい場合，自分で調査し，モデルを作成し，ソフトウエアを開発するという長い時間を省略できる．雛型モデルの理解を出発点とすれば，ブジネスマンや研究者の知的生産性を拡大する．また，学生が実際の問題を解決できる教育をして，社会人として送り出すことができます．数学に基礎のある学問，例えば統計や経営科学は，使いやすくて高機能な統計ソフトと数理計画法ソフトで「高度なユーザー教育」に切り替えるべきである．

・誰でも雛型モデルを利用して，知的生産性を上げ，身の回りの問題が解決できる．この訓練の後，LINDO社が提案するABC分析で雛型モデルにない問題に取り組めばよい．

・新村は，新しい判別理論を確立し（New Theory of Discriminant Analysis after R.Fisher, 2016, Springer），その応用研究として30年以上統計で成功していないBig Dataの代表である「癌の遺伝子解析」に世界で初めて成功した(Cancer Gene Analysis to Cancer Gene Diagnosis, 2017,Amazon Kindle)．

・筆者は，LINGOのDEAの雛型モデルを修正し，学生でも簡単に比較する代替案の問題点を発見し，解決し，改革改善を提案できる，一般人が利用できる問題解決学を体系化した．成果は，Amazonの本シリーズ1で『あらゆる評価を可視化する「DEA による問題の発見と解決」』を2017年に定価2.99ドル（333円）で出版済み．本シリーズは，Amazonの最低価格で出版し，多くの人に役立てたいと考えています．Unlimitedユーザーは無償です．

本編の上巻は，LINGOのマニュアル的な機能の説明が多く，初めて読む読者は読み飛ばすように訳注を付けた．中間は全て重要な雛型モデルの解説と出力の見方です．特に，ノーベル経済学賞のマコーウイッツは，シカゴ大学ビジネススクー

ルでこのテーマで学位を取ったので，専門家でなくても多くのビジネスマンがその概略をLINGOの雛型モデルと出力で理解しやすい．また，整数計画法は役に立つであろう．

2017年10月　柏にて

目次

**上巻**

下巻

# 第 9 章　多期間計画問題

## 9.1　はじめに

**線形計画法(LP:Linear Programming)**の最も重要な応用の一つに多期間計画がある.通常の計画問題は本質的には一期間を考える場合が多い.一期間の計画の場合は,未来の期間の決定と現在の期間の決定があたかも独立するような仮定のもとで,現在の期間の定式化を行なう.しかし,現在の期間におけるある製品の生産が必要額を上回った場合でも,余剰生産分は価値がないわけではなくて,通常は次の期間に利用される.

このように期間と期間の間の相互関連は,LP モデルで容易に表わすことができる.事実,実用上起こる大きな LP 問題は,ほとんど多期間モデルである.この「多期」という言葉に対して通常「ダイナミック」という言葉が共通の同義語として用いられる.例えば,多期間 LP はダイナミック・モデルとも呼ばれる.

多期間 LP を長年使用する例として,チーズ製造がある.生産の決定は,月または週で行わなれる.しかし多くのチーズは,数ヶ月単位で生産されるかもしれない.例えば,パルメザンチーズは 10 カ月間,貯蔵する必要がある.アメリカ人がスイスチーズと呼ぶものは,2 から 4 カ月貯蔵する.チェダーチーズの様々な等級は,貯蔵される週数で左右される.このような応用で,モデルを多期間で考えることは重要である.

時間に依存する計画モデルは,幾つかの期間に時間を分けて実世界を表す.単一の期間に対応するモデルの部分は,製品組合せ,配合,および他のモデルの組合せであることもある.これらの単一期間(静的モデル)は,下記の方法で多期間になる:

①各商品および期間のリンクまたは在庫変数:
　連結変数は 1 期間から次の期に移る商品の量を表す.

②各商品および期間の「物質バランス」か「ソース=使用量」制約式を使用する:
　この制約式の簡単な形式は,「初期在庫+生産量=販売量+期末在庫」である.

多期間モデルは,通常ローリングかスライド形式になる.これらの形式の場合,モデルは毎期間の初めに解かれる.最初の期間の解の提案が実行される.1 期間が経過すると,より良いデータと予測が利用できるようになり,モデルは 1 期間先に進む.第 2 期間であったものが第 1 期間として解かれる.これが全プロセスで繰り返される.このモデルをこの方法で使用するとき,実用的な問題が生じる.最新の情報が利用できるようになると同時に,この期間の「最適解」は,前の期間の「最適解」と大きく異なることもある.解を実行しなければならない人々は,この違いに当惑するかもしれない.スケジューリング・システムは緊張に苛まれ苦しむと言われている.配船計画,工場の操業/停止,朝食のシリアルの生産などのアプローチで(Brown, Dell & Woodd (1997)参照),成功したアプローチの鍵は,「参

照」解（例えば，前の期間の解）の指定である．**参照解から現在の解の偏差を最小にすることを，第 2 の目的関数**にする．これにゼロの重みを掛ければ，論理上の最適解を得る．第 2 の目的関数に非常に高い重みを置けば，参照解に戻される．適度な重みを第 2 の目的関数に掛ければ，標準的な会計で測定されるような低価格のよい妥協を得るが，参照解に近い．

全期間を，同じ長さにする必要はない，例えば，石油会社が来年の生産計画を立てるとき，期間を各 4 半期に対応させるのが懸命である．1 つの可能な区切りとして，冬季を 12 月 1 日から 3 月 15 日，春季を 3 月 16 日から 5 月 15 日，夏季を 5 月 16 日から 9 月 15 日，秋季を 9 月 16 日から 11 月 30 日にする．

林業や鉱山会社は，50 年を単位に計画を立てる．そのような場合には，最初の 2 期間は 1 年ごと，次の 2 期間は 2 年ごと，次の 2 期間は 3 年ごと，さらに次の 2 期間は 5 年ごと，および最終的な 3 期間は 10 年ごとにすることもある．

期間と期間の間の相互関連は，通常，在庫量を示す決定変数を導入して LP で定式化できる．この在庫量を表わす決定変数が，ある期間とそれに隣接した期間を結びつける．例として今，各期間である単一の製品をどれだけ生産するかという単一の決定変数があるとしよう．第 j 期における決定変数を $P_j$ で表わす．さらにこの第 j 期においてある製品を $D_j$ だけ売る契約をしたとする．第 j 期末におけるこの製品の在庫量を表わす決定変数を $I_j$ で定義する．この慣例によると，第 j 期の期首における在庫量は $I_{j-1}$ で表わされる．この LP の定式化で，各期において製品の利用に関する「製品の出荷＝製品の消費」という単一の制約式をもつことになる．第 2 期で利用できる製品の出荷は，期首在庫量 $I_1$ とその期における生産量 $P_2$ である．製品を消費するのは需要 $D_2$ と期末在庫 $I_2$ である．例えば，もし $D_2=60$，$D_3=40$ としたら，第 2 期の制約は

$I_1+P_2=60+I_2$ または $I_1+P_2-I_2=60$

となる．また第三期の制約は，

$I_2+P_3-I_3=40$

となる．ここで，$I_2$ がどのように期間と期間を結びつけているか．すなわち，第 2 期における制約式と第 3 期における制約式の両方にこの $I_2$ が表われることで，これらの期間を結びつけていることに注意してほしい．

問題の種類によっては，ある期間で出ていく正味の量は，必ずしも次の期間における正味の流入量に等しい必要はない．例えば，今ここで製品と呼んでいるものの代わりに，それが現金の場合，期間と期間を結合する変数は，短期の借入とか短期の貸出というものになる．第 2 期の末で 1 ドルの貸出をした場合は，今利率を期間あたり 5％とすると，第 3 期に入ってくる量は 1．05 ドルになる．

これに反して，もしこれまで「製品」と呼んでいたものが，ここである種の生産設備である場合，そして 1 期間あたりの予測される償却を 10％とすると，上記 2 つの制約式は，

$$0.90I_1 + P_2 - I_2 = 60$$

$$0.90I_2 + P_3 - I_3 = 40$$

に修正される．この場合，$P_I$ で示す変数は第 I 期に生産した数量になる．

次に示す例は単一生産，多期間計画の状況を単純化して説明する．

## 9.2 動的生産計画問題

ここで単一期間の計画ではなくて，幾つかの多期間の生産計画問題を考えよう．ある会社が単一製品を生産していて，次の各四半期でこの製品に対する需要は下記であると予測する．

| 春 | 夏 | 秋 | 冬 |
|---|---|---|---|
| 20 | 30 | 50 | 60 |

全ての需要を満たす場合，次の二つの極端な政策を考えることができる．

①各期でその期の需要分を生産して，一切在庫をもたない．

②各四半期で正確に 40 単位ずつ生産して，需要の変動を在庫で吸収させる．

在庫を抱えるには費用がかかり，また生産水準の変化でも費用がかかるため，通常，最小費用政策は上の①と②の組み合わせになる．すなわち若干の在庫を持ち，また生産水準も若干変動させることになる．

費用計算の目的から，この会社がある期から次の期で生産水準を変化させた場合，1 単位あたり 600 ドルかかる．これらの費用は，通常「生産変動費」と呼ばれる．また，期末の 1 単位の在庫に対して 700 ドルの在庫費がかかる．また初期在庫はゼロとし，初期生産水準は各期で 40 単位である．そして，四半期の末で，同じ生産水準にもどすことが要求されている．

上記①の在庫なし政策の費用は，

$$\$600 \times (20+10+20+10+20) = \$48,000.$$

になる．また上記②のような生産水準固定の政策の費用は，

$$\$700 \times (20+30+20+0) = \$49,000.$$

になる．

この最小費用を探す問題は，①と②の二つの政策の組み合わせであり，LP として定式化し解くことができる．

### 9.2.1 定式化

ここで，第 I 期末（I=1，2，3，4）の手持ち在庫量として $I_I$ という変数を定義する．また $P_I$ は I 期の生産量，$U_I$ と $D_I$ は第（I－1）から I 期にかけての生産量の増加分，または減少分を表わす．変数 $P_I$ が，明らかに決定変数である．各期の費用を簡単に計算するために，変数 $I_I, U_I, D_I$，を定義しておくとよい．そこで年間の費用を最小化するためには，在庫費の総和

$$\$700I_1 + \$700I_2 + \$700I_3 + \$700I_4$$

と生産変動費，

$$\$600U_1 + \$600U_2 + \$600U_3 + \$600U_4 + \$600U_5 + \$600D_1 + \$600D_2 + \$600D_3 + \$600D_4 + \$600D_5.$$

の総和を最小化しなければならない．第 4 半期末で，生産水準を 40 に戻すために $U_5$ と $D_5$ を加えた．

### 9.2.2 制約

どの多期間問題の場合でも，各期各製品に対して「素材の均衡」あるいは「出てゆくものと入ってくるものとを等しくする」という制約が必要になる．これらを言葉で表わせば，通常の形としては，

期首在庫＋生産量－期末在庫＝需要

となる．

今，考えている問題についての制約を代数的に書くと，

$$\begin{aligned} P_1 - I_1 &= 20 \\ I_1 + P_2 - I_2 &= 30 \\ I_2 + P_3 - I_3 &= 50 \\ I_3 + P_4 &= 60 \end{aligned}$$

となる．

ここで $I_4$ と $I_0$ は，最初の制約と最後の制約式には表われていないことに注目しよう．これは，初期在庫また最終在庫はゼロとなることが要求されているからである．この定式化のままで解を求めた場合，$U_1$ や $D_1$ などの変数をゼロよりも大きい値に強制するものは何もない．すなわち，この場合，解は単純な生産政策，$P_1=20$，$P_2=30$，$P_3=50$，$P_4=60$ となる．この政策は，最終期を除いて各期末で生産増加することを意味する．そこで，これら $U_1, U_2, U_3, U_4$ の変数が適切な値をとるには，制約を次のように修正する．

$$\begin{aligned} U_1 &\geqq P_1 - 40 \\ U_2 &\geqq P_2 - P_1 \\ U_3 &\geqq P_3 - P_2 \\ U_4 &\geqq P_4 - P_3 \end{aligned}$$

上の制約式では，生産が減少した時には適切な措置がとられていない．そこで類似の次のような 4 つの制約でこの問題を解決できる．

$$\begin{aligned} D_1 &\geqq 40 - P_1 \\ D_2 &\geqq P_1 - P_2 \\ D_3 &\geqq P_2 - P_3 \\ D_4 &\geqq P_3 - P_4 \end{aligned}$$

さて，各四半期末に生産水準が 40 になるいう要求を組み込むためには，この最終四半期末で変化を計る変数として $U_5$ と $D_5$ を追加する．そして $U_5$ と $D_5$ は次の制約で適切な値をとるように強制される．

$U_5 \geqq 40 - P_4$

$D_5 \geqq P_4 - 40$

先へ進む前に，上のような 10 個の制約の定式化から，次のような 5 個の制約の定式化ができることに留意しよう．ここで鍵となる点は，2 つの制約，

$U_2 \geqq P_2 - P_1$

$D_2 \geqq P_1 - P_2$

は単一の制約，

$U_2 - D_2 = P_2 - P_1$

に置き換えられる．このようにできる論拠は，代数的議論というよりは，むしろ経済的な理論になる．ここで，この定式化の目的は，もし $P_2 - P_1 \geqq 0$ ならば $U_2 = P_2 - P_1$ に強制し，もし $P_1 - P_2 \geqq 0$ ならば $D_2 = P_1 - P_2$ となるように強制することである．この定式化で，経済性からいって最適解では $U_2$ か $D_2$ のどちらか一つだけが正となるのが分かる．もしこの 2 番目の定式化で，$U_2$ と $D_2$ の両方が正になるようなことがあれば，その場合は両方の変数とも同じ量だけ減らすことができる．そして，どの制約もやぶらずに費用を減少できる．そこで完全で最もコンパクトな定式化は次のようになる．

```
MODEL:
!MInImIze Inventory + workforce change costs;
MIN=700*I1+700*I2+700*I3+700*I4
+600*U1+600*U2+600*U3+600*U4
+600*D1+600*D2+600*D3+600*D4
+600*U5+600*D5;
!Initial conditions on Inventory & Production;
[CNDBI] I0=0;
[CNDBP]P0=40;
 !Begining Inventory+Production=Demand+ending inventory;
[INV1]I0+P1=20+I1;
[INV2]I1+P2=30+I2;
[INV3]I2+P3=50+I3;
[INV4]I3+P4=60+I4;
 !ChangeUP-changeDown=Prod.thisperiod-Prod.prev.period;
[CHG1]U1-D1=P1-P0;
[CHG2]U2-D2=P2-P1;
[CHG3]U3-D3=P3-P2;
[CHG4]U4-D4=P4-P3;
[CHG5]U5-D5=P5-P4;
!Ending conditions;
```

```
[CNDEI]I4=0;
[CNDEP]P5=40;
END
```

解は次のようになる．

```
Optimal solution found at step:          7
Objective value:                  43000.00
VarIable           Value         Reduced Cost
      I1         5.000000          0.0000000
      I2        0.0000000           200.0000
      I3         5.000000          0.0000000
      I4        0.0000000          0.0000000
      U1        0.0000000           1200.000
      U2        0.0000000           250.0000
      U3         30.00000          0.0000000
      U4        0.0000000           250.0000
      D1         15.00000          0.0000000
      D2        0.0000000           950.0000
      D3        0.0000000           1200.000
      D4        0.0000000           950.0000
      U5        0.0000000           1200.000
      D5         15.00000          0.0000000
      I0        0.0000000          0.0000000
      P0         40.00000          0.0000000
      P1         25.00000          0.0000000
      P2         25.00000          0.0000000
      P3         55.00000          0.0000000
      P4         55.00000          0.0000000
      P5         40.00000          0.0000000
     Row    Slack or Surplus      Dual Price
       1         43000.00          -1.000000
   CNDBI        0.0000000          -950.0000
   CNDBP        0.0000000          -600.0000
    INV1        0.0000000           950.0000
    INV2        0.0000000           250.0000
    INV3        0.0000000          -250.0000
    INV4        0.0000000          -950.0000
    CHG1        0.0000000           600.0000
```

| | | |
|---|---|---|
| CHG2 | 0.0000000 | -350.0000 |
| CHG3 | 0.0000000 | -600.0000 |
| CHG4 | 0.0000000 | -350.0000 |
| CHG5 | 0.0000000 | 600.0000 |
| CNDEI | 0.0000000 | -1650.000 |
| CNDEP | 0.0000000 | 600.0000 |

解は次のように混合戦略になる．

$P_1 = P_2 = 25$；　$P_3 = P_4 = 55$．

このように LP で見い出した混合政策は，すでに述べた純粋政策の最良のものと比較しても 5,000 ドルも安いことになる．通常の現実的な多期間問題の場合には，製品が一種類ではなくて，複数の製品が係わってくる．次の例はこの方向で現実に一歩近づいて，数種類の製品を考慮するものである．

### 9.2.3　絶対値の表現

次が，生産の切り替え費用を表す：

600 * (@ABS (P1 - P0) + @ABS (P2 - P1) +… +@ABS (P5 -P4))；

これは LP を NLP に変える．LINDO SystemsInc．は，次の外見上明らかに非線形なものを線形にする方法を開発した．ある特定の条件下で，どの項目のどの表現も @ABS (表現)に取り替えることができる：

U-D=表現；

「ある特定の条件」とは，小さい値の@ABS (表現)の方が，大きい値の@ABS (表現)より好まれるということである．もし「表現」が正なら，U は「表現」と等しい．「表現」が負なら，「-表現」と等しくなる．

## 9.3　多期財務計画モデル

通常の多期間計画で，流動資産あるいは現金に類似の資産を管理することが重要な問題になる．もしも，現金保持を他の商品の在庫と同じように 1 つの在庫と考えると，上で考えた問題をもう少し進めることで，財務管理の意思決定問題を多期間モデルで表現できる．ここでカギとなる特色は，各期で「現金流入－現金流出＝0」という制約がある．次の簡単で現実的な例で，この種のモデルの主要な特徴を説明する．

### 9.3.1　キャシュフロー

**-未来の現金必要量の流れを満たすためのポートフォリオの作成-**

まず綿密な計画を作成し実行するため，今後 14 年間に次のような現金が必要になると仮定しよう．

| 年： | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| 1000$ | 10 | 11 | 12 | 14 | 15 | 17 | 19 |
| 年： | 8 | 9 | 10 | 11 | 12 | 13 | 14 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **1000$** | 22 | 24 | 26 | 29 | 31 | 33 | 36 |

このような計画が必要となる通常の例として，個人的な障害裁判がある．障害を受けた側が上記のような支払いの流れを将来にわたって受け取れるという話し合いが成立する場合が多い．他の例として，年金基金に必要な現金を満たすための債権のポートフォリオ，ある負債が別の現金の流れで置き換えられる，いわゆるバランスシートの無効化と呼ばれるものが挙げられる，

障害裁判の例の場合，管理を簡単にするために，両者とも上記の 15 回の支払いと等価の一時金払いを望むこととする．この一時金を受け取る側としては，普通預金のようなリスクの少ない低利息を使って，上記の現金の流れの現在価値に等しい額を受け取りたいとする．例えば，4％の利子率の場合は，この支払いの流れの現在価値は 230,437 ドルである．しかしながら，一時金を支払う側としては，もっと高い利息を使いたいと主張することがある．このような議論が首尾よく成立するためには，実際にそのような高い利子率を正当化する投資が可能であり，普通預金などよりも危険がないことを示さなければならない．このような場合の投資に，通常は政府の証券がある．一般に，ある任意の日で，そのような多くの投資の可能性がある．今ここで話を簡単にするために下記のような 2 つの投資機会があると仮定しよう．

| 証券 | 現費用 | 年間収益額 | 満期までの年数 | 満期時の支払額 |
|---|---|---|---|---|
| 1 | $980 | $60 | 5 | $1000 |
| 2 | $965 | $65 | 12 | $1000 |

支払い側は，ある額の一時支払い金で，そのうちどれだけを証券 1 に投資し，また証券 2 に投資するか，またどれだけを普通預金に投資するかについてお薦めの計画を提示する．そしてそれで各年の必要現金額が，この最小の一時支払い金で賄われことを提示する．この問題を解くには，次のような決定変数が役に立つ．

**$B_1$ ＝証券 1 に現在投資される額，これは額面で取り扱う．**

**$B_2$ ＝証券 2 に現在投資される額，これも額面で取り扱う．**

**$S_I$ ＝I 年において普通預金に投資される額．**

**L ＝初期の一時金額．**

ここで目的関数は初期の一時金を最小にすることである．そして各年で，現金の正味の流れをゼロとする制約が必要となる．そこで，余った現金は利息 4％で普通預金に投資する．全金額を 1,000 ドル単位で計ると，この問題の定式化は次のようになる．

```
MIN = L;
L - 0.98 * B1 - 0.965 * B2 - S0 = 10;
0.06 * B1 + 0.065 * B2 + 1.04 * S0 - S1 = 11;
0.06 * B1 + 0.065 * B2 + 1.04 * S1 - S2 = 12;
```

```
0.06 * B1 + 0.065 * B2 + 1.04 * S2 - S3 = 14;
0.06 * B1 + 0.065 * B2 + 1.04 * S3 - S4 = 15;
1.06 * B1 + 0.065 * B2 + 1.04 * S4 - S5 = 17;
0.065 * B2 + 1.04 * S5 - S6 = 19;
0.065 * B2 + 1.04 * S6 - S7 = 20;
0.065 * B2 + 1.04 * S7 - S8 = 22;
0.065 * B2 + 1.04 * S8 - S9 = 24;
0.065 * B2 + 1.04 * S9 - S10 = 26;
0.065 * B2 + 1.04 * S10 - S11 = 29;
1.065 * B2 + 1.04 * S11 - S12 = 31;
1.04 * S12 - S13 = 33;
1.04 * S13 - S14 = 36;
```

制約の各係数を PICTURE コマンドで表すと，この問題の構造がよく理解できる. A は 1 より大きく 10 未満の数字を表す. B は，10 より大きく 100 未満の数を表す. T は 1 未満であるが, 最低 0.1 である数を表す. U は 0.1 未満であるが，最低 0.01 である数を表す.

| | L | B1 | B2 | S0 | S1 | S2 | S3 | S4 | S5 | S6 | S7 | S8 | S9 | S10 | S11 | S12 | S13 | S14 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1: | 1 | | | | | | | | | | | | | | | | | | = | A |
| 2: | 1 | -T | -T | -1 | | | | | | | | | | | | | | | = | B |
| 3: | | U | U | A | -1 | | | | | | | | | | | | | | = | B |
| 4: | | U | U | | A | -1 | | | | | | | | | | | | | = | B |
| 5: | | U | U | | | A | -1 | | | | | | | | | | | | = | B |
| 6: | | U | U | | | | A | -1 | | | | | | | | | | | = | B |
| 7: | | A | U | | | | | A | -1 | | | | | | | | | | = | B |
| 8: | | | U | | | | | | A | -1 | | | | | | | | | = | B |
| 9: | | | U | | | | | | | A | -1 | | | | | | | | = | B |
| 10: | | | U | | | | | | | | A | -1 | | | | | | | = | B |
| 11: | | | U | | | | | | | | | A | -1 | | | | | | = | B |
| 12: | | | U | | | | | | | | | | A | -1 | | | | | = | B |
| 13: | | | U | | | | | | | | | | | A | -1 | | | | = | B |
| 14: | | | A | | | | | | | | | | | | A | -1 | | | = | B |
| 15: | | | | | | | | | | | | | | | | A | -1 | | = | B |
| 16: | | | | | | | | | | | | | | | | | A | -1 | = | B |

7 段目の $B_1$ の係数が 1.06 であることに留意する必要がある. これは額面 1000 ドルに対して元金の支払いが 1,000 ドル戻って来て，利子の支払いが 60 ドルで

あることを示す．変数 $S_{14}$（最終の支払いが済んだ後の普通預金への投資額）がこの問題に表われているが，このようなオプションをこの問題に与えることは意味がないと思えるかもしれない．しかし，この $S_{14}$ は実際には，最終期における現金の余りを示す変数である．しかし，それにもかかわらず，この問題を解いた段階で初期一時金払いを最小にするこの解で，最終期の終わりに現金が余るような解が得られることは異常ではない．なぜなら，中間期間に必要な資金を払うには，債券が最も経済的になるかもしれないし，それは最終期で債券が満期になった時，大きな元金の払い戻しで最も遠い最終期で，必要資金以上が戻ってくることが考えられる．この場合の解は次のようになる．

```
Optimal solution found at step:        14
ObjectIve valUe:                 195.6837
VarIable              ValUe        ReDUceD Cost
       L           195.6837           0.0000000
      B1           95.79577           0.0000000
      B2           90.15474           0.0000000
      S0           4.804497           0.0000000
      S1           5.604481           0.0000000
      S2           5.436464           0.0000000
      S3           3.261727           0.0000000
      S4          0.0000000           0.1069792
      S5           90.40358           0.0000000
      S6           80.87978           0.0000000
      S7           69.97503           0.0000000
      S8           56.63409           0.0000000
      S9           40.75951           0.0000000
     S10           22.24994           0.0000000
     S11          0.0000000           0.1412458
     S12           65.01479           0.0000000
     S13           34.61538           0.0000000
     S14          0.0000000           0.3796368
```

この一時金支払い額 195,683.7 ドルのうち，10,000 ドルがこの期の支払いに回され，4,804.50 ドルが普通預金に回されている．そして，0.98×95795.77＋0.965×90154.74＝180876.2 ドルが長期債券へ回ることになる．普通預金だけを考慮するのではなくより広い範囲の投資を考察することで，一時金払いの総額は 15％減って，34,750 ドルに減少した．

実際の解で，一時金額のほとんどが単一の証券に投資されることもありうる．例えば，証券 1 に投資される額は初期一時金支払い額の半分を超えてはいけないというような制約は次のように追加できる．

$$0.98B_1 - 0.5L \leqq 0$$

このような制約は，典型的なポートフォリオ制約と呼ばれるものである．

もう一つ状況を複雑にするのは，債券に投資される額 $B_1, B_2$ は端数になってはいけないということである．例えば，債券は 1,000 ドルを単位として売買がなされる．しかし，一般には適当な判断で，端数の値を最も近い整数の値にすれば，一時金支払い額はそれほど多く増加するものではない．例えば上記の例で，$B_1$ と $B_2$ を各 96 と 90 に設定しても，総費用は 195,726.50 ドルから 195,683.70 ドルに増加するだけである．これで $S_{14}$ はゼロでなくなり，そして最終の支払い余剰分はほぼ 40 ドルになる．

次は，集合を用いたモデルである：

```
MODEL:
!Name=PBOND,Bond Portfolio/cash matching problem:Given cash
needs in each future period, what collection of bonds should
we buy to cover these needs?;
SETS:
BOND/1..2/ : MATAT, ! Matures at Period;
             PRICE, ! Purchase Price;
             CAMNT, ! CoUupon Payout each Period;
              BUY;  ! Amount to buy of each bond;
PERIOD/1..15/:
     NEED,     ! Cash needed each Period;
SINVEST; ! Short term Investment each Period;
ENDSETS
DATA:
STRTE = .04;           ! Short term Interest rate;
MATAT = 6,   13;       ! Years to maturity;
PRICE = .980, .965;! Purchase Price in thousands;
CAMNT = .060, .065; ! Coupon amount in thousands;
NEED = 10, 11, 12, 14, 15, 17, 19, 20, 22, 24,
       26, 29, 31, 33, 36; ! Cash needed in
       thousands;
ENDDATA
!----------------------------------------------;
MIN = LUMP;
```

```
! First Period Is slightly special;
LUMP =
NEED(1) + SINVEST( 1) + @SUM( BOND: PRICE * BUY);
! For sUbseqUent Periods;
@FOR( PERIOD( I)| I #GT# 1:
 @SUM( BOND( J)| MATAT( J) #GE# I:
     CAMNT( J) * BUY( J)) +
  @SUM( BOND( J)| MATAT( J) #EQ# I:  BUY( J)) +
   ( 1 + STRTE) * SINVEST( I - 1) =
   NEED( I) + SINVEST( I););
! Can only bUy Integer bonDs;
@FOR( BOND( J): @GIN( BUY( J)););
END
```

出力は次のようになる.

```
Optimal solution found at step:        28
ObjectIve valUe:                 195.7265
Branch coUnt:                           3
    VarIable           Value        Reduced Cost
       STRTE      0.4000000E-01        0.0000000
        LUMP        195.7265           0.0000000
   MATAT( 1)        6.000000           0.0000000
   MATAT( 2)        13.00000           0.0000000
   PRICE( 1)       0.9800000           0.0000000
   PRICE( 2)       0.9650000           0.0000000
   CAMNT( 1)      0.6000000E-01        0.0000000
   CAMNT( 2)      0.6500000E-01        0.0000000
     BUY( 1)        96.00000           0.7622063
     BUY( 2)        90.00000           0.7290568
    NEED( 1)        10.00000           0.0000000
    NEED( 2)        11.00000           0.0000000
    NEED( 3)        12.00000           0.0000000
    NEED( 4)        14.00000           0.0000000
    NEED( 5)        15.00000           0.0000000
    NEED( 6)        17.00000           0.0000000
    NEED( 7)        19.00000           0.0000000
    NEED( 8)        20.00000           0.0000000
    NEED( 9)        22.00000           0.0000000
```

| | | |
|---|---|---|
| NEED( 10) | 24.00000 | 0.0000000 |
| NEED( 11) | 26.00000 | 0.0000000 |
| NEED( 12) | 29.00000 | 0.0000000 |
| NEED( 13) | 31.00000 | 0.0000000 |
| NEED( 14) | 33.00000 | 0.0000000 |
| NEED( 15) | 36.00000 | 0.0000000 |
| SINVEST( 1) | 4.796526 | 0.0000000 |
| SINVEST( 2) | 5.598387 | 0.0000000 |
| SINVEST( 3) | 5.432322 | 0.0000000 |
| SINVEST( 4) | 3.259615 | 0.0000000 |
| SINVEST( 5) | 0.0000000 | 0.8548042 |
| SINVEST( 6) | 90.61000 | 0.0000000 |
| SINVEST( 7) | 81.08440 | 0.0000000 |
| SINVEST( 8) | 70.17778 | 0.0000000 |
| SINVEST( 9) | 56.83489 | 0.0000000 |
| SINVEST( 10) | 40.95828 | 0.0000000 |
| SINVEST( 11) | 22.44661 | 0.0000000 |
| SINVEST( 12) | 0.1944784 | 0.0000000 |
| SINVEST( 13) | 65.05226 | 0.0000000 |
| SINVEST( 14) | 34.65435 | 0.0000000 |
| SINVEST( 15) | 0.4052172E-01 | 0.0000000 |

## 9.4 税金を考慮した財務計画モデル

次の問題では，上のポートフォリオ選択問題よりも複雑なものを取り扱う．そして，次に税金の効果を含み，それを検討する方法を説明する．ウィンストン・サレム開発管理社（WSDM 社）は，今後 3 年間で以下の投資計画を完成させようとしている．現在のところ WSDM 社は，投資のために 200 万ドルを持っている．今後 3 年間で，半年間ごとに WSDM 社は過去の投資から次のような収入を期待できる．すなわち，現在から数えて半年ごとに，$500,000，$400,000，$380,000，$360,000，$340,000，そして，3 年目の終わりには$300,000 である．また，WSDM 社が参入しようと考えている開発計画は 3 つある．第 1 の計画は，フォスター市場開発計画である．もし，WSDM 社が全面的に参加するならば，この開発計画は，今後 3 年間で，半年ごとに次のような現金の流れが期待できる．負の数字は投資を表わし，正の数字は収入を表わす．−$3000000，−$1000000，−$1800000，$4000000，$1800000，$1800000，$5500000 である．最後の数字は，3 年後の推定価値である．2 番目の計画は，ある中低所得者用住宅の運用を引き取って，若干の補修をして，3 年目の終わりには廃業する．この計画に全面的に参加した場合には，現金の流れは次のようになる．−$2000000，−$500000，$1500000，

$1500000, $1500000, $200000, −$1000000 である.3 番目の計画のディズニー・ユニバース・ホテル計画は，もし WSDM 社が全面的に参加すれば，次のような現金の流れとなる.ここでまた，最後の数字は，3 年目の終わりの推定価値である.−$2000000，−$2000000，−$1800000，$1000000，$1000000，$1000000, $6000000.WSDM 社は，半年あたり 3.5％の利子率で，半年単位で資金を借りることができる.どの時点でも，200 万ドルが借り入れの上限である.すなわち，借りている資金が 200 万ドルを超えることができないものとする.WSDM 社は，余った資金は半年あたり 3％の利子で投資できる.初めに，税金の問題を無視する.そして，3 年目の終わりで WSDM 社の実質の価値を最大化する問題を LP で定式化する.もし WSDM 社が，各プロジェクトに全面的に参加するのでない場合は，そのプロジェクトの現金の流れは各比率で減少するものとして計算する.

### 9.4.1 定式化と解

各変数を次で定義する.

F ＝フォスター市の計画に対する参画の比率

M ＝中低所得者用住宅運用への参画の比率

$D_I$ ＝ディズニー計画への参画の比率

$B_I$ ＝第 I 期における借入額（1,000 ドル単位）

$L_I$ ＝第 I 期における貸出額（1,000 ドル単位）

Z ＝6 期後における純価値（1,000 ドル単位）

この問題の定式化は，次のようになる（全ての数字は 1,000 ドル単位）.

```
MODEL:
MAX = Z; ! Max worth at end of final Period;
!Uses-sources = supplyofcashin each period;
 3000*F+2000*M+2000*D-B1+L1=2000;
1000*F+500*M+2000*D+1.035*B1-1.03*L1-B2+L2=500;
1800*F-1500*M+1800*D+1.035*B2-1.03*L2-B3+L3=400;
-400*F-1500*M-1000*D+1.035*B3-1.03*L3-B4+L4=380;
-1800*F-1500*M-1000*D+1.035*B4-1.03*L4-B5+L5=360;
-1800*F-200*M-1000*D+1.035*B5-1.03*L5-B6+L6=340;
Z-5500*F+1000*M-6000*D+1.035*B6-1.03*L6=300;
!Borrowing limits;
B1<=2000;
B2<=2000;
B3<=2000;
B4<=2000;
B5<=2000;
B6<=2000;
```

```
!We can invest at most 100% in a Project;
F<=1;
M<=1;
D<=1;
END
```

制約式の4から17までは，すべて各期における現金の流れの制約である．これらの制約は，各期において

（現金の流出）－（現金の流入）＝0

とする．例えば，最初の期で $L_I$ は現金の利用額であり，$B_I$ は現金の供給である．解は次の通りである．

```
Opsolution found at step:          11
Objective Value:                   7665.179
Variable            Value          Reduced Cost
       Z         7665.179             0.0000000
       F        0.7143414             0.0000000
       M        0.6372096             0.0000000
       D        0.0000000              452.3816
      B1         1417.443             0.0000000
      L1        0.0000000         0.8788487E-02
      B2         2000.000             0.0000000
      L2        0.0000000             0.3343139
      B3         2000.000             0.0000000
      L3        0.0000000             0.2509563
      B4         448.4490             0.0000000
      L4        0.0000000         0.5304549E-02
      B5        0.0000000         0.5149997E-02
      L5         2137.484             0.0000000
      B6        0.0000000         0.5000029E-02
      L6         3954.865             0.0000000
     Row    Slack or Surplus       Dual Price
       1         7665.179              1.000000
       2        0.0000000              1.819220
       3        0.0000000              1.757701
       4        0.0000000              1.381929
       5        0.0000000              1.098032
       6        0.0000000              1.060900
       7        0.0000000              1.030000
```

| | | |
|---|---|---|
| 8 | 0.0000000 | 1.000000 |
| 9 | 582.5567 | 0.0000000 |
| 10 | 0.0000000 | 0.3274043 |
| 11 | 0.0000000 | 0.2454662 |
| 12 | 1551.551 | 0.0000000 |
| 13 | 2000.000 | 0.0000000 |
| 14 | 2000.000 | 0.0000000 |
| 15 | 0.2856586 | 0.0000000 |
| 16 | 0.3627904 | 0.0000000 |
| 17 | 1.000000 | 0.0000000 |

よって WSDM 社は，フォスター市の計画に対して 0.7143414, 中低所得者用住宅運用計画は 0.672096, ディズニー計画に対して 0 の投資もしくは購入をするのが良いという結果が出た．この計画期間の終わりには純価値が 7,665,179 に増えることが分かった．

### 9.4.2 双対価格の解釈

最初の 7 個の各制約式の双対価格は，各期において，もう 1(×1000) ドル余分に利用できるようになったら，その結果，目的関数である Z すなわち最終期における純価値がどれだけ増えるかを表わす．例えば，1.81922 の表わす意味は，第 1 期にあと 1(×1000) ドル利用できるとしたら，最終期の純価値すなわち Z があと 1.82 (×1000) ドル増えるということである．

第 5 期にあと 1(×1000) ドルあったとしたら，その最終期における純価値は，1.03×1.03＝1.0609 (×1000) ドルに相当する．なぜならば，第 5 期にあと 1 単位余分にあっても，3％の利率であと 2 期間しか使えないからである．

第 4 期にあと 1(×1000) ドルの資金が利用できるとしたら，その分だけ第 4 期の借入金 B4 が少なくてすむ．そのため，第 5 期には，1.035(×1000) ドルだけ豊かになっているはずである．ところが，すでに観察したように，第 5 期のあと 1 ドルの資金は 1.0609 ドルの価値があるので第 4 期における 1 ドルの余分の資金は，1.035×1.0609＝$1.098032 の価値があることになる．

借入金についての双対価格の意味は，他の双対価格の意味と，次のようにつじつまを合わせることができる．第 2 期で，あと 1 ドル借入れできたとしたら，それは 1.7577 ドルの価値がある（第 2 制約の双対価格）．もし，この 1 ドルを借入したとしたら，第 3 期で 1.035 ドル支払わねばならない．それは，実際には 1.035×1.38193 の費用となる．従って，第 2 期のあと 1 ドルを借入れることを最終期の純価値で評価すると，1.7577−1.035×1.38193＝0.3274 となる．これは，第 2 期の借入制約の双対価格に一致する．

ある t 期における実質的な利子率，すなわち資本の費用 I を導くには，双対価格を使って，進んで資金を借り受けるに値する利率を求めればよい．t 期において 1

ドル借りれば，確かにｔ期で１ドル余分に使えるが，次の（ｔ＋1）期においては，（1＋Ｉ）ドル支払わなければならない，そこで，これらの２つの事項を均衡させなければならない．第１期について考えよう．第１期のあと１ドル余分の資金は，最終期の第６期では，１．81922 ドルの価値がある．しかし，第２期に（1＋Ｉ）ドル支払うことになり，それは最終期の価値にすれば（1＋Ｉ）×1．7577 ドルの費用となる．これらを均衡させれば次のようになる．

1．81922 ＝（1＋Ｉ）×1．7577

これより，

Ｉ ＝0．035

これは，第１期において，そもそもこの率で借りているのだから驚くべきことではない．しかし，ぎりぎりまで借りてはいけない．他の期でも，同様の分析を応用すれば，次のような実質利子率を算出できる

| 期間 | Ｉ | 期間 | Ｉ |
|---|---|---|---|
| 1 | 0．035 | 4 | 0．035 |
| 2 | 0．2719 | 5 | 0．03 |
| 3 | 0．2586 | 6 | 0．03 |

## 9．5　現在価値　対　LP の分析

プロジェクトを評価するための標準的な方法は，現金流動の流れの現在価値の計算である．これまで説明したように，LP の分析は現在価値（PV）の分析である．PV 分析の仮定は，お金が次のような場合もあることである：

a）単一の利率で，

b）もしくは限界なしで，

c）借りるか，貸し付ける．

ちょうど考慮されるこのような LP モデルは，PV 分析と同じ仮定がなされれば同じ推薦を与える．しかし LP 分析は貸出と別の借入を別の金利にできたり，借入れ期間と貸付期間がばらばらでもよい． もしくは，借入れと貸付の上限額を別に設定できる．

PV 分析のように，LP による分析は内部収益率がプロジェクトを評価するのに使用されているとき起きうる多数の収益率のあい昧性を避ける． $1，000，000 の初期投資を要求し，1 年後に$2，500，000 が払い戻され，そして 2 年後に終了費用として$1，550，000 支払うプロジェクトを考慮する．このプロジェクトは年間 13．82％と，36．18％の内部収益率がある．資本費用が年間 11％ならプロジェクトは魅力的だと言えるであろうか．資本費用が年間 12％ならば， PV と LP の両方ともこのプロジェクトを（正しく）拒否し， 24％ならば受け入れ，38％ならば拒否するであ

ろう．

### 9．6　税金の清算

ここで税金を考慮に入れることにする．次のような簡略化した状況を考えてみよう．どの期でも利益には 50％の税金がかかるものとする．ある期で損失がある場合，その 80％は税金の軽減のため，次の期に持ち越すことができるものとする．各期の 3 つの計画で，売り上げから支出を引いたものは，次のようである．この支出には，減価償却費も含まれている．

| 期間 | フォスター市 | 中低所得者住宅 | ディズニー | 既存 |
|---|---|---|---|---|
| 1 | −100，000 | −200，000 | −150，000 | 0 |
| 2 | −300，000 | −400，000 | −200，000 | 100，000 |
| 3 | −600，000 | −200，000 | −300，000 | 80，000 |
| 4 | −100，000 | 500，000 | −200，000 | 76，000 |
| 5 | 500，000 | 1，000，000 | 500，000 | 72，000 |
| 6 | 1，000，000 | 100，000 | 800，000 | 68，000 |
| 7 | 4，000，000 | −1，000，000 | 5，000，000 | 60，000 |

**将来の現金の流れに対して 20％の税金がかかるものとする．**税金を考慮した場合，前述の変数に加えて次のような変数を定義する．

$P_I$ ＝第 I 期における利益

$C_I$ ＝第 I 期における損失

ここで，定式化は次のように影響される．まず第 1 に，$P_I$ と $C_I$ を適切に計算する方程式を加えなければならない．第 2 に，税金の支払いにむけられる現金の流れの制約式を追加しなければならない．

この税金を計算する方程式は，次のようになる．

利益−損失＝売り上げ−支払い−0．8×（前期の損失）

代数的には，この第 2 期における方程式は次のようになる．

$$P_2-C_2=100+0.03L_1-300F-400M-200D-0.035B_1-0.8C_1$$

標準形は次のようになる．

$$P_2-C_2-0.03L_1+300F+400M+200D+0.035B_1+0.8C_1=100$$

全体像は，次のようになる．

```
MAX = Z;
!Cash flow constraints, Uses-sources= 0, Including the 50%
tax Usage;
3000*F+2000*M+2000*D-B1+L1+0.5*P1=2000;
1000*F+500*M+2000*D+1.035*B1-1.03*L1-B2+L2+0.5*P2=500;
```

```
1800*F-1500*M+1800*D+1.035*B2-1.03*L2-B3+L3+0.5*P3=400;
-400*F-1500*M-1000*D+1.035*B3-1.03*L3-B4+L4+0.5*P4=380;
-1800*F-1500*M-1000*D+1.035*B4-1.03*L4-B5+L5+0.5*P5=360;
-1800*F-200*M-1000*D+1.035*B5-1.03*L5-B6+L6+0.5*P6=340;
Z-5500*F+1000*M-6000*D+1.035*B6-1.03*L6+0.5*P7=300;
!TheborrowInglImIts;
B1<=2000;
B2<=2000;
B3<=2000;
B4<=2000;
B5<=2000;
B6<=2000;
!TheInvestInglImIts;
F<=1;
M<=1;
D<=1;
! Taxable Profit-Loss for each Period;
100*F+ 200*M+ 150*D +P1-C1= 0;
300*F+ 400*M+ 200*D+0.035*B1-0.03*L1+P2+0.8*C1-C2=100;
600*F+ 200*M+ 300*D+0.035*B2-0.03*L2+P3+0.8*C2-C3= 80;
100*F- 500*M+ 200*D+0.035*B3-0.03*L3+P4+0.8*C3-C4= 76;
-500*F-1000*M- 500*D+0.035*B4-0.03*L4+P5+0.8*C4-C5= 72;
-1000*F- 100*M- 800*D+0.035*B5-0.03*L5+P6+0.8*C5-C6= 68;
-4000*F+1000*M-5000*D+0.035*B6-0.03*L6+P7+0.8*C6-C7= 60;
```

解は次のとおりである．

```
ObjectIve valUe:                 5899.975
VarIable          Value          Reduced Cost
       Z       5899.9750          0.0000000
       F       0.4872107          0.0000000
       M       1.0000000          0.0000000
       D       0.0000000          945.00740
      B1       1461.6320          0.0000000
      L1       0.0000000          0.5111823E-02
      P1       0.0000000          0.4499472
      B2       2000.0000          0.0000000
      L2       0.0000000          0.1960928
      P2       0.0000000          0.3793084
```

| | | |
|---|---|---|
| B3 | 1046.9790 | 0.0000000 |
| L3 | 0.0000000 | 0.3167932E-02 |
| P3 | 0.0000000 | 0.2042549 |
| B4 | 0.0000000 | 0.2575563E-02 |
| L4 | 991.26070 | 0.0000000 |
| P4 | 0.0000000 | 0.1107492 |
| B5 | 0.0000000 | 0.2537532E-02 |
| L5 | 3221.6490 | 0.0000000 |
| P5 | 1072.6580 | 0.0000000 |
| B6 | 0.0000000 | 0.2499981E-02 |
| L6 | 4359.3480 | 0.0000000 |
| P6 | 751.86020 | 0.0000000 |
| P7 | 1139.6230 | 0.0000000 |
| C1 | 248.72110 | 0.0000000 |
| C2 | 696.29720 | 0.0000000 |
| C3 | 1039.3640 | 0.0000000 |
| C4 | 340.85670 | 0.0000000 |
| C5 | 0.0000000 | 0.1091125 |
| C6 | 0.0000000 | 0.1075000 |
| C7 | 0.0000000 | 0.5000000 |
| Row | Slack or Suroplus | Dual Price |
| 1 | 5899.9750 | 1.0000000 |
| 2 | 0.0000000 | 1.3218740 |
| 3 | 0.0000000 | 1.2860920 |
| 4 | 0.0000000 | 1.0678540 |
| 5 | 0.0000000 | 1.0456780 |
| 6 | 0.0000000 | 1.0302250 |
| 7 | 0.0000000 | 1.0150000 |
| 8 | 0.0000000 | 1.0000000 |
| 9 | 538.36780 | 0.0000000 |
| 10 | 0.0000000 | 0.1924019 |
| 11 | 953.02070 | 0.0000000 |
| 12 | 2000.0000 | 0.0000000 |
| 13 | 2000.0000 | 0.0000000 |
| 14 | 2000.0000 | 0.0000000 |
| 15 | 0.5127893 | 0.0000000 |
| 16 | 0.0000000 | 573.56060 |

| | | |
|---|---|---|
| 17 | 1.0000000 | 0.0000000 |
| 18 | 0.0000000 | -0.2109901 |
| 19 | 0.0000000 | -0.2637376 |
| 20 | 0.0000000 | -0.3296720 |
| 21 | 0.0000000 | -0.4120900 |
| 22 | 0.0000000 | -0.5151125 |
| 23 | 0.0000000 | -0.5075000 |
| 24 | 0.0000000 | -0.5000000 |

税金を考慮すると，LP 解が実質的に大きな変化を受けたことに注目すべきである．この新しい解で中低所得者用計画 X（プロジェクト X）に，より多くの資金が割り当てられ，フォスター市計画への投資は減少している．プロジェクト X にはキャッシュフローがあり，年間の利益の流れを滑らかにする手助けとなる．

## 9.7　動的ネットワークか多期間ネットワークか

ここまで，我々は主にネットワーク問題を定常状態か 1 つの期間問題として見てきた．例えば，パイプライン・ネットワーク・モデルを例として挙げると，解は，毎日連続的に起こる資源の流れ，もしくは一定の期間起こり，止まる流れとして解釈される．しかし本来は，期間ごとに変わりゆく流れの方が現実的であり，多期間や動的な解により興味がある．考慮すべきこれらの多期間フローは，あるアークを超えるのにフローが複数の期間を有することがある．動的ネットワークの例として以下が挙げられる．

① 多様なダムと川のシステムが挙げられる．水力発電や川の流れ，川や湖の水位の必要性を満たすべく色々な基準を満たすために，各期間で川系とダムから零れる水の量への興味がある．その期間は一日であったり，アークはダムからダムへの川の断面であったり，またはあるダムから次のダムへ水が流れるのに数期間もかかる場合もある．

② 危険の迫っている施設の避難もしくはある地域の避難対策として，短時間でなるべく多くの人を非難させられるルートを知っておく必要がある．建物からの避難の場合，期間は 10 秒であったりする． アークは廊下または階段の吹き抜けのドアからドアであったりする． アークごとに，一定時間にどれだけの人が入れるかという許容範囲が決まっている．

③ 飛行機やトラックの最速ルート．ネットワークの各アークは飛行機やトラックでの動き．アークのリードタイムは，アーク間を移動する乗り物の移動時間となる．

③ 動的ネットワークを代数的に表現するために次を定義する．

**パラメータ**：

$L(I, j)$ ＝ ある期間のリードタイム．フローがアーク内のノード I から j へ移動する時間

**変数**：

$x_{Ijt}$ = 期間 t 内に I からアーク Ij 全体をフローする．よって j から出てくるのは，期間 t+L(I,j)

$V_{jt}$ = 在庫は期間 t の最後にノード j に留まる．

基本的なノードのバランス方程式は，

**(期間 t の期末在庫はノード k にある) = (現在の期間の期末在庫は k)**

**+ (到着する積荷) - (機関 t で k から出される積荷)**

もしくは代数的に，

$$V_{kt} = V_{kt-1} + \Sigma_I \ x_{Ik(t-L(I,k))} - \Sigma_j \ x_{kjt}$$

である．

**例**：ある建物の避難問題を使って本旨を説明する．詳細については，集合ベースのLINGOモデルを参照． モデルはwww.lindo.com.Libraryからダウンロードできる（evacu8.lngというファイル名である）．**図9.1**には，これから避難路を計画する建物の数値的詳細が示されている．ノードは，人々が集合する，もしくはいる場所を示す．アークは，廊下や階段の吹き抜け等を示す．各ネットワークで各ノードそれぞれから避難させる人々の数は，ノードの下にイタリック体で示されている．期間は10秒間隔である．アークそれぞれの下にイタリック体で示される数字は，避難にかかる時間である． アークの上に示される数は，ある期間にアークに入れる人数制限である．ノードの上に示される数は，そのノードで待てる人数の上限である．

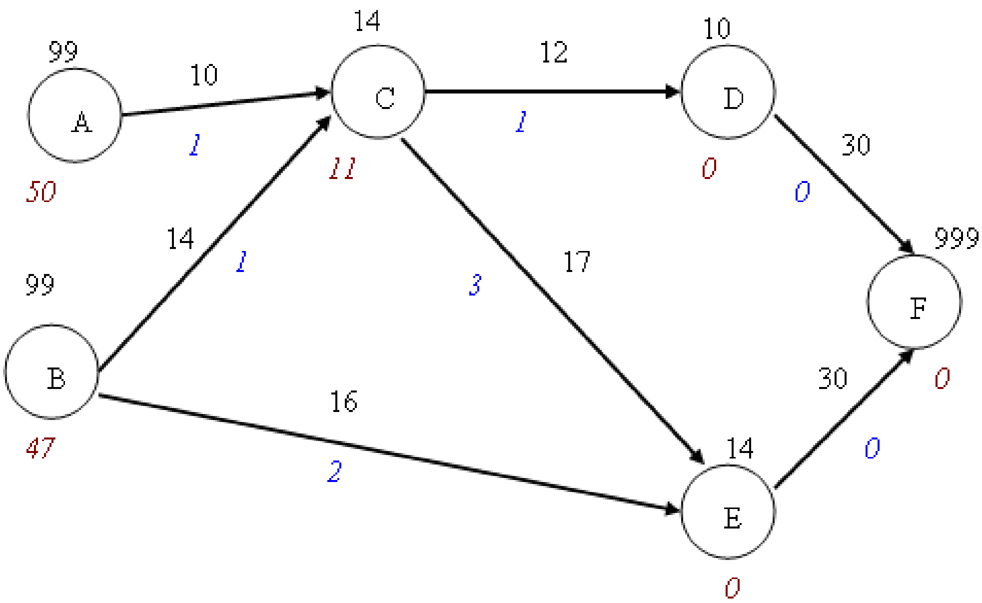


図 9.1　**避難路の構築**

ノードFは，外界を表す.例えば，最も速いノードAにいる50人が，安全な場所にたどり着ける最速期間は2期間で，A，C，D，Fを通る道である.しかし，ノードAにいる全ての人がここまで素早く避難できるわけではない．アークCからDは一度に12人しか通れないからである．また，アークBの人々も，アークCとDを使用しようとするかもしれない.アーク（C,E）は長いリードタイムがあるが，廊下が広いため許容人数が17人と多い．アーク（C,E）は，短めのリードタイムであるが，狭くて短い吹き抜け階段なため許容人数が少ない.

