7 被覆・人員配置・分割問題

7.1 はじめに

被覆問題は，サービス産業でよく利用される．要求される集合がカバー（満たされる）されることに特徴がある．種々の活動があるが，そのうちの幾つかに対応することで要求がみたされる．この問題を言葉で表すと次のようになる．

ここでの目的関数は，最小の費用で要求を満たす活動水準の組合せを選択することである．種々の問題の活動と要求の例を以下に示す．

問題	要求	活動
要員計画	日あるいは週単位で要求される人数	労働あるいは交替のパターン．これらのパターンは，幾つかの期間の要求を満たす．
ルーティング	訪問すべき顧客	幾つかの顧客を訪問する，さまざまな道順．
紙，木材，鋼板，服地などの原材料から製品の切断	最終製品のサイズ要求	種々の最終製品のサイズを原材料から切り出す切断のパターン．各パターンは，最終製品のすべてではない，幾つかの要求を満たす．

次の節で，これらの問題の幾つかをより詳細に紹介する．

7.1.1 人員配置問題

サービス施設における経営の主要部分は，スケジューリングと人員配置である．すなわち，何人の人間を使うかを決定することである．この問題は，電話会社のオペレータ部門，有料道路，大きな病院などのサービスを供給しなければならない施設において存在する．

この解のプロセスは3つの部分からなる．

①１日の時間あたり，または１週間の各曜日に必要とされる人数の予測．
②個人の働ける日と労働協約に基ずき働くことが可能なシフトパターンの識別．
③各シフトパターンで働く人数の決定，すなわち費用が最小かつ各期間で働く人の総人数が，①で決定された要求を満たすこと．

これら3ステップはどれも難しいが,LPは③を解決することを助けてくれる．

人員配置問題の最初の論文の一つがEdie(1954) である．彼はニューヨークの港湾当局が管理する料金所の職員の配置方法を開発した．Edie の論文は古いけれども，彼の議論は非常に適切で完全である．概要(p.138)を紹介する．「試みはリンカーン・トンネルで行なわれた．各料金徴集員のブースの割り当て，交替時間およびスケジュールに厳しく従うように指示されたメモ用紙が渡された．・・・集金の動作やブースの開閉が，通行料の集金が自然に行えるよう指示された．ブース数はわずかに余分で，過度に多すぎないように．・・・言うまでもなく，十分な満足を与えるものにする・・・」

7.1.2 北東有料道路の人員配置問題

シカゴ郊外の北東有料道路の料金所は，24時間に以下の人員を必要とする．

時間	必要とする人員
0時から午前6時	2
午前6時から10時	8
午前10時から正午	4
正午から午後4時	3
午後4時から6時	6
午後6時から10時	5
午後10時から24時	3

料金所の人間は4時間働き，1時間休み，又4時間働く．何時から働いてもよい．目的関数は雇う人数の最小化としてLPを定式化する．

(1)北東有料道路問題の定式化とその解

決定変数を定義する．

x_1 = 真夜中から働く人の数

x_2 = 午前1時から働く人の数

・

・

x_{24} = 午後11時から働く人の数

各時間帯に対し1つの制約が存在し，その時間における人員数は要求される数より多い．目的関数は雇われる人数の最小化である．

MIN=x_1+x_2+x_3+..+x_{24};

ST

x_1 +x_{24}+x_{23}+x_{22}+x_{20}+x_{19}+x_{18}+x_{17}>= 2;（夜中から午前1時）

x_2 +x_1 +x_{24}+x_{23}+x_{21}+x_{20}+x_{19}+x_{18}>=2;（午前1時から2時）

.