我々の目的は，何であるのか？ 明らかに分かっているものは，人々を避難させるのにかかる時間を最短にすることである.ここで面白い挑戦は，上記の問題で最大70秒で避難させることができるかどうかである.次の目的は，速く脱出できる人数を最大にすることである.X_i_j_tが期間tにiからjに移動している人数を示す.ここで,X_D_F_1+X_E_F_1をより大きく，X_D_F_9+ X_E_F_9をより小さくしたい.今回実際に使う目的関数は，以下の通りである.

```
Max = 10*(X_D_F_1+X_E_F_1)+9*(X_D_F_2 + X_E_F_2)
+8*(X_D_F_3+X_E_F_3)+7*(X_D_F_4 + X_E_F_4)+ etc.
```

すなわち，期間1に10人を外に出すという望ましい値を付け加え，期間2には9人という値を加えるという形で進める.このモデル **evacu8.lng** は，一般的な集合形式で書かれている.実際に使われている目的関数（上記参照）やData節の制約がどようになっているかを見たい場合，LINGO | Generate | Display modelをクリックしてみるとよい.

例えば，10期間あるとする.今回，これに対応する多期間ネットワークを書かないが，次のように考えることができる.とても幅の広い紙を10枚準備し，上記のネットワークと並べる.それから，アークそれぞれの矢印（i,j）をはずし，それをサブネットワークであるL（i,j）に再接続する.ネットワークの主な制約は，それぞれの期間における各ノードにあるフローバランス制約である.たとえば，期間5のノードEの制約は次のようになる.

```
[BAL_E_5] -X_B_E_3-X_C_E_2+X_E_F_5-V_E_4+V_E_5 = 0 ;
```

これは，以下と同等である.

```
V_E_5 = V_E_4 + X_B_E_3 + X_C_E_2 - X_E_F_5;
```

言葉で説明すると，期間5の終わりにアークEにいる人数は，期間4の終わりにいる人数と2期間前にノードBからEへ移動した人数，そしてノードCからEへ3期間前に移動した人数を足し，5期間前にノードEからFへ向かった人数を引いた数と等しくなる.

一般的に，evacU8.lngの集合形式だと制約は次のように書き表される.

```
! For every node k and time Period t;
@FOR(NXT(k,t)|t #GT# 1:
[BAL] V(k,t) = V(k,t-1)-@SUM(NXN(k,j):X(k,j,t))
```

+@SUM(NXN(I,k)|t-LT(I,k) #GT# 0: X(I,k,t-LT(I,k))); );

集合NXT( , )は，ノードkと期間tの組み合わせ全てであり，このネットワーク上に存在するkからjのアーク全ての集合を示す．このモデルは，各期間にそれぞれのノードで何人収容できるか，また各期間に各アークを移動する人数を制約することで完成する．例えば，期間4にアークBからEへの移動人数に上界を設定すると次のようになる．

[UFLO_B_E_4] X_B_E_4 <= 16 ;

期間6にノードDにいられる人数の上界は次のように設定する．

[USTOR_D_6] V_D_6 <= 10 ;

Evacu8.lngを解いてみると，実際に70秒で避難できることが分かる．例えば「非常口」などの配置を簡潔に分かり易くするためには，全ての人があるノードから同じルートを辿って避難するのが望ましいであろう．この解がこの追加的な管理上の制約を満たしているかどうかを確認したい．もう一つの動的ネットワークの例として，水力発電の川系システムが挙げられる．モデルは，**Dampoold.lng**というファイル名で保存されている．生産系の例は，**mrpcap.lng**を参照．

## 9.8 末端効果

ほとんどの多期間計画モデルの分析から，ある有限な時間を将来から切断してている．この切断方法が重要である．一般的にモデルの計画が終わった時に残された状態（在庫水準や資本投資）に注意する．もしモデルの計画を勝手に 5 年で終わらせてしまうとしたら，最適解は実質的に 5 年でどのようにビジネスを終わるかという最適解に変わる．Grinold (1983)は，end-of-horizon effect を緩和する色々な方法を分かりやすく説明している．下記に幾つかを紹介する．

①切り捨て：単にカットオフ・ポイントから先の期間を切り捨てる．

②Primal limit：最終期間の終わりに在庫水準というような適度な限界を設定する

③残存価額/双対価格：最終的な期間の終わりに在庫水準などに適切な残存価額を設定する．

④ 無限最終期間：モデルの最終期間を同じ決定が適用される無限期間とする．

最終期間が有限周期の初期と同等になるように，純現在価値割引が目的関数に使用される．これは，Cario 等(1994)が安田海上火災で用い，Peiser と Andrus がテキサスの不動産開発モデルにも使った．また Eppen, Martin,Schrage（1988）では GM のモデルに適用された．

### 9.8.1 腐敗性/貯蔵寿命制約

多くの製品，特に食品を取り扱う場合，ある程度の時間しか保存しておけないということを考えなければならない．例えば，輸血のための血液の場合，最大 21 日間しか保存できない．単一レベルの生産だと，これは簡単に表現できる．

$D_t$=期間 t における需要，$P_t$=期間 t における生産，$I_t$= 期間 t の最後にある在庫とすると，在庫バランス制約は次のようになる．

$$I_{t-1} + P_t = D_t + I_t$$

製品が廃棄される前に 1 期間持ち越せる場合，$I_t \le D_{t+1}$ という制約が加えられる．一般的に製品は最大 k 期間在庫で保存できるため，制約には $I_t \le D_{t+1} + D_{t+2} \cdots + D_{t+k}$ を追加する．

### 9.8.2 開始費用と停止費用

電力発電産業には始動停止問題として知られる意思決定問題がある．電力に対する需要は一日の流れの中でも変わる．電力会社は，需要が高まるにつれどの発電機を稼動させるか，また需要が低下した場合どの発電機を止めるかを決定しなければならない．一番の問題は，どれだけ稼動させておくかにかかわらず，発電機を稼動させるのに多大なる費用がかかるということである．より効率よく発電する発電機（例えば，石炭燃料）に限って，効率の低い発電機（ガス燃焼など）より稼動により費用がかかる．よって，短期間のみ電力の追加が必要な場合，ガス燃焼発電機を稼動させた方がより費用効率がよい．Eppen，Martin，Schrage(1988)も自動車工場の開始・停止でこれと類似した費用構造に遭遇した．典型的な開始費用の表現方法は，$y_{It}$=1 と $z_{It}$=1 の 2 つの変数を使う方法である．ユニット I が期間 t 稼動しているならば $y_{It}$ = 1 を，それ以外なら 0，ユニット I が期間 t に稼動し始めたならば $z_{It}$ = 1，それ以外なら 0 となる．

重要な制約はこうなる．

$$z_{It} \ge y_{It} - y_{It-1}.$$

よって，$y_{It}$ =1 であるが，$y_{It-1}$ = 0 の場合，$z_{It}$ は 1 となる．停止費用が必要な場合，これは別途計算するが似たような方法で計算できる．

## 9.9 非最適性

ある状況で，例えば上記のようにモデルの計画の終盤で，需要が周期的である（永久に毎週同じ周期を繰り返す）と結論するに十分である場合がある．ここで疑問として浮かび上がるのは，周期の期間が一定しているかどうかである．答えは「No」の場合もある．それは，需要が同じパターンを辿ったとしても，最大値をもたらす法則が，毎週毎週同じ周期を持つとは限らない．週ごとに違う動きがある方が望ましい場合もある．これを説明するために，第 8 章で紹介した飛行経路と割当問題を考えてみよう．元のデータに 2 種類の飛行機が各フライトでもたらす利益のデータを追加した．

利益寄与（$100）

| Flight | 出発地 | 目的地 | 出発時刻 | 到着時刻 | MD90 | B737 |
|---|---|---|---|---|---|---|
| F221 | ORD | DEN | 800 | 934 | 115 | 111 |
| F223 | ORD | DEN | 900 | 1039 | 109 | 128 |
| F274 | LAX | DEN | 800 | 1116 | 129 | 104 |
| F105 | ORD | LAX | 1100 | 1314 | 135 | 100 |
| F228 | DEN | ORD | 1100 | 1423 | 125 | 102 |
| F230 | DEN | ORD | 1200 | 1521 | 132 | 105 |
| F259 | ORD | LAX | 1400 | 1609 | 112 | 129 |
| F293 | DEN | LAX | 1400 | 1510 | 105 | 131 |
| F412 | LAX | ORD | 1400 | 1959 | 103 | 135 |
| F766 | LAX | DEN | 1600 | 1912 | 128 | 105 |
| F238 | DEN | ORD | 1800 | 2121 | 128 | 101 |

例えば，飛行パターン 221 の場合，MD90 機の方が B737 機よりも大幅に有益($11,500vs.$11,100)である．一方飛行パターン 259 だと，B737 の方が有益($12,900vs. $11,200)である．これらの飛行パターンは毎日行われる．MD90 機が 7 機と B737 機が 1 機あるとする．以前のように回送はないものとする．先ず 1 日周期の解を使うとする．第 8 章のモデルを適切に修正したものがこれである．

```
MODEL:
SETS:  ! Fleet routing and assignment (FLEETRAT);
 CITY :;   ! The cities involved;
 ACRFT:    ! Aircraft types;
  FCOST,   !  Fixed cost per day of this type;
  FSIZE;   !  Max fleet size of this type;
 FLIGHT:;
 FXCXC(FLIGHT,CITY,CITY) :
  DEPAT,   ! Flight departure time;
  ARVAT;   ! arrival time at dest.;
 AXC(ACRFT,CITY):
  OVNITE; ! Number staying overnight by type, city;
 AXF(ACRFT,FXCXC):
  X,       ! Number aircraft used by type, flight;
  PC;      ! Profit contribution by type, flight;
ENDSETS
DATA:
 CITY = ORD  DEN  LAX;
 ACRFT, FCOST, FSIZE =
  MD90   .01    7
```

```
  B737   .01    1;
FLIGHT=F221 F223 F274 F105 F228 F230 F259 F293 F412 F766
F238;
 FXCXC, DEPAT, ARVAT =
!      Flight  Origin Dest. Depart Arrive;
           F221   ORD   DEN    800    934
           F223   ORD   DEN    900   1039
           F274   LAX   DEN    800   1116
           F105   ORD   LAX   1100   1314
           F228   DEN   ORD   1100   1423
           F230   DEN   ORD   1200   1521
           F259   ORD   LAX   1400   1609
           F293   DEN   LAX   1400   1510
           F412   LAX   ORD   1400   1959
           F766   LAX   DEN   1600   1912
           F238   DEN   ORD   1800   2121;
 PC = ! Profit contribution of each vehicle*flight combo;
115 109 129 135 125 132 112 105 103 128 128
111 128 104 100 102 105 129 131 135 105 101;
ENDDATA
!----------------------------;
! Maximize profit contribution from flights minus
   overhead cost of aircraft in fleet;
 MAX = @SUM( AXF( I, N, J, K): PC( I, N, J, K) * X( I, N, J,
K))- @SUM( AXC( I, J): FCOST( I) * OVNITE( I, J));
!At any instant, departures in particular, the number of
cumulative  arrivals  must  be>=number  of  cumulative
departures;
! For each flight of each aircraft type;
 @FOR( ACRFT( I):@FOR( FXCXC( N, J, K):
! Aircraft on ground in morning +
   number aircraft arrived thus far >=
   number aircraft departed thus far;
OVNITE(I,J)+@SUM(FXCXC(N1,J1,K1)|K1#EQ#J#AND#
ARVAT(N1,J1,K1)#LT#DEPAT(N,J,K):
X(I,N1,J1,J))>=@SUM(FXCXC(N1,J1,K1)|J1#EQ#J#AND#
DEPAT(N1,J1,K1)#LE#DEPAT(N,J,K):
```

```
X(I,N1,J,K1));););
!This model does not allow deadheading,so at the end of the day,arrivals must equal departures;
 @FOR(ACRFT(I):
@FOR(CITY(J):
@SUM(AXF(I,N,J1,J):X(I,N,J1,J))=
@SUM(AXF(I,N,J,K):X(I,N,J,K)););););
!  Each flight must be covered;
   @FOR(FXCXC(N,J,K):
@SUM(AXF(I,N,J,K):X(I,N,J,K))=1;););
! Fleet size limits;
  @FOR(ACRFT(I):
@SUM(AXC(I,J):OVNITE(I,J))<=FSIZE(I););
! Fractional planes are not allowed;
   @FOR( AXF: @GIN( X); );
END
```

訳注：モデルは，Generateで次の通りである．

```
MODEL:
[_1]MAX=-0.01*OVNITE_MD90_ORD-0.01*OVNITE_MD90_DEN-0.01*OVNITE_MD90_LAX-0.01*OVNITE_B737_ORD-0.01*OVNITE_B737_DEN-0.01*OVNITE_B737_LAX+115*X_MD90_F221_ORD_DEN+109*X_MD90_F223_ORD_DEN+129*X_MD90_F274_LAX_DEN+135*X_MD90_F105_ORD_LAX+125*X_MD90_F228_DEN_ORD+132*X_MD90_F230_DEN_ORD+112*X_MD90_F259_ORD_LAX+105*X_MD90_F293_DEN_LAX+103*X_MD90_F412_LAX_ORD+128*X_MD90_F766_LAX_DEN+128*X_MD90_F238_DEN_ORD+111*X_B737_F221_ORD_DEN+128*X_B737_F223_ORD_DEN+104*X_B737_F274_LAX_DEN+100*X_B737_F105_ORD_LAX+102*X_B737_F228_DEN_ORD+105*X_B737_F230_DEN_ORD+129*X_B737_F259_ORD_LAX+131*X_B737_F293_DEN_LAX+135*X_B737_F412_LAX_ORD+105*X_B737_F766_LAX_DEN+101*X_B737_F238_DEN_ORD;
[_2]OVNITE_MD90_ORD-X_MD90_F221_ORD_DEN>=0;
[_3]OVNITE_MD90_ORD-X_MD90_F221_ORD_DEN-X_MD90_F223_ORD_DEN>=0;
[_4]OVNITE_MD90_LAX-X_MD90_F274_LAX_DEN>=0;
[_5]OVNITE_MD90_ORD-X_MD90_F221_ORD_DEN-X_MD90_F223_ORD_DEN-
```

X_MD90_F105_ORD_LAX>=0;
[_6]OVNITE_MD90_DEN+X_MD90_F221_ORD_DEN+X_MD90_F223_ORD_DEN-
X_MD90_F228_DEN_ORD>=0;
[_7]OVNITE_MD90_DEN+X_MD90_F221_ORD_DEN+X_MD90_F223_ORD_DEN+X
_MD90_F274_LAX_DEN-X_MD90_F228_DEN_ORD-
X_MD90_F230_DEN_ORD>=0;
[_8]OVNITE_MD90_ORD-X_MD90_F221_ORD_DEN-X_MD90_F223_ORD_DEN-
X_MD90_F105_ORD_LAX-X_MD90_F259_ORD_LAX>=0;
[_9]OVNITE_MD90_DEN+X_MD90_F221_ORD_DEN+X_MD90_F223_ORD_DEN+X
_MD90_F274_LAX_DEN-X_MD90_F228_DEN_ORD-X_MD90_F230_DEN_ORD-
X_MD90_F293_DEN_LAX>=0;
[_10]OVNITE_MD90_LAX-X_MD90_F274_LAX_DEN+X_MD90_F105_ORD_LAX-
X_MD90_F412_LAX_ORD>=0;
[_11]OVNITE_MD90_LAX-
X_MD90_F274_LAX_DEN+X_MD90_F105_ORD_LAX+X_MD90_F293_DEN_LAX-
X_MD90_F412_LAX_ORD-X_MD90_F766_LAX_DEN>=0;
[_12]OVNITE_MD90_DEN+X_MD90_F221_ORD_DEN+X_MD90_F223_ORD_DEN+
X_MD90_F274_LAX_DEN-X_MD90_F228_DEN_ORD-X_MD90_F230_DEN_ORD-
X_MD90_F293_DEN_LAX-X_MD90_F238_DEN_ORD>=0;
[_13]OVNITE_B737_ORD-X_B737_F221_ORD_DEN>=0;
[_14]OVNITE_B737_ORD-X_B737_F221_ORD_DEN-
X_B737_F223_ORD_DEN>=0;
[_15]OVNITE_B737_LAX-X_B737_F274_LAX_DEN>=0;
[_16]OVNITE_B737_ORD-X_B737_F221_ORD_DEN-X_B737_F223_ORD_DEN-
X_B737_F105_ORD_LAX>=0;
[_17]OVNITE_B737_DEN+X_B737_F221_ORD_DEN+X_B737_F223_ORD_DEN-
X_B737_F228_DEN_ORD>=0;
[_18]OVNITE_B737_DEN+X_B737_F221_ORD_DEN+X_B737_F223_ORD_DEN+
X_B737_F274_LAX_DEN-X_B737_F228_DEN_ORD-
X_B737_F230_DEN_ORD>=0;
[_19]OVNITE_B737_ORD-X_B737_F221_ORD_DEN-X_B737_F223_ORD_DEN-
X_B737_F105_ORD_LAX-X_B737_F259_ORD_LAX>=0;
[_20]OVNITE_B737_DEN+X_B737_F221_ORD_DEN+X_B737_F223_ORD_DEN+
X_B737_F274_LAX_DEN-X_B737_F228_DEN_ORD-X_B737_F230_DEN_ORD-
X_B737_F293_DEN_LAX>=0;
[_21]OVNITE_B737_LAX-X_B737_F274_LAX_DEN+X_B737_F105_ORD_LAX-
X_B737_F412_LAX_ORD>=0;

[_22]OVNITE_B737_LAX-X_B737_F274_LAX_DEN+X_B737_F105_ORD_LAX
+X_B737_F293_DEN_LAX-
X_B737_F412_LAX_ORDX_B737_F766_LAX_DEN>=0;
[_23]OVNITE_B737_DEN+X_B737_F221_ORD_DEN+X_B737_F223_ORD_DEN+
X_B737_F274_LAX_DEN-X_B737_F228_DEN_ORD-X_B737_F230_DEN_ORD-
X_B737_F293_DEN_LAX-X_B737_F238_DEN_ORD>=0;
[_24]-X_MD90_F221_ORD_DEN-X_MD90_F223_ORD_DEN-
X_MD90_F105_ORD_LAX+X_MD90_F228_DEN_ORD+X_MD90_F230_DEN_ORD-
X_MD90_F259_ORD_LAX+
X_MD90_F412_LAX_ORD+X_MD90_F238_DEN_ORD=0;
[_25]X_MD90_F221_ORD_DEN+X_MD90_F223_ORD_DEN+X_MD90_F274_LAX_
DEN-X_MD90_F228_DEN_ORD-X_MD90_F230_DEN_ORD-
X_MD90_F293_DEN_LAX+
X_MD90_F766_LAX_DEN-X_MD90_F238_DEN_ORD=0;
[_26]-X_MD90_F274_LAX_DEN+X_MD90_F105_ORD_LAX+
X_MD90_F259_ORD_LAX+X_MD90_F293_DEN_LAX-X_MD90_F412_LAX_ORD-
X_MD90_F766_LAX_DEN=0;
[_27]-X_B737_F221_ORD_DEN-X_B737_F223_ORD_DEN-
X_B737_F105_ORD_LAX+
X_B737_F228_DEN_ORD+X_B737_F230_DEN_ORD-X_B737_F259_ORD_LAX+
X_B737_F412_LAX_ORD+X_B737_F238_DEN_ORD=0;
[_28]X_B737_F221_ORD_DEN+X_B737_F223_ORD_DEN+X_B737_F274_LAX_
DEN-X_B737_F228_DEN_ORD-X_B737_F230_DEN_ORD-
X_B737_F293_DEN_LAX+
X_B737_F766_LAX_DEN-X_B737_F238_DEN_ORD=0;
[_29]-X_B737_F274_LAX_DEN+X_B737_F105_ORD_LAX+
X_B737_F259_ORD_LAX+X_B737_F293_DEN_LAX-X_B737_F412_LAX_ORD-
X_B737_F766_LAX_DEN=0;
[_30]X_MD90_F221_ORD_DEN+X_B737_F221_ORD_DEN=1;
[_31]X_MD90_F223_ORD_DEN+X_B737_F223_ORD_DEN=1;
[_32]X_MD90_F274_LAX_DEN+X_B737_F274_LAX_DEN=1;
[_33]X_MD90_F105_ORD_LAX+X_B737_F105_ORD_LAX=1;
[_34]X_MD90_F228_DEN_ORD+X_B737_F228_DEN_ORD=1;
[_35]X_MD90_F230_DEN_ORD+X_B737_F230_DEN_ORD=1;
[_36]X_MD90_F259_ORD_LAX+X_B737_F259_ORD_LAX=1;
[_37]X_MD90_F293_DEN_LAX+X_B737_F293_DEN_LAX=1;
[_38]X_MD90_F412_LAX_ORD+X_B737_F412_LAX_ORD=1;

```
[_39]X_MD90_F766_LAX_DEN+X_B737_F766_LAX_DEN=1;
[_40]X_MD90_F238_DEN_ORD+X_B737_F238_DEN_ORD=1;
[_41]OVNITE_MD90_ORD+OVNITE_MD90_DEN+OVNITE_MD90_LAX<=7;
[_42]OVNITE_B737_ORD+OVNITE_B737_DEN+OVNITE_B737_LAX<=1;
@GIN(X_MD90_F221_ORD_DEN);@GIN(X_MD90_F223_ORD_DEN);
@GIN(X_MD90_F274_LAX_DEN);@GIN(X_MD90_F105_ORD_LAX);
@GIN(X_MD90_F228_DEN_ORD);@GIN(X_MD90_F230_DEN_ORD);
@GIN(X_MD90_F259_ORD_LAX);@GIN(X_MD90_F293_DEN_LAX);
@GIN(X_MD90_F412_LAX_ORD);@GIN(X_MD90_F766_LAX_DEN);
@GIN(X_MD90_F238_DEN_ORD);@GIN(X_B737_F221_ORD_DEN);
@GIN(X_B737_F223_ORD_DEN);@GIN(X_B737_F274_LAX_DEN);
@GIN(X_B737_F105_ORD_LAX);@GIN(X_B737_F228_DEN_ORD);
@GIN(X_B737_F230_DEN_ORD);@GIN(X_B737_F259_ORD_LAX);
@GIN(X_B737_F293_DEN_LAX);@GIN(X_B737_F412_LAX_ORD);
@GIN(X_B737_F766_LAX_DEN);@GIN(X_B737_F238_DEN_ORD);
END
```

解は次の通りである：

```
Global optimal solution found at step: 106
Objective value:                 1323.940
                        Variable           Value
     X( MD90, F221, ORD, DEN)         1.000000
     X( MD90, F223, ORD, DEN)         1.000000
     X( MD90, F274, LAX, DEN)         1.000000
     X( MD90, F105, ORD, LAX)         1.000000
     X( MD90, F228, DEN, ORD)         1.000000
     X( MD90, F230, DEN, ORD)         1.000000
     X( MD90, F259, ORD, LAX)         1.000000
     X( MD90, F412, LAX, ORD)         1.000000
     X( MD90, F238, DEN, ORD)         1.000000
     X( B737, F293, DEN, LAX)         1.000000
     X( B737, F766, LAX, DEN)         1.000000
```

この解の毎日の利益貢献度は 1323.94*100=132,394 である.1 機しかない B73 がフライト 293 として DEN から 2 時に LAX へと発つ.また,フライト 766 としても LAX を 4 時に出発し,DEN へ向かっていることに注目すべきである.このモデルでは,一日おきに指定された空港に最初から MD90 機が 7 機および B737 機が 1 機揃っていることが条件となる.参照のため,B737 機を抜いてモデルを解析してみると,利益貢献が$132,094 となった.B737 機は一日に$200 しか利益に貢献して

いないことが分かる． 解を 2 日周期にしたらより良くなるだろうか．モデルへの入力を次のモデルのように変更してみる．2 日分の需要を入力する．2 日目のフライトは S で，初日のフライとは F で記す．そうしなければ 2 つのモデルは同一のものとなってしまう． 2 日周期の解は少なくても 2*132,394=$264,788 であるべきである．

```
MODEL:
SETS:  ! Fleet routing and assignment (FLEETRAT);
 CITY :;  ! The cities involved;
 ACRFT:   ! Aircraft types;
  FCOST,  !  Fixed cost per day of this type;
  FSIZE;  !  Max fleet size of this type;
 FLIGHT:;
 FXCXC( FLIGHT, CITY, CITY) :
  DEPAT,  ! Flight departure time;
  ARVAT;  ! arrival time at dest.;
 AXC( ACRFT, CITY):
  OVNITE; ! Number staying overnight by type, city;
 AXF( ACRFT, FXCXC):
  X,      ! Number aircraft used by type, flight;
  PC;     ! Profit contribution by type, flight;
ENDSETS
DATA:
 CITY = ORD  DEN  LAX;
 ACRFT, FCOST, FSIZE =
  MD90   .01    7
  B737   .01    1;
 FLIGHT = F221 F223 F274 F105 F228 F230 F259 F293 F412 F766
F238 S221 S223 S274 S105 S228 S230 S259 S293 S412 S766 S238;
 FXCXC, DEPAT, ARVAT =
!      Flight  Origin Dest. Depart Arrive;
         F221   ORD   DEN    800    934
         F223   ORD   DEN    900   1039
         F274   LAX   DEN    800   1116
         F105   ORD   LAX   1100   1314
         F228   DEN   ORD   1100   1423
         F230   DEN   ORD   1200   1521
         F259   ORD   LAX   1400   1609
```

```
        F293    DEN    LAX    1400    1510
        F412    LAX    ORD    1400    1959
        F766    LAX    DEN    1600    1912
        F238    DEN    ORD    1800    2121
        S221    ORD    DEN    3200    3334
        S223    ORD    DEN    3300    3439
        S274    LAX    DEN    3200    3516
        S105    ORD    LAX    3500    3714
        S228    DEN    ORD    3500    3823
        S230    DEN    ORD    3600    3921
        S259    ORD    LAX    3800    4009
        S293    DEN    LAX    3800    3910
        S412    LAX    ORD    3800    4359
        S766    LAX    DEN    4000    4312
        S238    DEN    ORD    4000    4521;
 PC = ! Profit contribution of each vehicle*flight combo;
115 109 129 135 125 132 112 105 103 128 128
115 109 129 135 125 132 112 105 103 128 128
111 128 104 100 102 105 129 131 135 105 101
111 128 104 100 102 105 129 131 135 105 101;
ENDDATA
```

解は次のとおりである.

```
Global optimal solution found at step:          103
Objective value:                           2718.930
                     Variable              Value
    X( MD90, F221, ORD, DEN)           1.000000
    X( MD90, F223, ORD, DEN)           1.000000
    X( MD90, F274, LAX, DEN)           1.000000
    X( MD90, F105, ORD, LAX)           1.000000
    X( MD90, F228, DEN, ORD)           1.000000
    X( MD90, F230, DEN, ORD)           1.000000
    X( MD90, F259, ORD, LAX)           1.000000
    X( MD90, F293, DEN, LAX)           1.000000
    X( MD90, F766, LAX, DEN)           1.000000
    X( MD90, F238, DEN, ORD)           1.000000
    X( MD90, S221, ORD, DEN)           1.000000
    X( MD90, S274, LAX, DEN)           1.000000
```

X( MD90, S105, ORD, LAX)　　　1.000000
X( MD90, S228, DEN, ORD)　　　1.000000
X( MD90, S230, DEN, ORD)　　　1.000000
X( MD90, S259, ORD, LAX)　　　1.000000
X( MD90, S412, LAX, ORD)　　　1.000000
X( MD90, S766, LAX, DEN)　　　1.000000
X( MD90, S238, DEN, ORD)　　　1.000000
X( B737, F412, LAX, ORD)　　　1.000000
X( B737, S223, ORD, DEN)　　　1.000000
X( B737, S293, DEN, LAX)　　　1.000000

利益は2718.93*100=$271,893となり,一日周期の解2*132,394=$264,788より多くなった.どのようにして結果が良くなったのであろうか. B737がどのように使われたかに注目してみよう.初日には, LAXからORDへフライト412で飛行した.これは2日目には,ORDからDENへフライト223で, それからDENからLAXへフライと293で戻る.2日目にならないとLAXには戻らない.この3つのフライトはMD90に対し,B737機にとっても有益である.2日周期にすることで, B737機は3つのとても有益なフライトを最低半分の時間飛べるようになった.よって需要パターンが1日であっても, 解を2日周期にすることは有益なこともある.

どのように解を制限してしまうことを防ぐかの良い考察はAdams (1986)の「conceptual blockbusting」に示されている. Orlin (1982)には, cyclic vehicle routing問題のより詳しい分析が載っている.

# 第 10 章　配合問題

## 10.1　はじめに

配合問題は次の特徴をもっている．

①2 つ以上の入力される原材料がある．

②使用原材料は 1 つ以上の特性値をもっている．

③使用原材料を配合して作られる 1 つ以上の最終製品がある．ただし，最終製品は要求された特性値を満足しなければならない．

最終製品の特性は，配合された原材料に含まれる特性の加重平均になる．

例えば，次の通りである．

| **最終製品** | **特性** | **原材料** |
|---|---|---|
| 食料・飼料 | 蛋白，炭水化物，脂肪含有量 | コーン，ダイズ，小麦 |
| ガソリン | オクタン価，揮発性，蒸気圧 | 原油 |
| 金属 | カーボン，クローム，マンガン含有量 | 原鉱石，スクラップ |
| 輸出穀物 | 水分，異物，ダメージ度合 | 種々の供給業者からの穀物 |
| 石炭 | 硫黄，灰分，水分 | 種々の生産地からの石灰 |
| ワイン | ビンテージ，種類，地域 | 種々の地域の原ワイン |
| 銀行のバランシート | 種々のタイプのローンの比率，投資対象 | 利用できるローンや投資の種類 |

配合モデルは主として次の産業で用いられる．

①食料・飼料の加工産業：家畜の飼料，ホットドック等

②金属工業：原材料やリサイクルやスクラップを用いた配合

③石油産業：一定成分をもつガソリンの配合

典型的な原材料の市場価格は，1 か月や 1 週間単位で変動している．賢い買手は，たとえばコーンを一番安い供給者から買うだろう．もっと賢い買手は，ダイズの価格がコーンより安いなら，ダイズをもっと使用する配合政策をとる．

単純配合問題は，単純な製品混合問題と同じである．この 2 つの違いは，製品混合問題では幾つかの製品の生産台数を決めるのに対し，単純配合問題では幾つかの原料を配合して 1 つの製品の作り方を決める．この問題は，食料，肥料，化学工業で用いられる．肉加工工場では，ソーセージの原料の配合を決めるために LP を使っている．様々な鋳造工場では，溶鉱炉に仕込む原料の配合の最低費用を見つけるため配合問題が使われている．

Fiels & Mcgee (1978) は，飼育場の家畜のための低費用な飼料配合の LP 問題を紹介している．このモデルは，日に 1000 回以上用いられている．

最初の一般的な LP に関する出版は，George Stigler (1945) の配合問題に関する定式化であった．この問題は，80 の食料から 12 の栄養に関する条件を満足させるダイエット食の製法を決めることである．スティグラーがこの問題を定式化

したときには，LP を解くためのシンプレックス法(単体法)が存在していなかった．また，この問題が LP の特別なケースであるとは，一般には認識されていなかった．この普遍性に気付いたスティグラーは，「線形条件に従った線形関数を最小化する直接の方法は見つからない」と述べた．彼が巧妙な議論で得た解は，後の単体法が発明されてから得られた解と数セントの差しかなかった．LP 解もスティグラーの解も，両方とも食べてうまくはない．両方とも主にキャベツ，小麦粉，乾燥インゲンマメと色どりのためにほうれん草を少し含んでいる．この食料で生きたいと思う人がいるかどうか，あるいはこれだけで生きて行けるかどうかさえ明かではない．これらの解は，美味しさの制約を考えていない．

### 10.2 配合問題の構造

簡単な食料配合問題を考えてみよう．我々はたんぱく質を 15％以上含んだ家畜の飼料を作らなくてはいけない．この食料はとうもろこし（タンパク質 6％）と大豆粉（タンパク質 35％）を配合して作る．C が配合物中のコーンのブッシェル数，S が大豆粉のブッシェル数とすれば，制約式は次のように非線形で表される．

$$\frac{0.06\ C + 0.35\ S}{C + S} \geq 0.15$$

両辺に分母をかけ，整理すると次の線形式になる．

$$0.06C+0.35S \geqq 0.15\ (C+S) \qquad \text{（たんぱく質含有量）}$$

または，$-0.09\ C + 0.20\ S \geq 0$.

例えば，脂肪，炭水化物，色，味，手触りなどの非線形な制約も，似たような方法で扱うことができる．

もし配合がバッチサイズを決定変数としたより大きな問題の一部分である場合，定式化はほんの少し複雑になる．この章の 2 番目の例はこの複雑さについての考察である．最初の例では，バッチサイズをあらかじめ指定した状況を考える．

#### 10.2.1 例：ピッツバーグ製鉄会社の配合問題

ピッツバーグ製鉄会社（PS）は，次の厳しい品質要求のある高炭素鋼の新製品を作る契約をした：

| | 以上 | 以下 |
|---|---|---|
| 炭素含有量 | 3.00％ | 3.50％ |
| クロム含有量 | 0.30％ | 0.45％ |
| マンガン含有量 | 1.35％ | 1.65％ |
| シリコン含有量 | 2.70％ | 3.00％ |

PS 社は，次の資材をバッチで配合して使用できる．

| | 費用／ﾎﾟﾝﾄﾞ | 炭素 ％ | クロム ％ | マンガン ％ | シリコン ％ | 使用可能な量 |
|---|---|---|---|---|---|---|
| 銑鉄 1 | 0.03 | 4 | 0 | 0.9 | 2.25 | 制限なし |
| 銑鉄 2 | 0.0645 | 0 | 10 | 4.5 | 15 | 制限なし |
| シリコン 1 | 0.065 | 0 | 0 | 0 | 45 | 制限なし |

| | | | | | | |
|---|---|---|---|---|---|---|
| シリコン 2 | 0.061 | 0 | 0 | 0 | 42 | 制限なし |
| 合金 1 | 0.1 | 0 | 0 | 60 | 18 | 制限なし |
| 合金 2 | 0.13 | 0 | 20 | 9 | 30 | 制限なし |
| 合金 3 | 0.119 | 0 | 8 | 33 | 25 | 制限なし |
| カーバイド | 0.08 | 15 | 0 | 0 | 30 | 20 lb. |
| 鉄鋼 1 | 0.021 | 0.4 | 0 | 0.9 | 0 | 200 lb. |
| 鉄鋼 2 | 0.02 | 0.1 | 0 | 0.3 | 0 | 200 lb. |
| 鉄鋼 3 | 0.0195 | 0.1 | 0 | 0.3 | 0 | 200 lb. |

この品質要求を満たす 1 トン（2000 ポンド）のバッチを配合したい．問題は，11 の原料を配合し，品質要求を満たして，費用を最小にすることである．ベテランの鉄鋼マンは，最少の費用の組合せは 9 個以上の原料を使わないことを問題視している．よいブレンドとは何であろうか？11 の価格および 4 つの品質要求の改善は，交渉可能である．どの価格および品質要求を交渉するのが良いだろうか．

配合品の化学成分は，単にある原料の化学成分の重み平均である．従って 40%の合金 1 と 60%の合金 2 を配合すると，例えばマンガンの化学成分は，0.4×60+0.60×9=29.4 になる．

### 10.2.2 PS 社の配合問題の定式化と解

PS 社配合問題は，11 変数で 13 制約式を持つ LP で定式化できる．11 の変数は，我々の選べる 11 の原料に相当する．制約式のうちの 4 つは，品質の下限で，他の 4 つは上限である．4 つの制約式は，シリコンカーバイドと鉄鋼の使用制限に関するものである．13 番目の制約式は，全ての原料の使用の合計の重さが 2000 ポンドを示す．

P1 を銑鉄の使用量とし，他の原料も同様とすると，1 ポンドあたりの費用の最小化問題は次のように示される．

```
MIN=0.03*P1+0.0645*P2+0.065*F1+0.061*F2+0.1*A1+0.13*A2+0.119
*A3+0.08*CB+0.021*S1+0.02*S2+0.0195*S3;
! Raw materIal avaIlabIlItIes;
CB <= 20;
S1 <= 200;
S2 <= 200;
S3 <= 200;
! 品質要求;
! カーボン;
.04*P1+0.15*CB+0.004*S1+0.001*S2+0.001*S3>=60;
.04*P1+0.15*CB+0.004*S1+0.001*S2+0.001*S3<=70;
!クローム;
0.1*P2+0.2*A2+0.08*A3>=6;
```

```
0.1*P2+0.2*A2+0.08*A3<=9;
!マンガン;
0.009*P1+0.045*P2+0.6*A1+0.09*A2+0.33*A3+0.009*S1+0.003*S2+0
.003*S3>=27;
0.009*P1+0.045*P2+0.6*A1+0.09*A2+0.33*A3+0.009*S1+0.003*S2+0
.003*S3<=33;
!シリコン;
0.0225*P1+0.15*P2+0.45*F1+0.42*F2+0.18*A1+0.3*A2+0.25*A3+0.3
*CB>=54;
0.0225*P1+0.15*P2+0.45*F1+0.42*F2+0.18*A1+0.3*A2+0.25*A3+0.3
*CB<=60;
!最終製品要求;
P1+P2+F1+F2+A1+A2+A3+CB+S1+S2+S3=2000;
```

このモデルを言葉で表すと次のようになる．

原料費を次の条件のもと最小にしなさい

①原料供給の範囲内(列 2-5)

②品質要求を満たす(列 6-13)

③最終製品の要求(列 14)

これを解くと，次の解を得る．

```
Optimal solution found at step:        11
Objective value:                 59.55629
Variable           Value        Reduced Cost
      P1        1474.264           0.0000000
      P2        60.00000           0.0000000
      F1       0.0000000          0.1035937E-02
      F2        22.06205           0.0000000
      A1        14.23886           0.0000000
      A2       0.0000000          0.2050311E-01
      A3       0.0000000          0.1992597E-01
      CB       0.0000000          0.3356920E-02
      S1        200.0000           0.0000000
      S2        29.43496           0.0000000
      S3        200.0000           0.0000000
     Row    Slack or Surplus      Dual Price
       1        59.55629            1.000000
       2        20.00000           0.0000000
       3       0.0000000          0.1771118E-03
```

| | | |
|---|---|---|
| 4 | 170.5650 | 0.0000000 |
| 5 | 0.0000000 | 0.5000000E-03 |
| 6 | 0.0000000 | -0.1833289 |
| 7 | 10.00000 | 0.0000000 |
| 8 | 0.0000000 | -0.2547314 |
| 9 | 3.000000 | 0.0000000 |
| 10 | 0.0000000 | -0.1045208 |
| 11 | 6.000000 | 0.0000000 |
| 12 | 0.0000000 | -0.9880212E-01 |
| 13 | 6.000000 | 0.0000000 |
| 14 | 0.0000000 | -0.1950311E-01 |

11 の原料の内，わずか 7 つしか使われていないことに注目して欲しい．

実際にはこのタイプの LP は，月に 2 回解かれていた．1 回目は減少費用や双対価格を含めて，原料を購入する人が購入のガイドとして使った．第 2 回目は，主に治金家が原料を配合するときに使われた．

溶鉱炉に酸素を送りこむことを考える．この酸素は炭素と完全に結びついて $CO_2$ となって出て行く．酸素 32 ポンドに対し炭素 12 ポンドの割合で燃える．酸素は 1 ポンドにつき 2 セントである．この付加的な選択の加わった問題を再び定式化してみると決定が変わるだろうか．元の解はカーボンの上限値より下限値に制約されているので，カーボンを燃やすための酸素注入の選択はあまり意味のないことは明白だ．カーボンを燃やすことは，たとえそれ自体は費用がかからなくても，解の全費用を増加させることになる．

## 10.3　製品混合問題における配合問題

配合問題の定式化の付加的な側面として，バッチサイズを決定変数とした例がある．前の例では，バッチサイズは明記されていた．次の例では，いかに安く製品を組み合わせするかによって，製品の量が決まる．このように製品混合の決定とバッチサイズの決定が同時に行われる．

この例は製油所におけるガソリン配合問題にも示唆を与える．ブタン，重ナフサ，触媒による改質ガソリンの 3 つの原料からガソリンを配合したい．合成されるガソリンの 4 つの特質とそのデータが重要である．費用，オクタン化，蒸気圧，揮発性の特徴は，下記の表にまとめられている．REG と PRM の費用が負なのは，販売（収入）することを意味する．

オクタン価は，ノッキングやピンギングに対するガソリンの抵抗の尺度である．蒸気圧と揮発性は関連が深い．蒸気圧は特に異常に暖かい春の日に，エンストのしやすさの尺度である．揮発性は，寒い日にエンジンがどれくらいスタートしやすいかの尺度である．

| 特質 | ブタン (BUT) | 改質ガソリン (CAT) | 重ナフサ (NAP) | レギュラーガソリン (REG) | 高級ガソリン(PRM) |
|---|---|---|---|---|---|
| 費用/単位 | 7.3 | 18.2 | 12.5 | −18.4 | −22 |
| オクタン価 | 120 | 100 | 74 | 89 ≦ oct ≦110 | 94 ≦ oct ≦ 110 |
| 蒸気圧 | 60 | 2.6 | 4.1 | 3 ≦ vp ≦ 11 | 3 ≦ vp ≦ 11 |
| 揮発性 | 105 | 3 | 12 | 17 ≦ vl ≦ 25 | 17 ≦ vl ≦ 25 |
| 利用可能量 | 1000 | 4000 | 5000 | | |

この計画では，わずか 1000 単位のブタンが利用可能である.レギュラーガソリンの利益は，原料費用を除いて 18.40 ドル/単位である.この例を少し単純化して，原料の相互関係は線形とする.例えば BUT と CAT を 50 対 50 で配合したとすればオクタンは，0.5*120+0.5*100=110，揮発性は 0.5*105+0.5*3=54 になる.実際には，この線形性はオクタン価で少しくずれる.

### 10.3.1 定式化

品質制約は，少し考慮が必要になる.ブタン，触媒作用の改質油，重ナフサと REG ガソリンのバッチの比率は，BUT/REG，CAT/REG，NAP/REG である.例えば，線形の神が我々に微笑むならば，REG のオクタン価の制約は，次のようになる：

$$(BUT/REG)\times 120+(CAT/REG)\times 100+(NAP/REG)\times 74 \geq 89.$$

BUT/REG のような比率は，線形でない.しかし，REG を両辺に掛けることで線形制約になる：

$$120\ BUT + 100\ CAT + 74\ NAP \geq 89\ REG$$

あるいは，次の標準形式になる.

$$120\ BUT + 100\ CAT + 74\ NAP - 89\ REG \geq 0.$$

### 10.3.2 上下限制約式の表現

全ての品質制約は，上下限の制約を持つ.オクタン価の上限制約は，明らかに次のようになる：$120BUT+100CAT+74NAP-110REG \leq 0$

スラック変数を加えることで，等式になる：

$$120BUT+100CAT+74NAP-110REG+SOCT=0.$$

SOCT=0 なら，上限に拘束される.$SOCT=110REG - 89REG = 21REG$ のとき，下限に拘束される.よって，上下限制約のコンパクトな次の制約式を得る.