x_7 +x_6 +x_5 +x_4 +x_2 +x_1 +x_{24}+x_{23}>=8 ;（午前6時から7時）

・

x_{24}+x_{23}+x_{22}+x_{21}+x_{19}+x_{18}+x_{17}+x_{16}≧3 ;（午後11時から夜中）

PICTUREから，シフトの間の1時間の休みの影響がわかる．

制約行	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	..	x_{17}	x_{18}	x_{19}	x_{20}	x_{21}	x_{22}	x_{23}	x_{24}	RHS
0時から1時.	1										1	1	1	1		1	1	1	≥2
1時から2時.	1	1										1	1	1	1		1	1	≥2
2時から3時.	1	1	1										1	1	1	1		1	≥2
3時から4時.	1	1	1	1										1	1	1	1		≥2
4時から5時.		1	1	1	1										1	1	1	1	≥2
5時から6時.	1		1	1	1	1										1	1	1	≥2
6時から7時	1	1		1	1	1	1										1	1	≥8
7時から8時	1	1	1		1	1	1	1										1	≥8
8時から9時	1	1	1	1		1	1	1	1										≥8
9時から10時		1	1	1	1		1	1	1										≥8
10時から11時			1	1	1	1		1	1										≥4
11時から12時				1	1	1	1		1										≥4
12時から13時					1	1	1	1											≥3
13時から14時						1	1	1	1										≥3
14時から15時							1	1	1										≥3
							etc											etc	

(2)　集合による定式化

LINGO でこの問題を集合で定式化すると，コンパクトになる．2 つの集合がある．最初は 1 日 24 時間であり，もう一つは 9 時間交替である．変数 X の値の剰余を求める@WRAP 関数を使用している．

```
MODEL:! 24 hour shift scheduling;
 SETS:!Each shift is 4 hours on, 1 hour off,4 hours on;
  HOUR/1..24/: X, NEED;
 ENDSETS
 DATA:
  NEED=2 2 2 2 2 2 8 8 8 8 4 4 3 3 3 3 6 6 5 5 5 5 3 3;
 ENDDATA
 MIN = @SUM( HOUR(I): X(I));
 @FOR( HOUR( I): ! People on duty in hour I are those who started 9 or
 less hours earlier, but not 5;
 @SUM(HOUR(J)|(J#LE#9)#AND#(J#NE#5):
X(@WRAP((I-J+1),24)))>= NEED(I));
END
```

非零の解は以下の通りで，目的関数は 15.75 である．

x_2 = 5	x_5 = 0.75	x_{11} = 1	x_{16} = 1
x_3 = 0.75	x_6 = 0.75	x_{14} = 1	x_{17} = 1
x_4 = 0.75	x_7 = 0.75	x_{15} = 2	x_{18} = 1

この答えは小数なので役にたたない．15.75 人は，整数解を必要とするから少なくとも 16 人を雇わなければならない．これを解決するため，次の@GIN 関数を END の前に挿入すればよい：@FOR(HOURS: @GIN(X));

これを解くと,次の非零の整数値をもち,目的関数が 16 の解を得る.

$x_2 = 4$　　$x_5 = 1$　　$x_{14} = 1$　　$x_{17} = 2$

$x_3 = 1$　　$x_6 = 1$　　$x_{15} = 1$　　$x_{18} = 1$

$x_4 = 1$　　$x_7 = 1$　　$x_{16} = 2$

この種の人員配置問題の最も大きい例の 1 つは，コールセンターである．例えば,クレジットカードのサービスセンター,カタログ通販のための電話受けつけ,などである．オマハやネブラスカの人口のかなりの人が，コールセンターで働いている.コールセンターの典型的な交替パターンは，8 時間労働において，15 分の休憩，30 分の昼休み，およびもう 15 分の休憩から構成されている.

7.1.3　人員配置の付加的機能

人員配置の実現には，先の 3 ステップに加えて，名簿作成，作業パターンの識別，作業パターンの選択が必要である．名簿作成では，特定の個人が特定の作業パターンの仕事をする．例えば航空会社では，個人（例えば，パイロット）は選ばれた仕事パターンを選択する.