1) $120BUT+100CAT+74NAP-110REG+SOCT=0.$

2) $SOCT \leq 21REG.$

たとえ多くの成分があるかもしれないが，第 2 の制約は 2 つの変数だけを含む.これは，上下限制約をコンパクトに表現するうまい方法である.同様な議論は，蒸

気圧と揮発性制約に用いることができる．最後に，全体が原料の合計に等しい制約を，レギュラーガソリンと高級ガソリン別に追加する．

全ての制約式を標準形と利益貢献度で表現にすると，次のモデルを得る：

```
ＭＯＤＥＬ：
ＭＡＸ＝２２＊Ｂ＿ＰＲＭ＋１８．４＊Ｂ＿ＲＥＧ－７．３＊ＸＢＵＴ＿ＰＲＭ－７．３＊ＸＢＵＴ＿ＲＥＧ－
１２．５＊ＸＮＡＰ＿ＰＲＭ－１２．５＊ＸＮＡＰ＿ＲＥＧ－１８．２＊ＸＣＡＴ＿ＰＲＭ－１８．２＊ＸＣＡＴ＿ＲＥＧ；
！ＳＵｂｊｅｃｔｔｏｒａｗｍａｔｅｒＩａｌａｖａＩｌａｂＩｌＩｔＩｅｓ；
［ＲＭＬＩＭＢＵＴ］ＸＢＵＴ＿ＰＲＭ＋ＸＢＵＴ＿ＲＥＧ＜＝１０００；
［ＲＭＬＩＭＣＡＴ］ＸＣＡＴ＿ＰＲＭ＋ＸＣＡＴ＿ＲＥＧ＜＝４０００；
［ＲＭＬＩＭＮＡＰ］ＸＮＡＰ＿ＰＲＭ＋ＸＮＡＰ＿ＲＥＧ＜＝５０００；
！ＦｏｒｅａｃｈｆＩｎＩｓｈｅＤｇｏｏＤ，ｂａｔｃｈｓＩｚｅｃｏｍＰＵｔａｔＩｏｎ；
［ＢＤＥＦ＿ＲＥＧ］Ｂ＿ＲＥＧ－ＸＮＡＰ＿ＲＥＧ－ＸＣＡＴ＿ＲＥＧ－ＸＢＵＴ＿ＲＥＧ＝０；
［ＢＤＥＦ＿ＰＲＭ］Ｂ＿ＰＲＭ－ＸＮＡＰ＿ＰＲＭ－ＸＣＡＴ＿ＰＲＭ－ＸＢＵＴ＿ＰＲＭ＝０；
！ＢａｔｃｈｓＩｚｅｌＩｍＩｔｓ；
［ＢＬＯ＿ＲＥＧ］Ｂ＿ＲＥＧ＞＝４０００；
［ＢＨＩ＿ＲＥＧ］Ｂ＿ＲＥＧ＜＝８０００；
［ＢＬＯ＿ＰＲＭ］Ｂ＿ＰＲＭ＞＝２０００；
［ＢＨＩ＿ＰＲＭ］Ｂ＿ＰＲＭ＜＝６０００；
！ＱＵａｌＩｔｙｒｅｓｔｒＩｃｔＩｏｎｓｆｏｒｅａｃｈｑＵａｌＩｔｙ；
［ＱＵＰＲＥＧＯＣ］－
１１０＊Ｂ＿ＲＥＧ＋ＳＯＣＴ＿ＲＥＧ＋７４＊ＸＮＡＰ＿ＲＥＧ＋１００＊ＸＣＡＴ＿ＲＥＧ＋１２０＊ＸＢＵＴ＿ＲＥＧ＝０；
［ＱＤＮＲＥＧＯＣ］－２１＊Ｂ＿ＲＥＧ＋ＳＯＣＴ＿ＲＥＧ＜＝０；
［ＱＵＰＲＥＧＶＡ］－
１１＊Ｂ＿ＲＥＧ＋ＳＶＡＰ＿ＲＥＧ＋４．１＊ＸＮＡＰ＿ＲＥＧ＋２．６＊ＸＣＡＴ＿ＲＥＧ＋６０＊ＸＢＵＴ＿ＲＥＧ＝０；
［ＱＤＮＲＥＧＶＡ］－３＊Ｂ＿ＲＥＧ＋ＳＶＡＰ＿ＲＥＧ＜＝０；
［ＱＵＰＲＥＧＶＯ］－
２５＊Ｂ＿ＲＥＧ＋ＳＶＯＬ＿ＲＥＧ＋１２＊ＸＮＡＰ＿ＲＥＧ＋３＊ＸＣＡＴ＿ＲＥＧ＋１０５＊ＸＢＵＴ＿ＲＥＧ＝０；
［ＱＤＮＲＥＧＶＯ］－８＊Ｂ＿ＲＥＧ＋ＳＶＯＬ＿ＲＥＧ＜＝０；
［ＱＵＰＰＲＭＯＣ］－
１１０＊Ｂ＿ＰＲＭ＋ＳＯＣＴ＿ＰＲＭ＋７４＊ＸＮＡＰ＿ＰＲＭ＋１００＊ＸＣＡＴ＿ＰＲＭ＋１２０＊ＸＢＵＴ＿ＰＲＭ＝０；
［ＱＤＮＰＲＭＯＣ］－１６＊Ｂ＿ＰＲＭ＋ＳＯＣＴ＿ＰＲＭ＜＝０；
［ＱＵＰＰＲＭＶＡ］－
１１＊Ｂ＿ＰＲＭ＋ＳＶＡＰ＿ＰＲＭ＋４．１＊ＸＮＡＰ＿ＰＲＭ＋２．６＊ＸＣＡＴ＿ＰＲＭ＋６０＊ＸＢＵＴ＿ＰＲＭ＝０；
［ＱＤＮＰＲＭＶＡ］－３＊Ｂ＿ＰＲＭ＋ＳＶＡＰ＿ＰＲＭ＜＝０；
［ＱＵＰＰＲＭＶＯ］－
２５＊Ｂ＿ＰＲＭ＋ＳＶＯＬ＿ＰＲＭ＋１２＊ＸＮＡＰ＿ＰＲＭ＋３＊ＸＣＡＴ＿ＰＲＭ＋１０５＊ＸＢＵＴ＿ＰＲＭ＝０；
［ＱＤＮＰＲＭＶＯ］－８＊Ｂ＿ＰＲＭ＋ＳＶＯＬ＿ＰＲＭ＜＝０；
```

```
END
```
次は，同じモデルを集合を用いて定式化したモデルである．

```
MODEL:
! General Blending Model(BLEND) in LINGO;
 SETS:
!Each raw material has availability & cost/Unit;
   RM/ BUT, CAT, NAP/: A, C;
! Each f.g. has min & max sellable, Profit
contr./Unit and batch size to be determined;
   FG/ REG, PRM/: D, E, P, B;
   ! There are a set of quality measures;
   QM/ OCT, VAP, VOL/;
!Each RM & QM combo has a quality level;
   RQ( RM, QM): Q;
!For each combo QM, FG there are upper &
lower limits on quality, slack on quality to be determined;
   QF( QM, FG): U, L, S;
!Each combination of RM and FG has an amount used,to be
determined;
   RF( RM, FG): X;
 ENDSETS
 DATA:
A=1000, 4000, 5000;!Raw material availabilities;
C =  7.3, 18.2, 12.5;     ! R. M. costs;
Q =  120,   60, 105, !Quality Parameters...;
     100,  2.6,   3,     ! R. M. by qUalIty;
      74,  4.1,  12;
D = 4000, 2000; ! Min needed of each F.G.;
E = 8000, 6000; !Max sellable of each F.G;
P = 18.4,   22; !Selling Price of each F.G.;
U =  110,  110, ! Upper limits on quality;
      11,   11, ! Quality by F.G.;
      25,   25;
L =   89,   94, !Lower limits on quality...;
       8,    8,       ! Quality by F.G.;
      17,   17;
 ENDDATA
```

```
!----------------------------------------;
! The model;
! For each raw material, the availabilities;
     @FOR( RM( I):
     [RMLIM] @SUM( FG( K): X( I, K)) < A( I););
     @FOR( FG( K):
!For each finished good, compute batch size;
      [BDEF] B( K) = @SUM( RM( I): X( I, K));
      ! Batch sIze limits;
           [BLO] B( K) > D( K);
           [BHI] B( K) < E( K);
    ! Quality restrictions for each quality;
    @FOR( QM( J):
[QUP]@SUM( RM(I): Q(I, J) * X(I, K)) + S( J,
          K) = U( J, K) * B( K);
[QDN] S(J, K) < (U(J, K) - L(J, K)) * B(K);); );
!We want to maximize Profit contribution;
[PROFIT] MAX = @SUM( FG: P * B)
        - @SUM( RM( I): C( I) * @SUM( FG( K): X( I, K)));
END
```

解は次の通りである.

```
 Objective value:                    48750.00
       Variable            Value          ReDduced Cost
        B( REG)          4000.000            0.0000000
        B( PRM)          4500.000            0.0000000
   S( OCT, REG)          84000.00            0.0000000
   S( OCT, PRM)          72000.00            0.0000000
   S( VAP, REG)          1350.424            0.0000000
   S( VAP, PRM)          7399.576            0.0000000
   S( VOL, REG)          17500.00            0.0000000
   S( VOL, PRM)          36000.00            0.0000000
   X( BUT, REG)          507.4153            0.0000000
   X( BUT, PRM)          492.5847            0.0000000
   X( CAT, REG)          1409.958            0.0000000
   X( CAT, PRM)          2590.042            0.0000000
   X( NAP, REG)          2082.627            0.0000000
   X( NAP, PRM)          1417.373            0.0000000
```

| Row | Slack or Surplus | Dual Price |
|---|---|---|
| RMLIM( BUT) | 0.0000000 | 27.05000 |
| RMLIM( CAT) | 0.0000000 | 6.650000 |
| RMLIM( NAP) | 1500.000 | 0.0000000 |
| BDEF( REG) | 0.0000000 | -22.65000 |
| BLO( REG) | 0.0000000 | -1.225000 |
| BHI( REG) | 4000.000 | 0.0000000 |
| QUP( REG, OCT) | 0.0000000 | -0.4750000 |
| QDN( REG, OCT) | 0.0000000 | 0.4750000 |
| QUP( REG, VAP) | 0.0000000 | 0.0000000 |
| QDN( REG, VAP) | 10649.58 | 0.0000000 |
| QUP( REG, VOL) | 0.0000000 | 0.0000000 |
| QDN( REG, VOL) | 14500.00 | 0.0000000 |
| BDEF( PRM) | 0.0000000 | -22.65000 |
| BLO( PRM) | 2500.000 | 0.0000000 |
| BHI( PRM) | 1500.000 | 0.0000000 |
| QUP( PRM, OCT) | 0.0000000 | -0.4750000 |
| QDN( PRM, OCT) | 0.0000000 | 0.4750000 |
| QUP( PRM, VAP) | 0.0000000 | 0.0000000 |
| QDN( PRM, VAP) | 6100.424 | 0.0000000 |
| QUP( PRM, VOL) | 0.0000000 | 0.0000000 |
| QDN( PRM, VOL) | 0.0000000 | 0.0000000 |
| PROFIT | 48750.00 | 1.000000 |

解は，高級ガソリンが最も利益に貢献するので，希少資源の BUT と CAT が許す限り，上質ガソリンを 4,500 で普通ガソリンを 4,000 生産する．集合ベースのモデは，データがモデル方程式から分離されている．従って，データが変っても，ユーザーはモデル方程式を変更する必要はない．

配合モデルは，ずっと幾年もの間精製所の標準的な操業手段である．最近，これらの LP モデルがより正確な非線形モデルの配合プロセスに取替えられている．Texaco 社がどう行ったかは，Rigby, Lasdon & Waren(1995)参照．

ガソリン配合モデルは，色々複雑な事例がある．例えば，上質ガソリンは，ベンダーがオクタン成分を揮発性の変化の範囲内で一定にして欲しいかもしれない．理由は，非燃料噴射型の自動車の加速装置を踏むと，未加工ガスが取り入れ口に取り入れられる．非常に揮発性のある部分が燃焼室に最初に達する．これらの構成要素にオクタンが低いものがあれば，ガソリンの「平均」オクタン価が高くてもノッキングが起きる．これは大部分のドライブが不変の速度であるカンザスのクロス・カントリーのハイウェーにある給油所よりも，都市の給油所で重要かもしれない．

## 10.4 品質要求の代替案の正しい選択

品質の特徴は，良さの尺度の代わりに，悪さの尺度で表現出来る．例えば，自動車の燃費があげられる．1 ガロンあたりの走行マイル数のかわりに，1 マイルあたりのガロン数で表すことができる．配合成分の品質（例えば車の燃費など）について考えるとき，配合成分全てを足す場合，良さの尺度であるか悪さの尺度であるかを統一することが大切だ．次に，この例を示そう．

連邦政府の規定では，ある年にある自動車会社で売られた車の全ての平均燃費が，18 マイル／ガロン以上でなければいけない．フォード社における仮説的なケースを考えてみよう．フォードは 4 種の車（マークＶ，フォード，グラナダ，フィエスタ）しか売らないと仮定する．これらの車についての個々のパラメータが下にあげられている．

| 自動車 | マイル/ガロン | 限界生産費用 | 売価 |
|---|---|---|---|
| フィエスタ | 30 | 13,500 | 14,000 |
| グラナダ | 18 | 14,100 | 15,700 |
| ForD | 16 | 14,500 | 15,300 |
| Mark V | 14 | 15,700 | 20,000 |

生産能力については，ある程度自由度があり，2 車種について適用できる．これらの制限は，次の通りである．

| 年間の容量 | 車種制限 |
|---|---|
| 250,000 | フィエスタ |
| 2,000,000 | グラナダ + ForDs |
| 1,500,000 | ForDs ＋ Mark V’s |

合計 300 万台売ることができる．フォードでは，どの車種をどの位売る計画を立てれば良いだろうか？マイル数制約による定式化は，次になる．

```
MAX = 500*FIESTA+1600*GRANADA+4300*MARKV+800*FORD;
  12*FIESTA-4 * MARKV-2 * FORD >= 0;
  FIESTA <= 250;
  GRANADA + FORD <= 2000;
  MARKV + FORD <= 1500;
  FIESTA + GRANADA  + MARKV   + FORD <= 3000;
```

車とドルは 1000 単位である．行 2 の最初の制約は次と同じになる．

$$\frac{30\ FIesta + 18\ GranaDa + 16\ ForD + 14\ Mark\ V}{FIesta + GranaDa + ForD + Mark\ V} \geq 18.$$

解は次の通りである.

Optimal solution found at step: 1

Objective value: 6550000.

| Variable | Value | Reduced Cost |
|---|---|---|
| FIESTA | 250.0000 | 0.0000000 |
| GRANADA | 2000.000 | 0.0000000 |
| MARKV | 750.0000 | 0.0000000 |
| FORD | 0.0000000 | 2950.000 |

| Row | Slack or Surplus | Dual Price |
|---|---|---|
| 1 | 6550000. | 1.000000 |
| 2 | 0.0000000 | -1075.000 |
| 3 | 0.0000000 | 13400.00 |
| 4 | 0.0000000 | 1600.000 |
| 5 | 750.0000 | 0.0000000 |
| 6 | 0.0000000 | 0.0000000 |

車種にかかわらず，1 年間の走行マイル数がどの車も同じだと仮定しよう．全車種の全車による合計走行マイル数を，使われたガソリンのガロン数で割ったものが 18 以上になるかという疑問がわく．ロスなく，おのおのの車が 1 マイル走ったとする．1 マイル走るのに使われるガソリンは 1／（マイル／ガロン）である．ゆえに全部の車が同じ距離を走ったら，車の使った燃料のガロンに対する走行マイル数の割合は，

(250+2000+750)/(250/30+2000/18+750/14)=17.3 マイル／ガロンとなり，18 マイル／ガロン以下の解を我々は得る．

最初の定式化は各車に同じ量のガソリンを割り当て，各車がその割当をすべて使いきることと同じことである．燃費の悪い車が少ない距離を走ることで，平均 18 マイル／ガロンとなったのである．2 番目の定式化は，ガロン/マイルを用いており，次の制約になる．このマイル数の制約は次のようになる．

$$\frac{Fiesta/30+Granada/18+Ford/16+MarkV/14}{Fiesta+Granada+Ford+MarkV} \leq 1/18$$

これを標準型になおすと

$$-.022222FIESTA+.0069444FORD+.015873MARKV \leqq 0$$

この制約でこの問題を解くと次のような解が得られる．

```
Optimal solution found at step:        0
Objective value:                 4830000.
Variable          Value        Reduced Cost
  FIESTA        250.0000          0.0000000
    FORD      0.0000000            2681.250
   MARKV        350.0000          0.0000000
 GRANADA        2000.000          0.0000000
```

2 番目の定式化では，明らかに利益が減少している．また最初の定式化は，連邦政府の規定を満足している．しかし自動車会社では，連邦政府の真意ではなく，実際に言ったことを実行して後で批判されるよりも，安全を見越して 2 番目の方法でマイル数を計算している．

例えば，1998 年の「アメリカの軽トラック」（いわゆるスポーツ多用途車）の規制は 20.7 マイル/ガロンで，乗用車の規制は 27.5 マイル/ガロンであった．生産した車が要件に満たない場合，1 ガロンあたり 10 マイルごとに，米国連邦政府は車両 1 台につき 5 ドルの罰金を課す．必要条件は，「モデル年度」に基づいています．これにより，ある年に目標を達成できない場合に，ある程度の柔軟性が自動車メーカーに与えられます．例えば，大きなシボレーなどのマイレージが悪い車両の「生産を停止する」ことができ，その後のすべてのコピーが次のモデル年度に属すると宣言します．これは現在のモデル年度の目標を達成するかもしれないが，問題を次のモデル年度に延期する．

## 10.5 配合品質の計算法

物の配合では，平均をとって効率を測定する必要がある．N 個の観測値（x1，x2，… .xN）があるとき，平均を計算する少なくとも 3 つの方法がある：

算術平均：$(x_1+x_2\ .\ .\ .\ +x_N)/N$

幾何学的：$(x_1\times x_2\ .\ .\ .\ \times x_N)\hat{}(1/N)$

調和平均：$1/[(1/x_1+1/x_2\ .\ .\ .\ +1/x_N)/N]$

算術平均は，有価証券の資産の平均リターンを計算するために適切である．しかし，有価証券の平均成長に興味があれば，私達はおそらく年次成長の幾何平均を使用する．最初の年に 1.5 で 2 年目が 0.67（初年度 50%で，2 年度-33%の収益率）の投資を考える．このとき，平均成長率を算術平均の(1.5+0.67)/2= 1.085 でなくて，調和平均を用いることが適切である．調和平均が適切なものに，前の自動車の例のように何かの平均の比を用いる場合である．

調和平均が適切な他の例に密度がある．食糧や飼料のような製品は，種々の密度をもつ異なった原料から特定の決められた密度の製品を製造したい．密度は通常容積ごとの重量(例えばグラム/$cm^3$)で測定される．決定変数が体積よりも密度で測定されていれば，調和平均が適切である．

### 10.5.1 例

2つの原料が，0.7 g/cc と0.9 g/cc の密度とする．各1グラムを一緒に配合すれば，密度はどうなるだろう？配合品の重量は2グラムである．体積は（1/0.7 + 1/0.9）である．従って，密度は2/(1/0.7 +1/0.9)=0.7875 g/cc である．算術平均の0.8より小さい．次の定義をする：

$X_I$ =配合品の原料 $I$ のグラム，

$t$ =望ましい密度の下限．

密度の下限制約（調和平均制約）は，次のようになる：

$(X_1 + X_2)/(X_1/0.7 + X_2/0.9) \geqq t$

or

$(1/t - 1/0.7)X_1 + (1/t - 1/0.9) X_2 \geqq 0$

### 10.5.2 一般平均

変換 f(q) で上の議論を一般化する．関数 f() は品質を線形化する関数である．基本的な考えは，実際に使われる品質の測定単位の多くが，任意に選ばれていることである（例えば，なぜ水の氷点は華氏スケールの 32 度であるか？）．それで，標準的に使われ品質の測定値が線形に配合できないとしても，それを行う変換を見つけることができる．そのような線形化は，産業界で一般的である．若干の例を示す：

①Rigby, Lasdon,& Waren (1995)は，テキサコ石油で配合された揮発油のリード蒸気圧を計算するとき，この考えを使った．$r_I$ がブレンドの構成要素 I のリード蒸気圧であるならば，

変換：$f(r_I) = r_I^{1.25}$ を使った．

構成要素 1 が 80 の蒸気圧，構成要素 2 が 100 の蒸気圧をもち，$x_I$ は構成要素 I の使用量とし，少なくとも 90 の蒸気圧で配合したい．制約は次のようになる：

$$80^{1.25}x_1 + 100^{1.25}x_2 \geq 90^{1.25}(x_1 + x_2),$$

or

$$239.26\,x_1 + 316.23x_2 \geq 277.21(x_1 + x_2).$$

② 化学製品の発火点は，それが燃え出す最も低い温度である．典型的なジェット燃料は，およそ 100 度 F の発火点を持つ．典型的な暖房油は，少なくとも 130 度 F の発火点を持つ．超音速 Sr-71 ジェット機で使われる燃料は，数百程度 F の発火点である．$P_I$ が構成要素 I の発火点であるならば，次の変換が発火点の近似線形式である：$f(P_I) = 10^{42}(P_I + 460)^{-14.286}$

たとえば，構成要素 1 が 100，構成要素 2 が 140 の発火点をもち，$x_I$ は構成要素 I の使用量として，少なくとも 130 の発火点の配合品を作りたいとき，次の制約になる：

$$10^{42}(100 + 460)^{-14.286}\,x_1 + 10^{42}(140+ 460)^{-14.286}\,x_2 \leq \geq$$

$10^{42}(130+460)^{-14.286}(x_1 + x_2)$,

or

$548.76\ x_1 + 204.8\ x_2 \geq 260\ (x_1 + x_2)$.

③アメリカ石油協会（API）の規格の「API 重量」で材料の重さを計りたい（Dantzig & Thapa (1997)）．API 重量は線形に配合できない．しかし，指定した重量は次の式で線形化できる：

$sg = 141.5/(API\ gravity + 131.5)$

材料の比重は，1 立方センチメートルの材料のグラムでの重さである．水は，10 の API 重量を持つ．

④ 液体の粘性は，センチストーク単位で計られる．特定の直径の穴の中を流れるために，華氏 122 度である．高い粘性の液体は，より簡単に流れない．$v_I$ が構成要素 I の粘性であるならば，次の変換は粘性をほぼ線形にする：

$f(v_I) = \ln(\ln(v_I + .08))$

例えば，構成要素 1 が 5，そして構成要素 2 が 25 の粘性をもち，$x_I$ は構成要素 I の容量とし，20 の粘性をもつ配合品を作る制約式は，次の通りである：

$\ln(\ln(5+.08)) \times x_1 + \ln(\ln(25+.08)) \times x_2 \leq$

$\ln(\ln(20+.08)) \times (x_1 + x_2)$,

or

$.4857 x_1 + 1.17 x_2 \leq 1.0985 (x_1 + x_2)$.

⑤ 長さ xi のグラスファイバーを通る光の伝送率や，期間 xi にわたる投資の成長率や，試行 $x_i$ で失敗しない確率 a は，次のような制約になる：

$a_1^{x1}\ a_2^{x2} \cdots a_n^{xn} \geq a_0$.

これは，対数をとることで線形になる（すなわち，$\ln(a_1) * x_1 + \ln(a_2) * x_2 + \cdots \ln(a_n) * x_n \geq \ln(a_0)$）．

例えば，年 10%の長期成長を期待している株と，年 6%のリスクの少ない債権があり，5 年間で 40%の利益を得たい．X1 と x2 を株と債権につぎ込む年数であるならば，次の制約を考えればよい：

$(1.10)^{x1}(1.06)^{x2} \geq 1.40$.

次のようにして，線形になる:

$\ln(1.10) x_1 + \ln(1.06) x_2 \geq \ln(1.40)$

**または**$.09531 x_1 + .05827 x_2 \geq .3364$,

$x_1 + x_2 = 5$.

これらの例は，行列を用いた変換を許すことで，複数の品質変換に利用できる．

## 10.6　配合制約の双対価格の解釈

配合制約の双価価格を有効に使うには，少し説明が必要である．例として高級ガソリン配合問題のオクタン価の下限制約について考えてみよう．

$-94PRN + 120BUT + 74NAP + 100CAT \geqq 0$

この制約の双対価格は，右辺が 0 から 1 に変わると利益の増加を表す．不運なことに，この変化は普通では考えられない．もっと典型的な変化は，オクタンの割合が 94 から 95 に変わることだ．変数 PRN の I 行の係数の小さな変化の影響の測定の方法は，I 行の双対価格と PRN の積を計算することだ．変数 RG とオクタンの制約についてこの値は，

7270.30*(-.204298)=-1485.31

である．これはオクタンの品質要求が 94 から 93 に減ると（95 に増えると），合計利益は 1485 増えて 44814 になる（1485 減って 41843 になる）．わずかな変化に対するこの近似値は，変化後の真の利益と言うこともできる．実際にオクタンの品質要求を 94 にして（あるいは 95 にして）LP を解き直すと，実際の利益は 44825 になる（43200 になる）．

この近似値は，一般に次のようにまとめることができる．もし配合のある品質要求を小量 e だけ変えたいとき，この変化の利益への影響は，e の大きさ×（制約の双対価格）×（バッチサイズ）で近似できる．この小量の変化の近似は変化後の利益とも言える．大きな変化に対しての近似はどの様な方向に対しても誤りとなる．

## 10.7　分数計画法

配合問題で，次の比率制約があった：

$$\frac{\sum_j q_i X_j}{\sum_j X_j} \geq q_0$$

そして，次のように線形にできた：

$$\sum_j q_j X_j \geq q_0 \sum X_j \quad \text{or} \quad \sum (q_j - q_0) x_j \geq 0$$

目的関数で，類似した特徴を扱えるだろうか？ それはできる．以下の形式の問題は，線形形式に変換できるだろうか？

(1) Maximize $\dfrac{u_o + \sum_j u_j X_j}{v_o + \sum_j v_j X_j}$

(2) subject to: $\sum_j a_{Ij} X_j = b_I$

$a_{I,j}, U_0, U_j, v_0, v_j$ は定数である．以下の変換で線形にすることができる：

$r = 1/(v_0 + \sum_j v_j X_j)$，$y_j = X_j r$ として $r>0$ とすると，目的関数は次のようになる：

(1') Maximize $U_0 r + \sum_j U_j y_j$

subject to:

$r = 1/(v_0 + \sum_j v_j X_j)$

(1.1') $r\ v_0 + r\ \sum_j v_j\ y_j = 1$
制約式(2)は：
$\sum_j a_{Ij}\ X_j = b_I$
次のようになる：
$\sum_j a_{Ij}\ X_j\ r = b_I\ r$
or
(2') $\sum_j a_{Ij}\ y_I - b_I\ r = 0$

## 10.8 多段階配合：プールする場合

配合問題を複雑にする要因は，全ての原材料が別々に保存されていない点である．このような状況は，多くの状況で起こる．経済的な理由のため，2つの成分は同じ場所で生産されるかもしれないし，1台のタンク車や1つのパイプラインで一緒に輸送されるかもしれない．もう一つの可能性は，2つの原材料は別々に届けられるが，一つの設備だけが配合場所で利用できる．一般に原材料を配合する多くの施設は，貯蔵装置の数に限りがある．例えば，穀物貯蔵装置は6台だけかもしれない．石油精製所のタンクは，6台だけかもしれない．6台以上の原料がある場合，別々に保存できない．石油産業では，これは**「プール問題」**と呼ばれている．

配合問題におけるプール問題は，非線形計画になる．ここで議論する問題は，Haverly (1978)参照．原料A，BとCは，不純物として3%，1%，2%の硫黄を含んでいる．これらは，2つの化学製品XとYを提供するために配合されるが，各2.5%と1.5%の硫黄成分の上限がある．XとYの単価は9ドルと15ドル/単位で，XとYを100/単位と200/単位まで購入できる．原料A，B，Cの単価は，6ドル，16ドル，10ドルである．問題は，利益を最大にする操業の決定である．

この配合問題を複雑にしている要因は，**図10.1**に示すように製品AとBが同じタンクに保存しなければならないという事実である．それで，AとBの量が決定されるまで，プールされた硫黄含有量は分からない．XとYの硫黄含有量は，プールされたものとCに含まれる硫黄含有量の合計である．XとYの硫黄制約はAとBの必要な量に影響を及ぼす．結局循環的になり，非線形性を引き起こす．

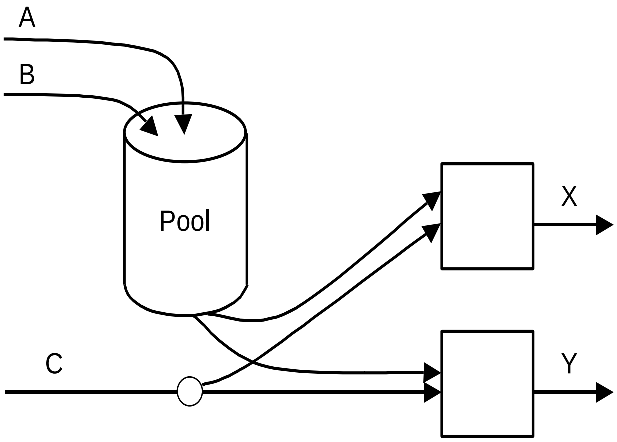

**図 10.1 プール問題**

このシステムを定義している制約方程式は，出力製品のために材料バランスと硫黄制約を含む．最初に材料バランス方程式を考慮しなさい．プールされた原料はすべて使用されると仮定して，次の関係がある：

A の購入量(AmountA) + B の購入量 =

X のプール分(POOLTOX) + Y のプール分．

出力製品は，次の制約がある：

X のプール分 + CtoX =X の生産量(AmountX)

Y のプール分 + CtoY = Y の生産量．

C の総量に関して： CtoX + CtoY = C の購入量．

新しい変数として Pools でプールされた硫黄分を表すと，X と Y へのプールされた硫黄分の制約を書くことが簡単になる．Pools を 0 と 100 の間で，他の全てのパーセンテージを同じスケールで表すと，これらの制約は次になる：

Pools × X のプール分 + 2 × $CtoX$ ≤ 2.5 × X の生産量

Pools × Y のプール分 + 2 × $CtoY$ ≤ 1.5 × Y の生産量

不等式の左側が製品の実際の硫黄分で，右辺は製品で許される硫黄の最大量である．プールされた硫黄のバランス方程式は：

3×A の購入量+B の購入量 =Pools×(A の購入量 + B の購入量)．

これは各原材料に含まれる硫黄分が，原材料の総量のうち Pools でもって表される硫黄分と等しいことを表す．すでに述べたように，製品の需要上限は次の通りである：

X の生産量 ≤ 100

Y の生産量 ≤ 200

全変数は非負で，1 ≤ Pools ≤ 3 である．

最後に，利益関数は，Cost$A$，Cost$B$，Cost$C$，Cost$X$，Cost$Y$ が適当なコスト係数であるならば，次の式になる：

Cost$X$ × X の生産量 + Cost$Y$ × Y の生産量 − Cost$A$ × A の購入量− Cost$B$ × B の購入量 − Cost$C$ × C の購入量

LINGO の定式化は次の通りである：

```
MODEL:
   COSTA = 6;
   COSTB = 16;
   COSTC = 10;
   COSTX = 9;
```

```
   COSTY = 15;
 MAX = COSTX * AMOUNTX + COSTY * AMOUNTY - COSTA * AMOUNTA -
 COSTB * AMOUNTB - COSTC * AMOUNTC;
   ! SoUrces = Uses for the Pool;
   AMOUNTA + AMOUNTB = POOLTOX + POOLTOY;
   ! SoUrces for fInal ProDUcts;
   POOLTOX + CTOX = AMOUNTX;
   POOLTOY + CTOY = AMOUNTY;
   ! Uses of C;
   AMOUNTC = CTOX + CTOY;
   ! BlenDIng constraInts for fInal ProDUcts;
   POOLS * POOLTOX + 2 * CTOX <= 2.5 * AMOUNTX;
   POOLS * POOLTOY + 2 * CTOY <= 1.5 * AMOUNTY;
   ! BlenDIng constraInt for the Pool ProDUct;
   3*AMOUNTA + AMOUNTB=POOLS*(AMOUNTA + AMOUNTB);
   ! DemanD UPPer lImIts;
   AMOUNTX <= 100;
   AMOUNTY <= 200;
END
```

この問題は，幾つもの局所最適解がありトリッキーである．LINGO は次の解を求めた．

```
Optimal solution found at step:        16
ObjectIve Value:                 400.0000
VarIable          Value        Reduced Cost
   COSTA        6.000000          0.0000000
   COSTB        16.00000          0.0000000
   COSTC        10.00000          0.0000000
   COSTX        9.000000          0.0000000
   COSTY        15.00000          0.0000000
 AMOUNTX       0.0000000          0.0000000
 AMOUNTY        200.0000          0.0000000
 AMOUNTA       0.0000000           2.000003
 AMOUNTB        100.0000          0.0000000
 AMOUNTC       100.00000          0.0000000
 POOLTOX       0.0000000           4.000026
 POOLTOY        100.0000          0.0000000
    CTOX       0.0000000          0.0000000
```

| | | |
|---|---|---|
| CTOY | 100.00000 | 0.0000000 |
| POOLS | 0.9999932 | 0.0000000 |

最適操業は，製品 *Y* を *B* と *C* の使用量と同じ量を作ることである．費用は$(16 + 10)/2 = $13 で，販売価格は$15 なので，利益は$2 である．200 単位生産するので，利益は$400 になる．このような非線形問題は，初期値で解が異なるという奇妙な性質がある．次のような INIT 節で，任意の初期値を定義できる：

訳注：大域的最適化オプションを用いれば，この議論は不要になる．

INIT 節で任意の初期値を定義できる．上の初期値で LINGO は次の解を得た：

```
INIT:
AMOUNTX = 0
AMOUNTY = 0
AMOUNTA = 0
AMOUNTB = 0
AMOUNTC = 0
POOLTOX = 0
POOLTOY = 0
   CTOX = 0
   CTOY = 0
  POOLS = 3
ENDINIT
```

Optimal solution found at step: 4

Objective Value: 100.0000

| Variable | Value | Reduced Cost |
|---|---|---|
| AMOUNTX | 100.0000 | 0.0000000 |
| AMOUNTY | 0.0000000 | 0.0000000 |
| AMOUNTA | 50.00000 | 0.0000000 |
| AMOUNTB | 0.0000000 | 2.000005 |
| AMOUNTC | 50.00000 | 0.0000000 |
| POOLTOX | 50.00000 | 0.0000000 |
| POOLTOY | 0.0000000 | 6.000000 |
| CTOX | 50.00000 | 0.0000000 |
| CTOY | 0.0000000 | 0.0000000 |
| POOLS | 3.000000 | 0.0000000 |
| COSTA | 6.000000 | 0.0000000 |
| COSTB | 16.00000 | 0.0000000 |

| | | |
|---|---|---|
| COSTC | 10.00000 | 0.0000000 |
| COSTX | 9.000000 | 0.0000000 |
| COSTY | 15.00000 | 0.0000000 |

この解で，原料 A と B を用い製品 X だけを生産し販売した．製品の費用は 8 ドル/単位で，利益は 1 ドル/単位である．100 単位が生産されたので，最終的な利益は 100 ドルである．この解は局所最適である．すなわち，この操業からの小さい変化は，利益を減らす．この解の周りに，より良い解はない．

我々の以前の解は 400 ドルの利益を生んでいるが，局所最適解である．しかし，他のいかなる可能な点もこれより大きい利益はないので，400 ドルは**大域的最適解**と呼ぶ．読者は，例えば A の使用量を増やし，B と C を減少すると他の局所最適解を見つけることができる．一般的に言って，初期値をゼロにセットしてはいけない．0 にどんな量を掛けても 0 であるので，最適化アルゴリズムの挙動に悪さをする 例えば，前の例で POOLS= 2 に変更すると，次の解に陥る：

```
Optimal solution found at step:        1
Objective value:                 0.0000000E+00
Variable           Value        Reduced Cost
 AMOUNTX        0.0000000          0.0000000
 AMOUNTY        0.0000000          0.0000000
 AMOUNTA        0.0000000           6.000000
 AMOUNTB        0.0000000           16.00000
 AMOUNTC        0.0000000          0.0000000
 POOLTOX        0.0000000          -10.00000
 POOLTOY        0.0000000          -10.00000
    CTOX        0.0000000          0.0000000
    CTOY        0.0000000          0.0000000
   POOLS         2.000000          0.0000000
   COSTA         6.000000          0.0000000
   COSTB         16.00000          0.0000000
   COSTC         10.00000          0.0000000
   COSTX         9.000000          0.0000000
   COSTY         15.00000          0.0000000
```

この出力は，局所最適解でない安定した点を選んだ．すなわち，目的関数で，アルゴリズムに使われるトレランスの範囲内で小さい変化をしても改良されない点である．この点は，1 次の最適条件を満たす．しかし，出発点を少し変更すれば，解は局所最適解を見つけ，場合により大域的最適解になるだろう．実際，全ての変数の 0 の初期値を 0.1 に設定すると，大域的最適解になる．このモデルは，LINGO の

Global オプションを使うことが有用有用である例である．これを用いれば，たやすく 400 という大域的最適解が必ず見つかる．

この問題で，多くの解は，等式制約を満たす．もちろん，5 つの等式制約（列 5 から列 8 と列 2）は，常に活発（Active）である．それに加えて，大域的最適解で，Y の硫黄分が上限制約になり，6 つの変数（*POOLS*，*CTOX*，*POOLTOX*，*AMOUNTA*，*AMOUNTY*，*AMOUNTX*）が上下限制約に拘束される．それゆえ，12 の活発な制約と，10 の変数だけがある．変数と同じだけの活発な制約がある場合，頂点解と呼ぶ．LP で頂点解になるが，NLP では正しくない．しかし，配合問題では NLP であっても頂点解はまれなことではない．

変数より多い活発な制約があるとき，頂点は退化であると言われる．この問題の大域的最適解で，2 つの活発な制約がある．POOL の上下限式を取り去ることで，冗長性が解消できる．制約 8 と変数の非負は不要である．制約式列 3 と 4 と，CTOX，CTOY，POOLTOX，POOLTOY の下限でほのめかされるので，AMOUNTX と AMOUNTY の下限を落とすことができた．これで変数と同じだけの活発な制約式をもつ頂点解を得た．他の制約も冗長である．読者は，それらを見つけてみよう．

# 第11章　IPの定式化と解法

“To be or not to be”is true.

-G.Boole-

## 11.1　はじめに

最適化の応用で，決定変数が整数であってほしい場合が多い．GM が 1,524,328.37 台のシボレーを生産すべき解を得た場合，これを四捨五入しても問題はない．しかし，航空会社が確保すべき飛行機の台数として 1.37 台という解が得られたら，この数値を一体どう扱うかが問題になる．多くの数理計画法モデルで，その幾つかの決定変数を整数に指定できれば非常に有意義である．よい商業用の最適化ソフト（ソルバーと呼ぶこともある）は，ユーザーが決定変数を整数値に制限できる．ユーザーがこの条件をソルバーに知らせる方法は異なっている．LINGO では X を整数変数に指定するのに@GIN（x）で行う．このように整数制約を含んだ問題を定式化し，その解を求めることを整数計画法（IP）と呼ぶ．

### 11.1.1　変数の種類

IP モデルは，変数の種類から以下の 2 つに分けられる．

① 純粋もしくは混合：純粋整数計画法は，全ての変数が整数に限定される．混合整数計画法は，幾つかの変数が整数に限定され，あとの変数は実数である．

② “0/1”変数もしくは一般整数変数：応用分野で整数値は 0 か 1 に限定されることが多い．従って，IP ソルバーによっては，整数変数はすべて“0/1”に限定している場合がある．

幾つかの変数を整数に指定することは，一見大した影響がないように見えるが，実際はきわめて強力な効果を持つ．モデルで整数変数が使われるのは，「あれか／これか」という選択を，“0/1”の変数で表わす時である．現実の IP の多くが，この“0/1”型である．

訳注：LINDO 社の製品の使用では，LP,QP,IP,NLP と確率計画法を読者が意識する必要がなく，最初に行われるモデルの分析でソルバーが自動的に必要な解法を選ぶ．

## 11.2 IPの標準的な応用

現実に良く出会う問題は，若干の組合せ的問題を除いては，LP 問題が多い．本章では，組合せ的性質を IP で定式化する方法について述べる．しかも，一般的な IP でなく，“0/1”型の IP が多いのは興味深い．

2 値変数は「開始する／停止する，作る／買う」などを表すのに使用できる．それらは時々「ハムレット」変数と言われる．「買うか，買わないか，それが問題だ」である．2 値変数はまた論理学者の George Boole を記念してブール変数と呼ぶこ

ともある．彼は 2 値だけを取る変数を操作するためにブール代数を開発した．ブール代数では，値は「真／偽」であるが，それは真を 1 に偽を 0 で表すちょっとした概念に跳躍すればよい．Boole が開発したこの方法は，現在では「Boole のように強力である」と賛辞されている．

### 11.2.1　一般整数変数の 2 進表示

有限の範囲にある一般的な整数変数は，一組の“0/1”で表すことができる．例えば x が整数（0，1，2，…，15）に制限されるとする．y1，y2，y3，y4 の 4 個の 0/1 変数を導入すると，X は（$y_1+2y_2+4y_3+8y_4$）で表すことができる．X の最大値が 31 なら 5 個の 0/1 変数があればよい．実際 k 個の 0/1 変数を使用すれば，最大（$2^k-1$）まで表すことができる．X の最大値の Log を取ると，必要な 0/1 変数の個数が分かる．この方法は一般的な非負の整数を表すことにも有効であるが，実際に適用することは効率的でない．

### 11.2.2　最小バッチサイズ問題

ある生産活動の水準を決めるため，規模の経済を無視できないとき，その活動の最小バッチサイズを決める必要が生じる．例えば，あなたが証券会社から債権を買う場合，少なくとも 100 単位購入しなければならないとしよう．このような条件は，次のように“0/1”変数で定式化できる．

**x ＝決定すべき活動水準（債権の発注数量）**

**y＝“0/1”変数，X＞0 のときのみ 1 の値を取る**

**B=X の最小バッチサイズ（すなわち 100）**

**U=X の値に関する既知の上限**

次の 2 つの制約が，最小バッチサイズ条件を表現している．

$x \leqq Uy$

$By \leqq x$

もし y=0 ならば，始めの制約が x=0 となることを強制する．もし y=1 ならば 2 番目の制約により，x が B 以上になるように強制する．定義 U は注意して決めなければならない．計算効率を高めるには，U は許される限り小さな値にするべきである．

ソルバーの中には，半連続変数 x を用いて，最小バッチサイズ要求を直接処理するものもある．x が半連続とは，x の値が 0 か $B \le x \le \infty$ で表される．この場合，11.2.1 で述べた 2 進数表現は用いない．

### 11.2.3　固定費問題

最小バッチサイズ問題と類似の状況として，ある生産活動の費用関数が，**図 11.1** のように固定費部分と量に比例する線形部分の和となっている場合がある．いま，変数 x，y と U を最小バッチサイズ問題の時と同様に定義して，x ＞0 の固定費をKとする．そして，定式化には次のような項が入ってくる．

MIN　Ky+Cx+...

ST　　　x ≦ Uy

この制約式と目的関数の項 $Ky$ で，$x$ が正の値を取ると，固定費 K が課せられる．ここでも計算効率上，許される限り小さな値を U に与えた方がよい．

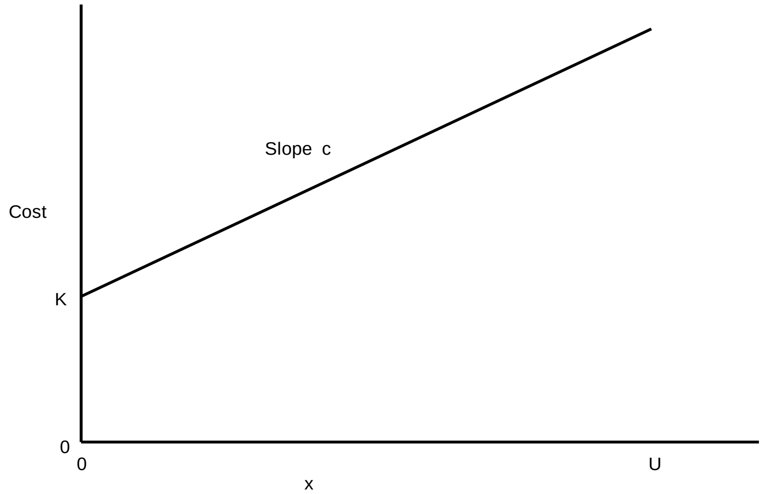


**図 11.1　典型的な固定費問題**

### 11.2.4　簡単な工場配置問題

簡単な工場配置（Simple Plant Location，SPL）問題は，よくみられる固定費用の問題であり，その定式化を次のように定義する．

**n　＝工場を配置する候補地の数**

**m　＝顧客地点，すなわち需要地の数，各需要地にはどれか 1 つの工場がサービスに当たるかを指定する．**

**k　＝開設される工場の数**

**$f_I$ ＝候補地 I で工場を運用する固定費（例えば，年当り）**

**I＝1，2，. . ，n**

**$c_{Ij}$＝顧客 j を候補地 I の工場に割り当てる費用（例えば，年当り）**

**I=1，2，. . ，n　j=1，2，. . ，m**

ここで問題は，どの候補地に工場を開設し，また各顧客地はどの工場からサービスを受けるかを決めることである．

SPL 問題の例としては，顧客が広い範囲に分布している状況での私書箱配置問題がある．この場合，工場候補地に相当するのが，銀行に運営して貰う私書箱の配置候補である．顧客地点は，会社のある大都市地域であり，例えば 100 個ほどの私書箱を必要としている．顧客はその支払を一番近い私書箱に郵送する．全ての支払を 1 ケ所に郵送せず，何か所かの私書箱を使う理由は，郵送の日数を短縮したいからである．いま，郵便で 6 千万ドル受け取る会社が，郵送時間を 2 日減らすと，この減少分の価値は資本費用を年 10％として，1 年当り約 3 万ドル改善できる．特定の地域 I に私書箱を持つ年当りの費用 $f_I$ は，その地域で処理する送料には関係ない．費用の項 c は，だいたい（資本の 1 日当りの費用）×（I と j の間の郵送日数）×（j からの年間送料（ドル表示））に等しい．

工場設置問題で，次の決定変数を定義する：

**$y_I$ ＝ 1（候補地 I に工場を開設するとき），0（開設しないとき）**

**$x_{Ij}$＝1（顧客 j が工場 I に割り当てられるとき），0（そうでないとき）**

この問題の IP のコンパクトな定式化は次の通りである．

$$\text{Min} \quad \sum_{i=1}^{n} f_I y_I + \sum_{i=1}^{n} \sum_{j=1}^{m} c_{Ij} x_{Ij} \qquad (1)$$

$$\text{st} \quad \sum_{i=1}^{n} x_{Ij} = 1 \quad \text{for } j = 1 \text{ to } m, \qquad (2)$$

$$\sum_{j=1}^{m} x_{Ij} \le m y_I \quad \text{for } I = 1 \text{ to } n, \qquad (3)$$

$$\sum_{i=1}^{n} y_I = k, \qquad (4)$$

$$y_I = 0 \text{ or } 1 \quad \text{for } I = 1 \text{ to } n, \qquad (5)$$

$$x_{Ij} = 0 \text{ or } 1 \quad \text{for } I = 1 \text{ to } n,\ j = 1 \text{ to } m. \qquad (6)$$

制約(2)は，各候補地 j が正確に 1 つの工場に割り当てられることを表す．制約(3)は，どれかの顧客が m ヶ所の範囲内で I に割り当てられるなら，工場を I に配置するよう強制する．この定式化で，この問題を解くと，こんな簡単な問題でもあきれるほどの計算時間がかかる．どこに問題があるかというと，制約(5)と(6)を除いて LP で解くと，多くの変数に端数（非整数）がでて，最適整数解とは似ても似つかぬ物が得られる．この種の条件下では，まずほとんどの整数解探索は，思うようにいかない．そこで「きつい」定式化で LP 解が整数解になりやすいようにするために，上記の制約（3）を次の制約で置き換える．

$$x_{Ij} \le y_I \quad \text{for } I = 1 \text{ to } n,\ j = 1 \text{ to } m. \qquad (3')$$

一見（3）式を(3’)で置き換えるのは，あまり意味がないように見える．いま，20 の工場候補地と 60 カ所の顧客地があるとすると，(3)式なら 20 個の制約であるのに，(3’)式になると，20*60=1200 個の制約になる．しかし，LP としては(3)式よりも(3’)式の方が自然に整数に成りやすいことが経験的に知られている．

### 11.2.5　容量制約のある工場配置問題

すでに説明した SPL 問題の仮定を変更して，各工場で処理できる需要量に限度があると，CPL(Capacitated Plant Location)問題となる．すなわち，CPL では各顧客の需要量が分かっていて，各工場候補地で対応できる総供給量に限度があると仮定する．次のようなパラメータを追加して，定義する必要がある．

**$D_j$＝顧客地 j の需要量**

**$K_I$＝候補地 I に開設する工場の供給容量**

IP モデルは次の通りである：

$$\text{Min} \quad \sum_{i=1}^{n} f_I y_I + \sum_{i=1}^{n} \sum_{j=1}^{m} c_{Ij} x_{Ij} \qquad (7)$$

$$\text{st} \quad \sum_{i=1}^{n} x_{Ij} = 1 \quad \text{for } j = 1 \text{ to } m \qquad (8)$$

$$\sum_{j=1}^{m} D_j x_{Ij} \le K_I y_I \quad \text{for } I = 1 \text{ to } n \qquad (9)$$

$$x_{Ij} \le y_I \quad \text{for } I = 1 \text{ to } n,\ j = 1 \text{ to } m. \qquad (10)$$
$$y_I = 0 \text{ or } 1 \quad \text{for } I = 1 \text{ to } n \qquad (11)$$
$$x_{Ij} = 0 \text{ or } 1 \quad \text{for } I = 1 \text{ to } n,\ j = 1 \text{ to } m. \qquad (12)$$

これは，「単一供給源型」の問題といわれる．変数 $x_{Ij}$ が，厳密に 0 か 1 に限定しているので，各顧客地はその全需要量を単一の工場から供給してもらわねばならない．「分割供給」を認めるなら，変数 $x_{Ij}$ は端数値になり，その解釈としては数値 $x_{Ij}$ は顧客地 j が工場 I から受け取る量の比率である．この場合，条件(12)がなくなる．

分割調達は通常望ましくない．例えば同じ小学校に通った生徒は同じ中学校に通うことを好むので，小学生を複数の中学校に割り当てることは好ましくない．

**工場配置問題の例**

カリフォルニアにある Zzyzx 社は，現在次の都市に倉庫を持っている．(A)ボルチモア，(B)チェイエネ，(C)ソルトレイクシティー，(D)メンフィス，(E)ウィチタ．これらの倉庫から，全米の顧客地域に商品を供給している．各地の顧客をまとめて次の諸都市に顧客地があると考えると都合がよい．(1)アトランタ，(2)ボストン，(3)シカゴ，(4)デンバー，(5)オマハ，(6)ポートランド．

いま，Zzyzx 社の倉庫が多いことが問題になっている．倉庫をいくつか閉鎖して，輸送費やサービス費を増やさないで，固定費を削減したい．月あたりの関連データは，以下のように整理されている．

| 費用(月，トンあたり) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 需要都市 | | | | | | 月供給容量 | 月固定費 |
| 供給都市 | 1 | 2 | 3 | 4 | 5 | 6 | | |
| A | $1,675 | $400 | $685 | $1,630 | $1,160 | $2,800 | 18 | $7,650 |
| B | 1460 | 1940 | 970 | 100 | 495 | 1200 | 24 | 3,500 |
| C | 1925 | 2400 | 1425 | 500 | 950 | 800 | 27 | 3,500 |
| D | 380 | 1355 | 543 | 1045 | 665 | 2321 | 22 | 4,100 |
| E | 922 | 1646 | 700 | 508 | 311 | 1797 | 31 | 2,200 |
| 月需要容量(トン) | 10 | 8 | 12 | 6 | 7 | 11 | | |

例えば，A（ボルチモア）の倉庫を閉鎖すると，毎月の固定費$7,650 を削減できる．もし，5（オマハ）の月需要の全てを E（ウイチタ）から受けとると，オマハに供給する輸送費は 7×311 =$2177 である．どの顧客の需要も，その全てを単一の供給地からまかなう必要はない．この種の「供給複線化」は，各倉庫の容量が限られていることから生じることが多い．例えば，チェイネは月当り 24 トンしか供給できない．Zzyzx 社は，どの工場を閉鎖すべきであろうか．

この問題を解く次の 4 つの方法の実績を比較する．

①「ゆるい IP」定式化，

②「きつい IP」定式化，

③欲張って開設していくヒューリスティック法：どの倉庫も開設していない状態からスタートして，最大費用削減をもたらす倉庫から順次開設していく．そして，もうこれ以上開設する意味がなくなったとき終了する．
④欲張りに閉鎖していくヒューリスティック法：全ての倉庫が開設されている状態からスタートして，最大削減をもたらすように，閉鎖倉庫を選ぶ．もうこれ以上費用削減をもたらさないときに終了する．

ヒューリスティック法の利点は，応用しやすい点にある．この 4 つの方法のパフォーマンスの比較を以下に示す．

| 方法 | 最良解 | 計算時間（秒） | 開設地 | LP による費用 |
|---|---|---|---|---|
| 「ゆるい IP」 | 46，031 | 3．38 | A，B，D | 35，662 |
| 「きつい IP」 | 46，031 | 1．67 | A，B，D | 46，031 |
| 開設していく<br>ヒューリスティック | 46，943 | nIl | A，B，D，E | - |
| 閉鎖していく<br>ヒューリスティック | 46，443 | nIl | A，C，D，E | - |

この比較で注目すべきは，「ゆるい IP」と「きつい IP」は，ともに同じ最適解を得るが，計算時間は「ゆるい IP」はほぼ 2 倍の時間がかかる点である．問題規模が大きくなると，この違いは劇的なものになる．

単一製品動的ロットサイズ問題は，次のようなパラメータで表される．

$n$ ＝ある製品の生産計画に対する期間の数

$D_j$＝期間 j における予想需要（ただし，j ＝1，2，．．．，n）

$f_I$＝期間 I において生産するときの固定費

$h_I$＝期間 I から期間（I＋1）に繰り越される製品 1 台当りの費用

もし次のように定義すると，この問題は簡単な工場配置問題として計算できる．

$$c_{Ij} = D_j \sum_{t=i}^{j-1} h_t .$$

すなわち $c_{Ij}$ は，期間 I の生産が期間 j の要求を満たす費用である．各期は，可能性のある工場サイトと顧客の両方で考えることができる．

さらに，もし期間 I で生産能力の上限 $K_I$ があるならば，この容量制限のある動的ロットサイズ問題は，制限のある工場配置問題の特別なケースになる．

**IP における双対価格と減少費用**

IP の双対価格と減少費用は，解釈に制限がでてくる．IP の初心者は，それを単に無視するのが最善である．興味のある方は，双対価格と減少費用の解釈は，整数

変数を最適解の値に固定し，それらをモデルから省いて LP モデルで考えればよい．従って，純粋な IP プログラム（全ての変数が整数）は，次の通りである：

・全ての双対価格がゼロである

・減少費用は，単にその変数の目的関数の係数（MAX 問題では符号が逆）である

混合整数プログラムの双対価格は興味深い．例えば，工場の立地問題は，どこに工場を立地するかは整数変数で，出荷量は連続量になる．このモデルの双対価格は，工場をどこに立地するかあらかじめ指定した後，輸送量の決定モデルから導かれたものになる．

## 11.2.6　　シナリオのあるモデルの代替アプローチ

私たちは，代替案に関して 2 つの異なる方法に直面する：①2 つ以上の代替案を選択し，どれが最善かを決める，または②自然や市場が 2 つ以上の代替案のいずれかを選択し，私たちは自然がどの代替案を選んだか分からないので，全ての代替案に対して最適策を準備し，自然が選んだ代替案が分かった後で対応する．ここでは，①だけを検討する．これを，シナリオ・アプローチや分離定式化（Disjunctive formulation）という．Balas（1979），または Martin（1999）の 16.2.3 節を参照．私たちは代替案を無視すると仮定した場合，変数は単に $x_1, x_2, \cdots, x_n$ とする．代替案 s を選ぶ条件をシナリオ s と呼ぶ．一般性を失わず，全変数を非負とする．個別の決定問題を定式化するシナリオアプローチは，次のようになる．

各シナリオに対して：

1）シナリオ s が選ばれたら関係する制約を書き出す．

2）これらの制約の全変数に s を追加し，他のシナリオの同じ変数と区別する．すなわち，シナリオ s の $x_j$ を $x_{sj}$ にする．

3）0/1 変数 $y_s$ を追加する．$y_s$ はシナリオ s が選ばれればモデルで $y_s=1$ に，選ばれなければ $y_s=0$ にする．

4）シナリオ s の各制約の右辺定数項に $y_s$ を掛ける．

5）シナリオ s の各制約の各変数 $x_{sj}$ に，$x_{sj} \le M * y_s$（M は大きな正の定数）を追加する．これで各変数 $x_{sj}$ は，シナリオ s が選ばれなければ強制的に 0 になる．

最後に $\sum_s y_s=1$ でもって全シナリオを一つにまとめ，シナリオのいずれかを選択する．各変数 $x_j$ に，$x_j=\sum_s x_{sj}$ を追加し，xj が選ばれたシナリオの適切な値になるようにする．例えば，ステップ 1 の直後に，$\sum_j a_{sj} * x_j \le a_{s0}$ の制約式があれば，ステップ 2-4 で $\sum_j a_{sj} * x_{sj} \le a_{s0} * y_s$ に変形する．もし，$y_s=0$ は $x_{sj}=0$ を意味するので（もし全ての $a_{sj}$ が非負で，$x_j$ が非負に制約されているなら），ステップ 5 の制約を強制する必要がない．

シナリオアプローチに似たやり方に，Adams & Sherali（2005）の RLT アプローチがある．次の節で，シナリオアプローチを紹介する．

## 11.2.7　区間線形関数による線形化

もし業者に見積もりを請求すると，**図 11.2** のような数量割引が提案されることがある．

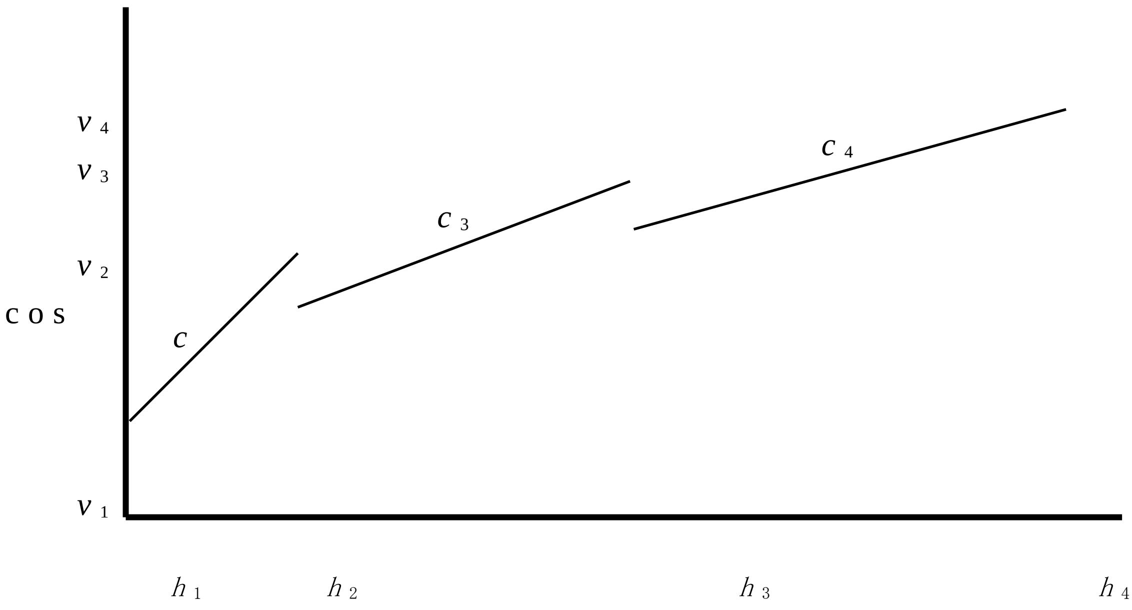


**図 11.2 区間線形費用曲線による数量割引**

定義： $h_s$, $v_s$ = 区間線形関数の区間 s の上限の座標値

$c_s$ = 区間 *s の傾き，*

**注**： 区間 1 は何も買わない区間である．この例では，区間線形関数は連続を仮定していない．どの区間でも基本的に$50を支払い，購入量に比例し1単位当たり次を支払う．

$2.00/単位 If 購入量 < 100,

$1.90/単位 If 100≤購入量< 1000,

$1.80/単位 If 1000≤購入量≤ 5000.

*$h_s$, $v_s$ , $c_s$ を定数とし， $h_s \le h_{s+1}$ を満たす：*

| h | v | c |
|---|---|---|
| *0* | *0* | *0* |
| *100* | *250* | *2* |
| *1000* | *1950* | *1.90* |
| *5000* | *9050* | *1.80;* |

$x$ は購入量を示す．前述のステップ 1 を用い区間（シナリオ 1）を選ぶと，

$cost = 0;$

$x = 0;$

区間/シナリオ 2 を選ぶと,

$\boldsymbol{cost = v_2 - c_2*(h_2 - x);}$ **[or 250- 2*(100-$x$)],**

$\boldsymbol{x \le h_2;}$ **[or $x \le 100$],**

$\boldsymbol{x \ge h_1;}$ **[or $x \ge 0$],**

同じ制約式が シナリオ(区間3と4)に適用される. $x = 99.44$ のような実数の場合, $x \le 100$ よりも$x \le 99$ を用いる.

ステップ 2-4 を適用する:

区間(シナリオ 1)が選ばれると

$cost_1 = 0;$

$x_1 = 0;$

区間(シナリオ 2)が選ばれると

$cost_2 = v_2 * y_2 - c_2 * h_2 * y_2 + c_2 * x_2;$ [or $cost_2 = 50 * y_2 + 2 * x_2$],

$x_2 \le h_2 * y_2;$ [or $x_2 \le 100 * y_2$],

$x_2 \ge h_1 * y_2;$ [or $x_2 \ge 0 * y_2$],

区間(シナリオ 3)が選ばれると

$cost_3 = v_3 * y_3 - c_3 * h_3 * y_3 + c_3 * x_3;$ [or$cost_3 = 50 * y_3 + 1.9 * x_3$],

$x_3 \le h_3 * y_3;$ [or $x_3 \le 1000 * y_3$],

$x_3 \ge h_2 * y_3;$ [or $x_3 \ge 100 * y_3$],

区間(シナリオ 4)が選ばれると

$cost_4 = v_4 * y_4 - c_4 * h_4 * y_4 + c_4 * x_4$ [or$cost_4 = 50 * y_3 + 1.8 * x_4$]

$x_4 \le h_4 * y_4;$ [or $x_3 \le 5000 * y_4$],

$x_4 \ge h_3 * y_4;$ [or $x_3 \ge 1000 * y_4$],

4つの区間を選ぶために

$y_1 + y_2 + y_3 + y_4 = 1;$

$y_1, y_2, y_3, y_4 = 0$ or $1;$

真の購入量と費用は次の通りである:

$x_1 + x_2 + x_3 + x_4 = x;$

$cost_1 + cost_2 + cost_3 + cost_4 = cost;$

例: よくある問題として, 数量割引をする供給業者に対して発注量を決める問題がある. 例えば, 発注する固定費として 150 ドルかかる. 製品 1000 ガロンまでは, ガロン当り 2 ドルで購入できる. 1000 ガロンを越えるとガロン当り 1.90 ドルに価格が下がる. そして, 5000 ガロンまでしか購入できない.

ここでパラメータは, 次のように計算できる.:

*$y_0 = 1$ (購入量が 0), 0(それ以外);*

$y_1$ = 1 (0<購入量 < 1000), 0(それ以外);,

$y_2$ = 1 (1000≤購入量), 0(それ以外);

$x_1$ = 1,000 ガロン未満の購入量;

$x_2$ = 1,000 ガロン以上の購入量;

モデルは次のとおりである:

```
MIN = cost + ...;
!s.t.;
! 区間/シナリオ 1;
   cost1 = 150*y1 + 2*x1;
   x1 <= 1000*y1;
! 区間/シナリオ 2;
   cost2 = (150+(2.0-1.9)*1000)*y2+1.9*x2;
   x2 >= 1000*y2;
   x2 <= 5000*y2;
!シナリオ/区間を選ぶ;
  y0+y1+y2=1;
 cost1+cost2=cost;
    x1+x2=x;
@BIN(y1); @BIN(y2);
! モデルを完成するため,他の制約と変数を加える;
```

### 11.2.8 分離可能な関数に変換

前の方法は,1 変数の非線形関数にだけ適用できる.多変数の関数を変形するために利用できる標準的な方法がある.従って,関数は変形された変数で分離可能になる.例えば次の 2 つの変数の積($x_1 * x_2$)を,分離してみよう.

次の線形制約式を考える:

$$y_1 = (x_1 + x_2)/2$$

$$y_2 = (x_1 - x_2)/2$$

そして $x_1 * x_2$ のすべてを $y_1^2 - y_2^2$ で置き替える($x_1 * x_2 = y_1^2 - y_2^2$).

これは次のように証明される.

$$y_1^2 - y_2^2 = (x_1^2 + 2 * x_1 * x_2 + x_2^2)/4 - (x_1^2 - 2 * x_1 * x_2 + x_2^2)/4$$

$$= 4 * x_1 * x_2 / 4 = x_1 * x_2$$

この例は,2 つの変数の積は,2 つの新しい変数を加え,その変数の平方で置き換えることができる.n 個の変数で n(n-1)/2 個の積を考えることができる.この積を置き換えるのに,n(n-1)個の新しい変数が必要になる.ある特定の条件の下で,上記の考えは(Cholesky 分解を使用して)n 個の新しい変数が必要になる.

## 11.3　整数計画解法の概要

IP を解く計算時間は，定式化の仕方で大きく異なる．そこで，IP の解法を知っておくことが役に立つ．解法としては，「切除平面法」と「分岐限定法（Branch-and-Bound Method，**B&B 法**）」がある．IP の一般的な解法に関しては，Nemhauser & Wolsey ら(1988)，Wolsey(1998)を参照．通常の IP は，いわゆる B&B 法を採用しているが，切除平面法も一部取り入れている．B&B 法は，平たく言えば知的列挙法とでも言うべきものである．B&B 法は，問題をまず LP として解く．もしその解が整数でないなら，それらの非整数変数を丸めるやり方の全てを知的に探索する．次の問題を例にして，分岐限定法を説明する．

**MAX=75*X1+6*X2+3*X3+33*X4;**

**774*X1+76*X2+22*X3+42*X4<=875;**

**67*X1+27*X2+794*X3+53*X4<=875;**

**@BIN( X1); @BIN( X2); @BIN( X3); @BIN( X4);**

計算機がこの問題の最適整数解を探索する過程を**図 11.4** で示す．まず，この問題を LP として，各制約式が，X1,X2,X3,X4≦1 のもとで解く．その解はラベル 1 の枠に示す．変数 X2 と X3 が整数でないので，これは受け入れられない．ここで，ひとまず X2 を取り上げ，次のような論理を展開する．最適解で，X2 は 0 か 1 でなければならない．そこで，元の問題を 2 つの部分問題に置き換える．その 1 つは，X2 を 1 に固定し（ノード 2 の枠），もう 1 つは x2 を 0 に固定する（ノード 8）．この 2 つの新しい IP を解いて，そのうちの良い方の解が元の問題の満足解に違いないと考える．この推論方法が，「分岐」という用語を用いている理由である．部分問題の生成は，図の分岐に対応している．

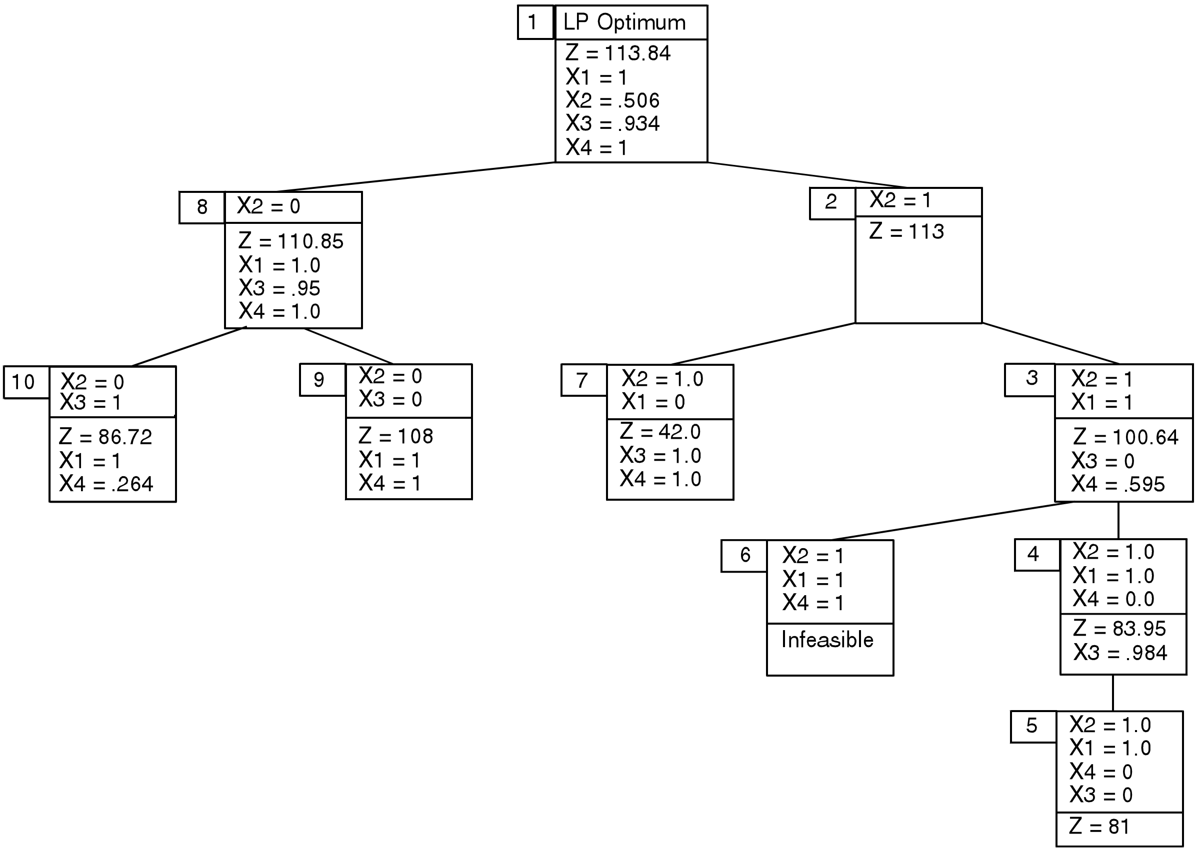


図 11.4　分岐限定探索木

各ノードの左上の数字が，部分問題を調べた順序を表わしている．変数 Z は目的関数値である．X2 を 1 に固定した部分（ノード 2）を LP で解くと，X1 と X3 が非整数になる．今度は X1 で上述の議論をして，更に 2 つの部分問題を作る．1 つは X1 を 0 に固定し（ノード 7），もう 1 つは X1 を 1 に固定する（ノード 3）．この過程を X4 と X3 で繰り返し，ノード 5 に至る．この時点で整数解 z=81 が得られる．しかし部分問題 6 から 10 までを見ないと，これが最適解かどうかは分からない．部分問題 6 は，X2，X1 および X4 をすべて 1 に設定すると可能解ではないので，もうこれ以上探索する必要はない．部分問題 7 も，これ以上探索する必要はない．なぜなら Z が 42 という値は，すでに得られている整数解よりも悪いからである．

ノード 9 で X3=0 にすると，Z=108 というさらによい解が得られた．ノード 10 は，分岐限定法の「限定」の方の意味を説明している．この解は整数ではないが，もうこれ以上探索する必要はない．というのは，Z=86.72 という解は，すでに得られている 108 という整数解よりも悪いからである．各ノードの Z の値は，その下にある子ノードの Z の値の上限を与えている．なぜなら，子ノードの部分問題は，すべてその上の親ノードの問題に更に制約を追加して得られる．制約を追加すれば，可能解集合は更に小さくなる．すなわち，探索樹木の下方に下がるに連れて，Z の値が改善されることはない．図の樹木は，探索の進め方の 1 つを表わしている．同じ問題で，次のように自由に選択できるもののうち，どれから進めるかで，異なる樹木で探索することになる．

（a）次に調べるノードの選択
（b）どの変数を使って枝別れするかの選択

例えば，もしノード 1 の後，直ちにノード 8 と 9 を調べると，Z=108 の解が早い段階で見つかった．さらにノード 4，5，6 は調べるのを省略できたはずである．なぜなら，ノード 3 の Z＝100.64 は，すでに見つかっている整数解（108）よりも悪いので，ノード 3 の子ノードを調べる必要がない．上記の例に示す樹木では，最初のノードは変数 X2 のとりうる値で枝別れしている．しかし，最初の枝別れ変数として，X3 あるいは X1 を選んでもよかったはずである．

探索の効率は，上記（a）と（b）の選択をいかに賢明にやるかに係わっている．（b）の分割は，1 変数の分岐が行われる．LP 解で $x = 1.6$ が得られると，$x \le 1$ と $x \ge 2$ の 2 つの部分問題に分岐する．分岐は，1 変数だけでなく任意の制約式でも良い．例えば，部分問題 $x_1+x_2+x_3 \le 0$ と $x_1+x_2+x_3 \ge 1$ を考えても良い．また，分岐は 2 値でなくてもよい．例えば $y_1+y_2+y_3=1$ の制約で，$y_1=1$，$y_2=1$，または $y_3=1$ という 3 つの部分問題に分岐してもよい．

（b）の選択では，「決め手になる」よい変数を選ぶのが望ましい．一般には LINGO がよい選択をするので，ユーザが探索過程の詳細を意識する必要はない．しかし，モデル選択で，一般的な分岐限定過程を念頭にいれておかなくてはならない．もしユーザが，ある整数変数 X が「決め手になる」ことを予め知っていたなら，その変数 X を定式化の最初の方に位置させて，その重要性を示せば，LINGO にとって有益な情報になる．このような一般的な理解にたつと，「きつい（tight）」LP 定式化の重要性が了解されよう．「きつい」LP 定式化とは，それを解いたときの目的関数値が IP による最適値に近いもののことである．部分問題の LP 解は，その下のノードの探索を打ち切る限界値として使われる．その限界値が貧弱であると，樹木の初期ノードをたくさん明示的に列挙して調べ上げねばならない．なぜなら，その限界値よりもよい値の子ノードが，その下にたくさんあるので，探索を打ち切ることができないからである．

### 11.4　IP の計算上の難しさ

IP を解くことは，難しい問題である．このことは，LP と対照的である．LP の場合，計算時間は変数の個数に比例し，また制約式の数の自乗に比例する．しかし IP の場合は，制約式が増えると計算時間は減るかもしれない．整数変数の数が増えると，計算時間は比較にならない早さで劇的に増大する．わずか 60 変数，6 制約の IP でも，それを解くのが困難な場合もある．

筆者注：次の問題を簡単に解けない商用ソフトも多い．

```
TITLE NO_TITLE;
[_1]MIN=-81*X_1-221*X_2-219*X_3-317*X_4
-385*X_5-413*X_6;
```

```
[_2]12228*X_1+36679*X_2+36682*X_3+48908*X_4
+61139*X_5+73365*X_6=89716837;
@GIN(X_1);@GIN(X_2);@GIN(X_3);@GIN(X_4);@GIN(X_5);
@GIN(X_6);
@BND(0,X_1,99999);@BND(0,X_2,99999);
@BND(0,X_3,99999);@BND(0,X_4,99999);
@BND(0,X_5,99999);@BND(0,X_6,99999);
```

LINGO は 1 秒以内で解いて，解は次のとおりである．

```
Global optimal solution found at iteration:0
Objective value:-540564.0
   Model Title: NO_TITLE
Variable              Value          Reduced Cost
     X_1           0.000000             -81.00000
     X_2           2445.000             -221.0000
     X_3           1.000000             -219.0000
     X_4           0.000000             -317.0000
     X_5           0.000000             -385.0000
     X_6           0.000000             -413.0000
Row      Slack or Surplus        Dual Price
 _1         -540564.0             -1.000000
 _2          0.000000              0.000000
```

LP と同様に，与えられた問題で代替的な IP による複数の定式化がありうる．しかし，IP の場合は，解けるか解けないかは，それをどう定式化するかが決定的な決め手になることが多い．良い定式化を行なうには，熟練が必要である．この章でこれから述べる多くの問題では，良い定式化をするかしないかが，解けるか解けないかの分かれ目となる．

### 11.4.1 NP 完全問題

IP は，NP 完全問題に属する．私達は NP を「多項式オーダーでない」と幾分緩く考えるかもしれない．これは計算が問題のサイズを多項式オーダーで解くアルゴリズムが知られていないことを意味する．サイズ n の問題を解く時間が，K をある正の整数として $n^k$ に比例している場合をいう．例えば，一組の n 個の数を分類することは，$n^2$ に比例した多項式時間（注意ぶかくすれば n*log(n)）になる．

しかし，n 個の 0/1 をもつ IP は，最悪の場合で $2^n$ に比例した指数時間になる．あらゆる問題を多項式時間で解くことを保証する IP アルゴリズムは提案されていない．NP 完全で P 完全は，「yes/no」で実行可能な問題である．yes/no の最適化問題の変形は次のような問題である．1250 以下の費用をもつこの問題に実行可能な

解があるだろうか.最適化問題で，私達は最低の費用で実行可能解がほしい.LP の実行可能解がある問題は，多項式時間で解答可能であることを Khachian (1979) は示した.LP は P 完全になる.実行可能な IP は，NP 完全である.これらの問題のあるものは，問題のサイズが多項式である他の NP 完全な問題に変えることが可能である.従って私達が問題 A を多項式時間の問題 B に変えて，多項式時間で B を解けば，多項式時間で A を解いたことになる. NP 完全問題は，誰かがこれらの問題の 1 つを解決するための速い(例えば，最悪の場合に多項式オーダーで)方法を開発すれば，その他全ての NP 完全問題の多項式時間アルゴリズムがあることになる.NP 完全性は，最悪の場合であり平均的なことではない.実用的な目的では，平均的な行動に主に興味がある.現状は多くの重要で実用的な IP 問題を解決する平均時間はかなり短いことである.誰かが時折非常に困難な IP 問題を示すかもしれないが，私達が多くの実用的な IP を急速に解決できるということを妨げない.多分現代数学の最も大きい未解決の問題は，NP 完全なクラスの問題が本来困難であるかどうかである.この質問は「P=NP か」のようにいい表わされる.これらの問題は実に困難であるか，または一般に速いアルゴリズムを発見するほど私達がスマートでないかのいずれかである.事実，Clay Mathematics Institute (www.claymath.org) は，100 万$の世紀の賞金を出している.NP 完全の分類に関する広範囲の議論は，Martin (1999)を参照.

**11.5 自然に整数解が得られるアルゴリズムの問題**

IP の解法は，LP として整数制約を無視して解いて，解が自然に整数になることを祈ることである.例えば，x が 0/1 の整数変数の場合，この制約を $0 \le x \le 1$ に置き換えて最初に解く.IP 問題の分析は，解決し易い IP 問題かどうか知ることが重要である.一般に LP 解と IP 解の目的関数の値が近かったら，IP の計算は容易である.この容易さは，LP 解のほとんが自然に整数になることである.従って，私達はどのような LP が自然に整数解が得られるか，知っていることが重要である.

自然に整数解が得られる LP 問題のクラスは，次のとおりである：

①ネットワーク型の LP,

②MRP か LeontIef 型の LP,

③行操作か双対を取ることで(a)か(b)に変形できる問題.

最初，ネットワークおよび MRP 型の LP の特徴を見てみることにする.

**11.5.1 ネーートワーク型 LP 再考**

①単純な上下限制約(例えば $x \le 3$)を無視して，各変数が高々2 つの制約式に現れ，②その係数が+1 と-1 の場合，ネットワーク型の LP といわれる.あるいは変数が 1 つの制約に現れる場合，係数が+1 か-1 の場合である.

*結論：* 右辺定数項が整数の場合，整数解がある.もし目的関数の係数が全て整数なら，整数の双対価格を持つ最適解がある.

**11.5.2 整数 LeontIef 制約式**

もし次の条件を満たすと，Leontief 制約あるいは MRP(Material Requirements Planning)という(Jeroslow,Martin,et al. (1992)).

・各制約式は等式である，

・各列は 1 つの+1 の係数をもつ，

・各列は 0 または負の整数値をもつ.

・右辺定数項は，非負の整数である.

**結果：**制約式が MRP を満たす LP は，整数の最適解になる.目的関数の係数がすべて整数なら，整数の双対価格になる最適解がある.

### 11.5.2　例： 1 期間 MRP 問題

ある自転車会社は，次の 3 つの製品を作る： UnIcycles (U)，普通自転車(r)，

およびTwInbIkes (t).各製品は色々な部品から組み立てられる： 座席(s)，車輪(w)，ハブ(h)，スポーク(P)，鎖(c)，およびリンク(l).各製品の完全な資材

表は次に示されている.括弧内の数字は，子ノードの何単位が親ノードの 1 個あたりに要求されるかを表す.

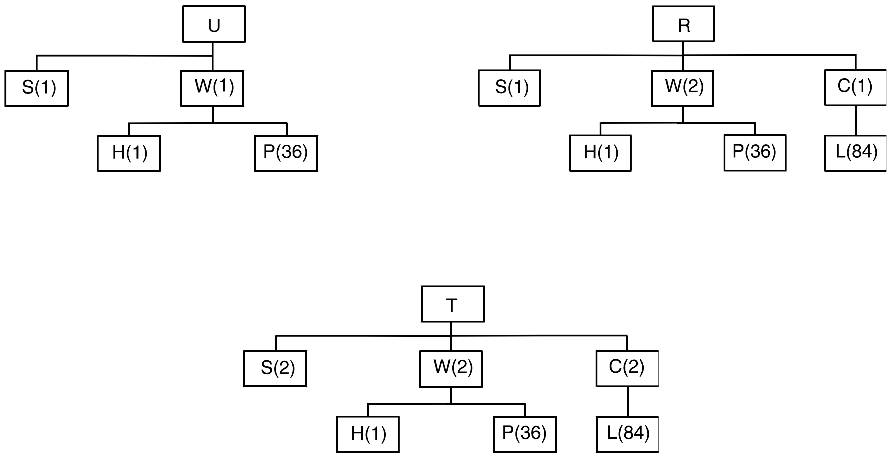


**図 11.5 自転車の MRP 構造**

現在の在庫はゼロであるが，100 台の Unicycles，500 台の普通自転車，200 台の Twinbikes を供給する必要がある.完成品および組立部品は，次の価格で製造されるか買うことができる：

| 項目： | U | R | T | S | W | C | H | P | L |
|---|---|---|---|---|---|---|---|---|---|
| 購入価格： | 2.6 | 5.2 | 3.1 | 0.25 | 1.4 | 0.96 | 0.19 | 0.07 | 0.05 |
| 組立費： | 1.04 | 1.16 | 1.9 | 0.2 | 0.22 | 0.26 | 0.16 | 0.04 | 0.03 |

組立費は，組み立て時に必要な費用である.組み立てに使われる部品の費用を含んでいない.需要を満たし，各項目を何単位購入すれば最小費用になるだろうか.LPモデルは次の通りである.

```
MODEL:
SETS:
TYPES/U, R, T/:M, B, MP, BP, NEED;
MATERIALS/S, W, C/:MM, MB, MMP, MBP;
SUBMATS/H, P, L/:SMM, SMB, SMP, SBP;
REQ(TYPES, MATERIALS): MATREQ;
MREQ(MATERIALS, SUBMATS): SMATREQ;
ENDSETS
DATA:
NEED=  100  500  200;!供給量;
MP  = 1.04 1.16  1.9;!組み立て価格;
BP  =  2.6  5.2  3.1;!購入価格;
MMP =   .2  .22  .26;!スポーク，車輪，チェーンの組み立て価格;
MBP =  .25  1.4  .96;!スポーク，車輪，チェーンの購入価格;
SMP =  .16  .04  .03;!ハブ，スポーク，リンクの組み立て価格;
SBP =  .19  .07  .05;!ハブ，スポーク，リンクの購入価格;
MATREQ  =1    1    0
         1    2    1
         2    2    2;
SMATREQ =0    0    0
         1   36    0
         0    0   84;
ENDDATA
MIN = @SUM(TYPES : M * MP + B * BP)
     + @SUM(MATERIALS : MM * MMP + MB * MBP)
     + @SUM(SUBMATS: SMM * SMP + SMB * SBP);
@FOR(TYPES: M + B = NEED);
@FOR(MATERIALS(I): MM(I) + MB(I) =
    @SUM(TYPES(J): M(J) * MATREQ(J, I)));
@FOR(SUBMATS(I): SMM(I) + SMB(I) =
    @SUM(MATERIALS(J): MM(J) * SMATREQ(J, I)));
END
```

訳注:Generate で次のモデルが生成される.

```
MODEL:
[_1]MIN=0.16*SMM_H+0.19*SMB_H+0.04*SMM_P+0.07000000000000001*
SMB_P+0.03*SMM_L+0.05*SMB_L+0.2*MM_S+0.25*MB_S+0.22*MM_W+1.4*
MB_W+0.26*MM_C+0.96*MB_C+1.04*M_U+2.6*B_U+1.16*M_R+5.2*B_R
+1.9*M_T+3.1*B_T;
[_2]M_U+B_U=100;
[_3]M_R+B_R=500;
[_4]M_T+B_T=200;
[_5]MM_S+MB_S-M_U-M_R-2*M_T=0;
[_6]MM_W+MB_W-M_U-2*M_R-2*M_T=0;
[_7]MM_C+MB_C-M_R-2*M_T=0;
[_8]SMM_H+SMB_H-MM_W=0;
[_9]SMM_P+SMB_P-36*MM_W=0;
[_10]SMM_L+SMB_L-84*MM_C=0;
END
```

次の図で，MPR の特徴を満たすことを確認しよう．

```
          U  U  R  R  T  T  S  S  W  W  C  C  H  H  P  P  L  L
          M  B  M  B  M  B  M  B  M  B  M  B  M  B  M  B  M  B
       1: A  A  A  A  A  A  T  T  T  A  T  T  T  T  U  U  U  U  MIN
UNICYCLE: 1  1     '        '        '        '        '        =  B
 REGULAR: '   ' 1  1   '    '   '    '   '    '   '    '   '    =  C
TWINBIKE:          '  1  1  '        '                          =  C
   SEATS:-1    -1  ' -2     1  1     '        '        '        =
  WHEELS:-1    -2  ' -2'    '   ' 1  1   '    '   '    '   '    =
  CHAINS:      -1  ' -2     '        '  1  1  '        '        =
    HUBS:          '        '    -1  '        1  1     '        =
  SPOKES: '   '    '   '    '    -B  '   '    '   ' 1  1   '    =
   LINKS:          '        '        ' -B     '        '  1  1  =
```

解は次のとおりである．

```
Optimal solution found at step:          0
Objective value:                  3440.000
Variable           Value        Reduced Cost
   M( R)        500.0000           0.0000000
   B( U)        100.0000           0.0000000
   B( T)        200.0000           0.0000000
  MM( S)        500.0000           0.0000000
```

| | | |
|---|---|---|
| MB( W) | 1000.000 | 0.0000000 |
| MB( C) | 500.0000 | 0.0000000 |

解は自然に整数である.Unicycles と Twinbike を全て OEM で購入する.普通自転車は，製造する座席と購入する車輪および鎖で組み立てる.もし LM≤300 を加えて製造するリンクの上限を 300 すれば，MRP の条件に反するので実数解を得る.

### 11.5.3　自然に整数解になる定式への変形

行操作は次のどちらである:

- 非ゼロの定数を式に掛ける,
- 1つの式を何倍かしたものを他の式に加える.

行操作は問題の実行可能領域も最適解も変えない.従って，行操作でネットワーク LP か MRP 型の LP に変形できれば，整数解になる.私達は実際に整数解を得るために変形をする必要はない.モデルが整数の右辺定数項を持ち，ネットワーク LP か MRP 型の LP であることを示せたら，整数解をもつ.

**例：**次の LP を考えよう:

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | P | P | P | P | P | P | P | P | P | P | | |
| | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 4 | | |
| | 4 | 3 | 2 | 1 | 4 | 3 | 2 | 4 | 3 | 4 | | |
| 1: | 9 | 6 | 4 | 3 | 6 | 4 | 3 | 4 | 3 | 3 | MIN | |
| 2: | 1 | 1 | 1 | 1 | | | | | | | = | 1 |
| 3: | 1 | 1 | 1 | | 1 | 1 | 1 | | | | = | 1 |
| 4: | 1 | 1 | | | 1 | 1 | | 1 | 1 | | = | 1 |
| 5: | 1 | | | | 1 | | | 1 | | 1 | = | 1 |

LP として解くと，次の解が得られる:

P12=P34=1; all others 0.

あらかじめ，整数解になることが分かるだろうか？もし(5')=(5)-(4);
(4')=(4)-(3);(3')=(3)-(2)という行操作を行えば，次の LP モデルを得る:

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | P | P | P | P | P | P | P | P | P | P | | |
| | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 4 | | |
| | 4 | 3 | 2 | 1 | 4 | 3 | 2 | 4 | 3 | 4 | | |
| 1: | 9 | 6 | 4 | 3 | 6 | 4 | 3 | 4 | 3 | 3 | MIN | |
| 2: | 1 | 1 | 1 | 1 | | | | | | | = | 1 |
| 3': | | | | -1 | 1 | 1 | 1 | | | | = | 0 |
| 4': | | | -1 | | | | -1 | 1 | 1 | | = | 0 |
| 5': | | -1 | | | | -1 | | | -1 | 1 | = | 0 |

これはネットワーク LP なので，自然な解をもつ.

**例：**　7つの活動で構成されたあるプロジェクトの最小の経過時間を見つけるた

めに次の LP を考える（PICTURE 形式で）:

```
      A  B  C  D  E  F
  1:-1        '     1  MIN
AB:-1  1      '        >=  3
AC:-1    '1   '    '   >=  2
BD:    -1     1        >=  5
BE:    -1     '  1     >=  6
CF: '     -1  '    '1  >=  4
DF:          -1     1  >=  7
EF:           '-1   1  >=  6
```

これはネットワーク型 LP（例えば，列 a，b，または f を考慮しなさい）でも，MRP でもない(例えば，列 a か f を考慮しなさい)．それにもかかわらず，解は自然な整数解を得る．

```
Optimal solution found at step:        0
Objective Value:                 15.00000
Variable           Value        Reduced Cost
       A       0.0000000           0.0000000
       B        3.000000           0.0000000
       C        2.000000           0.0000000
       D        8.000000           0.0000000
       E        9.000000           0.0000000
       F        15.00000           0.0000000
     Row    Slack or Surplus      Dual Price
       1        15.00000            1.000000
      AB       0.0000000           -1.000000
      AC       0.0000000           0.0000000
      BD       0.0000000           -1.000000
      BE       0.0000000           0.0000000
      CF        9.000000           0.0000000
      DF       0.0000000           -1.000000
      EF       0.0000000           0.0000000
```

私達はあらかじめ自然な整数解を予測できただろうか？モデルの PICTURE を見れば，各制約式に 1 つの+1 と 1 がある．従って，双対モデルはネットワーク LP であり，整数解になる．

## 11.6　割り当て問題および関連づけられた順序または経路問題

割り当て問題は，より複雑で実用的な問題の主要な要素として頻繁に見られる

簡単なLP問題である．割当問題は次のとおりである：

費用行列：　$c_{Ij}$ =対象Iに人jを割り当てる費用，

変数：　$x_{Ij}$ = 1（対象Iが人jに割り当てられる）．

次のモデルになる：

Minimize $\Sigma_I \Sigma_j c_{Ij} x_{Ij}$

subject to

$\sum_I x_{Ij}$ = 1　for each object $I$,

$\sum_j x_{Ij}$ = 1　for each person $I$,

$x_{Ij} \geq 0$.

この問題はLPで解くと，$x_{Ij}$ は自然に整数になる．割り当て問題に似ている経路や順序問題が数多くある．

**例：　割当の問題**

大きな航空会社は，ハブ空港で自社の運行計画を考える傾向がある．航空会社は，多数の飛行機をハブ空港にある時間帯（例えば，9AM から 10AM）に到着させ，特定の短い時間帯で離陸させたい（例えば，10AM から 11AM）．乗客は，出発地から到着地に行くために，ハブ空港で降りて少なとも 1 回乗り換える数多くの組合せがある．例えば，ユナイテッド航空はハブ空港としてシカゴ／オヘアを，Delta 航空はアトランタを，TWA はセントルイスを，アメリカン航空はダラス/フォート・ワースを使用する．ハブ空港使用の好ましい目的関数は，ハブで飛行機の乗り換え回数（と手荷物の移動量）を最小化することである．次の小さな例は，割り当てモデルをこの問題にいかに適用するかを説明している．

ある航空会社は，6 機の飛行機を 9 時から 9 時半の間オヘア空港に到着させる．同じ 6 機の飛行機は，9:40 と 10:20 の間に異なった目的地へ飛行する．出入りする飛行機の間で，乗り換える乗客の平均人数は下記の通りである．

| | L01 | L02 | L03 | L04 | L05 | L06 | 注 |
|---|---|---|---|---|---|---|---|
| I01 | 20 | 15 | 16 | 5 | 4 | 7 | *フライト I05 は，到着が遅いので L01 に接続できない．同様に，フライト I06 も L01, L02, L03 に接続できない* |
| I02 | 17 | 15 | 33 | 12 | 8 | 6 | |
| I03 | 9 | 12 | 18 | 16 | 30 | 13 | |
| I04 | 12 | 8 | 11 | 27 | 19 | 14 | |
| I05 | 0 | 7 | 10 | 21 | 10 | 32 | |
| I06 | 0 | 0 | 0 | 6 | 11 | 13 | |

決定する問題は，到着便をどの出発便に割り当てるかである．例えば到着便 I02 を出発便 L03 に割り当てると，33 人（と彼らの手荷物）が到着便に残る．どのように飛行機を割り当てれば，乗り換え乗客数と荷物を最小化できるか？

この問題は，割当問題として次のように定義ができる．

$x_{Ij}$ = 1 （到着便 $I$ を出発便 $j$ に割り当てる），
0 （その他の場合）．

目的関数は，飛行機を乗り換えない乗客の最大化で，次のように定式化される．

```
MODEL:   ! 割り当てモデル(ASSIGNMX);
SETS:
 FLIGHT;
 ASSIGN( FLIGHT, FLIGHT): X, CHANGE;
ENDSETS
DATA:
 FLIGHT = 1..6;
! The Value of assigning i to j;
 CHANGE = 20   15   16   5   4   7
          17   15   33  12   8   6
           9   12   18  16  30  13
          12    8   11  27  19  14
        -999    7   10  21  10  32
        -999 -999 -999   6  11  13;
ENDDATA
!--------------------------------;
! 割り当ての値の最大化;
MAX = @SUM(ASSIGN: X * CHANGE);
@FOR( FLIGHT( I):
!  各 I は，J に割り当てられる;
    @SUM( FLIGHT( J): X( I, J)) = 1;
!  各 I は，到着便にゑありあてられる;
    @SUM( FLIGHT( J): X( J, I)) = 1; );
END
```

訳注：Generate で自然表記のモデルを生成する．

```
MODEL:
[_1]MAX=20*X_1_1+15*X_1_2+16*X_1_3+5*X_1_4+4*X_1_5+7*X_1_6+1
7*X_2_1+15*X_2_2+33*X_2_3+12*X_2_4+8*X_2_5+6*X_2_6+9*X_3_1+1
2*X_3_2+18*X_3_3+16*X_3_4+30*X_3_5+13*X_3_6+12*X_4_1+8*X_4_2
+11*X_4_3+27*X_4_4+19*X_4_5+14*X_4_6-999*
X_5_1+7*X_5_2+10*X_5_3+21*X_5_4+10*X_5_5+32*X_5_6-999*X_6_1-
999*X_6_2-999*X_6_3+6*X_6_4+11*X_6_5+13*X_6_6;
[_2]X_1_1+X_1_2+X_1_3+X_1_4+X_1_5+X_1_6=1;
```

```
[_3]X_1_1+X_2_1+X_3_1+X_4_1+X_5_1+X_6_1=1;
[_4]X_2_1+X_2_2+X_2_3+X_2_4+X_2_5+X_2_6=1;
[_5]X_1_2+X_2_2+X_3_2+X_4_2+X_5_2+X_6_2=1;
[_6]X_3_1+X_3_2+X_3_3+X_3_4+X_3_5+X_3_6=1;
[_7]X_1_3+X_2_3+X_3_3+X_4_3+X_5_3+X_6_3=1;
[_8]X_4_1+X_4_2+X_4_3+X_4_4+X_4_5+X_4_6=1;
[_9]X_1_4+X_2_4+X_3_4+X_4_4+X_5_4+X_6_4=1;
[_10]X_5_1+X_5_2+X_5_3+X_5_4+X_5_5+X_5_6=1;
[_11]X_1_5+X_2_5+X_3_5+X_4_5+X_5_5+X_6_5=1;
[_12]X_6_1+X_6_2+X_6_3+X_6_4+X_6_5+X_6_6=1;
[_13]X_1_6+X_2_6+X_3_6+X_4_6+X_5_6+X_6_6=1;
END
```

魅力のない接続は禁止され，解は次のとおりである．

```
Optimal solution found at step:        9
Objective Value:                135.0000
Variable          Value          Reduced Cost
X( 1, 1)        1.000000          0.0000000
X( 2, 3)        1.000000          0.0000000
X( 3, 2)        1.000000          0.0000000
X( 4, 4)        1.000000          0.0000000
X( 5, 6)        1.000000          0.0000000
X( 6, 5)        1.000000          0.0000000
```

101 便と 104 便以外は，102 便と 103 便の間で，105 便と 106 便の間で入れ替えが起きる．解は IP モデルでないのに，どれも自然に整数である．

### 11.6.2 巡回セールスマン問題

有名な最適化問題の 1 つは，巡回セールスマン問題(TSP)である．それは選ばれる割り当てが旅行を構成するという付加的な条件のついた割当問題である．目的関数は移動した総距離を最小にすることである．Lawler 他(1985)は，この魅惑的な問題の力作である．TSP の 1 つの例は，電子回路板の製造で行われる．Danusaputro, Lee & Martin-Vega ら(1990)は，最適な電子回路の穴の空け順，すなわちドリルを最小時間動かして穴を開ける問題を取り上げている．同じような TSP は，部品を自動挿入機械で基盤に挿入する順序を決定したい製造業で起こる．

TSP は次のデータで表される：

$c_{Ij}$ = 都市 $I$ から都市 $j$ へ直接いく費用（例えば，距離）．

解は次の変数で表される：

$y_{Ij}$ = 1 （ $I$から $j$へ直接いく），0(それ以外).
目的関数： MIn $\Sigma_I \Sigma_j\ c_{Ij}\ y_{Ij}$ ；
制約式の指定の幾つかの方法を以下に述べる.

**部分ツアーの削除**

① 各都市 j に丁度一度だけ入る：

$\sum_{i\neq j}^{n} x_{Ij}$ = 1 for j = 1 to n,

②各都市 I を丁度一度だけ出る：

$\sum_{j\neq i}^{n} x_{Ij}$ = 1 for I = 1 to n,

③$y_{Ij}$ = 0 or 1, for I = 1, 2, …, n,
j = 1, 2, …, n, I≠ j：

① 都市 1 を含まない都市の集合 s の全ての部分集合で部分旅行は禁止
：$\Sigma_{I,j\in s}\ x_{Ij}$ ＜ |S| − 1 for 全ての部分集合 S,
|S|は集合 S のサイズ

上記の定式化は，Dantzig, Fulkerson & Johnson ら(1954)による.魅力のない部分旅行削除の特徴は，n 都市あれば約 2n の制約がある.

**負荷の累積的な定式化：**

もし，「Uj を都市 j の旅行の順番」と定義すると，私たちは制約式の数を減らすことができる.同じことであるが，もし各都市が 1 単位の何かを拾い上げる（または配信）ことを要求すると，「Uj は都市 j に立ち寄った後に拾い上げ（または配信）た累積の単位である.制約集合(4)を(5)で置き換える：

⑥ $U_j \geq U_I + 1 - (1 - x_{Ij})n$ for $j$= 2,3,4,… ; $j \neq I$.

制約式⑤によるアプローチは，Miller, Tucker & Zemlin(1960)による.タイプ⑤による制約式は $n^2$ であるが，④は⑤より厳しい.

もし④を用いないと，大規模な計算問題は手に負えないかもしれない.④は膨大な数の制約式があるが，それらのほんの幾つかが最適解の拘束に役立っている.従って，必要に応じて制約式④を追加していく反復アプローチは，驚くほどうまく動作する.Padberg＆Rinaldi ら (1987)は，この反復的なアプローチを使用し，2000 都市の TSP を解決できた.計算時間は大型 PC で数時間だった.

**マルチコモディティフローの定式化：**

前の定式化と類似しているが，各都市は幾つかの商品の 1 単位を必要としている.

次の変数を定義する： $x_{Ijk}$=1(I から j に運ばれる商品の単位，最終的には k に配達される).もし都市 1 から出発し，n 都市があるとすると，制約式④を以下のように置き換える：

For $k$ = 1,2,3,…,$n$：

$\Sigma_{j>1}\ X_{1jk} = 1$; （各単位は出発地から送られる）

$\Sigma_{I\neq k}\ X_{Ikk} = 1$; （各都市 $k$ は商品を手に入れる.）

For $j = 2, 3, \cdots, n$, $k = 1, 2, 3, \cdots, n$, $j\neq k$:

$\Sigma_I X_{Ijk} = \Sigma_{t\neq j}\ X_{jtk}$ （j に入ってくる商品は, j から他の都市 t に出て行く）

この定式化の欠点は，約 $n^3$ 個の制約および変数があることだ. この定式化の注目すべき特徴は，部分旅行削除定式化と同じように厳しい定式化がある点である. これは，Claus(1984)の研究による.

**Heuristics**

実用的な問題では，最適解の代わりに良い満足解で我慢することで，数時間かかる計算時間を数秒や数分で解けることである. TSP 問題の最も一般的なヒューリスティックな利用法は，Lin&Kernighan ら(1973)を参照. このヒューリスティック法は，この旅行で都市の賢い再順序づけで解を改善できる. 実用的な問題で，ヒューリスティックな解は，最適解より 2% 程度悪いだけであった. Bland&Shallcross ら（1989)は，PC 制御機械の操作から起こる 14,464 都市までの問題を記述している. これらの問題で Lin & Kernighan のヒューリスティックな解は，最適解より 1.7%以上悪くなるものはなかった.