人材派遣では，例えば最初の 1 時間に，少なくとも 3 人のスペイン語と 4 人の英語を話すスピーカーが必要かもしれない．派遣従業員のスキルが異なる場合があり，ある人は英語だけ，他の人は英語とスペイン語の両方に通じている.

例えばメールの処理では,繁忙期に優先度の低いものの処理を遅らせることで,人員不足に対処することもある.

ほとんど全ての状況下で，需要はややランダムなので，スタッフ要件は柔軟に対応すべきである．期間中に少なくとも 10 人が必要なのに，11 人雇うこともある．この場合，予想に反して緊急の仕事が入った場合，余剰人員が対応できる．18 章の待ち行列理論は，余剰人員の限界利益の見積りに頻繁に使用される.

7.2　分割とパターン選択問題

製紙産業では，製品は最初に経済的な生産サイズで作られる．これらのサイズは，最終製品として小さなサイズに分割される．最小費用でより小さなサイズにどのように分割するかの決定が,「分割問題」である.「一次元分割問題」の例として，機械の性能で材料が 72 インチの幅を要求すると仮定する．これをより小さな幅に分割する方法はいろいろあるが，そのうちの二つを図 7.1 に示す.

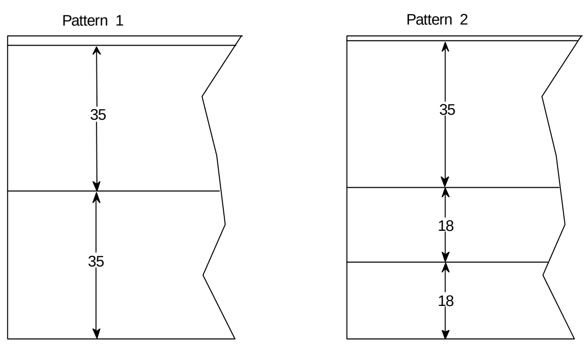

図 7.1　分割パターンの例

パターン 1 の端の無駄は 2 インチ (72 − 2 × 35 = 2) であり，パターン 2 の無駄は 1 インチ (72 − 2 × 18 − 35 = 1) である．しかしパターン 2 は，18 インチ

の長さは 35 インチの材料の長さに比べて 2 倍でないと有効ではない．よって，端の無駄と結果としての無駄の間の妥協点を見つけなければならない．

分割問題の解は，3 ステップに分けることができる．

①最終的に必要とされる幅の推測

②より小さな幅に分割する際の可能なパターンの決定

③②の各パターンのどれだけが必要かの決定．すなわち①における要求が最小費用で満たされるかどうか．LP はステップ③の実行で使用できる．

多くの巨大製紙会社は，分割問題の解法をベースにした LP 解を持っている．実際の分割問題は，端の無駄または結果の無駄の妥協に付け加えて様々な費用因子を含んでいる．LP の有効性は，これらの因子の重要性に依存している．次の例は，複雑な費用因子を伴なわない分割問題の基本的な特徴を示している．

7.2.1　クルドット社の分割問題

クルドット社は非常に幅広い家庭用器具，すなわち冷蔵庫やストーブなどを生産している．原材料費の重要な部分は，鉄板の購入である．現在，鉄板はコイルの形で購入し，72，48，36 インチの 3 種類の幅がある．製造工程では 8 種類の幅（60，56，42，38，34，24，15，10 インチ）が要求される．鉄板は同じ品質と厚さが要求される．

続いて起こる問題は，無駄を取り除くことである．例えば 72 インチ幅のコイルを 38 インチ幅のコイルと 15 インチ幅のコイル 2 つに切り離すやり方がある．ここで，無駄な 4 インチ幅のコイルが出てしまうことになる．

3 つの異なった幅の原材料の 1 フィート当りの価格は，36 インチ幅が 15 セント，48 インチ幅が 19 セント，72 インチ幅が 28 セントである．単純計算で，1 インチ×1 フィートあたりの価格は，36 インチ幅が 15/36＝0.41667 セント／(インチ×フィート)，48 インチ幅が 0.395833，72 インチ幅が 0.3888889 である．