**TSP 問題の例：**

P. ローズは現在失業中で，お金をもうけるため次のことを企画した. 彼はナショナル・リーグの観戦旅行に 18 人のグループのガイドをしたい. 彼は余裕がないので，できるだけ旅行費用を安くしたい. 旅行はシンシナチで始まり，シンシナチで終わる. 距離行列は次の通りである：

**解：**距離行列は対称的であることを利用し，部分旅行の削除法を紹介する. 決定変数は次のとおりである：

$Y_{Ij}=1$; （都市 I と j 間にリンクがある），0：（リンクがない）

| | Atl | Chi | Cin | Hou | LA | Mon | NY | Phi | Pit | StL | SD | SF |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Atlan-ta | 0 | 702 | 454 | 842 | 2396 | 1196 | 864 | 772 | 714 | 554 | 2363 | 2679 |
| Chica-go | 702 | 0 | 324 | 1093 | 2136 | 764 | 845 | 764 | 459 | 294 | 2184 | 2187 |
| Cinci. | 454 | 324 | 0 | 1137 | 2180 | 798 | 664 | 572 | 284 | 338 | 2228 | 2463 |
| Hous-ton | 842 | 1093 | 1137 | 0 | 1616 | 1857 | 1706 | 1614 | 1421 | 799 | 1521 | 2021 |
| L.A. | 2396 | 2136 | 2180 | 1616 | 0 | 2900 | 2844 | 2752 | 2464 | 1842 | 95 | 405 |
| Mont-real | 1196 | 764 | 798 | 1857 | 2900 | 0 | 396 | 424 | 514 | 1058 | 2948 | 2951 |
| New York | 864 | 845 | 664 | 1706 | 2844 | 396 | 0 | 92 | 386 | 1002 | 2892 | 3032 |
| Phild-pha. | 772 | 764 | 572 | 1614 | 2752 | 424 | 92 | 0 | 305 | 910 | 2800 | 2951 |
| Pitts-brg. | 714 | 459 | 284 | 1421 | 2464 | 514 | 386 | 305 | 0 | 622 | 2512 | 2646 |
| St. Louis | 554 | 294 | 338 | 799 | 1842 | 1058 | 1002 | 910 | 622 | 0 | 1890 | 2125 |
| San Diego | 2363 | 2184 | 2228 | 1521 | 95 | 2948 | 2892 | 2800 | 2512 | 1890 | 0 | 500 |
| San Fran. | 2679 | 2187 | 2463 | 2021 | 405 | 2951 | 3032 | 2951 | 2646 | 2125 | 500 | 0 |

従ってアトランタとシカゴ間にリンクがあれば，YATLCHI=1 になる．各ノードは，2 つのリンクで接続されなければならない．定式化は次の通りである：

**MIN　選ばれるリンクの費用**

**制約式：各都市は，それに結びついたリンク数 2 で選ばれる**

**　　　　各リンクは高々一度選ぶことができる．**

**訳注：このモデルは少し技巧的である．都市 I から都市 j に行くリンクと，都市 j から都 I に行く 2 つのリンクを，I＜j の場合，都市 I から都市 j に行くリンクとみなしている．これによって，2 都市間の距離を上三角行列の対角線の上だけの定義で行っている．**

LINGO のモデルは次に示されている．

```
MODEL:
SETS:
CITY;
ROUTE(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 1616
 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
```

訳注：Generateで次のモデルが生成される．

```
MODEL:
[_1]MIN=702*Y_CHI_ATL+454*Y_CIN_ATL+324*Y_CIN_CHI+842*Y_HOU_A
TL+1093*Y_HOU_CHI+1137*Y_HOU_CIN+2396*Y_LA_ATL+2136*Y_LA_CHI+
2180*Y_LA_CIN+1616*Y_LA_HOU+1196*Y_MON_ATL+764*Y_MON_CHI+798*
Y_MON_CIN+1857*Y_MON_HOU+2900*Y_MON_LA+864*Y_NY_ATL+845*Y_NY_
CHI+664*Y_NY_CIN+1706*Y_NY_HOU+2844*Y_NY_LA+396*Y_NY_MON+772*
Y_PHI_ATL+764*Y_PHI_CHI+572*Y_PHI_CIN+1614*Y_PHI_HOU+2752*Y_P
HI_LA+424*Y_PHI_MON+92*Y_PHI_NY+714*Y_PIT_ATL+459*Y_PIT_CHI+2
84*Y_PIT_CIN+1421*Y_PIT_HOU+2464*Y_PIT_LA+514*Y_PIT_MON+386*Y
_PIT_NY+305*Y_PIT_PHI+554*Y_STL_ATL+294*Y_STL_CHI+338*Y_STL_C
IN+799*Y_STL_HOU+1842*Y_STL_LA+1058*Y_STL_MON+1002*Y_STL_NY+9
10*Y_STL_PHI+622*Y_STL_PIT+2363*Y_SD_ATL+2184*Y_SD_CHI+2228*Y
_SD_CIN+1521*Y_SD_HOU+95*Y_SD_LA+2948*Y_SD_MON+2892*Y_SD_NY+2
800*Y_SD_PHI+2512*Y_SD_PIT+1890*Y_SD_STL+2679*Y_SF_ATL+2187*Y
_SF_CHI+2463*Y_SF_CIN+2021*Y_SF_HOU+405*Y_SF_LA+2951*Y_SF_MON
+3032*Y_SF_NY+2951*Y_SF_PHI+2646*Y_SF_PIT+2125*Y_SF_STL+500*Y
```

```
＿ＳＦ＿ＳＤ；
［＿２］Ｙ＿ＣＨＩ＿ＡＴＬ＋Ｙ＿ＣＩＮ＿ＡＴＬ＋Ｙ＿ＨＯＵ＿ＡＴＬ＋Ｙ＿ＬＡ＿ＡＴＬ＋Ｙ＿ＭＯＮ＿ＡＴＬ＋Ｙ＿ＮＹ＿ＡＴＬ
＋Ｙ＿ＰＨＩ＿ＡＴＬ＋Ｙ＿ＰＩＴ＿ＡＴＬ＋Ｙ＿ＳＴＬ＿ＡＴＬ＋Ｙ＿ＳＤ＿ＡＴＬ＋Ｙ＿ＳＦ＿ＡＴＬ＝２；
［＿３］Ｙ＿ＣＨＩ＿ＡＴＬ＋Ｙ＿ＣＩＮ＿ＣＨＩ＋Ｙ＿ＨＯＵ＿ＣＨＩ＋Ｙ＿ＬＡ＿ＣＨＩ＋Ｙ＿ＭＯＮ＿ＣＨＩ＋Ｙ＿ＮＹ＿ＣＨＩ
＋Ｙ＿ＰＨＩ＿ＣＨＩ＋Ｙ＿ＰＩＴ＿ＣＨＩ＋Ｙ＿ＳＴＬ＿ＣＨＩ＋Ｙ＿ＳＤ＿ＣＨＩ＋Ｙ＿ＳＦ＿ＣＨＩ＝２；
［＿４］Ｙ＿ＣＩＮ＿ＡＴＬ＋Ｙ＿ＣＩＮ＿ＣＨＩ＋Ｙ＿ＨＯＵ＿ＣＩＮ＋Ｙ＿ＬＡ＿ＣＩＮ＋Ｙ＿ＭＯＮ＿ＣＩＮ＋Ｙ＿ＮＹ＿ＣＩＮ
＋Ｙ＿ＰＨＩ＿ＣＩＮ＋Ｙ＿ＰＩＴ＿ＣＩＮ＋Ｙ＿ＳＴＬ＿ＣＩＮ＋Ｙ＿ＳＤ＿ＣＩＮ＋Ｙ＿ＳＦ＿ＣＩＮ＝２；
［＿５］Ｙ＿ＨＯＵ＿ＡＴＬ＋Ｙ＿ＨＯＵ＿ＣＨＩ＋Ｙ＿ＨＯＵ＿ＣＩＮ＋Ｙ＿ＬＡ＿ＨＯＵ＋Ｙ＿ＭＯＮ＿ＨＯＵ＋Ｙ＿ＮＹ＿ＨＯＵ
＋Ｙ＿ＰＨＩ＿ＨＯＵ＋Ｙ＿ＰＩＴ＿ＨＯＵ＋Ｙ＿ＳＴＬ＿ＨＯＵ＋Ｙ＿ＳＤ＿ＨＯＵ＋Ｙ＿ＳＦ＿ＨＯＵ＝２；
［＿６］Ｙ＿ＬＡ＿ＡＴＬ＋Ｙ＿ＬＡ＿ＣＨＩ＋Ｙ＿ＬＡ＿ＣＩＮ＋Ｙ＿ＬＡ＿ＨＯＵ＋Ｙ＿ＭＯＮ＿ＬＡ＋Ｙ＿ＮＹ＿ＬＡ＋Ｙ＿ＰＨ
Ｉ＿ＬＡ＋Ｙ＿ＰＩＴ＿ＬＡ＋Ｙ＿ＳＴＬ＿ＬＡ＋Ｙ＿ＳＤ＿ＬＡ＋Ｙ＿ＳＦ＿ＬＡ＝２；
［＿７］Ｙ＿ＭＯＮ＿ＡＴＬ＋Ｙ＿ＭＯＮ＿ＣＨＩ＋Ｙ＿ＭＯＮ＿ＣＩＮ＋Ｙ＿ＭＯＮ＿ＨＯＵ＋Ｙ＿ＭＯＮ＿ＬＡ＋Ｙ＿ＮＹ＿ＭＯＮ
＋Ｙ＿ＰＨＩ＿ＭＯＮ＋Ｙ＿ＰＩＴ＿ＭＯＮ＋Ｙ＿ＳＴＬ＿ＭＯＮ＋Ｙ＿ＳＤ＿ＭＯＮ＋Ｙ＿ＳＦ＿ＭＯＮ＝２；
［＿８］Ｙ＿ＮＹ＿ＡＴＬ＋Ｙ＿ＮＹ＿ＣＨＩ＋Ｙ＿ＮＹ＿ＣＩＮ＋Ｙ＿ＮＹ＿ＨＯＵ＋Ｙ＿ＮＹ＿ＬＡ＋Ｙ＿ＮＹ＿ＭＯＮ＋Ｙ＿ＰＨ
Ｉ＿ＮＹ＋Ｙ＿ＰＩＴ＿ＮＹ＋Ｙ＿ＳＴＬ＿ＮＹ＋Ｙ＿ＳＤ＿ＮＹ＋Ｙ＿ＳＦ＿ＮＹ＝２；
［＿９］Ｙ＿ＰＨＩ＿ＡＴＬ＋Ｙ＿ＰＨＩ＿ＣＨＩ＋Ｙ＿ＰＨＩ＿ＣＩＮ＋Ｙ＿ＰＨＩ＿ＨＯＵ＋Ｙ＿ＰＨＩ＿ＬＡ＋Ｙ＿ＰＨＩ＿ＭＯ
Ｎ＋Ｙ＿ＰＨＩ＿ＮＹ＋Ｙ＿ＰＩＴ＿ＰＨＩ＋Ｙ＿ＳＴＬ＿ＰＨＩ＋Ｙ＿ＳＤ＿ＰＨＩ＋Ｙ＿ＳＦ＿ＰＨＩ＝２；
［＿１０］Ｙ＿ＰＩＴ＿ＡＴＬ＋Ｙ＿ＰＩＴ＿ＣＨＩ＋Ｙ＿ＰＩＴ＿ＣＩＮ＋Ｙ＿ＰＩＴ＿ＨＯＵ＋Ｙ＿ＰＩＴ＿ＬＡ＋Ｙ＿ＰＩＴ＿Ｍ
ＯＮ＋Ｙ＿ＰＩＴ＿ＮＹ＋Ｙ＿ＰＩＴ＿ＰＨＩ＋Ｙ＿ＳＴＬ＿ＰＩＴ＋Ｙ＿ＳＤ＿ＰＩＴ＋Ｙ＿ＳＦ＿ＰＩＴ＝２；
［＿１１］Ｙ＿ＳＴＬ＿ＡＴＬ＋Ｙ＿ＳＴＬ＿ＣＨＩ＋Ｙ＿ＳＴＬ＿ＣＩＮ＋Ｙ＿ＳＴＬ＿ＨＯＵ＋Ｙ＿ＳＴＬ＿ＬＡ＋Ｙ＿ＳＴＬ＿Ｍ
ＯＮ＋Ｙ＿ＳＴＬ＿ＮＹ＋Ｙ＿ＳＴＬ＿ＰＨＩ＋Ｙ＿ＳＴＬ＿ＰＩＴ＋Ｙ＿ＳＤ＿ＳＴＬ＋Ｙ＿ＳＦ＿ＳＴＬ＝２；
［＿１２］Ｙ＿ＳＤ＿ＡＴＬ＋Ｙ＿ＳＤ＿ＣＨＩ＋Ｙ＿ＳＤ＿ＣＩＮ＋Ｙ＿ＳＤ＿ＨＯＵ＋Ｙ＿ＳＤ＿ＬＡ＋Ｙ＿ＳＤ＿ＭＯＮ＋Ｙ＿Ｓ
Ｄ＿ＮＹ＋Ｙ＿ＳＤ＿ＰＨＩ＋Ｙ＿ＳＤ＿ＰＩＴ＋Ｙ＿ＳＤ＿ＳＴＬ＋Ｙ＿ＳＦ＿ＳＤ＝２；
［＿１３］Ｙ＿ＳＦ＿ＡＴＬ＋Ｙ＿ＳＦ＿ＣＨＩ＋Ｙ＿ＳＦ＿ＣＩＮ＋Ｙ＿ＳＦ＿ＨＯＵ＋Ｙ＿ＳＦ＿ＬＡ＋Ｙ＿ＳＦ＿ＭＯＮ＋Ｙ＿Ｓ
Ｆ＿ＮＹ＋Ｙ＿ＳＦ＿ＰＨＩ＋Ｙ＿ＳＦ＿ＰＩＴ＋Ｙ＿ＳＦ＿ＳＴＬ＋Ｙ＿ＳＦ＿ＳＤ＝２；
［＿１４］Ｙ＿ＣＨＩ＿ＡＴＬ＜＝１；
［＿１５］Ｙ＿ＣＩＮ＿ＡＴＬ＜＝１；
［＿１６］Ｙ＿ＣＩＮ＿ＣＨＩ＜＝１；
［＿１７］Ｙ＿ＨＯＵ＿ＡＴＬ＜＝１；
［＿１８］Ｙ＿ＨＯＵ＿ＣＨＩ＜＝１；
［＿１９］Ｙ＿ＨＯＵ＿ＣＩＮ＜＝１；
［＿２０］Ｙ＿ＬＡ＿ＡＴＬ＜＝１；
［＿２１］Ｙ＿ＬＡ＿ＣＨＩ＜＝１；
［＿２２］Ｙ＿ＬＡ＿ＣＩＮ＜＝１；
［＿２３］Ｙ＿ＬＡ＿ＨＯＵ＜＝１；
［＿２４］Ｙ＿ＭＯＮ＿ＡＴＬ＜＝１；
［＿２５］Ｙ＿ＭＯＮ＿ＣＨＩ＜＝１；
```

```
[_26]Y_MON_CIN<=1;
[_27]Y_MON_HOU<=1;
[_28]Y_MON_LA<=1;
[_29]Y_NY_ATL<=1;
[_30]Y_NY_CHI<=1;
[_31]Y_NY_CIN<=1;
[_32]Y_NY_HOU<=1;
[_33]Y_NY_LA<=1;
[_34]Y_NY_MON<=1;
[_35]Y_PHI_ATL<=1;
[_36]Y_PHI_CHI<=1;
[_37]Y_PHI_CIN<=1;
[_38]Y_PHI_HOU<=1;
[_39]Y_PHI_LA<=1;
[_40]Y_PHI_MON<=1;
[_41]Y_PHI_NY<=1;
[_42]Y_PIT_ATL<=1;
[_43]Y_PIT_CHI<=1;
[_44]Y_PIT_CIN<=1;
[_45]Y_PIT_HOU<=1;
[_46]Y_PIT_LA<=1;
[_47]Y_PIT_MON<=1;
[_48]Y_PIT_NY<=1;
[_49]Y_PIT_PHI<=1;
[_50]Y_STL_ATL<=1;
[_51]Y_STL_CHI<=1;
[_52]Y_STL_CIN<=1;
[_53]Y_STL_HOU<=1;
[_54]Y_STL_LA<=1;
[_55]Y_STL_MON<=1;
[_56]Y_STL_NY<=1;
[_57]Y_STL_PHI<=1;
[_58]Y_STL_PIT<=1;
[_59]Y_SD_ATL<=1;
[_60]Y_SD_CHI<=1;
[_61]Y_SD_CIN<=1;
[_62]Y_SD_HOU<=1;
```

```
[_63]Y_SD_LA<=1;
[_64]Y_SD_MON<=1;
[_65]Y_SD_NY<=1;
[_66]Y_SD_PHI<=1;
[_67]Y_SD_PIT<=1;
[_68]Y_SD_STL<=1;
[_69]Y_SF_ATL<=1;
[_70]Y_SF_CHI<=1;
[_71]Y_SF_CIN<=1;
[_72]Y_SF_HOU<=1;
[_73]Y_SF_LA<=1;
[_74]Y_SF_MON<=1;
[_75]Y_SF_NY<=1;
[_76]Y_SF_PHI<=1;
[_77]Y_SF_PIT<=1;
[_78]Y_SF_STL<=1;
[_79]Y_SF_SD<=1;
END
```

このモデルを LP で解くと，次の解が得られる．

```
Optimal solution found at step:        105
Objective value:     5020.000
     Variable               Value          Reduced Cost
Y( CIN, ATL)       1.000000             0.0000000
Y( CIN, CHI)       1.000000             0.0000000
Y( HOU, ATL)       1.000000             0.0000000
Y( NY, MON)        1.000000             0.0000000
Y( PHI, NY)        1.000000             0.0000000
Y( PIT, MON)       1.000000             0.0000000
Y( PIT, PHI)       1.000000             0.0000000
Y( STL, CHI)       1.000000             0.0000000
Y( STL, HOU)       1.000000             0.0000000
Y( SD, LA)         1.000000             0.0000000
Y( SF, LA)         1.000000             0.0000000
Y( SF, SD)         1.000000             0.0000000
```

5020 マイルの費用であり，**図 11.6** で図示される．

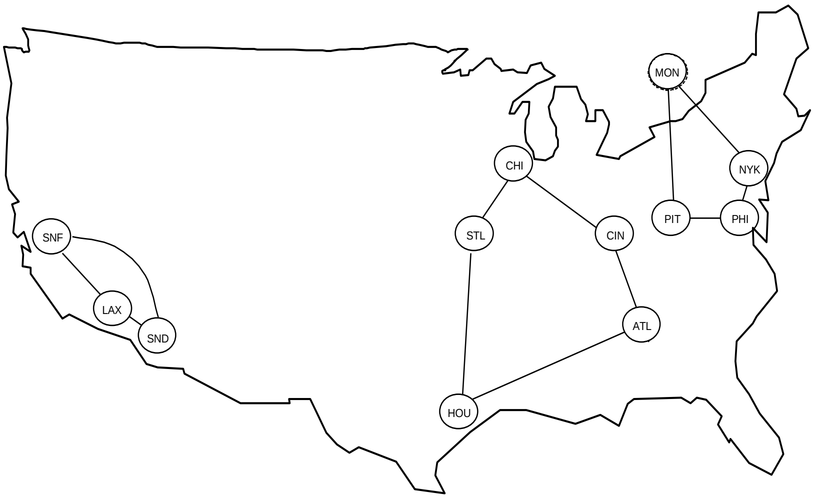


図 11.6

不幸にして，解は 3 つの部分旅行に分かれた.次の制約を加えて，部分旅行をなくしたい:

!SUBTOUR ELIMINATION;

Y(SF, LA) + Y(SD, LA) + Y(SF, SD) <= 2;

LP の解は 6975 で,図 11.7 になる.

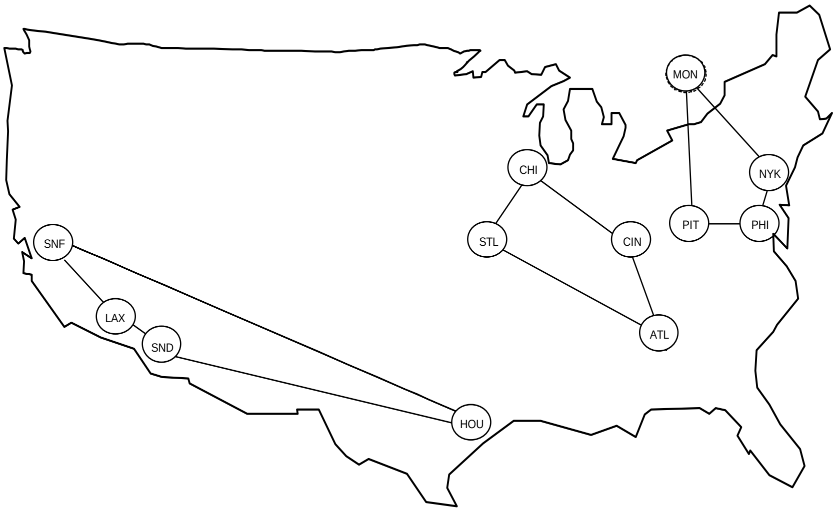


図 11.7

さらに次の制約式を加え，南西部の部分旅行の削減を試みる.

Y(LA,SD)+Y(SD,HOU)+Y(SF,LA)+Y(SF,HOU)<=3;

この後，部分旅行の削減のため次のような制約をつけ加えていく.

Y(CIN,ATL)+Y(CIN,ATL)+Y(CIN,CHI)+Y(HOU,ATL)+Y(STL,CHI)+Y(STL,HOU)<=4;

Y(NY,MON)+Y(PHI,NY)+Y(PIT,MON)+Y(PIT,PHI)<=3;

Y(SD,LA)+Y(SF,LA)+Y(SF,SD)<=2;
Y(NY,MON)+Y(PHI,MON)+Y(PHI,NY)<=2;
Y(CIN,ATL)+Y(PIT,CHI)+Y(PIT,CIN)+Y(STL,ATL)+Y(STL,CHI)<=4;
Y(SD,HOU)+Y(SD,LA)+Y(SF,HOU)+Y(SF,LA)<=3;
Y(CIN,ATL)+Y(CIN,CHI)+Y(STL,ATL)+Y(STL,CHI)<=3;
Y(PHI,MON)+Y(PHI,NY)+Y(PIT,MON)+Y(PIT,NY)<=3;
Y(CIN,ATL)+Y(MON,CHI)+Y(NY,MON)+Y(PHI,NY)+Y(PIT,CIN)+PIT,PHI)+Y(STL,ATL)+Y(STL,CHI)<=7;
Y(SD,LA)+Y(SF,HOU)+Y(SF,SD)+Y(LA,HOU)<=3;

上の制約式を加えて実行すると，**図 11.8** の$7,577 の完全な旅行が得られる．分枝限定法を用いず，LP で計算できた．

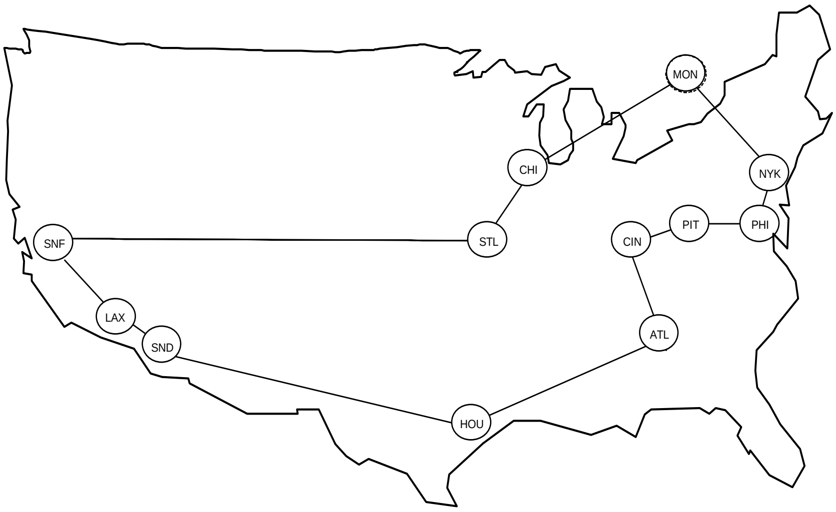


**図 11.8**

P．ローズの行ったヒューリスティック法はどうであろうか？最も明らかなヒューリスティック法は，「一番近い未訪問都市」に行くことである．シンシナチで始まり，次に一番近い未訪問都市に行き，最後にシンシナチに戻れば，総距離は 8015 マイルで，最適解より約 6％悪いだけである．

**任意の都市を巡る TSP**

あらゆる都市を訪問するという条件を落とせば，「**任意都市の TSP 問題**」になる．これは vj が仕事 j からの利益で，$c_{Ij}$ が仕事 I から仕事 j への転換費用であるような仕事の順序づけ問題である．

$y_j$ = 1（都市 j を訪問する場合），0（それ以外）

もし vj が都市 j を訪問する価値なら，目的関数は次の通りである：

Minimize　$\sum_I \sum_j c_{Ij}\ x_{Ij} - \sum v_j\ y_j$ .

制約式は次の通りである．

② 各都市 j は高々一度訪問できる：$\sum_{I \neq j} x_{Ij} = y_j$

② 都市 j に入れば，そこを出なければならない：$\Sigma_{k \neq j} x_{jk} = y_j$

③ ホームベース 1 を含まない都市の部分集合 s に対し，部分旅行は許されない．

$\Sigma_{I,j \in S} x_{Ij} \leq |S| - 1$，$|S|$は集合 $S$ のサイズ．

例えば，ホームベースを含む n 都市があれば，サイズ 3 の(n−1)(n−2)/(3×2)個のサブセットがある．

④ ③の代わりに，次の置き換えがある．

$U_j \geq U_I + 1 - (1 - x_{Ij})n$ for $j = 2, 3, \cdots, n$.

$U_j$は旅行で都市 j の訪問番号である．制約式 ③は④より条件が厳しい．

### 11.6.3 容量のある多数 TSP/車両旅程問題

重要で実用的な問題は，中央ターミナルからの車の旅程である．例として大都会の新聞配達のためのトラックの旅程である．各 TSP は有限な容量があるため，多数の TSP 問題として考える．この問題は，ときどき小口混載経路問題と呼ばれる．モデルは次の通りである：

v =車の容量

Dj =都市または停止地 j の要求

各都市 j は，ji>1 を満たし一度訪問しなければならない： $\sum_j x_{Ij} = 1$

各都市 I>j は，一度は出なければならない： $\sum_i x_{Ij} = 1$

部分旅行は禁じられている： $\sum_{i,j \in T} x_{Ij} \leq |S| - 1$,

積み過ぎ無し:1 を含む都市 T の各集合で，積載負荷以下を代入：

$$\sum_{i,j \in T} x_{Ij} \leq |T| - k,$$

k =(1 つの負荷を減らすために T から省く都市の最小値)．

このモデルは，例えば 25 都市くらいのサイズで最適化し解決できる．より大きく複雑で実用的な問題は，Clarke & Wright(1964) のヒューリスティック法を用いれば，すばやく満足解を求める出発点を見つけることができる．

**例題：**

次は，一般的な車両経路問題（Vehicle Routing Problem，VRP)の LINGO モデルである．VRP は，配送/顧客のピックアップ/補修など，多くのサービスにみられる．容量制限のある車両は，デポを出発しサービスを要求する顧客を回った後，デポに帰る．ここでは，デポの都市 1 から他の 9 都市をサービスする問題を考える．

```
MODEL:  ! (VROUTE);
! The Vehicle Routing Problem (VRP) occurs in many service
```

```
systems such as delivery, customer pick-up, repair and
maintenance. A fleet of vehicles, each with fixed
capacity, starts at a common depot and returns to the
depot after visiting locations where service is demanded.
Problems with more than a dozen cities can take lots of
time.This instance involves delivering the required amount
of goods to 9 cities from a depot at city 1;
SETS:
CITY/ Chi Den Frsn Hous KC LA Oakl Anah Peor Phnx/: Q,U;
! Q(I) = amount required at city I(given),
must be delivered by just 1 vehicle.
U(I) = accumulated deliveries at city I ;
CXC( CITY, CITY): DIST, X;
! DIST(I,J) = distance from city I to city J
X(I,J) is 0-1 variable,
= 1 if some vehicle travels from city I to J,
else 0 ;
ENDSETS
DATA:
! city 1 represents the common depot, i.e. Q( 1) = 0;
Q= 0    6    3    7    7   18    4    5    2    6;
! distance from city I to city J is same from J to I,
distance from city I to the depot is 0,
because vehicle need not return to the depot ;
DIST=  ! To City;
!Chi  Den Frsn Hous   KC   LA Oakl Anah  Peor Phnx  From;
0  996 2162 1067  499 2054 2134 2050  151 1713!  Chicago;
0    0 1167 1019  596 1059 1227 1055  904  792!  Denver;
0 1167    0 1747 1723  214  168  250 2070  598!  Fresno;
0 1019 1747    0  710 1538 1904 1528  948 1149!  Houston;
0  596 1723  710    0 1589 1827 1579  354 1214!  K. City;
0 1059  214 1538 1589    0  371   36 1943  389!  L. A.;
0 1227  168 1904 1827  371    0  407 2043  755!  Oakland;
```

```
0 1055  250 1528 1579   36  407    0 1933  379! Anaheim;
0  904 2070  948  354 1943 2043 1933    0 1568! Peoria;
0  792  598 1149 1214  389  755  379 1568    0;! Phoenix;
! VCAP is the capacity of a vehicle ;
VCAP = 18;
ENDDATA
!------------------------------------------------------------;
! The objective is to minimize total travel distance;
MIN = @SUM( CXC: DIST * X);
! for each city, except depot....;
@FOR( CITY( K)| K #GT# 1:
! a vehicle does not travel inside itself,...;
X( K, K) = 0;
! a vehicle must enter it,... ;
@SUM( CITY( I)| I #NE# K #AND# ( I #EQ# 1 #OR#
Q( I) + Q( K) #LE# VCAP): X( I, K)) = 1;
! a vehicle must leave it after service ;
@SUM( CITY( J)| J #NE# K #AND# ( J #EQ# 1 #OR#
Q( J) + Q( K) #LE# VCAP): X( K, J)) = 1;
```

```
! U( K) = amount delivered on trip up to city K
>= amount needed at K but <= vehicle capacity;
@BND( Q( K), U( K), VCAP);
! If K follows I, then can bound U( K) - U( I);
@FOR( CITY( I)| I #NE# K #AND# I #NE# 1: U( K) >=
U( I) + Q( K) - VCAP + VCAP*( X( K, I) + X( I, K))
- ( Q( K) + Q( I)) * X( K, I););
! If K is 1st stop, then U( K) = Q( K);
U(K)<=VCAP-(VCAP-Q(K))*X(1,K);
! If K is not 1st stop...;
U( K) >=
Q( K)+ @SUM( CITY( I)| I #GT# 1: Q( I) * X( I, K)););
! Make the X's binary;
@FOR( CXC( I, J): @BIN( X( I, J)) ;);
! Must send enough vehicles out of depot;
@SUM( CITY( J)| J #GT# 1: X( 1, J)) >=
@FLOOR((@SUM( CITY( I)| I #GT# 1: Q( I))/ VCAP) + .999);
END
```

解は次の通りである．

```
Optimal solution found at step:        973
Objective value:                  6732.000
              Variable           Value
         X( CHI, HOUS)        1.000000
           X( CHI, LA)        1.000000
          X( CHIPEOR)        1.000000
          X( CHIPHNX)        1.000000
          X( DEN, CHI)        1.000000
         X( FRS数AKL)        1.000000
         X( HOUS, CHI)        1.000000
           X( KC, DEN)        1.000000
           X( LA, CHI)        1.000000
         X( OAKL, CHI)        1.000000
        X( ANAH, FRSN)        1.000000
          X( PEOR, KC)        1.000000
        X( PHNX, ANAH)        1.000000
```

リンクをたどることで，4つの次の旅行があることが分かる：
シカゴ-ヒューストン，シカゴ- LA，シカゴ- PeorIa- KC-デンバー，シカゴ-フェニックス-アナハイム-フレズノ-オークランド．

**結合されたDCの位置／車両経路指定**

しばしば，新しい工場か流通センター(DC)の開設と関連付けられる車両経路指定問題として，最低の費用で顧客に役立つDCからのサービスを考える．問題の完全な解は，DCの位置および旅程問題を同時に解決する．次のIPによる定式化は1つのアプローチである．

**パラメータ**

$F_I$ =位置IにDCをもつ固定費，

$C_j$ =経路jの使用費用，

$a_{Ijk}$ = 1（経路jがDCIを基点とし，顧客kをサービスする）．
0（各経路に一つのDCがある．）

**決定変数**

$y_I$ = 1（DC Iを使用），0（それ以外），

$x_j$ = 1（経路jを使用），0（それ以外），

**モデル：**

**Minimize** $\Sigma_I F_I y_I + \Sigma_j c_j x_j$

**subject to**

**（需要制約） 各顧客 $k$ に対して：** $\Sigma_I \Sigma_j a_{Ijk} x_j = 1$

**（強制制約）各 $DC_I$ と顧客 $k$ に対して：** $\Sigma_j a_{Ijk} x_j \leq y_I$

**11.6.4 最小スパニング樹**

nノードのスパニング・ツリーは，(n-1)個のアークがあり，ノードの各ペアの間にちょうど一つのパスがある．最小費用のスパニング・ツリーは，通信網の設計でみられる．

ノード1を木の根とする．1からIを通ってjへの道があれば，$x_{Ij}= 1$，他の場合は $x_{Ij} = 0$．定式化は次のとおりである：

Minimize $\Sigma_I \Sigma_j c_{Ij} x_{Ij}$

subject to

① $\Sigma_I \Sigma_j x_{Ij} = n - 1$

② {1，2，…，n} のサブセットs毎に

$\Sigma_{I,j\in S} x_{Ij} \leq |S| - 1$ ，$x_{Ij} = 0$ or $1$

①と②の代替案は，次の各ノードに順序 nj を割り当てる次の制約式の集合である：

$$\sum_{i \neq j} x_{ij} = 1, \quad \text{for } j = 2, 3, 4, \cdots, n,$$

$$U_j \geq U_I + 1 - (1 - x_{Ij})n \quad \text{for } j = 2, 3, 4, \cdots, n.$$

$$U_j \geq 0$$

この場合，$U_j$ はノード j とノード 1 の間のアークの数である．
純粋なスパニング・ツリー問題であれば，Kruskal (1956)の「どん欲なアルゴリズム」は最適解を見つける早道である．

### 11.6.5 線形注文問題

TSP と同じような問題に，線形注文問題がある．n 個の対象の厳密な順序を見つけたい．適用例は，スポーツのトーナメントのランク付け，マーケティングでの物の選好順位，1 台の機械の仕事順，投入産出行列の産業の順序，考古学の歴史的な対象の順位，などである．Grötschel(1985)を参照．これは，マーケッティングに用いられるコンジョイント分析に似ている．重大な入力データは，費用項目の $c_{Ij}$ である．オブジェクト I は，提案された順序でオブジェクト j の前にいつでも現われれば，$c_{Ij}$ は生じる費用である．決定変数は次のとおりである： $x_{Ij} = 1$ (直接または間接的であれ，$I \neq j$ に対して，I が j に先行)
問題は次の通りである：

$$\text{Minimize} \quad \sum_i \sum_j c_{Ij} \, x_{Ij}$$

subject to

$$① \quad x_{Ij} + x_{jI} = 1 \quad \text{for all } I \neq j$$

もし I が j に，j が k に先行すれば，I は k に先行することを意味する．これは次の制約式を強制する．

$$② \quad x_{Ij} + x_{jk} + x_{kI} \leq 2$$

$$\text{for all } I, j, k \text{ wIth } I \neq j, I \neq k, j \neq k.$$

モデルは，$x_{jI} = 1 - x_{Ij}$ (j > I) に注意して，$x_{jI}$ を消去すると次のようになる．

$$①' \quad 0 \leq x_{Ij} \leq 1.$$

$$②' \quad x_{Ij} + x_{jk} - x_{Ik} + s_{Ijk} = 1 \quad \text{for all } I < j < k$$

$$0 \leq s_{Ijk} \leq 1$$

n から 3 個のオブジェクトを選ぶ方法は，$n!/((n-3)!\ 3!) = n \times (n-1) \times (n-2)/6$ 個ある．そこで制約式の概算は，$n^3/6$ になる．

**例**

10 個のドイツ，チェコおよび米国の飲み物が，公平なドイツのテスターによって

好みテストがされた．各 10*9/2 = 45 のペアが，6 人のテスターで評価された．次のモデルの行列 C の要素 c（I，j）は，飲料 I を飲料 j より好む 6 人の回答数である．私達が飲料の順位付けをしたいと思えば，適切な目的関数は順位がテスターの一対比較の順位と一致する数を最大にすることである．

```
MODEL:
! Linear ordering of objects or products,
   based on pairwise comparisons(LINERORD);
SETS:
PROD:RANK;!Eachproductwillgetarank;
PXP(PROD,PROD):C;
ENDSETS
DATA:
PROD=KONIG,FURST,PILSURQ,GUNZB,RIEGELE,
PAULA,JEVER,BECKS,WARST,BUD;
! Some data on German beverages;
 C= ! Times that object I was preferred over J;
0	2	2	3	3	5	5	5	4	4
4	0	3	3	4	3	2	3	2	2
4	3	0	3	5	4	3	2	4	4
3	3	3	0	5	6	3	4	4	3
3	2	1	1	0	1	4	4	5	3
1	3	2	0	5	0	5	4	1	4
1	4	3	3	2	1	0	2	1	3
1	3	4	2	2	2	4	0	4	2
2	4	2	2	1	5	5	2	0	4
2	4	2	3	3	2	3	4	2	0;
ENDDATA
!----------------------------------------;
SETS:
PIP(PROD,PROD)|&1#LT#&2:
X;!X(I,J)=1 if I precedes J in our ranking;
PIPIP(PROD,PROD,PROD)
|&1#LT#&2#AND#&2#LT#&3:S;
ENDSETS
! Maximize the number of times our pairwise
  ordering matches that of our testers;
MAX=@SUM(PIP(I,J):C(I,J)*X(I,J)+C(J,I)*(1-X(I,J)));
```

```
! The rankings must be transitive, that is,
  If I->J and J->K, then I->K;
@FOR(PIPIP(I,J,K):
!   Note N*(N-1)*(N-2)/6 of these!;
X(I,J)+X(J,K)-X(I,K)+S(I,J,K)=1;
@BND(0,S(I,J,K),1););
@FOR(PIP:@BIN(X););! Make X’s 0 or 1;
! Count number products before product I( + 1);
@FOR(PROD(I):
RANK(I)=1+@SUM(PIP(K,I):X(K,I))
+@SUM(PIP(I,K):1-X(I,K)););
END
```

これを解くと，最適解 168 を得る．これは(10*9/2)*6 = 270 個の一対比較から，LINGO は 168 回の完全な順序に同意したことになる．

```
Optimal solution found at step:          50
Objective value:                   168.0000
Branch count:                             0
      Variable            Value        Reduced Cost
   RANK( KONIG)        3.000000          0.0000000
   RANK( FURST)        10.00000          0.0000000
RANK( PILSURQ)         2.000000          0.0000000
   RANK( GUNZB)        1.000000          0.0000000
RANK( RIEGELE)         7.000000          0.0000000
   RANK( PAULA)        5.000000          0.0000000
   RANK( JEVER)        9.000000          0.0000000
   RANK( BECKS)        8.000000          0.0000000
   RANK( WARST)        4.000000          0.0000000
     RANK( BUD)        6.000000          0.0000000
```

このランキングに従えば，GUNZB は第 1 位(最も好まれる)で，FURST は第 10 位である(最も好まれない)．代替案があることに注意することは重要である．これは 270 回から 168 回の入力の一対比較に一致させる代替的な順序づけがあるかもしれない．実際は，PILSURQ が第 1 順位になる 168 という値をもつ別の解を示すことができる

### 11.6.6 二次割当問題

二次割当問題は，線形割当問題と同じ制約式である．但し，目的関数は次のような 2 つの変数の積を含んでいる．

$$\mathrm{MIn}\quad \sum_i\sum_j\sum_k\sum_l c_{Ijkl}\, x_{Ij}\, x_{kl}$$

subject to:

For each $j$: $\sum_i x_{Ij} = 1$

For each $I$:

$$\sum_j x_{Ij} = 1$$

この問題の例に，次のようなものがある.

(a) 設備レイアウト： $D_{jl}$ が部屋 j と部屋 l 間の物理的な距離. $s_{Ik}$ は部門 I および k 間の通信量. 部門 I が部屋 j 割り当てられれば $x_{Ij}=1$ である. そして次の目的関数を最小化したい：

$$\sum_i\sum_j\sum_k\sum_l x_{Ij}\, x_{kl}\, D_{jl}\, s_{Ik}$$

(b) 車をターミナルでゲートに割り当て： $D_{jl}$ は航空会社のターミナルのゲート j とゲート I の距離（駅の乗客，またはトラックターミナル間の距離）. $s_{Ik}$ は車 I と車 k の間で移しかえる必要がある貨物の重量や乗客数. そして車 I が(入力や出力)ゲート j に割り当てられれば $x_{Ij}=1$. そして次の最小化したい.

$$\sum_i\sum_j\sum_k\sum_l x_{Ij}\, x_{kl}\, D_{jl}\, s_{Ik}$$

(c) 周波数の割り当て： $D_{Ij}$ が送信機 I と j 間の物理的な距離. $s_{kl}$ は k と l 間の周波数の距離. $P_I$ は送信機 I の出力. そして $\max\{P_I P_j\}\ (1/D_{Ij})(1/s_{kl})$ を，送信機 I を周波数 k に，送信機 j を周波数 l に割り当てられた場合に小さくしたい.

(D) VLSI 回路のレイアウト： VLSI（大規模集積回路)の設計の第一歩は，回路の様々な場所に様々な必要な部品を割り当てることである. 詳細は SarrafzaDeh&Wong(1996)参照. Steinberg (1961)は，電子回路を相互に連結するワイヤーの長さを最小にする電子部品の割り当てる例を紹介している. 回路設計では，回路は 2 つから 6 つの部分に仕切られる. $D_{jl}$ は領域 j と領域 l 間の物理的な距離. $s_{Ik}$ は部品 I と k の間で要求される結合数. 部品 I が領域 j に割り当てられれば $x_{Ij}=1$. そして次の最小化をしたい：

$$\sum_i\sum_j\sum_k\sum_l x_{Ij}\, x_{kl}\, D_{jl}\, s_{Ik}$$

(e)タイプ回転円盤の設計：タイプ回転円盤文字と数字を次のように配置する，

①最も頻繁に現れる物を一緒にする，②例えば，q と U のように，よく一緒に現れる文字は，タイプ回転円盤上の近くに配置する．
(f)ディスクファイルの割振り：ファイル I と j が同じディスクに割り当る場合，$w_{Ij}$ が干渉を表すなら，ディスクにファイルを割り当てたいと思う干渉の合計を最小にしたい．

2 次割当問題は悪名高く困難な問題である．誰かがそのような問題を頼めば，問題が真の二次割当問題でないことを示すために全力を尽すべきである．難しさの 1 つの徴候は，解が自然に整数にならないからである．

二次割当問題の最初の記述の 1 つは，Koopmans & Beckmann ら(1957)である．このため，この問題は時々KooPmans-Beckmann 問題として知られている．このモデルを使用して，大国の相互作用する設備の位置を見つけるために用いた．Elshafei (1977)は，このモデルを病院の設置に用いた．特に，病院の 19 の部門を 19 の物理的な地域に割り当てるために用いた．Elshafei の目的関数は，患者が部門間を歩かなければならない総距離の和を最小にすることである．この病院では，患者に 1 年に 13,973,298 メートルの距離の移動を要求した．最適割当は 8,606,274 メートルの総距離になり，38%以上の入院患者の移動の減少になった．

小さい二次割当問題は，変形によって線形な IP に変えることができる：

$z_{Ijkl}$ で積 $x_{Ij}$ $x_{kl}$ を置き換えることで，目的関数は線形になる：

$$\text{Min} \quad \sum_i \sum_j \sum_k \sum_l c_{Ijkl} \, z_{Ijkl}$$

もし $N$ 部門と $N$ 箇所の場所があれば，$N \times N$ 個の変数 $x_{Ij}$，と $N \times N \times N \times N$ 個の変数 $z_{Ijkl}$ がある．このモデルはすぐに巨大になるので，幾つかの削減方法がある：

① $c_{Ijkl}$ $x_{Ij}$ $x_{kl}$ と $c_{klIj}$ $x_{kl}$ $x_{Ij}$ は，次のように一つにまとめることができる：

$$(c_{Ijkl} + c_{klIj}) \, x_{kl} \, x_{Ij}$$

これで変数 $z$ と関連する制約式を減らせる．

② ある種の割り当ては事前に削除できる（例えば，大きな設備は小さな場所に割り当てない）．多くのクロス項 $c_{Ijkl}$ は 0 なので(例えば，設備の I と j 間に交流がない)，関連する変数 $z$ を導入しない．

$x_{Ij}$ と $x_{kl}$ が 1 の場合のみ，t $z_{Ijkl} = 1$ である．Sherali & Adams ら(1999)は，次のタイプの制約がこの要求を主張している：

$I$, $k$, $l$ に対して：　$x_{kl} = \sum_{j, j \neq l} z_{ijkl}$

対象 $k$ が場所 $l$ に割り当てられると，他の対象 $I$（$I \neq k$）に対して，I に割り当てられる場所 $j$（$j \neq l$）がある．次の記述は，飛行機を空港のどのゲートに割り当てるかを決定するためのものである．乗客が飛行機の乗り換えの重みつき費用を最小化する．

単一の変数 $z_{Ijkl}$ で $x_{Ij}$ と $x_{kl}$ の積を取替えなさい．

もし，$c_{Ijkl} > 0$ なら，次の制約式を加える．

$z_{Ijkl} \geq x_{Ij} + x_{jl} - 1$．

もし，$c_{Ijkl} \leq 0$ なら，次の制約式を加える．

$z_{Ijkl} \leq x_{Ij}$，$z_{Ijkl} \leq x_{kl}$．

次は上記の LINGO の実施である

```
MODEL:
! Quadratic assignment problem(QASGNM6);
!  Given transfers between flights, distance between
  gates assign flights to gates to minimize total
  transfer cost;
 SETS:  ! Symmetric Quadratic Assignment Problem;
  FLIGHT/1..6/;
  GATE/ E3 E4 E5 F3 F4 F5/;! TermInal 2 O'Hare gates;
  GXG( GATE, GATE)| &1 #LT# &2: T;
! Inter gate times(symmetric);
  FXF( FLIGHT, FLIGHT)| &1 #LT# &2: N;
! Transfers between flights;
  FXG( FLIGHT, GATE):   X;
! Flight to gate assignment variable;
 ENDSETS
 DATA:
  T =  70  40  60  90  90  ! Time between gates;
           50 100  80 110
              100  90 130
                   60  40
                        30;
  N =  12   0  12   0   5
           30  35  20  13  ! NO. Units between flights;
               40  20  10
                     0  6
                       14;
 ENDDATA
```

```
!------------------------------------------------------------;
! Warning: may be very slow for NO. objects > 6;
  SETS:  ! Warning: this set gets big fast!;
   TGTG(FLIGHT,GATE,FLIGHT,GATE)|&1#LT#&3#AND#
&2#LT#&4:Y;
  ENDSETS
! Min the cost of transfers * Distance;
   MIN=@SUM(TGTG(B,J,C,K):
Y(B,J,C,K)*N(B,C)*T(J,K););
! Each flight, B, must be assigned to a gate;
   @FOR(FLIGHT(B):
@SUM(GATE(J):X(B,J))=1;);
! Each gate, J, can receeIve at most one flight;
  @FOR(GATE(J):
@SUM(FLIGHT(B):X(B,J))=1;);
! Force Y( B, J, C, K) = 1 If B assigned to J and C to K;
!  Assumes the T() and N() matrices are non negative;
@FOR(TGTG(B,J,C,K):
Y(B,J,C,K)>=X(B,J)+X(C,K)+X(B,K)+X(C,J)-1;);
! Make the X's Integer;
    @FOR( FXG: @GIN( X););
! The following are optional cuts...,
   Each Pair of flights B and C must be assigned;
 @FOR(FXF(B,C):
@SUM(GXG(J,K):Y(B,J,C,K))=1;);
!  Each Pair of gates J anD K must get something;
  @FOR(GXG(J,K):
@SUM(FXF(B,C):Y(B,J,C,K))=1;);
NGATES=@SIZE(GATE);
! If flight B is assigned(or not) to J...;
  @FOR( FXG( B, J):
!  You have to incur the transportatIon;
   @SUM(TGTG(B,J,C,K):Y(B,J,C,K))
+@SUM(TGTG(C,J,B,K):Y(C,J,B,K))
+@SUM(TGTG(B,K,C,J):Y(B,K,C,J))
+@SUM(TGTG(C,K,B,J):Y(C,K,B,J))
=1+(NGATES-2)*X(B,J););
```

```
END
```

解は次のとおりである.

```
Optimal solution found at step:        9680
Objective value:     13490.00
Branch count:   12
                        Variable              Value
                      X( 1, E4)          1.000000
                      X( 2, F4)          1.000000
                      X( 3, F3)          1.000000
                      X( 4, F5)          1.000000
                      X( 5, E3)          1.000000
                      X( 6, E5)          1.000000
```

飛行機 1 がゲート *E4 に*,飛行機 2 はゲート *F4* などである.接続による乗客の総時間は 13,490 になる.この定式化はかなり厳しい.分岐が LP 解から整数解になることが要求されていない.

## 11.7 グループ化,一致,被覆,分割そしてパッキング問題

次の本質的な構造をもつ問題のクラスがある:

1)m 個の対象の集合がある

2)それらはサブセットに分かれ,ある規準で最適化される

| Objects | Group | グループの基準 |
|---|---|---|
| 寮の住人 | ルームメイト | 喫煙と禁煙で割り振る. |
| 顧客への配送 | 旅行 | 総重量が車の積載量より少ない.同じ旅行客はお互い近くにいる. |
| 学会のセッション | 同じ時間枠に予定されるセッション | 同じテーマの 2 つのセッションはない.十分なサイズの部屋がある. |
| 試験 | 同じ時間枠の試験 | 同じ時間に受ける試験は一つ |
| スポーツ | フォーサム(ゴルフやテニスのダブルス). | メンバーは同じ力量. テニスの混合ダブルスにおける性別の適切な組み合わせ |
| 色分けされる州 | 色分けされた全ての州 | 同じグループや色の州は隣接できない |
| 集積回路で結ばれる 2 点 | 回路基盤の下にある接続層 | 層の接続経路は交差しない.回路長を短くする. |

各対象がほとんど 1 グループに属する場合，パッキング問題と呼ばれる．例えば②の配達問題で，私達が明日配達しても良いと確信していれば，顧客優先順位の低い顧客の配達を明日にできる．もし各対象が丁度 1 グループに属する場合，**仕切り問題**と呼ばれる．例えば，⑦の電子回路の設計で，ある特定の組の点が接続する必要があるなら，基盤の下の丁度 1 つの接続層に割り当てられなければならない．各対象が少なくとも 1 グループに属していれば，**被覆問題**と呼ばれる．グループサイズが 2 以下に限られるパッキングか仕切り問題は，**マッチング問題**と呼ばれる．この問題には，高速な特殊解法がある．被覆問題と密接に関連した問題に**切断問題**がある．それは製紙，印刷，織物および鉄鋼企業で起こる．この問題では，私達は大きな原料を最終製品のさまざまなサイズを満たす切断パターンを定めたいと思う．

問題を分けることは容易だが，不適切なアプローチを取ると最適解を見つけることは困難かもしれない．問題を分類する 2 つの共通アプローチがある：

(1) 割り当て方法，

(2) 仕切り方法．前者は小さい問題に便利であるが，対象の数が大きくなればすぐに無理になる．

(1) 各対象 $j = 2, \cdots, N$ に対し： $X_{j,\ j-1} + X_{j+1,\ j-1} + \cdots\ X_{M,\ j-1} = 0$

$I = 2, 3, \cdots . N$, $j = 2, 3, . . . ,$ min $\{I, M\}$ に対し，次の制約式を加える： (2) $X_{Ij} \leq X_{I-1,1} + X_{I-1,2} + X_{I-1,\ j-1}$.

### 11.7.1　割当問題としてモデル

一般的な分類問題のための最も明らかなモデルは，次の 0/1 の決定変数の定義を基本とする：

$X_{Ij}$ = 1（対象 j がグループ I に割り当てる），0（それ以外）

このモデルの欠点は，多くの対象モデルがあり，多くの代替案の最適解があることである．それらの全ては，本質的である．例えば，ゴルファーa，b，c をグループ 1 に，ゴルファーe，f，g，h をグループ 2 に割り当てることと，ゴルファーa，b，c および D をグループ 2 にゴルファーe，f，g および h をグループ 1 に割り当てることは本質的に同じである．これらの代替的な最適解は，典型的な IP になり計算時間がかかる．

私達はこの対象性を削除し，最適性の損失無しで次の制約に同意すればこれらの代替的最適性を除去できる：

①対象 1 はグループ 1 にだけ割り当てることができる；

②対象 2 はグループ 1 または 2 に割り当てられ，もし対象 1 が 1 に割り当てられた場合は 1 にのみ割り当てられる．

② 一般的に，オブジェクト I がグループ I に割り当てられる場合に限り，対象 j は I<j になるグループ I に割り当てられる．

これは次のことを意味する：

$X_{II}$ = 1（対象 I がそのグループで一番小さなインデックスであり，かつ XIj が *I ≤ j のときのみ定義される）*

現在，我々はグループ問題の幾つかのみて，それらを解く方法を示す．

**11.7.2 マッチングの問題（グループサイズが 2）**

ルームメイトの割当問題は，グループサイズが 2 の単純なグルーピング問題である．この例は，多くの大学で学校の 1 年間の最初の年あるいは一年生が入学する前に，問題解決が迫られている．寮の部屋は，通常 1 年生の 2 人に同室を割り振る．生徒のペアはどうすれば良いだろうか？ 1 つのアプローチは，可能な限りのあらゆる組み合わせの生徒に対して，その組み合わせの利点を表すスコアを計算することである．考慮するスコアには，次のようなものがある：喫煙者と非喫煙者，夜遅く勉強するのが好きな学生と早起きして午前中に勉強する学生を一緒にしないことである．6 人の学生（ジョー，ボブ，チャック，エド，エヴァン，ショーン）の可能な全てのペアであるコアを計算してみよう．

**「スカラーモデル」**

```
! MaxImIze total score of PaIrs selecteD;
 MAX=9*X_JOE_BOB+7*X_JOE_CHUCK+4*X_JOE_ED
+6*X_JOE_EVAN+3*X_JOE_SEAN+2*X_BOB_CHUCK
+8*X_BOB_ED+X_BOB_EVAN+7*X_BOB_SEAN
+3*X_CHUCK_ED+4*X_CHUCK_EVAN+9*X_CHUCK_SEAN
+5*X_ED_EVAN+5*X_ED_SEAN+6*X_EVAN_SEAN;
 ! Each stUDent mUst be In exactly one PaIr;
 [JOE]X_JOE_BOB+X_JOE_CHUCK+X_JOE_ED
+X_JOE_EVAN+X_JOE_SEAN=1;
[BOB]X_JOE_BOB+X_BOB_CHUCK+X_BOB_ED
+X_BOB_EVAN+X_BOB_SEAN=1;
[CHUCK]X_JOE_CHUCK+X_BOB_CHUCK+X_CHUCK_ED+X_CHUCK_EVAN+X_CHUCK_SEAN=1;
[ED]X_JOE_ED+X_BOB_ED+X_CHUCK_ED+X_ED_EVAN+X_ED_SEAN=1;
[EVAN]X_JOE_EVAN+X_BOB_EVAN+X_CHUCK_EVAN+X_ED_EVAN+X_EVAN_SEN
    =1;
[SEAN]X_JOE_SEAN+X_BOB_SEAN+X_CHUCK_SEAN
+X_ED_SEAN+X_EVAN_SEAN=1;
 ! AssIgnments mUst be bInary, not fractIonal;
@BIN(X_JOE_BOB);@BIN(X_JOE_CHUCK);@BIN(X_JOE_ED);
@BIN(X_JOE_EVAN);@BIN(X_JOE_SEAN);@BIN(X_BOB_CHUCK);
@BIN(X_BOB_ED);@BIN(X_BOB_EVAN);@BIN(X_BOB_SEAN);
@BIN(X_CHUCK_ED);@BIN(X_CHUCK_EVAN);@BIN(X_CHUCK_SEAN);
```

```
@BIN(X_ED_EVAN);@BIN(X_ED_SEAN);@BIN(X_EVAN_SEAN);
```

解は次の通りである．

```
 Global optimal soluthon found.
 Objecthve value:      23.00000
 Extended solver steps:     0
 Total solver iterations:  0
Variable           Value         Reduced Cost
X_JOE_BOB       0.000000          -9.000000
X_JOE_CHUCK     0.000000          -7.000000
X_JOE_ED        0.000000          -4.000000
X_JOE_EVAN      1.000000          -6.000000
X_JOE_SEAN      0.000000          -3.000000
X_BOB_CHUCK     0.000000          -2.000000
X_BOB_ED        1.000000          -8.000000
X_BOB_EVAN      0.000000          -1.000000
X_BOB_SEAN      0.000000          -7.000000
X_CHUCK_ED      0.000000          -3.000000
X_CHUCK_EVAN    0.000000          -4.000000
X_CHUCK_SEAN    1.000000          -9.000000
X_ED_EVAN         0.000000          -5.000000
X_ED_SEAN         0.000000          -5.000000
X_EVAN_SEAN       0.000000          -6.000000
Row    Slack or SUrPlUs      DUal PrIce
   1       23.00000            1.000000
JOE        0.000000            0.000000
BOB        0.000000            0.000000
CHUCK      0.000000            0.000000
ED         0.000000            0.000000
EVAN       0.000000            0.000000
SEAN       0.000000            0.000000
```

変数 X_JOE_BOB があるのに，変数 X_BOB_JOE がない．これはドアの表示順を考慮せず，単に 2 人の順不同なペアに注目しているからである．寄宿舎は 6 学生でなく，60 人や 600 人という規模である．この場合集合を用いて定式化する必要がある．次の LINGO モデルは集合の利用法を説明している．モデルではペアの順番は考慮していない．LINGO は，集合を用いて全てのメンバー（この場合のペア）を設定する次の便利な条件を用いる．

```
 PXP( PERSON, PERSON)| &1 #LT# &2: VALUE, X;
```

PXP( PERSON, PERSON)は，集合PXPに6人の全ての組み合わせの6*6 人のペアが含まれる．条件節の「 |&1 #LT# &2」は，ペアの最初の学生を表すインデックス「&1」は，2番目の学生のインデックス「&2」より小さいことを示す．

「集合モデル」

```
MODEL: ! (roomates.lng);
 SETS:
  PERSON;
! Joe rooms wIth Bob means the same as
  Bob rooms wIth Joe, so we neeD only the
   UPPer trIangle;
  PXP( PERSON, PERSON)| &1 #LT# &2: VALUE, X;
 ENDSETS
 DATA:
  PERSON = Joe  Bob  ChUck  ED  Evan Sean;
   ValUe =  9     7     4     6    3   ! Joe;
                  2     8     1    7   ! Bob;
                        3     4    9   ! ChUck;
                              5    5   ! ED;
                                  6 ; ! Evan;
  ENDDATA
! MaxImIze the valUe of the matchIngs;
  MAX = @SUM(PXP(I,J):Value(I,j)* X(I,J));
! Each Person aPPears In exactly one match;
   @FOR( PERSON( K):
     @SUM(PXP(K,J):X(K,J))+@SUM(PXP(I,K):X(I,K))=1;);
! No tImesharIng;
  @FOR(PXP(I,J): @BIN(X(I,J)));
END
```

制約式「@SUM(PXP(K,J):X(K,J))+@SUM(PXP(I,K):X(I,K))=1」は，2つの項目がある．最初の条件は，kが最初のペアで，2番目はkが2番目のペアであるものの和になる．例えば，上のスカーラーモデルは，エドは最初の2つのペアで最初に，3つのペアで2番目に表れている．

次の解で，23の値が見つかった．

```
Variable            Value
X( JOE, EVAN)       1.000000
X( BOB, ED)         1.000000
X( CHUCK, SEAN)     1.000000
```

ジョーはエバンと，ボブはエドと，ショーンはチャックとペアを組んでこのモデルを組む．このモデルは，数百人規模の大きなオブジェクトに拡張できる．

マッチングは別の観点から，これ以降の「安定的なマッチング」を参照．

### 11.7.3 2つ以上のメンバーのいるグループ

以下に示す例は，最近遭遇した発電会社と石炭供給業者の問題である．あなたは石炭の供給業者であり，消費者が管理経営する「消費者電気事業（PTTP）」と非独占的な契約を有している．あなたは，艀でPTTPに石炭を供給する．PTTPとあなたの契約は，あなたが渡す石炭はトンあたり少なくとも13000のBTU’s（British thermal Unit），0.63%未満の硫黄，6.5%未満の灰， 7%未満の湿気を含んでいなければいけない．PTTPは上記の条件を満たさなかったら，艀を受け入れない．現在利用できる艀のデータを示す．

| 艀 | BTU/ton | いおう% | 灰% | 湿度% |
|---|---|---|---|---|
| 1 | 13029 | 0.57 | 5.56 | 6.2 |
| 2 | 14201 | 0.88 | 6.76 | 5.1 |
| 3 | 10630 | 0.11 | 4.36 | 4.6 |
| 4 | 13200 | 0.71 | 6.66 | 7.6 |
| 5 | 13029 | 0.57 | 5.56 | 6.2 |
| 6 | 14201 | 0.88 | 6.76 | 5.1 |
| 7 | 13200 | 0.71 | 6.66 | 7.6 |
| 8 | 10630 | 0.11 | 4.36 | 4.6 |
| 9 | 14201 | 0.88 | 6.76 | 5.1 |
| 10 | 13029 | 0.57 | 5.56 | 6.2 |
| 11 | 13200 | 0.71 | 6.66 | 7.6 |
| 12 | 14201 | 0.88 | 6.76 | 5.1 |

艀の1，5，10だけがPPTの条件を満たすので，取引に問題がある．どうすれば良いだろう？PTTPとの契約の細部を読んだ後，上記の条件をどのように解釈するかをPTTPと議論を開始した．議論の後で，PTTPは上記の条件を解釈し直し，3台までの艀を一組として受け入れることに同意した．すなわち，N組の艀を受け入れて平均品質が上記の品質要求を満たせば，n個の艀を受け入れ可能である．艀をいかに組み立てるか指定できる．各艀は，1つあるいは複数である場合もある．同じ組の艀は，同じ搬送にしなければいけない．

元のデータを見て，私達は12台の艀があるのに，最初の4つの艀で表される4つの明確なタイプの艀がある．現実には，これを期待する：石炭のタイプと関連した特定の鉱山に対応する各艀のタイプがある．

艀のグループ化問題を割当の問題としてモデル化することは，比較的簡単である．必要な決定変数は．「x（I，j）：グループjに割り当てられた艀Iの数」である．

私達は同じタイプの艀の間を区別しない条件を保った．12 台の艀があるが，私達はデータを見ないで 6 グループに制限できることを知っている．推論は次のとおりである： 7 つの空でないグループがあることを仮定しなさい．少なくともグループの 2 つは，一つの手である．2 組の 1 枚札が実行可能なら，それらの結合で得られるグループがある．従って，私達は次の LINGO モデルを書いてもいい．

```
MODEL:
SETS:
  MINE: BAVAIL;
  GROUP;
  QUALITY: QTARG;
!  Composition of each type of MINE load;
  MXQ( MINE, QUALITY): QACT;
!assignment of wich MINE to which group;
!no distinction between types;
  MXG( MINE, GROUP):X;
ENDSETS
DATA:
  MINE =   1..4;
      ! Barges available of each type(or mine);
  BAVAIL = 3 4 2 3;
  QUALITY =  BTU,  SULF, ASH, MOIST;
      ! Quality targets as upper limits;
  QTARG = - 13000  0.63  6.5   7;
      ! Actual qualities of each mine;
  QACT =-13029  0.57  5.56 6.2
           -14201  0.88  6.76 5.1
           -10630  0.11  4.36 4.6
           -13200  0.71  6.66 7.6;
! We need at most six groups;
  GROUP = 1..6;
  GRPSIZ = 3;
ENDDATA
! Maximize no. of barges assigned;
MAX = @SUM( MXG: X);
! Upper limit on group size;
 @FOR( GROUP(J): @SUM( MINE( I): X(I, J))
      <= GRPSIZ;);
```

```
! Assign no more of a type than are available;
 @FOR( MINE(I): @SUM( GROUP( J): X( I, J))
      <= BAVAIL( I));
! The blending constraints for each group;
 @FOR( GROUP(J):
  @FOR( QUALITY ( H):
   @SUM( MINE( I): X( I, J) * QACT( I, H)) <=
   @SUM( MINE( I): X( I, J) * QTARG( H));));
! barges must be integers;
@FOR( MXG: @GIN( X));
END
```

解から，PTTP に 3 隻でなく，10 艘を売ることである．

```
Objective Value:      10.00000
Variable            Value
X( 1, 1)         1.000000
X( 2, 2)         2.000000
X( 3, 2)         1.000000
X( 1, 4)         2.000000
X( 4, 4)         1.000000
X( 2, 5)         2.000000
X( 3, 5)         1.000000
```

例えば，グループ 1 は単に 1 つのタイプ 1 の艀である．グループ 2 はタイプ 2 の 2 艘とタイプ 3 の 1 艘である．実際のアプリケーションでは，約 60 艘であった．この問題を解決するには時間がかかるので，次の節でこの問題に対応する「分割する方法」について説明する．

**分割定式化による解法**

IP は解を得ることが困難であるが，このような問題に直面するときに便利で有用なルールを 2 つ紹介する：

1）区別できないものを区別しない；

2）部分問題を事前に解く

この問題は，マッチングかグループ化（割り当ての反対概念）アプローチで，「手で」解決することができる．「副問題を事前に解く」を適用して，私達は 4 つのタイプから選ばれる 3 台より少ない艀の全ての実行可能な組合せを列挙してもいい．「区別しないを適用」規則を適用し，私達は（1，1）と（2，2，2）のような組合せを考慮しない．なぜならそのようなセットは，1 枚札セットが（例えば，（1）および（2））実行可能である場合のみ実行可能である．従って，質を無視して，4 組の 1 枚札セット，6 組の 2 枚札セット，4 つの明瞭な 3 枚札がある（例えば，（1，2，3））．そし

て 12 個のペアの 3 枚札(例えば，(1，1，2))が合計 26 の組合せがある．唯一の実行可能な組合せが次のとおりであり，手で示すことは難しくない：(1)，(1，1，4)，および(2，2，3)． 従って，販売される艀の数を最大にするために私達が解決したいと思う IP は次のとおりである

```
Max = S001 + 3 * S114 + 3 * S223;
 S001 + 2 * S114 <= 3 ;
  !( type 1 の艀数);
   2 * S223  <= 4 ;
  !( type 2 の艀数);
  S223  <= 2 ;
  !( type 3 の艀数);
  S114  <= 3 ;
  !( type 4 の艀数);
```

これは容易に解けて，*S001* = 1，*S114* = 1，*S223* = 2 で目的関数は 10 になる．

与えられたデータで，高々10 個の艀で輸送する．それらをマッチさせる方法の一つは，品質要求を満足する．次の解が得られた．

| 艀 | 集合の平均品質 | | | |
|---|---|---|---|---|
| | BTU% | 硫黄% | 灰% | 湿度% |
| 1 | 13029 | 0.57 | 5.56 | 6.2 |
| 4，5，10 | 13086 | 0.6167 | 5.927 | 6.667 |
| 2，3，6 | 13010 | 0.6233 | 5.96 | 4.933 |
| 8，9，12 | 13010 | 0.6233 | 5.96 | 4.933 |

### 11.7.4　可変数のメンバーによるグループ（割り当て版）

多くのアプリケーションで，グループの大きさが可変であることが多い．金融会社は，金融サービスのオブジェクト（例えば，住宅ローン）をパッケージにして販売したい．パッケージの特徴は，少なくとも合計金額が 100 万ドル以上になることである．私たちはこの条件を満たすパッケージ数を最大にしたい．まず最初に，割り当て問題で定式化するが，次の命令ががキーである．

```
OXO( OBJECT, OBJECT)| &1 #LE# &2: X;
```

これは変数 X(I，J) のインデックスが必ず I ≤ J に制限している．この変数の解釈は次のようになる．

**X(I，J) = 1 （対象 J が対象 I と同じグループに割り当てられる），**

**X(I，I) = 1 （対象 I がそのグループの一番小さなインデックス）**

```
MODEL:
! Object bundling model. (OBJBUNDL);
```

```
!A broker has a nUmber of loans of size from $55,000 to
$946,000.
 The broker would like to group the loans into packages
 so that each package has at least $1M in it,
 and the number of packages is maximized;
! Keywords: bundling, financial, set packIng;
SETS: OBJECT: VALUE, OVER;
   OXO( OBJECT, OBJECT)| &1 #LE# &2: X;
 ENDSETS
 DATA:
 OBJECT=ABCDEFGHIJKLMNPQR;
 VALUE=910 870 810 640 550 250 120 95 55 200 321 492 567 837
193 364 946;
! The value in each bundle must be >= PKSIZE;
     PKSIZE = 1000;
 ENDDATA
!-----------------------------------------------;
! Definition of variables;
!  X( I, I) = 1 If object I is lowest numbered
              object in its package;
!  X( I, J) = 1 If object j is assigned to package I;
! Maximize number of Packages assembled;
  MAX = @SUM( OBJECT( I): X( I, I));
  @FOR( OBJECT( K):
! Each object can be assigned to at most one package;
     @SUM( OXO( I, K): X( I, K)) <= 1;
! A Package must be at least PSIZE in size;
     @SUM( OXO( K, J): VALUE( J) * X( K, J))
          - OVER( K) = PKSIZE * X( K, K););
! The X( I, J) must = 0 or 1;
  @FOR( OXO( I, J): @BIN( X( I, J)););
END
```

解は次のとおりである:

```
Variable           Value
X( A, A)        1.000000
X( A, H)        1.000000
X( B, B)        1.000000
```

X( B, F)　　　1.000000
X( C, C)　　　1.000000
X( C, J)　　　1.000000
X( D, D)　　　1.000000
X( D, Q)　　　1.000000
X( E, E)　　　1.000000
X( E, L)　　　1.000000
X( G, G)　　　1.000000
X( G, K)　　　1.000000
X( G, M)　　　1.000000
X( I, I)　　　1.000000
X( I, R)　　　1.000000
X( N, N)　　　1.000000
X( N, P)　　　1.000000

8 個のパッケージ（*AH*, *BF*, *CJ*, *DQ*, *EL*, *IR*, *JN*, *GKM*, *NP*）が作られた．たまたま全ての対象がどこかのパッケージに含まれている．全ての対象を 8 個のパッケージに含める代替案がある．そこで，他の代替案による最適解として第 2 の基準が考えられる（例えば，最大のパッケージはできるだけ 100 万に近い）． 最悪のパッケージは，120 万のため 100 万の目標を超える BF である．

### 11.7.5　メンバーが可変なグループ化（詰め合わせ版）

代替アプローチは，全ての可能なあるいは興味のある実行可能なグループをまず数え上げ，次のモデルを解くことである．

**Maximize　選んだグループの値**
**subject to:各対象は，選んだグループの一つに必ず含まれる**

この定式化は，割り当てによる定式化で容易に解けるという利点がある．この優位性は，グループサイズが 3 以上の大きなモデルで発揮される．もし n 個の対象とグループサイズが k の場合，n!/(k!（n−k)!）個の明確なグループがある．例えば，n = 50 で k = 3 の場合，19,600 個の候補のグループがある．

この定式化は，「混合（composite）変数」のアイデアを用いている．これは，元の定式化が計算困難な問題で役に立つ．特定の混合変数を 1 にすることは，元の変数の特定の組み合わせを 1 にすることである．その変数の実行可能な組み合わせに対応する混合変数だけを作る．これは，元のモデルを LP で緩めて解くことで現れる実数解の多くを削減する．混合変数のアイデアは，「列生成」と呼ばれることもある．ネットワークモデルのパスによる定式化は，混合変数を使用した例である．

**例： 金融商品のパッケージ化（改定版）**

この問題をパッキング・アプローチで解決するためのモデルは，100 万以上の条件を満たす全ての可能なパッケージまたはグループを構築することである．IP/LP の一般形式は次のとおりである：

**MAX 選択したパッケージの値**

**ST：**

**各オブジェクトは選択されたパッケージの少なくとも一つに現れる**

定式は以下のとおりで，パッケージを疎な集合で表す．4 個以上の対象を含むパッケージを検討する必要はないと仮定する．この定式化の魅力的な点は，各可能なグループに任意のスコアを与えることができる．特に次の定式は，融資目標額の＄ 100 万を超えているものに超過分の 2 乗をペナルティとして課している．

```
MODEL:
! Object bundling model.  (OBJBUNDH);
! A broker has a number of loans of size from $55,000 to
$946,000.
 The broker would like to group the loans into packages
 so that each package has at least $1M in it, preferably
 not much more,
 and the number of packages is maximized;
! Keywords: bundling, financial, set packIng;
 SETS:
 OBJECT: VALUE;
 ENDSETS
 DATA:
  OBJECT=A  B  C  D  E  F  G  H  I  J  K L  M  N  P  Q  R;
  VALUE = 910 870 810 640 550 250 120 95 55 200 321 492 567
837 193 364 946;
! The valUe In each bUnDle mUst be >= PKSIZE;
    PKSIZE = 1000;
 ENDDATA
 SETS:
 !Enumerate all 2,3, and 4 object unordered sets ≤ Package
size;
  BNDL2(OBJECT,OBJECT)|&1 #LT# &2 #AND#
        (VALUE(&1)+VALUE(&2)) #GE# PKSIZE:X2,OVER2;
  BNDL3(OBJECT,OBJECT,OBJECT)|&1 #LT# &2 #AND# &2 #LT# &3
    #AND# (VALUE(&1)+VALUE(&2)+VALUE(&3) #GE# PKSIZE):
```

```
    X3,OVER3;
  BNDL4(OBJECT,OBJECT,OBJECT,OBJECT)|&1 #LT# &2
    #AND#&2 #LT# &3 #AND# &3 #LT# &4 #AND# (( VALUE(&1) +
    VALUE(&2)+VALUE(&3)+VALUE(&4))#GE# PKSIZE):X4,OVER4;
 ENDSETS
!----------------------------------------------;
!ComPUte the overage of each bUnDle;
  @FOR( BNDL2(I,J):
    OVER2(I,J)=VALUE(I)+VALUE(J)-PKSIZE;);
  @FOR(BNDL3( I,J,K):
    OVER3(I,J,K)=VALUE(I)+VALUE(J)+VALUE(K)-PKSIZE);
  @FOR( BNDL4( I,J,K,L):
    OVER4(I,J,K,L)=VALUE(I)+VALUE(J)+VALUE(K)+VALUE(L)-
PKSIZE;);
! Maximize score of packages assembled. Penalize a package
that is over the minimum package size;
  MAX=@SUM(BNDL2(I,J):X2(I,J)*(1-(OVER2(I,J)/PKSIZE)^2))
      +@SUM( BNDL3( I,J,K):
           X3(I,J,K)*(1-(OVER3(I,J,K)/PKSIZE)^2))
      + @SUM( BNDL4(I,J,K,L):
          X4(I,J,K,L) * (1-(OVER4(I,J,K,L)/PKSIZE)^2));
  @FOR( OBJECT(M):
! Each object M can be in at most one of the selected
bdndles;
    @SUM( BNDL2( I, J)| I #EQ# M #OR# J #EQ# M: X2( I, J))
  + @SUM( BNDL3( I, J, K)| I #EQ# M #OR# J #EQ# M #OR# K
#EQ# M: X3( I, J, K))
  + @SUM( BNDL4( I, J, K, L)|
        I #EQ# M #OR# J #EQ# M #OR# K #EQ# M #OR# L #EQ# M:
          X4(I,J,K,L)) <= 1;);
! The X's must = 0 or 1;
  @FOR( BNDL2( I, J): @BIN( X2( I, J));); 
  @FOR( BNDL3( I, J, K): @BIN( X3( I, J, K)););
  @FOR( BNDL4( I, J, K, L): @BIN( X4( I, J, K, L)););
END
```

解は次の通りである.

```
Global optimal solution found at iteration: 19
```

```
Objective value: 7.989192
VarIable             ValUe
X2( A, H)          1.000000
OVER2( A, H)       5.000000
X2( B, P)          1.000000
OVER2( B, P)       63.00000
X2( C, F)          1.000000
OVER2( C, F)       60.00000
X2( D, Q)          1.000000
OVER2( D, Q)       4.000000
X2( E, L)          1.000000
OVER2( E, L)       42.00000
X2( I, R)          1.000000
OVER2( I, R)       1.000000
X2( J, N)          1.000000
OVER2( J, N)       37.00000
X3( G, K, M)       1.000000
OVER3(G,K,M)      8.000000
```

この結果は，前の割り当てによる解より幾分バランスが良い．一番大きな超過は，120,000 でなく 63,000 である． これはグループ化の方法が，大きなパッケージにペナルティを与えるようにしたからである．

### 11.7.3　メンバーが可変なグループ化（切断版）

パッキング・アプローチが有効な分野に，製紙や製鉄業における切断問題がある．第 7 章の例を再び用いる．私たちは，3 つの異なる原材料幅から手動で 8 個の最終製品の切断パターンをすべて列挙する．下の定式は自動的に設定可能な全てのパターンを列挙する．各原材料に対して，最終製品の合計幅が原材料幅以下になるような全ての可能な 1 から 7 の最終製品のグループを列挙している．

この定式化の 1 つの重要な機能は，例えば 20 以上の対象で計算時間を短くするショートカットがある点だ．ショートカットを説明するため次の 3 つの宣言文を考察しよう：

```
! Enumerate all possible cutting patterns with 1 fg;
  rxf(rm,fg)| lenf(&2) #le# lenr(&1): x1;
! Enumerate all possible patterns with 2 fg;
  rxf2( rxf, fg) |
     &2 #le# &3 #anD# (lenf(&2) + lenf(&3) #le#
lenr(&1)): x2;
! Enumerate all possible patterns with 3 fg;
```

```
      rxf3( rxf2, fg)| &3 #le# &4
           #anD# (lenf(&2) + lenf(&3)+ lenf(&4) #le#
    lenr(&1)): x3;
```

「rxf(rm,fg)」は，LINGO に 1 原材料と 1 最終製品の全ての組み合わせを作成する．条件「 | lenf(&2) #le# lenr(&1) 」は，最終製品の幅が原材料幅より大きくなる原材料(Index &1) と最終製品(Index &2) の組み合わせを禁じている． 例えば(R36, F38) の組み合わせは， rxf のメンバーでない． R36 がペアの最初の項目である rxf の集合には，次の 4 組 {( R36, F34), (R36, F24), ( R36, F15), ( R36, F10)} が含まれる．

次に，2個の最終製品の全ての実現可能な組み合わせを作成する次の宣言文を考える．:

rxf2( rm, fg, fg) | &2 #le# &3 #anD# (lenf(&2) + lenf(&3) #le# lenr(&1))

条件「&2 #le# &3」は，最終製品の順序に関心を持たないこと，すなわち同じ最終製品が2個組みあわされることを許す．条件「lenf(&2) + lenf(&3) #le# lenr(&1)」は，集合rxf2の要素が実現可能であることを制約する．この宣言は有効であるが，用いない．なぜだろうか？ 代わりに宣言rxf2( rxf, fg)を用いる．後者を用いるのは計算上の理由からである．後者で集合rxf の要素と最終製品の全ての組み合わせを考える．原材料がr36の場合を考える． もしrxf2(rm,fg,fg) を用いると，LINGO は2組の最終製品の8*8=64個の組み合わせを考え，次の4個{(r36,f24,f10), (r36,f15,f15),(r36,f15,f10), (r36,f10,f10)}を残す． もしrxf2(rxf,fg)を用い，原材料がR36の時，LINGOは4*8=32組の組み合わせを考えればよい． 集合rxfは，ペアの最初のメンバーがR36であるのは4組しかないためである．rxf3以上になると，計算時間は劇的に改善される．

```
! Cutting stock solver(cutgent);
! Keywords: cutting stock;
SETS:
! Each raw material has a size(length) and quantity;
  rm: lenr, qr;
! Ditto for each finished good;
  fg: lenf, qf;
ENDSETS
DATA:
! Describe the raw materials available;
  rm, lenr, qr =
  R72  72   9999
  R45  48   9999
```

```
  R36  36   9999;
! Describe the finished goods needed;
  fg, lenf, qf =
  F60  60   500
  F56  56   400
  F42  42   300
  F38  38   450
  F34  34   350
  F24  24   100
  F15  15   800
  F10  10  1000;
ENDDATA
SETS:
! Enuerate all possible cutting patterns with 1 fg;
  rxf(rm,fg)| lenf(&2) #le# lenr(&1): x1;
! Enumerate all possible patterns with 2 fg;
 rxf2(rxf,fg) |&2 #le#&3#anD#
 (lenf(&2) + lenf(&3) #le# lenr(&1)): x2;
! Enumerate all possible patterns with 3 fg;
  rxf3(rxf2,fg)|&3#le#&4#anD#(lenf(&2)+lenf(&3)+lenf(&4)#le#
 lenr(&1)): x3;
! Enumerate all possIble patterns wIth 4 fg;
  rxf4(rxf3,fg)| &4 #le# &5 #anD#
   (lenf(&2)+lenf(&3)+lenf(&4)+lenf(&5)#le#lenr(&1)): x4;
! Enumerate all possible patterns with 5 fg;
rxf5(rxf4,fg)|&5#le#&6#anD#(lenf(&2)+lenf(&3)+lenf(&4)+lenf(
&5)+lenf(&6)#le# lenr(&1)): x5;
! Enumerate all possible patterns with 6 fg;
 rxf6(rxf5,fg)|&6#le#&7#anD#(lenf(&2)+lenf(&3)+
lenf(&4)+lenf(&5)
      +lenf(&6)+lenf(&7) #le# lenr(&1)): x6;
ENDSETS
! Minimize length of material used;
MIN = @SUM(rxf(r,f1):lenr(r)*x1(r,f1))
    + @SUM(rxf2(r,f1,f2):lenr(r)*x2(r,f1,f2))
    + @SUM(rxf3(r,f1,f2,f3):lenr(r)*x3(r,f1,f2,f3))
    +@SUM(rxf4(r,f1,f2,f3,f4):lenr(r)*x4(r,f1,f2,f3,f4))
```

```
    +@SUM(rxf5(r,f1,f2,f3,f4,f5):
      lenr(r)*x5(r,f1,f2,f3,f4,f5))
    +@SUM(rxf6(r,f1,f2,f3,f4,f5,f6):
   lenr(r)*x6(r,f1,f2,f3,f4,f5,f6));
! We have to satisfy each finished good demand;
 @FOR( fg(f):
     @SUM(rxf(r,f): x1(r,f))
   + @SUM(rxf2(r,f1,f2)| f #eq# f1: x2(r,f1,f2))
   + @SUM(rxf2(r,f1,f2)| f #eq# f2: x2(r,f1,f2))
   + @SUM(rxf3(r,f1,f2,f3)| f #eq# f1: x3(r,f1,f2,f3))
   + @SUM(rxf3(r,f1,f2,f3)| f #eq# f2: x3(r,f1,f2,f3))
   + @SUM(rxf3(r,f1,f2,f3)| f #eq# f3: x3(r,f1,f2,f3))
   +@SUM(rxf4(r,f1,f2,f3,f4)|f #eq# f1:x4(r,f1,f2,f3,f4))
   +@SUM(rxf4(r,f1,f2,f3,f4)|f #eq# f2:x4(r,f1,f2,f3,f4))
   +@SUM(rxf4(r,f1,f2,f3,f4)|f #eq# f3: x4(r,f1,f2,f3,f4))
   +@SUM(rxf4(r,f1,f2,f3,f4)|f #eq# f4:x4(r,f1,f2,f3,f4))
   +@SUM(rxf5(r,f1,f2,f3,f4,f5)|
     f#eq#f1:x5(r,f1,f2,f3,f4,f5))
   +@SUM(rxf5(r,f1,f2,f3,f4,f5)|f#eq#f2:
    x5(r,f1,f2,f3,f4,f5))
   +@SUM(rxf5(r,f1,f2,f3,f4,f5)|f#eq#f3:
     x5(r,f1,f2,f3,f4,f5))
   +@SUM(rxf5(r,f1,f2,f3,f4,f5)|f#eq#f4:
    x5(r,f1,f2,f3,f4,f5))
   +@SUM(rxf5(r,f1,f2,f3,f4,f5)|f#eq#f5:
     x5(r,f1,f2,f3,f4,f5))
   + @SUM(rxf6(r,f1,f2,f3,f4,f5,f6)| f #eq# f1:
     x6(r,f1,f2,f3,f4,f5,f6))
   + @SUM(rxf6(r,f1,f2,f3,f4,f5,f6)|f #eq# f2:
     x6(r,f1,f2,f3,f4,f5,f6))
   +@SUM(rxf6(r,f1,f2,f3,f4,f5,f6)|f #eq# f3:
     x6(r,f1,f2,f3,f4,f5,f6))
   + @SUM(rxf6(r,f1,f2,f3,f4,f5,f6)|f #eq# f4:
    x6(r,f1,f2,f3,f4,f5,f6))
   + @SUM(rxf6(r,f1,f2,f3,f4,f5,f6)|f #eq# f5:
   x6(r,f1,f2,f3,f4,f5,f6))
   +@SUM(rxf6(r,f1,f2,f3,f4,f5,f6)|f #eq# f6:
```

```
    x6(r,f1,f2,f3,f4,f5,f6)) >= qf(f););
! We cannot use more raw material than is available;
  @FOR( rm( r):
     @SUM(rxf(r,f): x1(r,f))
   + @SUM(rxf2(r,f1,f2): x2(r,f1,f2))
   + @SUM(rxf3(r,f1,f2,f3): x3(r,f1,f2,f3))
   + @SUM(rxf4(r,f1,f2,f3,f4): x4(r,f1,f2,f3,f4))
   + @SUM(rxf5(r,f1,f2,f3,f4,f5): x5(r,f1,f2,f3,f4,f5))
  +@SUM(rxf6(r,f1,f2,f3,f4,f5,f6):x6(r,f1,f2,f3,f4,f5,f6))
        <= qr(r););
 ! Can only run integer quantities of each pattern;
     @FOR(rxf:  @GIN(x1));
     @FOR(rxf2: @GIN(x2));
     @FOR(rxf3: @GIN(x3));
     @FOR(rxf4: @GIN(x4));
     @FOR(rxf5: @GIN(x5));
     @FOR(rxf6: @GIN(x6));
```

もしLINGOのGenerateコマンドでモデルをスカラー表現すれば，56インチの制約式は次のようになることが分かる．:

```
   X2_R72_F56_F15+X2_R72_F56_F10+X1_R72_F56>=400 ;
```

Solve をクリックし，次の解を得る：

```
Global optimal solution found at iteration:       31
   Objective value:        119832.0
    Variable                            Value
X2( R72, F60, F10)                  500.0000
X2( R72, F56, F15)                  400.0000
X2( R72, F38, F34)                  350.0000
X3( R72, F42, F15, F15)             186.0000
X3( R72, F38, F24, F10)             100.0000
X4( R72, F42, F10, F10, F10)        114.0000
X4( R45, F15, F10, F10, F10)        2.000000
X6( R72, F15, F15, F10, F10, F10, F10)     13.00000
```

### 11.7.4　メンバーが可変グループか（車両ルーティング）

次の車両ルーティングの例は，列またはグループを生成する一環として，最適化計算を行う例である．11.6.3 節で用いた車両ルーティングの変形である．このモデルの大部分は，7 箇所に停止する全ての最小の実行可能な運行を考える．実現可能な運行は，各停車地に配達される荷物の合計が車両の容量 18 パレットを超えないこ

とである.最小化は，走行距離を最小化する停止地の順番を決めることである.実行可能な最小の運行計画が与えられると，以下の単純な IP モデルを解けばよい：

**MIN　選択した旅行の費用を；**

**ST：　各停止地に対し：1 つの旅行だけがこの停車地を含む；**

この小さな例では，15 の停止地を持つので，15 の制約と，実現可能な最小の旅行数に等しい約 7300 個の 0/1 の整数変数をもつ.

実現可能な最小の旅行（PSET2,PSET3 など）と，それらの関連する距離（D2( ), D3( ), など）をどう作成するかの技巧的な方法を紹介する.このため D4(I,J,K,L)変数を定義する. D4(I,J,K,L)変数は，デポを出発し，停止地 I,j,k を訪問後 L にいく最小の距離を表す.もし，DIST(I,J) が距離行列であり，D3 が D4 と同じように表されるなら，Held and Karp(1962)は，D4 がダイナミック・プログラミングで表されることを示した.

D4(I,J,K,L)=min[D3(I,J,K)+DIST(K,L),D3(I,K,J)+DIST(J,L),
D3(J,K,I)+DIST(I,L)]

完全な定式以下のとおりである.

```
MODEL:   ! (vrgenext);
! The Vehicle Routing Problem (VRP) occurs in many service
systems such as delivery, customer pick-up, repair and
maintenance. A fleet of vehicles, each with fixed
capacity, starts at a common depot and returns to the
depot after visiting locations where service is demanded.
This LINGO model generates all feasible one vehicle routes
and then chooses the least cost feasible multi-vehicle
combination;
SETS:
 CITY: Q;
 ! Q(I) = amount required at city i(given),
       must be delivered by just 1 vehicle;
 CXC( CITY, CITY): DIST;
 ! DIST(I,J) = Distance from city i to city J
ENDSETS
DATA:
CITY= ChI Den Frsn Hous KC LA Oakl Anah Peor Phnx Prtl Rvrs
Sacr SLC Sntn Sbrn;
! Amount to be Delivered to each customer;
Q= 0 6 3 7 7 18 4 5 2 6 7 2 4 3 3 2 ;
! city 1 represents the common depot, i.e. Q( 1) = 0;
```

```
! Distance from city I to city J is same(but need not be)
from J to I;
DIST=  ! To City;
!ChI  Den Frsn Hous KC LA Oakl Anah Peor Phnx Prtl Rvrs Sacr
SLC Sntn SBrn From;
   0  996 2162 1067  499 2054 2134 2050  151 1713 2083 2005
2049 1390 1187 1996 ! Chcago;
 996    0 1167 1019  596 1059 1227 1055  904  792 1238 1010
1142  504  939 1001 ! デンバー;
2162 1167    0 1747 1723  214  168  250 2070  598  745  268
162  814 1572  265 ! フレスノ;
1067 1019 1747 0  710  1538  1904  1528   948  1149  2205  1484
1909 1438  197 1533 ! ヒューストン;
 499  596 1723  710    0 1589 1827 1579  354 1214 1809 1535
1742 1086  759 1482 ! K-City;
2054 1059  214 1538 1589    0  371   36 1943  389  959   54
376  715 1363   59 ! L.A.;
2134 1227  168 1904 1827  371    0  407 2043  755  628  425
85  744 1729  422 ! Oaklnd;
2050 1055  250 1528 1579   36  407    0 1933  379  995   45
412  711 1353   55 ! Anahm;
 151  904 2070  948  354 1943 2043 1933    0 1568 2022 1889
1958 1299 1066 1887 ! Peoria;
1713  792  598 1149 1214  389  755  379 1568    0 1266  335
760  648  974  333 ! Phnix;
2083 1238  745 2205 1809  959  628  995 2022 1266    0 1001
583  767 2086  992 ! Prtlnd;
2005 1010  268 1484 1535   54  425   45 1889  335 1001    0
430  666 1309   10 ! Rvrsde;
2049 1142  162 1909 1742  376   85  412 1958  760  583  430
0  659 1734  427 ! Scrmto;
1390  504  814 1438 1086  715  744  711 1299  648  767  666
659    0 1319  657 ! SLC;
1187  939 1572  197  759 1363 1729 1353 1066  974 2086 1309
1734 1319    0 1307 ! Sant;
1996 1001  265 1482 1533   59  422   55 1887  333  992   10
427  657 1307    0 ! Sbrn;;
```

```
! VCAP Is the caPacIty of a vehIcle In 40「x48「 Pallets;
VCAP = 18;
ENDDATA
SETS:
! Enumerate all sets of various sizes of cities that are load feasible;
 SET2(CITY,CITY)|&1#GT#1#AND#&1#LT#&2
#AND#(Q(&1)+Q(&2)#LE#VCAP)::
SET3(SET2,CITY)|&2#LT#&3
#AND#(Q(&1)+Q(&2)+Q(&3)#LE#VCAP)::
SET4(SET3,CITY)|&3#LT#&4
#AND#(Q(&1)+Q(&2)+Q(&3)+Q(&4)#LE#VCAP)::
SET5(SET4,CITY)|&4#LT#&5
#AND#(Q(&1)+Q(&2)+Q(&3)+Q(&4)+Q(&5)#LE#VCAP)::
SET6(SET5,CITY)|&5#LT#&6#AND#
(Q(&1)+Q(&2)+Q(&3)+Q(&4)+Q(&5)+Q(&6)#LE#VCAP)::
SET7(SET6,CITY)|&6#LT#&7#AND#(Q(&1)+Q(&2)+Q(&3)+Q(&4)+Q(&5)+
Q(&6)+Q(&7)#LE#VCAP)::
!Enumerate all partially ordered sets with a specific city as the last one;
PSET2(CITY,CITY)|&1#GT#1#AND#&1#NE#&2
#AND#(Q(&1)+Q(&2)#LE#VCAP):D2,X2;
PSET3(SET2,CITY)|&1#NE#&3#AND#&2#NE#&3#AND#(Q(&1)+Q(&2)+Q(&3
)#LE#VCAP):D3,X3;
PSET4(SET3,CITY)|&1#NE#&4#AND#&2#NE#&4#AND#&3#NE#&4
#AND#(Q(&1)+Q(&2)+Q(&3)+Q(&4)#LE#VCAP):D4,X4;
PSET5(SET4,CITY)|&1#NE#&5#AND#&2#NE#&5#AND#&3#NE#&5
 #AND#&4#NE#&5#AND#(Q(&1)+Q(&2)+Q(&3)+Q(&4)+Q(&5)#LE#VCAP):D
5,X5;
PSET6(SET5,CITY)|&1#NE#&6#AND#&2#NE#&6#AND#&3#NE#&6#AND#&4#N
E#&6#AND#&5#NE#&6#AND#(Q(&1)+Q(&2)+Q(&3)+Q(&4)+Q(&5)+Q(&6)#L
E#VCAP):D6,X6;
PSET7(SET6,CITY)|&1#NE#&7#AND#&2#NE#&7#AND#&3#NE#&7#AND#&4#N
E#&7#AND#&5#NE#&7#AND#&6#NE#&7#AND#(Q(&1)+Q(&2)+Q(&3)+Q(&4)+
Q(&5)+Q(&6)+Q(&7)#LE#VCAP):D7,X7;
ENDSETS
```

```
! Compute shortest distance to visit all cities in PSET, and
ending UP at last city in each partially ordered set, Using
Held & Karp DP. Essential idea:
   DS(S,t)=minimumdistancetovisitallcitiesin
Sandthenendupatt.Therecursionis:
DS(S,t)=min{kinS:DS(S-k,k)+DIST(k,t)};
@FOR(PSET2(I,J):
D2(I,J)=DIST(1,I)+DIST(I,J);
@BIN(X2););
@FOR(PSET3(I,J,K):
!@SMIN is the min of a list of scalers. D3(I,J,K)=min cost
of starting at 1,visiting I and J in some order, and then K;
D3(I,J,K)=@SMIN(D2(I,J)+DIST(J,K),D2(J,I)+DIST(I,K));
@BIN(X3););
@FOR(PSET4(I,J,K,L):
 !D4(I,J,K,L)=min cost of starting at 1, visiting I, J, & K
in some order, and then L;
  D4(I,J,K,L) =
    @SMIN( D3(I,J,K)+DIST(K,L),
           D3(I,K,J)+DIST(J,L),
           D3(J,K,I)+DIST(I,L));
   @BIN(X4););
@FOR(PSET5(I,J,K,L,M):
  D5(I,J,K,L,M) =
   @SMIN( D4(I,J,K,L)+DIST(L,M),
          D4(I,J,L,K)+DIST(K,M),
          D4(I,K,L,J)+DIST(J,M),
          D4(J,K,L,I)+DIST(I,M));
    @BIN(X5););
@FOR(PSET6(I,J,K,L,M,N):
  D6(I,J,K,L,M,N) =
    @SMIN( D5(I,J,K,L,M)+DIST(M,N),
           D5(I,J,K,M,L)+DIST(L,N),
           D5(I,J,L,M,K)+DIST(K,N),
           D5(I,K,L,M,J)+DIST(J,N),
           D5(J,K,L,M,I)+DIST(I,N));
  @BIN(X6););
```

```
@FOR(PSET7(I,J,K,L,M,N,P):
  D7(I,J,K,L,M,N,P) =
      @SMIN( D6(I,J,K,L,M,N)+DIST(N,P),
             D6(I,J,K,L,N,M)+DIST(M,P),
             D6(I,J,K,M,N,L)+DIST(L,P),
             D6(I,J,L,M,N,K)+DIST(K,P),
             D6(I,K,L,M,N,J)+DIST(J,P),
             D6(J,K,L,M,N,I)+DIST(I,P));
  @BIN(X7););
! and finally, the optimization model... Min cost of routes
chosen, over complete routes ending back at 1;
   Min=
+@SUM(PSET2(I,J)|J#EQ#1:D2(I,J)*X2(I,J))
+@SUM(PSET3(I,J,K)|K#EQ#1:D3(I,J,K)*X3(I,J,K))
+@SUM(PSET4(I,J,K,L)|L#EQ#1:D4(I,J,K,L)*X4(I,J,K,L))
+@SUM(PSET5(I,J,K,L,M)|M#EQ#1:D5(I,J,K,L,M)*X5(I,J,K,L,M))
+@SUM(PSET6(I,J,K,L,M,N)|N#EQ#1:D6(I,J,K,L,M,N)
   *X6(I,J,K,L,M,N))+@SUM(PSET7(I,J,K,L,M,N,P)P#EQ#1:
D7(I,J,K,L,M,N,P)*X7(I,J,K,L,M,N,P));
! Each city must be on exactly one complete route;
@FOR(CITY(I1)|I1#GT#1:
+@SUM(PSET2(I,J)|J#EQ#1#AND#(I#EQ#I1):X2(I,J))
+@SUM(PSET3(I,J,K)|K#EQ#1#AND#(I#EQ#I1#OR#J#EQ#I1):
X3(I,J,K))+@SUM(PSET4(I,J,K,L)|L#EQ#1#AND#
(I#EQ#I1#OR#J#EQ#I1#OR#K#EQ#I1):X4(I,J,K,L))
+@SUM(PSET5(I,J,K,L,M)|M#EQ#1#AND#
(I#EQ#I1#OR#J#EQ#I1#OR#K#EQ#I1#OR#L#EQ#I1):
X5(I,J,K,L,M))+@SUM(PSET6(I,J,K,L,M,N)|N#EQ#1#AND#
(I#EQ#I1#OR#J#EQ#I1#OR#K#EQ#I1#OR#L#EQ#I1
#OR#M#EQ#I1):X6(I,J,K,L,M,N))
+@SUM(PSET7(I,J,K,L,M,N,P)|P#EQ#1#AND#
(I#EQ#I1#OR#J#EQ#I1#OR#K#EQ#I1#OR#L#EQ#I1
#OR#M#EQ#I1#OR#N#EQ#I1):X7(I,J,K,L,M,N,P))=1;);
```

たった4秒で次の解を得た.

```
Global optimal solution found at iteration: 134
   Objective value:    17586.00
                Variable                            ValUe
```

X2( LA, CHI)　1.000000
X3( KC, PEOR, CHI)　1.000000
X4( DEN, HOUS, SNTN, CHI)　1.000000
X5( FRSN, OAKL, PRTL, SACR, CHI)　1.000000
X6(ANAH,PHNX,RVRS,SLC,SBRN,CHI)　1.000000

明らかな疑問は，「この方法がどこまでスケールアップできるか？」である．大きな課題は，可能性ある膨大な数の旅行を生成し保存する方法である．重要な考察は，旅行あたりの停止地数が関係する．もしこれが小さくて，例えば 3 つであれば，この旅行は我々の手におえます．典型的な車両ルーティング問題は，100 地点の停止地がある．100 個から 3 個の停止地を選ぶ最短距離の旅行の可能性の数は，100!/[3!97!] = 161,700 であり，これは IP で十分対応できます．

### 11.8　変数の製品を一直線に並べること

すでに 0/1 の整数変数の積が，例えば y1×y2 や y12 が簡単な操作で線形表現に変換できることを見てきた．この変換は，0/1 の整数変数に限らず連続変数にも一般化できる．例えば，モデルに x × y があったとする．y は 0/1 の整数変数で x は非負の連続変数とする．この非線形式を，線形式に置き換えたい．($M_x$) を x の上限，($M_y$) を積 $x \times y$ の上限， $P = x \times y$ とすれば，次の線形制約は正しい P の値を与える：

$$P \le x$$
$$P \le M_y \times y$$
$$P \ge x - M_x \times (1 - y)$$

Hanson & Martin ら(1990)は，商品をまとめた場合の価格設定に有用であることを示した．商品の一括価格は，大量ディスカウントにみられる．この例として，
(a)航空運賃，ホテル，レンタカー，旅行および食事，または
(b)PC，モニター，プリンターおよびハードディスクである．映画配給業者は，映画を個別でリースするより，一括してリースした方が利益が増えることを Stigler (1963)は示した．一括政策は，販売人がバンドルを組み立てることは容易であり，買い手がアンバンドルすることが困難であることを仮定している．さもなければ，転売者は割引でバンドルを買い，次に値上げで個々の商品を販売できる

### 11.8.1 例： 製品のバンドル

MicrolanD ソフトは，最近優秀なワプロソフトを開発し，すでに開発済みの表計算ソフトを補完しようとしている．MIcrolanD は熟慮した割引き政策で，2 製品の組合せを提供する．幾つかの展示会で製品を展示した後，MicrolanD は次のマーケッティング戦略を開発した．

| 市場 | サイズ（万） | 種々の製品混合を幾つかの市別に販売し価格政策を最適化する 表計算単独 | ワープロ単独 | 抱き合わせ |
|---|---|---|---|---|
| ビジネス | 7 | 450 | 110 | 530 |
| 官公庁 | 5 | 75 | 430 | 480 |
| 教育 | 6 | 290 | 250 | 410 |
| 家庭 | 4.5 | 220 | 380 | 390 |

私達は単に「顧客」として各市場区分を参照する．経済理論は，顧客が最大の「顧客余剰(customer surplus)」を与える製品を買うことを提案する．顧客余剰は，「顧客が製品のために喜んで支払う価格（reservation prIce，留保価格）」から「商品の市価」を引いたものである．例えば，表計算だけ，ワプロだけ，および抱き合わせの価格が，400，150，および 500 とすると，ビジネス／科学市場は消費者余剰が 50，－40，30 なので表計算だけ購入する．

この状態の一般的なモデルは次で定義される：

$R_{Ij}$ = 顧客 I のバンドル j の留保価格，

$N_I$ =「顧客のサイズ I」（市場 I の個々の顧客数），

$s_I$ =顧客 I が達成する消費者余剰，

$y_{Ij}$ = 1 顧客 I がバンドル j を買う．それ以外は 0，

$x_j$ = 販売人によるバンドル j の価格．

私達は空のバンドルをその他のバンドルとして扱うことで，顧客は必ずその一つを買うことにする．

Microland 社の営業は $x_j$ を選ぶことを望む：

$$\text{Maximize} \quad \sum_i \sum_j N_I \; y_{Ij} \; x_j$$

各顧客が丁度 1 バンドル買うという事実は下記のもとで実施される：

各顧客 I に対して： $\sum_j y_{Ij} = 1$

各顧客 I は，次の消費者余剰を達成する：$s_I = \sum_j (R_{Ij} - x_j) y_{Ij}$

各顧客 I は，消費者余剰が最大になるのバンドル j を買うであろう．これは次の顧客 I とバンドル j の制約で行われる：

$$s_I \geq R_{Ij} - x_j$$

目的関数関数および消費者余剰の制約式の難しさは，それらは積 $y_{Ij}\ x_j$ を含む点である．前の小さい例を参考に，$y_{Ij}\ x_j$ を $P_{Ij}$ によっておき替える．$M_j$ が $x_j$ の上限で，$P_{Ij}$ $P_{Ij} = y_{Ij}\ x_j$ とすれば次の制約式が必要である：

$P_{Ij} \leq x_j$

$P_{Ij} \leq R_{Ij}\ y_{Ij}$

$P_{Ij} \geq x_j - (1 - y_{Ij})M_j.$

モデルにこの修正を行うと，次のモデルを得る：

Maximize $\sum_i \sum_j N_i\ P_{ij}$

subject to:

各顧客 *I* に対し

$\sum_j y_{ij} = 1;$

顧客 *I*，と 抱き合わせ *j* に対し *j*:

$s_i \geq R_{ij} - x_j;$

各顧客 *I* に対し：

$s_i = \sum_j (R_{ij}\ y_{ij} - P_{ij});$

非線形状態 $P_{ij} = y_{ij}\ x_j$ を実施するためには：

$P_{ij} \leq x_j$

$P_{ij} \leq R_{ij}\ y_{ij}$

$P_{ij} \geq x_j - (1 - y_{ij})M_j.$

全ての i および j に対し：

$y_{ij}$ = 0 or 1

LINGO モデルは次の通りである：

```
MODEL:
SETS:
MARKET/B,L,E,H/:S,N;
ITEM/NONE,SO,WO,BOTH/:X;
CROSS(MARKET,ITEM):R,Y,P;
ENDSETS
DATA:
N = 7, 5, 6, 4.5;
R =    0 450 110 530
       0  75 430 480
       0 290 250 410
       0 220 380 390;
ENDDATA
MAX=@SUM(CROSS(I,J):P(I,J)*N(I));
!Make the PICK variables 0/1;
```

```
@FOR(CROSS:@BIN(PICK));
!Each customer buys exactly one bundle;
@FOR(MARKET(I):@SUM(ITEM(J):Y(I,J))=1);
!Each customer's achieved surplus, SURP, must be at
 least as good from every possible bundle;
@FOR(ITEM(I):@FOR(MARKET(J):X(I)+S(J)
>=R(J,I)));
!Compute the achieved surplus;
@FOR(MARKET(I):@SUM(ITEM(J):P(I,J)
-R(I,J)*Y(I,J))+S(I)=0);
! Each product variable Pij must be.. ;
!    <= Xj ;
!    <= Rij * Yij ;
!    >= Xj   M + M * Yij ;
M=600;!Maxpriceofanybundle;
@FOR(CROSS(I,J):P(I,J)<=X(J);
P(I,J)<=Y(I,J)*R(I,J);
R(I,J)>=X(J)-M+M*Y(I,J););
! Price of bundle should be < sum of component
 prices;
X(BOTH)<=X(SO)+X(WO);
! Price of bundle should be > any one component;
X(BOTH)>=X(SO);X(BOTH)>=X(WO);
END
```

Microlad 社問題の解は，次の価格になる：

| | 表計算だけ | ワープロだけ | 抱き合わせ |
|---|---|---|---|
| バンドル価格 | 410 | 380 | 410 |

従って，ビジネスと法律/教育市場は，両製品のバンドルを買う.家庭市場はワプロだけ買う.Microlad 社の総収入は 90,900,000 である.興味がある読者は，もしバンドルをしないなら，最も高い収入が 67,150,000 になることを示しなさい

## 11.8 論理的な条件の表現

条件を論理式を使用して示すことが便利な応用分野がある.論理変数は真か偽の値だけで取ることができる.同様に論理変数を含む論理式は，真か偽だけを扱う.論理式には，2 つの主要な論理演算子.AND.と.OR.がある.

論理式（A.AND.B）は，a および b の両方が真であるときだけ真であり.

論理式（A.OR.B）は，a および b の少なくとも 1 つが真であときだけ真である.

「aが真なら，bは真でなければならない」という包含を論理演算子（⇒）は「A⇒B」のように表す.

論理変数は二値変数で表される：

TRUE（真）は，1によって表される

FALSE(偽)は，0によって表される.

もしa，b，およびcが0/1変数なら，次の制約式の組合せが様々で基本的な論理式を表すのに使用することができる：

| **論理式** | **数学的論理式** |
|---|---|
| $C = A$.AND.$B$ | $C \le A$<br>$C \le B$<br>$C \ge A + B - 1$ |
| $C = A$.OR.$B$ | $C \ge A$<br>$C \ge B$<br>$C \le A + B$ |
| $A \Rightarrow C$ | $A \le C$ |

# 第13章　ポートフォリオ最適化

## 13.1 序論

ポートフォリオ分析と証券投資とを結びつけるのは，通常，期待収益とリスクという2つの規準である．投資家は，前者の規準すなわち収益率については高い値を，そして後者の規準であるリスクについては低い値を求めるだろう．ところでリスクの測度は複数存在するが，中で最も利用されるリスク測度は収益率の分散である．この測度には幾つかの問題があるが，以下ではこの測度をしっかり吟味することから始めたい．

## 13.2 マーコウィッツの平均/分散ポートフォリオ・モデル

マーコウィッツ(1959)やロイ(1952)によって紹介されたポートフォリオ・モデルでは，投資家は投資ポートフォリオを構築する際に2つのポイントを考慮すると仮定している．それらが，期待収益とその分散（すなわち，リスク）である．分散は，実現した収益の期待収益周りの散らばり具合を表わし，その際に実現した収益の値が期待収益よりも高いかあるいは低いかは問題にしない．この点で，マーコウィッツ・モデルは若干の問題がある．何故ならば，典型的な投資家は恐らく期待収益よりも低い側への変動，すなわち下方リスクのみを問題とするからである．マーコウィッツ・モデルは，代表的な2つの情報を必要とする．それらが，（1）投資先の候補である証券の期待収益の推定値，および（2）それらの収益の共分散行列である．共分散行列は，証券ごとの投資収益の変動の違いだけでなく，ある証券の収益率が他の投資対象の収益率とどのような関係にあるのかについても明らかにする．以下では，中級程度の統計学のテキストで扱われる分散や共分散の概念を前提としている．マーコウィッツ・モデルの利点の一つは，それが効率的な**2次計画法**(**QP**: Quadratic Programming)で解く事ができる点である．2次計画法とは，目的関数が2次関数で制約条件が一次式になるモデルに適用された名前である．従って目的関数は，高々$x^2$や$x \times y$のような2変数の積からなる項を含む．

2次計画法は，**線形計画法**(**LP**: Linear Programming)のアルゴリズムに小幅な修正を加えるだけ適応できるので，計算上，魅力的な手法である[1]．厳密性を欠く説明ではあるが，2次関数の1階微分が線形関数となる点で，唯一小幅な修正が必要である．ところで，LINGOには一般的な**非線形問題**(**NLP**: Non Linear Programming)を解く機能（general nonlinear solver）があるため，2次関数に限定することは有用ではあるが重要でない．

### 13.2.1 例

ここでは，マーコウィッツ(1959)に掲載されたデータを使用する．エッペン，

[1] 具体的な考え方は、LINDOのマニュアルでしか説明されていない．

グールド，＆ シュミット(1991)も同じデータを使用している．次の**表 13.1** は，3 社の配当を含む 12 年間の株価成長の推移である．

**表 13.1　株価の成長倍率**

| 年 | S&P500 | ATT | GMC | USX |
|---|---|---|---|---|
| 43 | 1.259 | 1.300 | 1.225 | 1.149 |
| 44 | 1.198 | 1.103 | 1.290 | 1.260 |
| 45 | 1.364 | 1.216 | 1.216 | 1.419 |
| 46 | 0.919 | 0.954 | 0.728 | 0.922 |
| 47 | 1.057 | 0.929 | 1.144 | 1.169 |
| 48 | 1.055 | 1.056 | 1.107 | 0.965 |
| 49 | 1.188 | 1.038 | 1.321 | 1.133 |
| 50 | 1.317 | 1.089 | 1.305 | 1.732 |
| 51 | 1.240 | 1.090 | 1.195 | 1.021 |
| 52 | 1.184 | 1.083 | 1.390 | 1.131 |
| 53 | 0.990 | 1.035 | 0.928 | 1.006 |
| 54 | 1.526 | 1.176 | 1.715 | 1.908 |

なお後で参照するため，スタンダード・アンド・プアーズの S&P500 株価指数も同時に載せておいた．例えば，ATT の株価は 1 年目に 30%上昇し，GMC の株価は 2 年目に 29%上昇している．標準的な統計パッケージを使い，12 年間のデータに基づいて計算したこれら 3 種類（ATT, GMC, および USX）の株価の共分散行列は**表 13.2** のようになる．

**表 13.2　分散共分散**

| | ATT | GMC | USX |
|---|---|---|---|
| ATT | 0.01080754 | 0.01240721 | 0.01307513 |
| GMC | 0.01240721 | 0.05839170 | 0.05542639 |
| USX | 0.01307513 | 0.05542639 | 0.09422681 |

同じデータから推定した ATT, GMC, および USX の**期待収益率（年率）**はそれぞれ，**0.0890833, 0.213667, および 0.234583** である．なお，収益率の計算は配当額を含んでいる．相関行列を使えば，2 つの確率変数がどのように一緒に変動しているのかをより明らかにできる．2 つの確率変数の間の相関係数は，2 変数

間の共分散をそれぞれの標準偏差の積で割ったものに等しい．ここで考察している 3 種類の投資対象の相関行列は，**表 13.3** の通りである．相関係数は-1 から+1 の間の値をとり，その値が+1 のとき 2 変数間の相関が高くなる．GMC と USX の相関が高いことに注意しよう．その一方で，ATT も GMC や USX と共に変動する傾向があるが，GMC と USX が共に変動するほど相関は高くない．

**表 13.3　相関行列**

| | ATT | GMC | USX |
|---|---|---|---|
| ATT | 1.0 | | |
| GMC | 0.4938955 | 891.0 | |
| USX | 0.4097277 | 180.7472291 | 211.0 |

ポートフォリオを構成するこれら 3 銘柄の比率を，記号 ATT，GMC，USX で表わす．さらに，年率 15%の収益を要求することとしよう．このモデルは，次のように書くことができる：

```
MODEL:
! 期末におけるポートフォリオの分散の最小化;
[VAR]MIN=.01080754*ATT*ATT+.01240721*ATT*GMC+.01307513*ATT*U
SX+.01240721*GMC*ATT+.05839170*GMC*GMC+.05542639*GMC*USX+.01
307513*USX*ATT+.05542639*USX*GMC+.09422681*USX*USX;
! 期首における予算を 100%利用する;
[BUD] ATT + GMC + USX = 1;
! 期末時点の資産について要求する条件;
[RET]1.089083*ATT+1.213667*GMC+1.234583*USX>=1.15;
END
```

このとき，2 つの制約条件が事実上同じ単位であることに注意されたい．すなわち，1 番目の条件は事実上「期首時点での在庫」に関わる制約であり，2 番目の条件は「期末時点での在庫」に関わる制約である．従って，期待収益に関わる制約は次にように表わすこともできる．

```
.0890833 * ATT + .213667 * GMC + .234583 * USX >= .15
```

これも完全に正しい表現ではあるが，2 つの制約条件を集約したこの表現は期末状態と期首状態を必ずしも同じ方法で表現しているとは言えない．よって，表現の一貫性を好む読者であれば，2 本の制約式を用いた前者の表現方法を好むかもしれない．

同様ではあるが，以下は SETS を用いた別の定式化である：

```
MODEL:
 SETS:
  ASSET: AMT, RET;
  COVMAT(ASSET, ASSET): VARIANCE;
 ENDSETS
```

```
 DATA:
  ASSET =AT  GMC  USXG;
! 共分散行列と期待収益率;
  VARIANCE = .01080754 .01240721 .01307513
             .01240721 .05839170 .05542639
             .01307513 .05542639 .09422681;
       RET = 1.0890833  1.213667  1.234583;
    TARGET = 1.15;
 ENDDATA
! 期末におけるポートフォリオの分散の最小化;
[VAR]MIN=@SUM(COVMAT(I,J):AMT(I)*AMT(J)*VARIANCE(I,J));
! 期首における予算を 100%利用する;
[BUDGET] @SUM( ASSET: AMT) = 1;
! 期末時点の資産について要求する条件;
[RETURN] @SUM( ASSET: AMT * RET) >= TARGET;
END
```

このモデルを解くと，以下の解を得る：

```
   Optimal solution found at step:          4
   Objective value:                0.2241375E-01
     Variable          Value         Reduced Cost
       TARGET        1.150000           0.0000000
   AMT( ATT)        0.5300926           0.0000000
   AMT( GMC)        0.3564106           0.0000000
   AMT( USX)        0.1134968           0.0000000
   RET( ATT)         1.089083           0.0000000
   RET( GMC)         1.213667           0.0000000
   RET( USX)         1.234583           0.0000000
          Row    Slack or Surplus      Dual Price
          VAR        0.2241375E-01       1.000000
       BUDGET        0.0000000          0.3621387
       RETURN        0.0000000         -0.3538836
```

この解が推奨するポートフォリオの構成は，ATT が約 53%，GMC が約 36%，そして USX が 11%程度になる．このとき，ポートフォリオの期待収益率は 15%で，その分散は 0.02241381，標準偏差は約 0.1497123 である．

このモデルは，単純に年収益率に基づく統計データを用いている．しかし実際には， 年データよりも月次データを利用して共分散を計算する事の方がより普通と思われる．また意思決定者による期待収益率の推定は，過去のデータよりも，資産の

期待収益の動向に関わる直近の情報に基づいているかもしれない．あるいは，共分散と期待収益の推定に細心の注意を払う可能性もある．例えば，標準偏差の推定に，直近のデータを使用することが可能かもしれない．かなりの過去まで遡る大規模なデータを用いて，相関行列を推計することも可能である．従って，相関行列と共分散行列との関係から共分散行列を導出することもできる．

### 13.3 二元的な目的：効率的フロンティアとパラメトリック分析

ある投資家は，リスクとリターンとの間に「正しい」トレードオフを決定する正確な方法はないという．従って我々は，専らそれらのトレードオフに関心がある．もしこの投資家がより高いリターンを要求するのであれば，一般的に言って，より高いリスクで「その代価を払わなければならない」．ファイナンスの専門用語的に言えば，**リターンとリスクの効率的フロンティア**を導出することである．期待収益の値を 0.0890833 から 0.234583 までとして，このときの最小分散ポートフォリオのモデルを解くと，3 銘柄からなるポートフォリオは**図 13.1** のようなトレードオフを表わす曲線になる．ここで，要求された期待収益が 1.21894 を超えたところで曲線に「ひざ」が出来ている点に注意されたい．これは，この箇所で ATT がポートフォリオから除外されてしまうために起きる現象である．

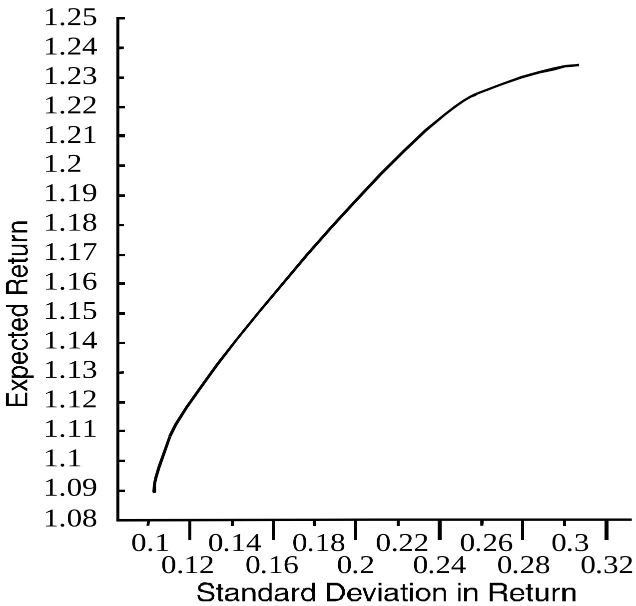


図 13.1 効率的フロンティア　（縦軸：期待収益率，横軸：収益率の標準偏差）

#### 13.3.1 安全資産を含んだポートフォリオについて

投資対象として利用可能な資産の 1 つが安全資産である場合，最適ポートフォリオ構成は特に簡単な形式になる．ここで，無リスクかつ年率 5%で投資する機会（例：政府短期財務省証券）を追加してみよう．従って，ここまでに分析してきた

前述の例に，分散がゼロで他の資産との共分散もゼロとなる第 4 の投資物件が追加されたことになる．さらに，年率 5%での投資額には制限が無いものとしよう．ここで，「要求する期待収益率を 15%から 5%に変更するとき，ポートフォリオ構成はどのように変化するだろうか?」という問題を考えてみる．

以下のモデルは，マーコウィッツのオリジナルモデルをわずかに一般化したものである．ここで，第 4 の投資対象（短期財務省証券，TBILL）が追加された点に注意されたい:

```
MODEL:
! 安全資産（政府短期財務省証券，TBILL）を追加する;
! 期末におけるポートフォリオの分散の最小化;
   [VAR] MIN = .01080754 * ATT * ATT +.01240721 * ATT * GMC
   +.01307513 * ATT * USX +.01240721 * GMC * ATT +.05839170
   * GMC * GMC +.05542639 * GMC * USX +.01307513 * USX * ATT
   +.05542639 * USX * GMC +.09422681 * USX * USX;
!  期首における予算を 100%利用する;
[BUD]  ATT + GMC + USX + TBILL = 1;
! 期末時点の資産について要求する条件;
[RET] 1.089083 * ATT + 1.213667 * GMC + 1.234583 * USX +
1.05 * TBILL >= 1.15;
END
```

あるいはまた，SETS を用いた定式化も可能であり，これは次のようになる:

```
MODEL:
SETS:
 ASSET: AMT, RET;
 COVMAT(ASSET, ASSET): VARIANCE;
ENDSETS
DATA:
ASSET=  ATT, GMC, USX, TBILL;
! 共分散行列;
  VARIANCE = .01080754 .01240721 .01307513   0
             .01240721 .05839170 .05542639   0
             .01307513 .05542639 .09422681   0
              0         0         0          0;
      RET = 1.0890833 1.213667 1.234583, 1.05;
      TARGET = 1.15;
ENDDATA
   ! 期末におけるポートフォリオの分散の最小化;
```

```
[VAR]MIN=@SUM(COVMAT(I,J):AMT(I)*AMT(J)*VARIANCE(I,J));
! 期首における予算を100%利用する;
[BUDGET] @SUM(ASSET: AMT) = 1;
! 期末時点の資産について要求する条件;
[RETURN] @SUM( ASSET: AMT * RET) >= TARGET;
END
```

上記の問題を解くと，以下のようになる：

```
Optimal solution found at step:        8
Objective value:                0.2080344E-01
Variable          Value         Reduced Cost
     ATT       0.8686550E-01    -0.2093725E-07
     GMC       0.4285285         0.0000000
     USX       0.1433992        -0.2218303E-07
   TBILL       0.3412068         0.0000000
     Row    Slack or Surplus      Dual Price
     VAR       0.2080344E-01      1.000000
     BUD       0.0000000          0.4368723
     RET       0.0000000         -0.4160689
```

ここで目標収益率が 15%であるにもかかわらず，ポートフォリオ全体の 34%以上が収益率 5%の無リスク資産に投資されている点に注意されたい．さらに，分散すなわちリスクが約 0.0224 から約 0.0208 に低下している点にも注意が必要である．

では，目標収益率を 5%に向けて低下させるとどうなるであろうか？ 目標収益率が 5%のとき，ATT，GMC，および USX の比率がゼロになることは明確だろう．解の単純な形は，3 銘柄（ATT，GMC，USX）の比率は同じ割合を保った状態で，無リスク資産とリスク資産の配分を変更することである．そこで，中間点を確認してみよう．次の例は，目標収益率を 10%にまで低下させたときの解である：

```
Optimal solution found at step:        8
Objective value:                0.5200865E-02
Variable          Value         Reduced Cost
     ATT       0.4342898E-01     0.0000000
     GMC       0.2142677         0.2857124E-06
     USX       0.7169748E-01     0.1232479E-06
   TBILL       0.6706058         0.0000000
     Row    Slack or Surplus      Dual Price
     VAR       0.5200865E-02      1.000000
     BUD       0.0000000          0.2184348
     RET       0.2384186E-07     -0.2080331
```

この解は，我々の推測に合致している．すなわち：

目標収益率を変化させても，リスク資産間の相対的投資比率は変化せず ATT のように 0.086 から 0.043 に半減し，無リスク資産が 0.34 から 0.67 と倍増し，リスク資産の 3 社は半減し，その間の配分が変化する．

この解から，丸め誤差を除けば，3 銘柄（ATT，GMC と USX）に投資された配分は，どちらの解でも同じである．従って，リスク選好が異なる 2 人の投資家であっても，彼らのポートフォリオに含まれるリスクのある株式の相対比率は同じであり，無リスク資産に割り当てられる比率が異なるだけである．実際，これらの例から観測された現象は，一般的に成立する．従って，異なる株式間での資金配分の意思決定は，株式への投資総額を所与とすれば，リスク選好の問題から分離することができる．トービンは 1981 年にノーベル賞を受賞したが，その主たる貢献は「分離定理」と呼ばれるこの特徴に気付いたことであった．よって，この特徴にあなたも気付いたならば，あなたもノーベル賞級の力量があるということだ．

### 13.3.2 シャープ比

（無リスク資産を含まない）リスク資産から構成されたあるポートフォリオ p について：

$R_p$=ポートフォリオ p の期待収益率，

$s_p$=ポートフォリオ p の収益率の標準偏差

$r_0$=無リスク資産の収益率

リスクとリターンが 2 つの指標であるのに対して，ポートフォリオ p の魅力を示す 1 つの指標がシャープ比（**Sharp ratio**）であり，以下の式で定義されている：

**Sharp ratio** $=(R_p - r_0) / s_p$

この式の意味を言葉で表現すれば，この指標は全ての資金を無リスク資産へ投資する場合に比べて，1 単位の追加的なリスクを受容することで，どれだけ追加的なリターンを得る事ができるのかを測っている．

この比率を最大化するポートフォリオには，ある明瞭な利点が存在する．ここで，次の記号を定義しよう：

**t = 我々が要求する目標収益率，**

**$w_p$ = 資産のうち，ポートフォリオ p に投資する比率（従って，残りは無リスク資産に投資されることになる）**

我々の収益目標を達成するために，収益率は次の式を満たさなくてはならない：

$(1 - w_p) * r_0 + wp * R_p = t$.

他方，投資全体の標準偏差は：

$R_p * s_p$

である．収益率に関する制約式を wp について解けば：

$w_p = (t - r_0) / (R_p - r_0)$.

を得る．従って，ポートフォリオの標準偏差を：

$w_p * s_p = [(t - r_0)/(R_p - r_0)] * s_p.$
で表わすことができる．ここでポートフォリオの標準偏差を最小化することは：
$\text{Min} [(t - r_0)/(R_p - r_0)] * s_p$
or
$\text{Min} [(t - r_0) * s_p/(R_p - r_0)].$
であり，このことは：
$\text{Max} (R_p - r_0)/s_p.$
と同値である．よって，リスクとリターンの選好に関わらず，リスク資産へ投資する資金は，リスク資産のポートフォリオのシャープ比を最大化するように投資すべきである．

以下は，無リスク利子率が5%である場合の例である：

```
MODEL:
!シャープ比を最大化する;
MAX=
(1.089083*ATT+1.213667*GMC+1.234583*USX-1.05)/
((.01080754*ATT*ATT+.01240721*ATT*GMC
+.01307513*ATT*USX+.01240721*GMC*ATT
+.05839170*GMC*GMC+.05542639*GMC*USX
+.01307513*USX*ATT+.05542639*USX*GMC
+.09422681*USX*USX)^.5);
!期首における予算を100%利用する;
[BUD]ATT+GMC+USX=1;
END
```

この問題の解は，次のようになる：

```
Optimal solution found at step:        7
Objective value:                  0.6933179
Variable            Value         Reduced Cost
     ATT        0.1319260         0.1263448E-04
     GMC        0.6503984         0.0000000
     USX        0.2176757         0.1250699E-04
```

ATT，GMC，およびUSXの相対的比率は，収益率5%の無リスク資産を明示的に含んだ先ほどのモデルとまったく同じである点に注意されたい．例えば，丸め誤差を除けば，相対的な比率は次のように同じになる：

$.1319262/.6503983 = 0.08686515/.4285286.$

## 13.4 ポートフォリオ・モデルの重要なバリエーション

単純なマーコウィッツ・モデルを利用するとき，気になる問題がある．それらは

次のような問題である：
a）考慮する資産の数が多くなるにつれて，共分散行列のサイズが非常に大きくなる．例えば証券数が 1000 個の場合，共分散行列の要素は単純計算で 100 万個，行列の対称性を考慮しても最低 50 万個ある．
b）データ更新（例えば，毎週）のたびにモデルを適用すると，ポートフォリオについて細かい，事によると重要でない「見直し」を頻繁にすることになる．
c）各証券の保有上限額についての制限が全くないこと．現実には，どのような資産であっても，例えばポートフォリオ総量の 5%を超えることが出来ないなどの法的あるいは制度上の規制が存在する．ポートフォリオ・マネージャーによっては，1 日の取引高を上限に設定することもあるだろう．保有額の規模が大きいと，仮にポートフォリオ・マネージャーが早急に保有株式を処分したいと思っても，売却で市場価格がかなり影響を受けることがあるからである．

分散構造を簡素化する方法として，2 つの手法が提案されている．それらが「シナリオ・アプローチ」と「ファクター・アプローチ」である．また，ポートフォリオの構成を頻繁に見直すべきか否かという問題については，取引費用の存在を明示的に扱うことが有用である．

### 13.4.1 取引費用があるポートフォリオについて

上述のモデルからは，新情報（すなわち，期待収益率と分散の新たな推定値）を入手したとき，どの程度頻繁にポートフォリオを調整すべきかについての示唆がほとんどない．仮に新情報を入手するたびに上述のモデルを適用するならば，我々は絶えずポートフォリオを調整することになる．この状況は，株式ブローカーにとっては手数料収入を得る機会となるため幸いである．しかしこの件には本来二次的な目的とすべきであろう．重要なことは，売買に伴うコストが存在することである．手数料費用は明らかに費用であるが，一見明らかでないもののビッド・アスク・スプレッド[2]も事実上は売買に伴う取引費用なのである．

以下で解説する手法では，取引費用を期首に支払うことを仮定している．ただし，支払時点を期末に変更することも簡単である．基本的なポートフォリオ・モデルに加えた主要な修正内容は次の通りである：
a）各証券について，「購入額」と「売却額」を示す変数を 2 つ追加する．
b）手数料を考慮した予算制約式に変更する．
c）各証券について，次の条件式を追加する：
証券 i への出資額 = （初期時点での証券 i の保有額）+（証券 i の購入額）–（証券 i の売却額）

---

[2] ビットアスクスプレッドとは、先物取引における「ビッドレート」と「オファーレート」の差のことを指す．ビットアスクスプレッドが小さいほど、その流動性が高いとされる．逆に流動性が低いとビットアスクスプレッドは開きやすくなる．（日経 225 先物取引ガイド）

### 13.4.2 例

購入額あるいは売却額の 1%を取引手数料として支払うと仮定する．現在のポートフォリオ構成は，ATT 株が 50%，GMC 株が 35%，そして USX 株が 15%とする．この構成比率は，現時点でほぼ最適比率である．このとき，さらなる調整は必要であろうか？　この問題は，以下のように表わすことができる：

```
MODEL:
 [VAR]MIN=.01080754*ATT*ATT+.01240721*ATT*GMC+.01307513*ATT*
    USX+.01240721*GMC*ATT+.05839170*GMC*GMC+.05542639*GMC*USX
    +.01307513*USX*ATT+.05542639*USX*GMC+.09422681*USX*USX;
[BUD]ATT+GMC+USX+.01*(BA+BG+BU+SA+SG+SU)=1;
[RET]1.089083*ATT+1.213667*GMC+1.234583*USX>=1.15;
[NETA]ATT=.50+BA-SA;
[NETG]GMC=.35+BG-SG;
[NETU]USX=.15+BU-SU;
END
```

制約式[BUD]は，資金を余すことなく利用することを示す．あるいは，[BUD]式に[NETA]，[NETG]，[NETU]の 3 式を代入することで，別の解釈も可能である．すなわち：

```
[BUD].01 * (BA + BG + BU + SA + SG + SU) + BA + BG + BU=SA +
SG + SU;
```

となるため，[BUD]制約は，手数料総額と購入金額の合計が売却金額に等しくなることを意味していると解釈できるの．

参考までに，以下に SETS 形式を用いたモデルの設定を示しておく：

```
MODEL:
SETS:
 ASSET: AMT, RETURN, BUY, SELL, START;
 COVMAT( ASSET, ASSET):VARIANCE;
ENDSETS
DATA:
  ASSET = ATT, GMC, USX;
 VARIANCE = .0108075 .0124072 .0130751
            .0124072 .0583917 .0554264
            .0130751 .0554264 .0942268;
   RETURN = 1.089083 1.213667 1.234583;
    START = .5 .35 .15;
   TARGET = 1.15;
ENDDATA
```

```
[VAR]MIN=@SUM(COVMAT(I,J):AMT(I)*AMT(J)*VARIANCE(I,J));
[BUD]@SUM(ASSET(I):AMT(I)+.01*(BUY(I)+SELL(I)))=1;
[RET]@SUM(ASSET:AMT*RETURN)>=TARGET;
@FOR(ASSET(I):[NET]AMT(I)=START(I)+BUY(I)-SELL(I););
END
```

このモデルの解は以下の通り：

```
 Optimal solution found at step:        4
 Objective value:                  0.2261146E-01
 Variable           Value        Reduced Cost
      ATT       0.5264748           0.0000000
      GMC       0.3500000           0.0000000
      USX       0.1229903           0.0000000
       BA       0.2647484E-01       0.0000000
       BG       0.0000000           0.4824887E-02
       BU       0.0000000           0.6370753E-02
       SA       0.0000000           0.6370753E-02
       SG       0.0000000           0.1545865E-02
       SU       0.2700968E-01       0.0000000
      Row    Slack or Surplus      Dual Price
      VAR       0.2261146E-01       1.000000
      BUD       0.0000000           0.3185376
      RET       0.0000000          -0.3167840
     NETA       0.0000000           0.3185376E-02
     NETG       0.0000000          -0.1639511E-02
     NETU       0.0000000          -0.3185376E-02
```

ATT 株については若干の買い増しを，GMC 株はそのままで，USX 株については若干の売却をこの解は示唆している．

### 13.4.3 税金を考慮したポートフォリオについて

税金の存在は投資分析をより複雑にするが，考慮しなくてはならない要因であろう．ポートフォリオへの税金の影響を例示するために，バンガード社が運用している 2 種類の類似する"グロース-アンド-インカム[3]"タイプの**表 13.4** のポートフォ

[3] 安定成長型ファンドと訳される．第一の目標として元本の長期的な成長（グロース）を目指し、第二の目標として配当収入（インカム）の確保を狙うミューチュアルファンド．安定した配当実績と将来的な利益成長力を有する大型株を購入していくのが普通だが、成長株よりも割安株のほうが高配当が得られるため、グロース志向というよりもバリュー志向の投資スタイルに近い．リスクの点では、S&P500 などの市場平均と歩調を合わせた動きをする傾向がある．

リオを 1 年間保有する場合を考えてみよう．ポートフォリオ S は税金を考慮せずに，その一方でポートフォリオ T は税引き後の運用成績を念頭に置いて管理された：

**表 13.4 ″グロース-アンド-インカム″タイプのポートフォリオ**

| ポートフォリオ | 分配金 | | 初期時点 | |
|---|---|---|---|---|
| | 配当 | キャピタル・ゲイン | 一株あたり株価 | 収益率 |
| S | $0.41 | $2.31 | $19.85 | 33.65% |
| T | $0.28 | $0.00 | $13.44 | 34.68% |

この例では恐らく偶然ではあるが，税金の影響を考慮したポートフォリオ T の方が，事実上，税引き前収益率も高くなっている．従って，税引き後，ポートフォリオ T の魅力はさらに高まる．配当所得とキャピタル・ゲイン所得への税率を共に 30%としよう．このとき，ポートフォリオ S が期末に支払う税金は投資金額 1 ドルあたり 0.3 *(0.41 + 2.31) /19.85 = 4.1 セントになるが，ポートフォリオ T は 0.3 * 0.28/13.44 = 0.6 セントだけである．

以下では，マーコウィッツ・モデルの一般化し，税金の影響を考慮する．ここでは，モデルの入力変数として，以下の項目が特に必要となる：

a) 各証券の株式保有数，

b) 証券ごとの一株あたり支払価格，および

c) 証券ごとの一株あたり推定配当額

このモデルと税金の影響を考慮しないモデルとでは，投資対象とする証券が相互に同じ程度に魅力的であるならば，次のような点で異なる結果に到達するだろう：

①配当課税を避けるため，配当を支払わない株式を購入する

②配当を支払う株式を売却する，そして

③キャピタル・ゲイン課税を避けるため，購入価格が高かった株式を売却する

これらは，2 つの証券がその他の点においてまったく同一であることを前提としている．以下では，モデルの完成度を高めるために取引費用を含め，さらに共分散行列の代わりに相関行列を使って証券の動きを表現する方法について例示する：

```
MODEL：
! ビッド・アスク・スプレッドと税金を考慮したマーコウィッツ・ポートフォリオ・モデルの一般化 (PORTAX) キーワード: マーコウィッツ, ポートフォリオ, 税金, 取引費用;
SETS：
ASSET：RET,START,BUY,SEL,APRICE,BUYAT,SELAT,DVPS,STD,X;
ENDSETS
```

プラス面は、市場全体に比べてボラティリティー（価格変動リスク）が小さいこと．マイナス面は、一般にトータルリターン（総合利回り）ベースでの飛び抜けた成績はあまり期待できないこと．(Traders Shop 参照)

```
DATA:
! オリジナルのマーコウィッツの例に基づくデータ;
 ASSET = TBILL ATT GMC USX;
! 期待成長倍率;
   RET =1.05 1.089083 1.21367 1.23458;
! 資産ごとの収益率の標準偏差;
   STD =0 .103959 .241644 .306964;
! 株式数で表わした初期時点でのポートフォリオ構成;
   START =10 50 70 350;
! 一株当たり購入価格;
   APRICE =1000 80 89  21;
! 現時点での一株あたり買値(ビッド)と売値(アスク);
   BUYAT  = 1000 87 89 27;
   SELAT  = 1000 86 88 26;
! 一株当たり配当金(推定値);
   DVPS =0 .5 0 0;
! 税率;
   TAXR =.32;
! 要求された成長倍率;
   TARGET =1.15;
 ENDDATA
 SETS:
  TMAT( ASSET, ASSET) | &1 #GE# &2: CORR;
 ENDSETS
 DATA:
! 相関行列;
CORR=1
    0 1
    0 0.4938961 1
    0 0.4097276 0.7472293 1 ;
 ENDDATA
!-----------------------------------------------;
!  ポートフォリオ収益率の分散を最小化する;
[OBJ] MIN=@SUM(ASSET(I):(X(I)*SELAT(I)*STD(I))^2)+
2*@SUM(TMAT(I,J)|I#NE#J:CORR(I,J)*X(I)*SELAT(I)*STD(I)
*X(J)*SELAT(J)*STD(J));
!予算制約(売却額で購入額と税金を補うこと);
```

```
[BUDC]@SUM(ASSET(I):SELAT(I)*SEL(I)-BUYAT(I)*BUY(I))>=TAXES;
[TAXC]TAXES>=TAXR*@SUM(ASSET(I):
DVPS(I)*X(I)+SEL(I)*(SELAT(I)-APRICE(I)));
!税引き後の収益率についての制約（ただし，売却時まで株価値上がり分への課税は考慮しない）;
[RETC]@SUM(ASSET(I):
RET(I)*X(I)*SELAT(I))-TAXES>=
TARGET*@SUM(ASSET(I):START(I)*SELAT(I));
!各証券の在庫関係を表わす式;
@FOR(ASSET(I):
[BAL]X(I)=START(I)+BUY(I)-SEL(I););
END
```

### 13.4.4 共分散構造を簡略化するためのファクター・モデルについて

シャープ(1963)は，株価の変動に有意な影響を与える「市場要因」の存在を示すことで，株価のランダムな挙動を表わすモデルを飛躍的に簡素化する方法を紹介した．ここでの市場要因とは，ダウ・ジョーンズ工業株平均や S&P500 株価指数，あるいは日経平均株価などである．次の記号を定義しよう：

$M$ = 市場ファクター，

$m_0$ = $E(M)$，

$s_0^2$ = $var(M)$，

$e_i$ = 証券 $i$ に固有なリターンの変動，

$s_i^2$ = $var(e_i)$．

ここで，シャープの近似は次の条件を仮定している（ただし E( ) は期待値を意味している)：

$E(e_i) = 0$

$E(e_i\ e_j) = 0$　　　$i \neq j$ のとき

$E(e_i\ M) = 0$.

このとき，シャープの単一ファクター・モデルによれば，1 ドルを証券 $i$ へ投資するときの収益率を次のように表わすことができる：

$u_i + b_i M + e_i$.

2 つのパラメータ $u_i$ と $b_i$ は，例えば証券 $i$ の収益率を市場ファクターへ最小二乗回帰することで推計が可能である．なお，パラメータ $b_i$ はこの証券のベータ値と呼ばれる．ここで：

$X_i$ = 資産 $i$ への投資額

を定義し，ポートフォリオ収益率の分散を次のように定義しよう：

$var[\sum X_i\ (u_i + b_i M + e_i)]$

$= var(\sum X_i\ b_i\ M) + var(\sum X_i\ e_i)$

$= (\sum X_i\ b_i)^2\ s_o{}^2 + \sum X_i{}^2\ s_i{}^2.$

従って，この問題は以下のように書くことができる：

**最小化の目的関数：** $Z^2\ s_o{}^2 + \sum X_i{}^2\ s_i{}^2$

**制約条件：**

$Z - \sum X_i\ b_i = 0$

$\sum X_i = 1$

$\sum X_i\ (u_i + b_i\ m_o) \geq r.$

従って，制約と変数を一つずつ追加することで，非対角要素に膨大な非ゼロ要素を含む共分散行列を，対角要素のみの共分散行列に集約できた．

実際には，市場関連リスク（システィマティック・リスク）を表すのに恐らく半ダースほどの要素が必要だろう．つまり，証券の収益は多くのインデックスやファクターと相関していると考えられる．典型的なファクターは，S&P500 などの市場指標や利子率，インフレ率，国防費，エネルギー価格，GNP，景気循環との相関，および各種鉱工業生産指数などである．例えば，債券価格は金利動向から強い影響を受けるだろう．

### 13.4.5 ファクター・モデルの例

ファクター・モデルは，証券収益率の分散を，一つあるいはそれ以上のファクターに起因する部分と，これらのファクターとは独立な部分との合計として表わす．ファクター・モデルを例示するために，ここでは重回帰分析を用いて，ATT，GMC，および USX の収益率を，同じ期間の S&P500 の収益率に回帰してみよう．モデルとその解は次のようになる：

```
MODEL:
! マルチ・ファクター・ポートフォリオ・モデル;
 SETS:
  ASSET: ALPHA, SIGMA, X;
  FACTOR: RETF, SIGFAC, Z;
  AXF(ASSET, FACTOR): BETA;
 ENDSETS
 DATA:
! 市場ファクター(s);
  FACTOR = SP500;
! 市場ファクター(s)の平均値と標準偏差;
   RETF = 1.191460;
   SIGFAC = .1623019;
! 証券ごとに市場ファクターへ回帰する;
!回帰式:Return(i)=Alpha(i)+Beta(i)*SP500+error(i);
ASSET =ATT  GMC  USX;
```

```
ALPHA = .563976   -.263502  -.580959;
BETA  = .4407264  1.23980   1.52384;
SIGMA = .075817    .125070   .173930;
! 要求する収益水準;
   TARGET = 1.15;
 ENDDATA
!-----------------------------------------------------;
! ポートフォリオ収益率の分散を最小化する;
[OBJ]MIN=@SUM(FACTOR(J):(SIGFAC(J)*Z(J))^2)
+@SUM(ASSET(I):(SIGMA(I)*X(I))^2);
! ポートフォリオのベータを計算する;
 @FOR(FACTOR(J):Z(J)=@SUM(ASSET(I):BETA(I,J)*X(I));); 
! 予算制約;
   @SUM( ASSET: X) = 1;
! 収益が満たすべき制約条件;

@SUM(ASSET(I):X(I)*ALPHA(I))+@SUM(FACTOR(J):Z(J)*RETF(J))>=T
ARGET;
END
```

以下は，解の一部である．：

| Variable | Value | Reduced Cost |
|---|---|---|
| TARGET | 1.150000 | 0.0000000 |
| X(ATT) | 0.5276550 | 0.0000000 |
| X(GMC) | 0.3736851 | 0.0000000 |
| X(USX) | 0.9865990E-01 | 0.0000000 |
| Z(SP500) | 0.8461882 | 0.0000000 |

| Row | Slack or Surplus | Dual Price |
|---|---|---|
| OBJ | 0.0229409 | 1.000000 |
| 2 | 0.0000000 | 0.3498846 |
| 3 | 0.0000000 | 0.3348567 |
| 4 | 0.0000000 | -0.3310770 |

ポートフォリオ構成がわずかに異なっている点に注意されたい．　しかしながら，ポートフォリオの分散の推定値は元のポートフォリオと極めて近い値である．

### 13.4.6 不確実性を表すシナリオ･モデルについて

シナリオ・アプローチでは，不確実性をモデル化するとき，将来起こりうる状況を少ない数の「シナリオ」で表現できるという仮定をおく．通常使われる，最も大雑把な状況の分類は楽観的，普通，悲観的の 3 状態であろう．シナリオ・アプロー

チの基礎となる着想のいくつかは，確率的計画法（stochastic programming）として知られる手法に由来している．この件については，例えば Madansky(1962)などを参照されたい．シナリオ・アプローチの大規模なポートフォリオへの適用例については，マーコウィッツ&ペロルド (1981)やペロルド(1984)がある．また，確率的計画法に関する一般的方法論については Infanger(1992)を参照のこと．なお，エッペン，マーチン，シュラージ[4](1988) は，シナリオ・アプローチを自動車産業の産出量計画に用いている．

ここで，次の記号を定義しよう：

**$P_s$ = シナリオ s が生じる確率，**

**$u_{is}$ = シナリオ s が発生したときの試算 i の収益率，**

**$X_i$ = 資産 i への投資，**

**$Y_s$ = シナリオが s であるときに実現する収益の平均からの乖離；**

**$= \sum_i X_i (u_{is} - \sum_q P_q u_{iq})$.**

この問題の代数的な表現は次のようになる．：

**最小化の目的関数： $\sum_s P_s Y_s^2$**

**制約条件：**

**$Y_s - \sum_i X_i (u_{is} - \sum_q P_q u_{iq}) = 0$（シナリオが s であるときに実現する収益の平均からの乖離）**

**$\sum_i X_i = 1$（予算制約）**

**$\sum_i X_i \sum_s P_s u_{is} \geq r$（要求する収益率）．**

ここで証券 i に固有な変動を $v_i^2$ とすれば，目的関数は次のようになる：

**最小化の目的関数：$\sum_i X_i^2 v_i^2 + \sum_s P_s Y_s^2$**

この式の重要な特色は，定式化の際に追加的な制約が必要ではあるものの，共分散行列が対角要素だけの，非常にまばらな行列になるという点である．

また一般に，ポートフォリオを構成する各証券の比率に，上限を設定したいと考えるかもしれない．もしも上限の設定や各証券に固有な変動の特定が無ければ，最適化の結果，シナリオ数と同数までの証券に投資することになるだろう．

### 13.4.7 例：不確実性を表わすためのシナリオ・モデル

ここでは再度，マーコウィッツのデータを使用する．ただし，12 年間を単純にそれぞれ異なる互いに独立なシナリオとして扱うことにする．このような場合，扱うデータ量を考えれば，以下のモデルの定式化では LINGO の‘SETS’形式を使用する方が便利である：

```
MODEL:
! シナリオ・ポートフォリオ・モデル;
```

---

[4] シュラーゲが正しいと日本人研究者から指摘を受けているが、初めからシュラージと言っていて、先方も訂正しないのでこのままにする。

```
SETS:
 SCENE/1..12/:PRB,R,DVU,DVL;
ASSET/ATT,GMT,USX/:X;
SXI(SCENE,ASSET):VE;
ENDSETS
DATA:
 TARGET = 1.15;
! オリジナルのマーコウィッツの例に基づくデータ;
 VE =
    1.300    1.225    1.149
    1.103    1.290    1.260
    1.216    1.216    1.419
    0.954    0.728    0.922
    0.929    1.144    1.169
    1.056    1.107    0.965
    1.038    1.321    1.133
    1.089    1.305    1.732
    1.090    1.195    1.021
    1.083    1.390    1.131
    1.035    0.928    1.006
    1.176    1.715    1.908;
! すべてのシナリオが等確率で発生することを想定;
 PRB= .08333 .08333 .08333 .08333 .08333 .08333
      .08333 .08333 .08333 .08333 .08333 .08333;
ENDDATA
! 目標とする最終値;
 [RET] AVG >= TARGET;
! 期末ポジションの期待値を計算する;
    AVG = @SUM( SCENE: PRB * R);
    @FOR(SCENE( S):
! 平均値からの乖離を計算する;
      DVU(S) - DVL(S) = R(S) - AVG;
! 異なるシナリオのもとでの値を計算する;
      R(S) = @SUM( ASSET( J): VE( S, J) * X( J)));
! 予算制約;
 [BUD] @SUM(ASSET: X) = 1;
 [VARI] VAR = @SUM(SCENE: PRB * ( DVU + DVL)^2);
```

```
[SEMIVARI] SEMIVAR = @SUM(SCENE: PRB * (DVL) ^2);
[DOWNRISK] DNRISK = @SUM(SCENE: PRB * DVL);
! 目的関数を VAR, SEMIVAR, あるいは DNRISK のいずれかに設定する;
[OBJ] MIN = VAR;
END
```

上記の問題を解くと，解の一部は以下のようになる：

```
Optimal solution found at step:         4
Objective value:                   0.2056007E-01
Variable            Value        Reduced Cost
 X( ATT)        0.5297389          0.0000000
 X( GMT)        0.3566688          0.0000000
 X( USX)        0.1135923          0.0000000
     Row    Slack or Surplus      Dual Price
     RET        0.0000000         -0.3246202
     BUD        0.0000000          0.3321931
     OBJ        0.2056007E-01      1.000000
```

この解に見覚えはないだろうか．既にお気づきの読者がいるかも知れないが，（丸め誤差の部分を除けば）この解は，（上述のシナリオ・モデルと同じ 12 年間のデータから計算した）共分散行列を使った当初のモデルから得たポートフォリオと同じである．事実，これは一般的な結論である．すなわち，もしも共分散行列と期待収益率が通常の統計学の公式を用いてオリジナルのデータから計算されたのであれば，同じデータに基づく共分散モデルとシナリオ・モデルは厳密に同じポートフォリオに到達する．

注意深い読者は，シナリオ・モデルの目的関数の値(0.02056)が共分散モデルのそれ(0.02241)よりも若干小さめになっている点に気付いたかも知れない．さらに非常に明敏な読者は，12×0.02054597/11 が丸め誤差を除けば 0.002241 に一致する点にも気付いているだろう．従って，目的関数の値の差異は単に，通常の統計パッケージが分散や共分散を計算するときサンプル数の N ではなく N − 1 で割る事に起因している．よって，より一般的な結論は，もしも共分散行列を N − 1 ではなく N で割って計算すると，共分散モデルとシナリオ・モデルは，目的関数の値も含めて同じ解に到達する．

## 13.5 分散以外のリスクの測度について

最も一般的なリスク測度は，分散(またはその平方根である標準偏差)であろう．この測度は，その収益率が対称的な分布に従う資産やいわゆる「効率的な」市場で取り引きされる資産のリスクを測る指標としては妥当な測度である．しかしながら，これらの 2 つの特徴が成立しないなら，分散には幾つかの欠陥が存在する．図

13.2 にある 4 種類の成長倍率の分布例をもとに考えてみよう.

投資 A, B, C は, いずれも期待成長が 1.10 倍 (従って期待収益率は 10%) で分散が 0.04 (あるいは平均周りの標準偏差は 0.20) であるため, 分散という観点からは同等と言える. しかしながら, 危険回避的な投資家にとっては, おそらく 3 種類の投資対象は同等ではないだろう. 分布(A)のもとでは, 初期投資からの損失は一切発生せず, さらに 20%の確率で投資が 1.5 倍 (すなわち収益率が 50%) になる確率もある. 他方で分布(C)のもとでは, 初期時点の価値の 0.7 倍 (従ってマイナス 30%の収益率) になる確率が 20%ある. よって, 危険回避的な投資家は投資 A を最も好み, そして投資 C を最も避ける傾向があるだろう. この例から明らかなように, 収益の分布が左右対称でない場合, 分散はリスクの測度としては必ずしも適切ではないことが分かる:

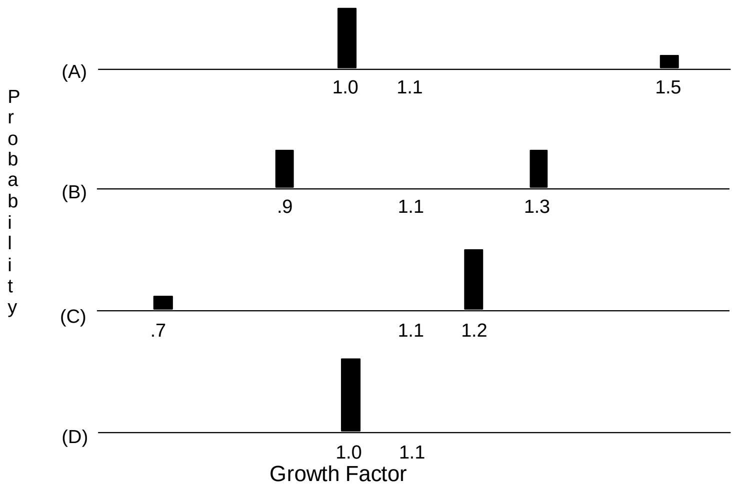


**図 13.2 成長倍率 (Growth Factor) の分布例**
**(縦軸:確率, 横軸:成長倍率)**

投資 D は, 投資 A よりも劣位な関係にあるため, 非効率な投資対象である. 仮に利用可能な投資対象が A と D だけであり, 我々の目的が最低 5%の期待収益率 (あるいは期待成長倍率が 1.05 倍) を達成する最小分散ポートフォリオの構築であるとする. このとき, A と D それぞれに 50%ずつ投資することが解になり, その結果, 分散は 0.01 (従って, 標準偏差は 0.1) になる. もしも投資 A に 100%投資した場合, 標準偏差は 0.20 になる. それにもかかわらず, 我々は投資 A へ 100%投資を選択するだろう. D よりも A に投資すれば, 収益率がより不確実になることは間違い無い. しかしながらいかなる状況においても, A の利益は D と同等あるいはそれ以上となる. 従って, 分散を規準にすると, 非効率的な投資を選択してしまう可能性がある.

主要な株式市場などのアクティブで効率的な市場では, 投資 A が投資 D よりも優位であることに投資家が気付くため, D のような投資は見当たらないだろう. 従って, 投資 D の市場価格は, その収益率と競合する他の投資対象の水準にまで低下す

ることになる．ところが，新たな物理施設（new physical facilities[5]）への投資を決定する場合，すべての投資候補を効率的にするような市場圧力が存在しないため，このような場合には分散というリスク測度はより不適切な指標になるだろう．

### 13.5.1 バリュー・アット・リスク（VaR）について

1994年，J.P.モルガンはRiskMetricsの導入によって，「バリュー・アット・リスク（VaR）」という概念を普及させた．VaRを利用するためには，次の2つの数値を指定しなければならない：

1）保有期間（リスク・ホライズン）．典型的には1日あるいは10日間といった具合に，損失のリスクを懸念する期間．そして，

2）確率の閾値．典型的には5%（あるいは1%）といった具合に，この以上を害ある結果とみなす水準．

従ってVaRは，最大5%（あるいは1%）の確率のもとでの，1日あたりの損失額と定義される．VaRに関する包括的なサーベイはJorion（2001)を参照されたい．VaRが注目される理由の一つに，バーゼル合意の一部として，欧州では銀行が抱えるポートフォリオ・リスクの測定手法として,本手法が推薦されていることがあげられる．銀行は，VaRで計測したリスクに比例して，資本準備金（capital reserve）を保有しなければならない．

**例：**明日までの1日で，我々のポートフォリオの価値が1万2000ドル上昇すると考えているとする．しかしながら実際には，ポートフォリオの価値は標準偏差が1万ドルの正規分布に従っているとする．従って，正規分布表から分布の左裾5%の確率に対応する値が，平均値よりも1標準偏差の1.644853倍だけ下方の結果になることが分かる．よって:

12000 -1.644853 * 10000 = -4448.50.

となり，VaRは$4448.50になる．

### 13.5.2 VaRの例

ATT/GMC/USCモデルに，VaRアプローチを適用してみよう．ここで，リスク許容度を5%とし，ポートフォリオのVaRを最小化することを目的とする．この問題は，閾値を最大化し，ポートフォリオの価値がその閾値以下になる確率を最大5%にするという問題と同値になる．

**分析：** 年末時点でポートフォリオの価値が正規分布に従っているものとする．このとき，左裾に5%の確率を与える境界値は，平均値から左方向へ1.64485標準偏差の点に対応する．このとき，バリュー・アット・リスクの最小化問題は，（平均値－1.64485×標準偏差）を最大化するようにポートフォリオの平均値と標準偏差

---

[5] physical facilities は、検索すると米国の大学関係者によく使われているが、翻訳でうまく説明できない．

を選択する問題と同じになる．この問題は，以下のように解くことが可能である：

```
MODEL: ! マーコウィッツ-VaR ポートフォリオ・モデル(PORTVAR);
 SETS:
  STOCKS: X, RET;
  COVMAT(STOCKS, STOCKS): VARIANCE;
 ENDSETS
 DATA:
  STOCKS = ATT  GMC  USX;
! 共分散行列と期待収益率;
  VARIANCE = .01080754 .01240721 .01307513
             .01240721 .05839170 .05542639
             .01307513 .05542639 .09422681;
       RET = 1.0890833  1.213667  1.234583;
    STARTW =1.0;  ! 期首時点の資産額;
       RHO =.05;  ! リスク許容度（ただしこの値は0.5未満である）;
 ENDDATA
!-----------------------------------------------------------;
! このリスク許容度に対応する標準偏差を計算する;
    RHO = @PSN(Z);
  @FREE(Z);
! バリュウ・アット・リスクの逆を最大化する;
[VAR] Max = ARET + Z * SD;
  ARET=@SUM(STOCKS:X*RET);
SD=(@SUM(COVMAT(I,J):X(I)*X(J)*VARIANCE(I,J)))^.5;
!  期首における予算を100%利用する;
[BUDGET] @SUM(STOCKS:  X) = STARTW;
END
```

この問題の解は次の通り：

| Variable | Value | Reduced Cost |
|---|---|---|
| RHO | 0.050000 | 0.0000000 |
| Z | -1.644853 | 0.0000000 |
| ARET | 1.109300 | 0.0000000 |
| SD | 0.111585 | 0.0000000 |
| X( ATT) | 0.843034 | 0.0000000 |
| X( GMC) | 0.125330 | 0.0000000 |
| X( USX) | 0.031636 | 0.0000000 |
| RET( ATT) | 1.089083 | 0.0000000 |

| | | |
|---|---|---|
| RET( GMC) | 1.213667 | 0.0000000 |
| RET( USX) | 1.234583 | 0.0000000 |
| Row | Slack or Surplus | Dual Price |
| 1 | -0.4163336E-16 | -1.081707 |
| VAR | 0.9257590 | 1.000000 |
| 3 | -0.2220446E-15 | 1.000000 |
| 4 | 0.0000000 | -1.644853 |
| BUDGET | 0.0000000 | 0.9257590 |

もしも ATT 株のみに投資するならば，ポートフォリオの分散は 0.01080754 になることに注意されたい．従って，標準偏差は 0.103959 であり，VaR は 1 – （1.089083 – 1.644853 * 0.103959） = 0.0818 となる．

このポートフォリオは，期待収益率と標準偏差の加重平均（ただし，標準偏差の加重はマイナス）を最大化しているため，効率的と言える．従って，もしもこれよりも期待収益率が高くかつ標準偏差が小さいポートフォリオがあれば，上記の解は上のモデルの目的関数を最大化することにはならない．

ここでもしリスク許容度を： RHO =0.1988 とすれば，本質的に ATT/GMC/USX 問題で考察したオリジナルのポートフォリオを得ることに注意されたい．

### 13.5.3 VaR の異常[6]

唯一つの数値でリスクを表現したいとき，VaR は有用で分かり易い指標である．しかしながら，変則的な特徴を考慮せずに，VaR を使用すべきではない．VaR に対する最も明白な批判は，この指標がポートフォリオの収益分布のある特定のパーセンタイル・ポイントだけにしか注目していない点である．つまり，低い確率で発生する事態が，実際どの程度害があるかについては注目していない．例えば，2 種類のポートフォリオ（P1 と P2）には，共に最大 5%の確率で 100 万ドルあるいはそれ以上の損失を生じる可能性があるとしよう．従って P1 と P2 の VaR は同じ値である．しかしながら，P1 は確率 5%できっかり 100 万ドルの損失が発生するのに対し，P2 は確率 1%で 100 万ドルまたは確率 4%で 1000 万ドルの損失が発生するものとしよう．ほとんどの人々は，P2 の方を危険なポートフォリオとみなすだろう．しかしながらこれらを同値とみなしてしまう VaR の欠点が，次の問題を引き起こしてしまうのである：

a）［良いニュースから生じる変則性］ポートフォリオを構成する証券の一つについてその予想利得を上方へ修正した場合，VaR を最小化するには直ちにその証券を売却するべきである．

b）［多様化が悪という変則性］ポートフォリオ X1 を有する銀行 1 がポートフォリオ X2 を有する銀行 2 を吸収合併した場合を考えよう．このとき，VaR(X1 + X2)

---

[6] 原著の 5 版以降は、13.5.3 と 13.5.4 は置き換えられている。

> VaR(X1) + VaR(X2)となるため，VaR によれば多様化によってリスクが上昇したことになるのである．

まず最初に，（a）の変則性について例示しよう．極めて保守的な投資家ならば，シナリオごとの最小収益率を最大化することでリスクに対応しようとするかもしれない．このケースは，VaR アプローチにおいて，任意なリスク許容度を 0 に近い値に設定した状態と同値である．この設定からいくつかの興味深い結果が得られる．ここでは A と C のみが投資可能であるとし，次の**表 13.5** に示した 2 種類のシナリオを考えてみよう：

**表 13.5　2 種類のシナリオ**

| シナリオ | 確率 | A からの利得 | C からの利得 |
|---|---|---|---|
| 1 | 0.8 | 1.0 | 1.2 |
| 2 | 0.2 | 1.5 | 0.7 |

最低可能な資産の最大化を望むのであれば，与えられた確率が正の値をとる限り，シナリオの確率は重要ではない．従って，以下の線形計画法は適切である：

```
MAX = WMIN;
!  当初の予算制約;
 A + C = 1;
!  シナリオ 1 のもとでの資産;
-WMIN+A+1.2*C>=0;
!  シナリオ 2 のもとでの資産;
-WMIN+1.5*A+0.7*C>=0;
```

この問題の解は，次のようになる：

```
Objective value:                    1.100000
Variable              Value        Reduced Cost
    WMIN           1.100000           0.0000000
       A          0.5000000           0.0000000
       C          0.5000000           0.0000000
```

2 つの投資対象はいずれも期待収益率が 10%であるから，予想される成長倍率が 1.10 となることは驚くべきことではない．すなわち，収益率は 10%となる．驚くべきは，リスクが全くない点である．従って初期投資額の 1 ドルを 50 セントずつ A と C に投資するならば，どちらのシナリオが現実に起きても初期投資額は 1.10 ドルまで増えるのである．

次に，「シナリオ 1 が起きたら，投資 C の利得は 1.2 ではなく 1.3 になる」という情報を非常に信頼できる友人から得たとしよう．これは確かに良いニュースである．何故ならば，投資 C の期待収益だけが上昇し，下方リスクについては何ら変化しないからである．では，この情報に対する反応はどうあるべきだろうか？　このモデルを以下のように修正してみよう：

```
MAX = WMIN;
```

```
! 当初の予算制約;
A + C = 1;
! シナリオ1のもとでの資産;
- WMIN + A + 1.3 * C > 0;
! シナリオ2のもとでの資産;
- WMIN + 1.5 * A + 0.7 * C > 0;
```

この問題を再度解くと，その解は次のようになる：

```
Objective value:                    1.136364
Variable            Value       Reduced Cost
    WMIN         1.136364          0.0000000
       A        0.5454545          0.0000000
       C        0.4545455          0.0000000
```

結局，C への投資を減らすことになり，少し違和感のある結果となっている．これではまるで友人が「株式 C に関するこの非常に好ましいニュースに市場が反応する前に，株式 C を売却してしまおう」と話を続けたように思えるだろう．何故，このような変則な事態になるのであろうか？ 問題は，可能な利得のうちのただ一つのポイントで，良し悪しを判断していることになる．この場合，それは考えうる最悪の結果である．この件に関しては，Clyman (1995)を参照されたい．

次に，(b)の変則性，すなわち「多様化が悪」という変則性について例示しよう．ポートフォリオ X1 と X2 はともに初期時点の資産が 100，期末時点における資産 w の分布は独立かつ均一で Prob{w = 80} = .04 および Prob{w = 110} = .96 とする．従ってリスク許容度を 5%とするならば，これら 2 つのポートフォリオの VaR はゼロである．何故ならば，損失が発生する確率は 5%以下である．ここでこれら 2 つのポートフォリオを合成すると，初期時点での資産価値は 200 であり，期末時点での資産価値の確率は Prob{w = 160} = .0016，Prob{w = 190} = .0768，および Prob{w = 220} = .9216 となるため，5%水準での VaR は 200 - 190 = 10 になる．つまり合併後の銀行の VaR は，合併前の個々の銀行の VaR の合計よりも大きくなる．従って VaR ルールによれば，合併後の銀行は以前よりも多くの安全資産を保有する必要がある．

### 13.5.4 条件付き VaR (Conditional Value at Risk, CVaR) について

低い確率で発生する事象がどの程度問題であるのかについて，まったく関心を払わない点が VaR の欠点であることは既に指摘した．条件付き VaR（あるいは CVaR）は，VaR のこの欠点を修正したものである．詳細は，Palmquist, Uryasev, and Krokhmal (2002)を参照されたい．ここで再度，ポートフォリオ P1 は確率 5%で 100 万ドルの損失，ポートフォリオ P2 は確率 1%で 100 万ドル，4%で 1000 万ドルの損失が予想されているとする．リスク許容度 5%のもとでの VaR は P1 と P2 ともに 100 万ドルとなるため，P1 と P2 は同等である．他方，条

件付き VaR（CVaR）は，VaR の閾値を越えた損失について，その損失額を明示的に考慮する方法である． 条件付き VaR と同様に，CVaR でもリスク許容度 ρ（例えば 5%）の設定が必要になる．さらにオプションとして，期待収益に関する選好パラメータ$\alpha \geq 0$ の設定も可能である．ポートフォリオの期末資産を確率変数 w で表わすとき，CVaR は，ポートフォリオの期末価値と VaR 閾値，t，および期末時点のポートフォリオの価値が VaR 目標値を下回る額とを加重したものになる．代数的には，CVaR の目的関数は次の式で表わされる：

$$\text{Max } \alpha E(w) + \rho t - E(\max[0, t - w]).$$

ところで，変数 t は他の制約には含まれない．最適状態において，変数 t が $\text{Prob}\{w \leq t\} \leq \rho$なる条件を満足することを示すのは容易である．何故ならば，最適ポートフォリオの VaR は（初期時点の資産－ t）となるからである．以下のモデルは，CVaR ポートフォリオの例である：

```
MODEL:
! シナリオ・ポートフォリオ・モデル;
! 条件付き VaR の最小化;
 SETS:
  SCENE: PRB, W, DVU, DVL;
  ASSET:  X;
  SXI(SCENE, ASSET): VE;
 ENDSETS
DATA:
 RHO = 0.1; ! リスク許容度;
 ALPHA = 0;
 TARGET = 1.15;
 SCENE = 1..12;
 ASSET =ATT GMC USX;
! オリジナルのマーコウィッツの例に基づくデータ;
VE =
   1.300    1.225    1.149
   1.103    1.290    1.260
   1.216    1.216    1.419
   0.954    0.728    0.922
   0.929    1.144    1.169
   1.056    1.107    0.965
   1.038    1.321    1.133
   1.089    1.305    1.732
   1.090    1.195    1.021
```

```
   1.083     1.390     1.131
   1.035     0.928     1.006
   1.176     1.715     1.908;
! すべてのシナリオが等確率で発生することを想定;
 PRB= .0833333 .0833333 .0833333 .0833333 .0833333
      .0833333 .0833333 .0833333 .0833333 .0833333
      .0833333 .0833333;
ENDDATA
! 各シナリオのもとでのポートフォリオの価値を計算する;
  @FOR(SCENE(S):W(S) = @SUM(ASSET(J):VE(S,J) * X(J));
! Measure deviations from CVaR target T;
DVL( S) - DVU( S) = T - W(S) ;);
! 予算制約;
 [BUD] @SUM( ASSET(i): X(i)) = 1;
! 期末時点でのポジションの期待値を計算する;
 [DEFAVG] AVG = @SUM(SCENE(s): PRB(s) * W(s));
! Ending value >= target;
 [RET] AVG >= TARGET;
! CVaR を最小化する;
[OBJ]MAX=ALPHA*AVG+RHO*T-@SUM(SCENE(s):PRB(s)*DVL(s));
END
```

解の一部は以下のようになる：

```
     Objective value:     0.09534855
                 Variable           Value
                      RHO       0.1000000
                    ALPHA        0.000000
                   TARGET        1.150000
                        T        1.017901
                      AVG        1.150000
                    W( 1)        1.236780
                    W( 2)        1.168732
                    W( 3)        1.300991
                    W( 4)        0.940602
                    W( 5)        1.029482
                    W( 6)        1.017901
                    W( 7)        1.077774
                    W( 8)        1.358208
```

| | |
|---|---|
| W( 9) | 1.061111 |
| W( 10) | 1.103096 |
| W( 11) | 1.022858 |
| W( 12) | 1.482470 |
| X( ATT) | 0.581326 |
| X( GMC) | 0.000000 |
| X( USX) | 0.418674 |

ポートフォリオの価値は初期時点で1だったので，このポートフォリオのVaRは1 – T = −0.017901である．シナリオは全部で12種類あるが，うちの4番目のシナリオだけでは最終資産価値がT = 1.017901以下になっている．従って，12分の1，または結果の10%未満において，ポートフォリオの最終価値が1.017901未満になる．

## 13.6 シナリオ・モデルと下方リスクの最小化について

収益率の分散の最小化は：

1) 現実の収益が正規分布に従う場合，あるいは

2) ポートフォリオの所有者の効用関数が2次関数である場合

のいずれかの場合で適切である．

実際には，どちらの条件でも，それらの成立を示すことは難しい．従って，より直感的なリスク測度を利用することを考えてみよう．そのような測度の一例が下方リスクであり，直感的に分かるように，収益率がある指定された目標水準よりも低くなる部分の量を測る指標となる．このアプローチを説明するために，以下の変数を定義しよう：

T = 使用者が指定する目標閾値．リスクを無視すれば，典型的にこの値は，期待収益率の最大値よりも小さく，最悪のシナリオのもとでの期待収益率よりも大きな値をとる．

Ys = シナリオ s のもとでの収益が目標を下回る量．

$=\max\{0, T-\sum X_i u_{is}\}$

モデルの代数的な表現は次のようになる：

最小化する目的関数：∑PsYs!予想される下方リスクを最小化する

制約条件：（それぞれのシナリオ s のもとでの目標を下回る乖離分を計算する）：

$Ys - T + \sum X_i u_{is} \geq 0$

$\sum X_i = 1$（予算制約）

$\sum X_i \sum P s u_{is} \geq r$（要求する収益率）

これは，線形計画問題である点に注意されたい．

### 13.6.1 半分散（semi-variance）と下方リスク

分散に代わってしばしば提案されるリスクの測度は，下方リスクの測度である．半分散（semi－variance）は，そのような測度の 1 つである．これは本質的には分散と同じだが，平均値よりも下方向の乖離だけをリスクとみなす測度である．従って，シナリオ・モデルはこのような測定に適している．上述のシナリオ・モデルは，わずかな修正で半分散モデルに変換できる．すなわち，変数 Y を平均値以下の乖離を表わす変数として再定義し，平均値以上の場合はゼロをとることにする．以下は，このモデルの結果である：

```
MODEL:
! シナリオ・ポートフォリオ・モデル;
! 半分散を最小化する;
 SETS:
  SCENE/1..12/: PRB, R, DVU, DVL;
  ASSET/ ATT,  GMT,  USX/:  X;
  SXI( SCENE, ASSET): VE;
 ENDSETS
DATA:
 TARGET = 1.15;
! オリジナルのマーコウィッツの例に基づくデータ;
VE =
   1.300    1.225    1.149
   1.103    1.290    1.260
   1.216    1.216    1.419
   0.954    0.728    0.922
   0.929    1.144    1.169
   1.056    1.107    0.965
   1.038    1.321    1.133
   1.089    1.305    1.732
   1.090    1.195    1.021
   1.083    1.390    1.131
   1.035    0.928    1.006
   1.176    1.715    1.908;
! すべてのシナリオが等確率で発生することを想定;
 PRB= .0833333 .0833333 .0833333 .0833333 .0833333
      .0833333 .0833333 .0833333 .0833333 .0833333
      .0833333 .0833333;
ENDDATA
! 異なるシナリオのもとでの値を計算する;
```

```
  @FOR(SCENE(S):R(S) = @SUM(ASSET(J):VE(S,J) * X(J));
! 平均値からの乖離を計算する;
DVU(S)-DVL(S)=R(S)-AVG;);
! 予算制約;
[BUD]@SUM(ASSET:X)=1;
! 期末時点のポジションの期待値を計算;
[DEFAVG]AVG=@SUM(SCENE:PRB*R);
! 期末時点の価値の目標;
[RET]AVG>TARGET;
! 半分散を最小化する;
[OBJ]MIN=@SUM(SCENE:PRB*DVL^2);
END
```

この問題の結果は，次の通り：

```
   Optimal solution found at step:          4
   Objective value:                     0.8917110E-02
   Variable              Value          Reduced Cost
      R( 1)           1.238875            0.0000000
      R( 2)           1.170760            0.0000000
      R( 3)           1.294285            0.0000000
      R( 4)          0.9329399            0.0000000
      R( 5)           1.029848            0.0000000
      R( 6)           1.022875            0.0000000
      R( 7)           1.085554            0.0000000
      R( 8)           1.345299            0.0000000
      R( 9)           1.067442            0.0000000
     R( 10)           1.113355            0.0000000
     R( 11)           1.019688            0.0000000
     R( 12)           1.479083            0.0000000
    DVU( 1)          0.8887491E-01        0.0000000
    DVU( 2)          0.2076016E-01        0.0000000
    DVU( 3)          0.1442846            0.0000000
    DVU( 4)          0.0000000            0.3617666E-01
    DVU( 5)          0.0000000            0.2002525E-01
    DVU( 6)          0.0000000            0.2118756E-01
    DVU( 7)          0.0000000            0.1074092E-01
    DVU( 8)          0.1952993            0.0000000
    DVU( 9)          0.0000000            0.1375965E-01
```

| | | |
|---|---|---|
| DVU( 10) | 0.0000000 | 0.6107114E-02 |
| DVU( 11) | 0.0000000 | 0.2171863E-01 |
| DVU( 12) | 0.3290833 | 0.0000000 |
| DVL( 1) | 0.0000000 | 0.8673617E-09 |
| DVL( 2) | 0.0000000 | 0.8673617E-09 |
| DVL( 3) | 0.0000000 | 0.8673617E-09 |
| DVL( 4) | 0.2170601 | 0.0000000 |
| DVL( 5) | 0.1201515 | 0.0000000 |
| X( ATT) | 0.5757791 | 0.0000000 |
| X( GMT) | 0.3858243E-01 | 0.0000000 |
| X( USX) | 0.3856385 | 0.0000000 |
| Row | Slack or Surplus | Dual Price |
| BUD | 0.0000000 | 0.1198420 |
| DEFAVG | 0.0000000 | -0.9997334E-02 |
| RET | 0.0000000 | -0.1197184 |
| OBJ | 0.8917110E-02 | 1.000000 |

ここで目的関数の値が，分散モデルの半分未満である点に注意されたい．ここでは（上方ではなく）下方への乖離のみを考慮しているので，目的関数の値は最大でも分散モデルの半分になる．分散モデルとの際立つ違いは，ポートフォリオの構成のうち，かなりの資金が GMC から USX にシフトしている点である．しかし，オリジナルのデータをみれば ATT 株が低収益であった期間の USX 株が GMC 株よりも高収益であるため，このシフトは驚きに値しないだろう．

### 13.6.2 下方リスクと平均絶対偏差（MAD）

下方リスクを決定するための閾値が平均収益ならば，下方リスクを最小にすることは，平均に関して平均絶対偏差(MAD)を最小にすることと同じである．何故ならば，平均に関する（絶対値をとる前の）乖離の合計はゼロでからである．従って，平均より上方の乖離の合計値は，平均より下方の乖離の合計値と等しくなる．このことは，絶対偏差の合計は，常に平均より下方の乖離の合計値の 2 倍になることを意味している．よって，平均より以下の下方リスクを最小にすることは，まさに平均値以下で絶対偏差の合計を最小にすることと同じである．Konno and Yamazaki (1991) は，MAD 測度を用いて，東証上場銘柄によるポートフォリオを構築している．

### 13.6.3 共分散行列に直接基づいたシナリオについて

オリジナルのデータではなく，共分散行列のみが利用可能な場合を考えよう．このような状況でも，共分散行列構造に合致するようなシナリオを構築することが可能である．以下は,4 種類のシナリオで達成可能な 3 銘柄（ATT, GMC, と USX）からなるポートフォリオの収益例である．これらのシナリオは，12.8.2 節の手法

を用いて，オリジナルデータから計算した共分散行列構造と合致するように構成されている：

```
MODEL:
SETS:
! 各証券は変動し，かつ平均収益率の情報は入手済み;
 ASSET/ATT, GMC, USX/:  X, RET;
! シナリオごとの収益率の分散とシナリオが発生する確率;
 SCEN/1..4/: Y, P;
! シナリオごとの各証券の収益率;
 COVMAT( SCEN, ASSET):ENTRY;
ENDSETS
DATA:
 ENTRY = .9851237 1.304437  1.097669
        1.193042  1.543131  1.756196
         .9851237  .8842088 1.119948
        1.193042  1.122902   .9645076;
  RET = 1.089083  1.213667  1.234583;
 P = .25 .25 .25 .25;
ENDDATA
! 分散を最小化する;
MIN=@SUM(SCEN:Y*Y*P);
! 4つのシナリオそれぞれのもとで収益率を計算する;
@FOR(SCEN(I):Y(I)-@SUM(ASSET(J):ENTRY(I,J)*X(J))+MEAN=0);
! 予算制約;
@SUM(ASSET:X)=1;
! 平均を定義あるいは計算する;
@SUM(ASSET:X*RET)=MEAN;
MEAN > 1.15; ! 目標収益率;
! それぞれの収益の分散は負の値をとる場合がある;
@FOR(SCEN: @FREE(Y));
END
```

この問題を解くと，次の見慣れた解を得る：

```
  Optimal solution found at step:          4
  Objective value:                 0.2241380E-01
         Variable           Value       Reduced Cost
             MEAN        1.150000          0.0000000
          X( ATT)       0.5300912          0.0000000
```

| | | |
|---|---|---|
| X( GMC) | 0.3564126 | 0.0000000 |
| X( USX) | 0.1134962 | 0.0000000 |
| RET( ATT) | 1.089083 | 0.0000000 |
| RET( GMC) | 1.213667 | 0.0000000 |
| RET( USX) | 1.234583 | 0.0000000 |
| Y( 1) | -0.3829557E-01 | 0.0000000 |
| Y( 2) | 0.2317340 | 0.0000000 |
| Y( 3) | -0.1855416 | 0.0000000 |
| Y( 4) | -0.7894565E-02 | 0.0000000 |
| P( 1) | 0.2500000 | 0.0000000 |
| P( 2) | 0.2500000 | 0.0000000 |
| P( 3) | 0.2500000 | 0.0000000 |
| P( 4) | 0.2500000 | 0.0000000 |
| ENTRY( 1, ATT) | 0.9851237 | 0.0000000 |
| ENTRY( 1, GMC) | 1.304437 | 0.0000000 |
| ENTRY( 1, USX) | 1.097669 | 0.0000000 |
| ENTRY( 2, ATT) | 1.193042 | 0.0000000 |
| ENTRY( 2, GMC) | 1.543131 | 0.0000000 |
| ENTRY( 2, USX) | 1.756196 | 0.0000000 |
| ENTRY( 3, ATT) | 0.9851237 | 0.0000000 |
| ENTRY( 3, GMC) | 0.8842088 | 0.0000000 |
| ENTRY( 3, USX) | 1.119948 | 0.0000000 |
| ENTRY( 4, ATT) | 1.193042 | 0.0000000 |
| ENTRY( 4, GMC) | 1.122902 | 0.0000000 |
| ENTRY( 4, USX) | 0.9645076 | 0.0000000 |
| Row | Slack or Surplus | Dual Price |
| 1 | 0.2241380E-01 | 1.000000 |
| 2 | 0.0000000 | 0.1914778E-01 |
| 3 | 0.0000000 | -0.1158670 |
| 4 | 0.0000000 | 0.9277079E-01 |
| 5 | 0.0000000 | 0.3947280E-02 |
| 6 | 0.0000000 | 0.3621391 |
| 7 | 0.0000000 | -0.3538852 |
| 8 | 0.0000000 | -0.3538841 |

ここで得た目的関数の値とATT，GMCおよびUSXへの資金配分は，本質的に最初に吟味したポートフォリオ問題と同値になっている点に注意されたい．

## 13.7　ヘッジング，マッチングおよびプログラム売買について

### 13.7.1　ポートフォリオ・ヘッジング

ポートフォリオ B をベンチマーク・ポートフォリオとする．ここで，収益率は B と同じだが，B よりもリスクが低いポートフォリオ C を構築することを「ポートフォリオ B をヘッジする」という．通常，ポートフォリオ B は除去不可能な要因を含んでいる．従って，既存の要因と負の相関をもつ証券を購入したいと思うだろう．以下はヘッジング・ポートフォリオの構築が有用となる例である：

a）今後 3 か月間で燃料を大量に購入する必要がある航空会社が，予期せぬ燃料価格の上昇という事態を回避したいと考えている場合．

b）今後 2 か月間で 20 万ドル相当のとうもろこしの生産を予定している農家が，（現時点でのとうもろこし価格に満足しているため）現在の時価に「ロック・イン」したいと思っている場合．

### 13.7.2　ポートフォリオ・マッチング，トラッキング，およびプログラム売買について

ベンチマーク・ポートフォリオ B に対して，B と同様の確率的な動きを示すが B に含まれている投資対象の一部が除外された新たなポートフォリオ C を構築したとしよう．このとき，この C をマッチング・ポートフォリオあるいはトラッキング・ポートフォリオと呼ぶ．次の例は，マッチング・ポートフォリオの構築が有用となる例である：

a）（例えば S&P500 指数のように）よく知られた指数と同レベルで運用したいポートフォリオ・マネージャーが，諸般の事情から指数に含まれる一部の株式を購入できないような場合．

b）迅速かつ低コストで取引可能な裁定取引者が，（市場における証券のミス・プライシングなどの）市場の非効率性から利益を得ようとする場合．例えば，ある明確に定義されたポートフォリオに対して，その将来の動きと完全に合致するポートフォリオを，今日，この裁定取引者が（ミス・プライシングが消滅する前に）低コストで構築できるならば，裁定利益の機会を得ることができるのである．

c）年金生活者の主たる関心事はインフレリスクである．このような場合，インフレ連動型ポートフォリオの構築が望まれるであろう．

(a)の例として，軍用兵器の売上が総収入の 2%超となる企業や，原子力発電所を直接所有あるいは操業する企業，もしくは核燃料サイクル関連のビジネスに関わる企業の株式をポートフォリオに含めないいわゆる「緑の」ミューチュアル・ファンドと呼ばれるポートフォリオがある．

例えば以下の**表 13.6** は，6 種類のバンガード・ポートフォリオの運用成績を，それらがベンチマークとするポートフォリオの運用成績と比較したものである（バンガード(1995)を参照）．

**表 13.6　6 種類のバンガード・ポートフォリオの運用成績(6 か月間の収益**

率，1995 年 6 月 30 日時点）

| バンガード社のポートフォリオ | ポートフォリオの成長率 | 比較対象指標の成長率 | インデックス名 |
|---|---|---|---|
| 500 Portfolio | +20.1% | +20.2% | S&P500 |
| Growth Portfolio | +21.1 | +21.2 | S&P500/BARRA Growth |
| Value Portfolio | +19.1 | +19.2 | S&P500/BARRA Value |
| Extended Market Portfolio | +17.1% | +16.8% | Wilshire 4500 Index |
| SmallCap Portfolio | +14.5 | +14.4 | Russell 2000 Index |
| Total Stock Market Portfolio | +19.2% | +19.2% | Wilshire 5000 Index |

運用成績に関してはポートフォリオごとにかなりの差異があるが，それぞれがベンチマークとする指標の運用成績とはかなりの程度合致している点に注意されたい．

## 13.8 ベンチマーク・ポートフォリオの構築方法について

ヘッジングあるいはマッチング・ポートフォリオの構築方法には，さまざまなアプローチが用いられている．マッチング・ポートフォリオについての直感的な方法は，マーコウィッツ・モデルの一般化である．つまり，ここでの目的は，ターゲット・ポートフォリオとトラッキング・ポートフォリオとの収益格差の分散を最小化することになる．

ベンチマーク・ポートフォリオのヘッジングあるいはマッチング・ポートフォリオについて考える際に有用な方法は，ベンチマークあるいはそれと負の相関を持つ要因を我々のポートフォリオに強制的に含めることである．仮に，ベンチマークがS&P500 などの単一指数であるとしよう．ここで分散をリスク測度とするならば，マッチングおよびヘッジング・ポートフォリオは以下の手順で構築できる：

①ベンチマークは，他の証券と同様に共分散行列の一部へ組み込むが，予算制約には含めない．なお，ここで投資予算 1 ドルを，制御可能なベンチマーク以外の証券へ投資することを仮定する．

②（例えば，S&P500 を複製する）マッチング・ポートフォリオを構築するためには，ベンチマークのファクターを-1 に設定すればよい．これで共分散行列の要素のなかでベンチマーク・ファクターに関連する非対角要素は行，列ともに無視される．事実上，ベンチマークをショートすることになる．もしもこのとき分散全体がゼロになるならば，ベンチマークの確率的変動に完璧にマッチさせたことになる．

③（例えば，S&P500 と負の相関をもつような）ヘッジング・ポートフォリオを構築するためには，ベンチマークのファクターを+1 に設定すればよい．これにより，

ポートフォリオの残余部分がベンチマーク・ファクターと逆行するように構成する．ここで予算制約を除外してみよう．この場合の解は，ベンチマーク 1 ドルについて最良なヘッジングあるいはマッチングを得るためにどれだけ制御可能なポートフォリオに投資すべきかを示す．

以下のモデルは，マーコウィッツ手法を，何らかのベンチマークを「相殺」するようなヘッジング・ポートフォリオの構築へ応用した例である．ここでは例えば，意思決定者が GMC に勤務しているため，彼の財産が GMC の資産に連動してしまうような状況を考えている．このような場合，彼は GMC を負の相関をもつようなポートフォリオを構築したいと考える可能性がある：

```
MODEL:
! 一般的なマーコウィッツ・ポートフォリオ・ヘッジング・モデル(PORTHEDG);
! ここでは 1 番目の証券あるいはベンチマークとなる証券を，その他の証券でヘッジすることを目的とする;
 SETS:
ASSET/GMCATTUSX/:RET,X;
TMAT(ASSET,ASSET)|&1#GE#&2:COV;
ENDSETS
 DATA:
! 期待収益率;
   RET = 1.21367, 1.089083, 1.23458;
! 共分散行列;
COV =
.05839170
.01240721  .01080754
.05542639  .01307513  .09422681;
! 要求する収益率;
   TARGET = 1.15;
 ENDDATA
!-----------------------------------------------;
! ポートフォリオ収益率の分散を最小化する;
[OBJ]MIN=(@SUM(ASSET(I):COV(I,I)*X(I)^2)+
2*@SUM(TMAT(I,J)|I#NE#J:COV(I,J)*X(I)*X(J)));
! ポートフォリオ中の 1 番目の証券;
X(1)=1;
! 予算制約 (1 番目の証券以外に適応される制約条件);
[BUDGET]@SUM(ASSET(I)|I#GT#1:X(I))=1;
! 収益率が満たすべき条件 (1 番目の証券以外に適応される制約条件);
```

```
[RETURN]@SUM(ASSET(I)|I#GT#1:RET(I)*X(I))>=TARGET;
END
```

解は以下の通り：

```
Optimal solution found at step:          4
Objective value:                 0.1457632
Variable            Value        Reduced Cost
 X( GMC)         1.000000          0.0000000
 X( ATT)        0.5813178          0.0000000
 X( USX)        0.4186822          0.0000000
```

従って，（GMC との関係が強い）USX よりも ATT への比重を高めることになる．

次のモデルは，ベンチマークを複製（あるいはベンチマークのマッチング・ポートフォリオ）を構築するためにマーコウィッツ手法を拡張した例である．この例では，マッチングの対象が S&P500 であるのに対し，投資対象は ATT，GMC，および USX だけに制限している：

```
MODEL:
! 一般化したマーコウィッツ・ポートフォリオ・マッチング・モデル(PORTMTCH);
! 1 番目の証券あるいはベンチマークとなる証券を，その他の証券にマッチさせることを目的とする;
 SETS:
  ASSET/ SP500 ATT GMC USX/: RET, X;
  TMAT(ASSET, ASSET) | &1 #GE# &2: COV;
 ENDSETS
 DATA:
! 期待収益率;
   RET =1.191458  1.089083, 1.21367, 1.23458;
! 共分散行列;
COV =
.02873661
.01266498   .01080754
.03562763   .01240721   .05839170
.04378880   .01307513   .05542639   .09422681;
! 要求する収益率;
   TARGET = 1.191458;
 ENDDATA
!-------------------------------------------------;
! ポートフォリオ収益率の分散を最小化する;
```

```
[OBJ]MIN=(@SUM(ASSET(I):COV(I,I)*X(I)^2)
+2*@SUM(TMAT(I,J)|I#NE#J:COV(I,J)*X(I)*X(J)));
! マッチングは，ベンチマークをカラ売りすることと同じである；
   X( 1) = -1;
   @FREE( X( 1));
! 予算制約 (1番目の証券以外に適応される制約条件);
 [BUDGET] @SUM( ASSET( I)| I #GT# 1: X( I)) = 1;
! 収益率が満たすべき条件 (1番目の証券以外に適応される制約条件);
 [RETURN] @SUM( ASSET( I)| I #GT# 1:
 RET( I) * X( I)) >= TARGET;
END
```

この問題の解は次の通り：

```
   Optimal solution found at step:          4
   Objective value:                 0.5245968E-02
    Variable           Value        Reduced Cost
   X( SP500)        -1.000000         0.0000000
     X( ATT)        0.2276635         0.0000000
     X( GMC)        0.4781277         0.0000000
     X( USX)        0.2942089       -0.1266506E-07
```

### 13.8.1 ベンチマーク・ポートフォリオへのシナリオ・アプローチの適用

シナリオ・アプローチを使うのであれば，ヘッジング・モデルは以下のようになる：

```
MODEL: ! (PRTSHDGE);
! シナリオ・ポートフォリオ・モデル (ただし，1番目の資産をヘッジする)；
! 分散を最小化する；
SETS:
  SCENE/1..12/: PRB, R, DVU, DVL;
  ASSET/  GMT, ATT,  USX/:  X;
  SXA( SCENE, ASSET): VE;
ENDSETS
DATA:
! オリジナルのマーコウィッツの例に基づくデータ；
 VE =
   1.225   1.300    1.149
   1.290   1.103    1.260
   1.216   1.216    1.419
   0.728   0.954    0.922
```

```
   1.144   0.929    1.169
   1.107   1.056    0.965
   1.321   1.038    1.133
   1.305   1.089    1.732
   1.195   1.090    1.021
   1.390   1.083    1.131
   0.928   1.035    1.006
   1.715   1.176    1.908;
! すべてのシナリオが等確率で発生することを想定;
 PRB= .0833333 .0833333 .0833333 .0833333 .0833333
      .0833333 .0833333 .0833333 .0833333 .0833333
      .0833333 .0833333;
! 要求する収益率;
   TARGET = 1.15;
ENDDATA
! リスクの最小化;
 [OBJ] MIN = @SUM( SCENE: PRB * ( DVL + DVU) ^ 2);
! ポートフォリオでは資産 1 を必ず保有しなくてはならない;
  X( 1) = 1;
! 各シナリオのもとで，ヘッジング・ポートフォリオの値を計算する;
  @FOR( SCENE( S):
    R( S)=@SUM( ASSET( J)| J #GT# 1: VE( S, J) * X( J));
! ヘッジ + ベンチマークからの乖離を計算;
DVU( S) - DVL( S) =  ( R(S) + VE( S, 1))/ 2 - TARGET;);
! 予算制約 (1 番目の証券以外に適用される制約条件);
 [BUDGET] @SUM( ASSET( J)| J #GT# 1: X( J)) = 1;
! 期末時点におけるポジションの期待値を計算する;
 [DEFAVG] AVG = @SUM( SCENE: PRB * R);
! 目標とする期末時点の値;
 [RET] AVG > TARGET;
END
```

解は以下のようになる:

```
   Optimal solution found at step:        4
   Objective value:                0.3441714E-01
   Variable          Value        Reduced Cost
    X( GMT)        1.000000          0.0000000
    X( ATT)       0.5813256          0.0000000
```

```
   X( USX)          0.4186744                  0.0000000
```

このポートフォリオが，マーコウィッツ・モデルと同じ結果になっている点に注意しよう．

シナリオ・モデルによる S&P500 のマッチング・ポートフォリオの構築は以下の通りである：

```
 MODEL:
  ! シナリオ・モデル（1 番目の資産にマッチさせる）（PRTSMTCH）;
  ! 分散を最小化する;
   SETS:
    SCENE/1..12/: PRB, R, DVU, DVL;
    ASSET/ SP500  ATT  GMT  USX/:  X;
    SXA( SCENE, ASSET): VE;
   ENDSETS
  DATA:
  ! オリジナルのマーコウィッツの例に基づくデータ;
   VE =
  !  S&P500   ATT    GMC    USX;
   1.258997    1.3  1.225  1.149
   1.197526  1.103  1.29   1.26
   1.364361  1.216  1.216  1.419
   0.919287  0.954  0.728  0.922
    1.05708  0.929  1.144  1.169
   1.055012  1.056  1.107  0.965
   1.187925  1.038  1.321  1.133
   1.31713   1.089  1.305  1.732
   1.240164  1.09   1.195  1.021
   1.183675  1.083  1.39   1.131
   0.990108  1.035  0.928  1.006
   1.526236  1.176  1.715  1.908;
  ! すべてのシナリオが等確率で発生することを想定;
  PRB= .0833333 .0833333 .0833333 .0833333 .0833333
       .0833333 .0833333 .0833333 .0833333 .0833333
       .0833333 .0833333;
  ! 要求する収益率;
    TARGET = 1.191458;
  ENDDATA
  ! リスクを最小化する;
```

```
[OBJ] MIN = @SUM(SCENE: PRB * ( DVL + DVU) ^ 2);
! シナリオごとにポートフォリオの値を計算;
 @FOR(SCENE( S):R( S) = @SUM( ASSET( J)| J #GT# 1: VE( S,
J) * X( J));
! ベンチマークからの乖離を計算;
 DVU( S) - DVL( S) = ( R(S) - VE( S, 1)); );
! (残りの資産に適用される)予算制約;
 [BUDGET] @SUM( ASSET( J)| J #GT# 1: X( J)) = 1;
! 期末s時点のポジションの期待値を計算;
 [DEFAVG] AVG = @SUM( SCENE: PRB * R);
! 期末時点で目標とする値;
 [RET] AVG > TARGET;
END
```

この問題の解は，以下の通り：

```
 Optimal solution found at step:          7
 Objective value:                 0.4808974E-02
  Variable            Value        Reduced Cost
 X( SP500)        0.0000000           0.0000000
   X( ATT)        0.2276583           0.0000000
   X( GMT)        0.4781151           0.0000000
   X( USX)        0.2942266           0.0000000
```

ここでも，マーコウィッツ・モデルと同じポートフォリオを得た点に注意されたい．

2種類のシナリオ・モデルでは，ベンチマークに対するリスクの測度としてともに分散を使用している．しかし，下方リスクなどのように，非対称のリスク測度を用いるように変更することも簡単である．

### 13.8.2 効率的なベンチマーク・ポートフォリオ

あるポートフォリオのリターンとリスクが，他のいかなるポートフォリオよりも高いリターンと低いリスクを与えるとき，このポートフォリオは効率的フロンティア上にあるという．

ここで，次の記号を定義しよう：

$r_i$ = 証券 i の期待収益率，

t = ポートフォリオ収益率についての任意な目標水準

資産 i に加重 $m_i$ を置くあるポートフォリオが効率的であるならば，このポートフォリオはある目標値 t について以下の問題の解になっている：

**最小化の目的関数： リスク**

**制約条件：**

**$\Sigma_{i=0}^{n}\ m_i = 1$ （予算制約条件）**

$\Sigma_{i=0}^{m} r_i m_i \geq t$　（**収益目標の制約条件**）.

ポートフォリオ・マネージャーの運用成績は，頻繁に何らかのベンチマーク・ポートフォリオと比較・評価している．ここでベンチマーク・ポートフォリオにおける証券 i の加重を記号 $b_i$ で表わすことにしよう．ベンチマーク・ポートフォリオが効率的フロンティア上に無いとき，興味ある疑問は「ベンチマークに対する効率的ポートフォリオの相対リスクを最小化するという基準のもとで，ベンチマーク・ポートフォリオにもっとも近い効率的フロンティア上のポートフォリオのウェートは何か？」であろう．

リスク測度がポートフォリオの分散で，安全資産が存在し，無リスク金利で無制限の借入が可能で，かつカラ売りが許される状況であれば，答えは極めて単純である．安全資産の加重を記号 $m_0$ で表わそう．この場合，目標収益率に応じて $m_0$ のみが変動するような，資産 i の加重が $m_i$ になるいわゆる「市場」ポートフォリオが存在する．具体的には，$i = 1, 2, \ldots, n$ で $m_i$ なる定数が存在し，資産 i の加重は単に $(1 - m_0) \times m_i$ となる．ここで以下の記号を定義しよう：

$q = 1 - m_0$ = 市場ポートフォリオの加重，

$R_i$ = 証券 i のランダムな収益率

すると，ベンチマーク・ポートフォリオに比べた効率的ポートフォリオの分散は次のようになる：

$$\mathrm{var}\left(\sum_{i=1}^{n} R_i [q m_i - b_i]\right)$$

$$= \sum_{i=1}^{n} (q m_i - b_i)^2 \,\mathrm{var}(R_i) + 2 \sum_{j>i} (q m_i - b_i)(q m_j - b_j)\,\mathrm{Cov}(R_i, R_j).$$

この式を q で微分し，ゼロに等しいとすることで，次の結果を得る：

$$q = \frac{\displaystyle\sum_{i=1}^{n} m_i b_i \,\mathrm{var}(R_i) + \sum_{j>i} (m_i b_j m_j b_i)\,\mathrm{Cov}(R_i, R_j)}{\displaystyle\sum_{i=1}^{n} m_i^2 \,\mathrm{var}(R_i) + 2\sum_{j>i} m_i m_j \,\mathrm{Cov}(R_i, R_j)}$$

例えば，安全資産の加重が $b_0$ のときベンチマーク・ポートフォリオが効率的フロンティア上に乗るのであれば，$b_i = (1 - b_0) m_i$ となり従って $q = 1 - b_0$ である．従って恐らく，$\{b_0, b_1, \ldots, b_n\}$ で定義されるベンチマーク・ポートフォリオよりも高い運用成績を要求されるポートフォリオ・マネージャーは，事実上，上述の q で与えられる効率的ポートフォリオのパフォーマンスと比較して評価されるべきであろう．

### 13.8.3 ポートフォリオ問題の効率的な設定方法

数理モデルの解析に要する時間は，モデルの設定方法で劇的に異なる場合がある．この現象は，整数計画法ではよく知られた問題であり，以下では非線形計画で同様の現象を例示したい．またポートフォリオ最適化問題で，数学的には同値だが，異なる3種類のモデルも提示する．

モデル1

$$\text{Minimize} \ \sum_{i=1}^{n}\sum_{j=1}^{n} q_{ij}\ x_i\ x_j$$

subject to

$$\sum_{j=1}^{n} x_j = 1$$

$$\sum_{j=1}^{n} r_j\ x_j = r_0$$

**モデル2**

共分散行列の対称性という性質を活用すれば，目的関数は次のように変更できる：

$$\text{Min} \ \sum_{i=1}^{n} x_i \ (q_{ii}\ x_i + 2\sum_{j=i+1}^{n} q_{ij}\ x_j)$$

subject to

$$\sum_{j=1}^{n} x_j = 1$$

$$\sum_{j=1}^{n} r_j\ x_j = r_0$$

**モデル3**

目的関数の中で変数 $x_i$ と掛ける項は分離して計算できるため，次のモデルへの変更が可能である：

$$\text{Minimize} \ \sum_{i=1}^{n} x_i\ w_i$$

subject to

For each $i$;

$$w_i = q_{ii}\ x_i + 2\sum_{j=i+1}^{n} q_{ij}\ x_j, \quad w_i \text{ a free variable}$$

$$\sum_{j=1}^{n} x_j = 1$$

$$\sum_{j=1}^{n} r_j \ x_j = r_0$$

ここでは，ニューヨーク証券取引所に上場する（IBM，ゼロックス，AT&T など）19 銘柄の運用成績データを用いてこれら 3 種類のモデルで数値解を計算する．なお，ここではモデルを通常の非線形計画問題とみなして解いている．従って，ここではこれらのモデルが 2 次計画問題である点は活用していない．

次の**表 13.7** は，モデルごとに解を得るまでの所要時間（秒）をまとめたものである：

**表 13.7　NLP の 3 モデルの計算時間の比較**

| モデル | 所要時間（秒） |
|---|---|
| 1 | 2.16 |
| 2 | 1.5 |
| 3 | 0.82 |

何故，所要時間が劇的に異なるのであろうか？

モデル(1)に対するモデル(2)の優位性は明白であろう．モデル(1)の目的関数の評価には約 2×n×n 個の積が必要であるが，モデル(2)にはおよそ n+n×n/2 個だけである．

モデル(3)の積は，実質的にモデル(2)と同数である．ところがそのうちの約 n×n/2 個は線形制約である．従って制約条件の数は劇的に増加しているが，これらは線形制約であり，線形制約を効率的に処理する技術は既に開発されているからである．

### 13.9 2 次計画法のためのコレスキー分解

モデル(3)と類似するが，より簡潔な別のモデルが存在する．共分散行列 $\{q_{ij}\}$ をコレスキー分解することで，共分散行列の「平方根」に相当する下三角行列 $\{L_{ij}\}$ を得ることができる．これを用いて新たなモデルを次のように表記できる：

**Minimize** $\sum_{j=1}^{n} w_j^{\,2}$

**subject to**

**For each** $j$:

$w_j = \sum_{i=j+1}^{n} L_{ij} \ x_j$, $w_j$ **a free variable**

$$\sum_{j=1}^{n} x_j = 1$$

$$\sum_{j=1}^{n} r_j \ x_j = r_0$$

このモデルが，目的関数に含まれる変数が（2n 個ではなく）n 個だけである点を

除けば，モデル(3)と概ね同じ構造である点に注意されたい．
　行列表記を用いれば，この式の変形を説明することは容易である．共分散行列 Q と下三角行列 L とが次の条件を満たすとしよう．
　L L’ = Q，ただし L’ は転置行列を示している
我々の目的は，以下の目的関数の最小化である
　x Q x’ = x L L’ x’
ここで，w = x L とすれば，我々の目的を単純に次のように表わすことができるだろう．
目的関数の最小化： w w’
制約条件：w = x L.
　以下は，コレスキー分解を用いた LINGO モデルを我々の 3 資産問題に応用した例である：

```
MODEL: ! コレスキー分解を用いたポートフォリオ・モデル;
SETS:
 ASSET: AMT, RET, CW;
 COVMAT( ASSET, ASSET): VARIANCE;
 MAT(ASSET,ASSET)| &1 #GE# &2: L; ! コレスキー分解;
ENDSETS
DATA:
   ASSET =     ATT       GMC         USX;
! 共分散行列と期待収益率;
VARIANCE = .01080754 .01240721 .01307513
           .01240721 .05839170 .05542639
           .01307513 .05542639 .09422681;
     RET = .0890833  .213667   .234583;
ENDDATA
! 分散を最小化する;
[VAR] MIN = @SUM( ASSET( I): CW( I) * CW( I));
! 期首における予算を 100%利用する;
[BUDGET] @SUM( ASSET: AMT) = 1;
! 期末時点の資産について要求する条件;
[RETURN] @SUM( ASSET: AMT * RET) > .15;
! 分散への寄与を計算する, CW();
@FOR(ASSET(J):@FREE(CW(J));
CW(J)=@SUM(MAT(I,J):L(I,J)*AMT(I)););
! LL'= VARIANCE を満たすコレスキー要因 (L) を計算する;
@FOR(ASSET(I):
```

```
@FOR(MAT(I,J)|J#LT#I:
L(I,J)=(VARIANCE(I,J)-@SUM(MAT(I,K)|
K#LT#J:L(I,K)*L(J,K)))/L(J,J););
L(I,I)=(VARIANCE(I,I)-@SUM(MAT(I,K)|
K#LT#I:L(I,K)*L(I,K)))^.5;);
END
```

以下は，解の一部である：

```
Optimal solution found at step:        4
Objective value:                  0.2241375E-01
         Variable           Value        Reduced Cost
         AMT( ATT)       0.5300926         0.0000000
         AMT( GMC)       0.3564106         0.0000000
         AMT( USX)       0.1134968         0.4492217E-08
         RET( ATT)       0.8908330E-01     0.0000000
         RET( GMC)       0.2136670         0.0000000
         RET( USX)       0.2345830         0.0000000
          CW( ATT)       0.1119192         0.0000000
          CW( GMC)       0.9671834E-01     0.0000000
          CW( USX)       0.2309568E-01     0.0000000
VARIANCE( ATT, ATT)      0.1080754E-01     0.0000000
VARIANCE( ATT, GMC)      0.1240721E-01     0.0000000
VARIANCE( ATT, USX)      0.1307513E-01     0.0000000
VARIANCE( GMC, ATT)      0.1240721E-01     0.0000000
VARIANCE( GMC, GMC)      0.5839170E-01     0.0000000
VARIANCE( GMC, USX)      0.5542639E-01     0.0000000
VARIANCE( USX, ATT)      0.1307513E-01     0.0000000
VARIANCE( USX, GMC)      0.5542639E-01     0.0000000
VARIANCE( USX, USX)      0.9422681E-01     0.0000000
       L( ATT, ATT)      0.1039593         0.0000000
       L( GMC, ATT)      0.1193468         0.0000000
       L( GMC, GMC)      0.2101144         0.0000000
       L( USX, ATT)      0.1257716         0.0000000
       L( USX, GMC)      0.1923522         0.0000000
       L( USX, USX)      0.2034919         0.0000000
              Row    Slack or Surplus    Dual Price
              VAR      0.2241375E-01    -1.000000
           BUDGET      0.0000000          0.8255034E-02
```

RETURN　　0.0000000　　-0.3538836

## 13.10 問題

1.　3つの株式，IBM，GM，およびGeorgiaPacific(GP)からなる株式ポートフォリオを考えてみよう．以下の表は，これらの株式の収益率（年率）の共分散行列である．

| | IBM | GM | GP |
|---|---|---|---|
| IBM | 10 | 2.5 | 1 |
| GM | 2.5 | 4 | 1.5 |
| GP | 1 | 1.5 | 9 |

従って，これらの株式に等金額を投資する場合，その分散の値は10+4+9+2(2.5+1+1.5)に比例することになる．IBM,GM,およびGPの予想収益率は，年率でそれぞれ9%と,6%と5%である．これら3つの株式を使って，ポートフォリオ収益率が年率で最低7%となるような最小分散ポートフォリオを見つけよ．ただし，IBMへの投資比率は最大でも80%，GPへの投資比率は最低でも10%とする．

2.あなたの現在のポートフォリオの50%がIBMで残りの50%がGPであるとしよう．さらに,売買の取引コストは,取引量の1%とする．これらを考慮するように，問題1の設定を変更せよ．

3.投資ファンドの管理者は,今後1年間の経済状態について，3つの異なったシナリオを仮定している．これらのシナリオを青，黄，赤と呼び，それらの主観確率を0.7,0.1,および0.2とする．管理者は，これら3つのシナリオのもとで，株式,債券,不動産,および金の配分を定めたポートフォリオを決定したいと考えている．資産とシナリオごとの年収益率の推定値は以下の通りである：

| | 株式 | 債券 | 不動産 | 金 |
|---|---|---|---|---|
| 青 | 9 | 7 | 8 | -2 |
| 黄 | −1 | 5 | 10 | 12 |
| 赤 | 10 | 4 | -1 | 15 |

期待収益率を最低でも6.5%とする最小分散ポートフォリオの問題を設定し，資産配分の問題を解きなさい．

4.本章で議論したATT/GMC/USXのポートフォリオ問題を考えてみよう．実際に数値解を求めた例で要求あるいは目標としていた収益率は15%であった．

a)ここで要求収益率を16%としよう．本文中の数値解の出力結果から，新しいポートフォリオの収益率の標準偏差について何が予測できるだろうか?

b)無リスクで年率5%の投資機会が利用可能となった場合についても本文中で例示した．なおこの4番目の投資機会は，分散も，他の証券との共分散もゼロであった．ではここで，無リスク利子率を年率5%ではなく4%であると仮定しよう．以前と同様，4%での投資額には制限が無いものとする．（要求収益率を年率で15%とした）オリジナルの問題への本文中の出力結果だけを使い，要求収益率15%のもとで，この新たな選択肢の意味について述べよ．

c）手元に 10 万ドルの投資資金があるとしよう．（出力結果の解を 10 万倍するのではなく）適切な数値が出力されるためには，もともとの ATT/GMC/USX モデルにどのような修正を加えるべきか．

d）（c）の結果に基づいて投資した場合，ポートフォリオの期末時点における推定標準偏差の値はいくつになるか．