コイルは適切な方法で切り離される．3 つの幅の原材料を切り離す効率的なやり方を以下に表にしてある．

例えばパターン C4 は 72 インチ幅のコイルを，24 インチ幅 1 つと 10 インチ幅 4 つ，無駄な 8 インチが残るように切り離すことである．

この計画期間で，要求される様々な幅の長さは，次の通りである．

幅	60「	56「	42「	38「	34「	24「	15「	10「
要求フィート数	500	400	300	450	350	100	800	1000

この期間に使用できる原材料は，72 インチのコイルで 1600 フィート，48 インチと 36 インチで 10,000 フィートである．様々な幅や要求を満たし，最小費用になるように分割されるパターンのフィート数を決定するモデルを定式化せよ．あなたは 36 インチ幅の材料の使用量を，前もって推測できるだろうか？

7.2.2　クルドット分割問題の定式化とその解

分割パターン表に現われる記号 A1，A2，...，E4 を，それに対応するパターンのフィート数を表わすものとする．

原材料の切断パターン

パターン名称	60「	56「	42「	38「	34「	24「	15「	10「	無駄
	要求される幅の切断数 72-インチ原材料								
A_1	1	0	0	0	0	0	0	1	2
A_2	0	1	0	0	0	0	1	0	1
A_3	0	1	0	0	0	0	0	1	6
A_4	0	0	1	0	0	1	0	0	6
A_5	0	0	1	0	0	0	2	0	0
A_6	0	0	1	0	0	0	1	1	5
A_7	0	0	1	0	0	0	0	3	0
A_8	0	0	0	1	1	0	0	0	0
A_9	0	0	0	1	0	1	0	1	0
B_0	0	0	0	1	0	0	2	0	4
B_1	0	0	0	1	0	0	1	1	9
B_2	0	0	0	1	0	0	0	3	4
B_3	0	0	0	0	2	0	0	0	4
B_4	0	0	0	0	1	1	0	1	4
B_5	0	0	0	0	1	0	2	0	8
B_6	0	0	0	0	1	0	1	2	3
B_7	0	0	0	0	1	0	0	3	8
B_8	0	0	0	0	0	3	0	0	0
B_9	0	0	0	0	0	2	1	0	9
C_0	0	0	0	0	0	2	0	2	4
C_1	0	0	0	0	0	1	3	0	3
C_2	0	0	0	0	0	1	2	1	8
C_3	0	0	0	0	0	1	1	3	3
C_4	0	0	0	0	0	1	0	4	8
C_5	0	0	0	0	0	0	4	1	2
C_6	0	0	0	0	0	0	3	2	7
C_7	0	0	0	0	0	0	2	4	2
C_8	0	0	0	0	0	0	1	5	7
C_9	0	0	0	0	0	0	0	7	2

48-インチ原材料									
D_0	0	0	1	0	0	0	0	0	6
D_1	0	0	0	1	0	0	0	1	0
D_2	0	0	0	0	1	0	0	1	4
D_3	0	0	0	0	0	2	0	0	0
D_4	0	0	0	0	0	1	1	0	9
D_5	0	0	0	0	0	1	0	2	4
D_6	0	0	0	0	0	0	3	0	3
D_7	0	0	0	0	0	0	2	1	8
D_8	0	0	0	0	0	0	1	3	3
D_9	0	0	0	0	0	0	0	4	8
36-インチ原材料									
E_0	0	0	0	0	1	0	0	0	2
E_1	0	0	0	0	0	1	0	1	2
E_2	0	0	0	0	0	0	2	0	6
E_3	0	0	0	0	0	0	1	2	1
E_4	0	0	0	0	0	0	0	3	6

計算のために，以下のものをつけ加えて定義することは有効である．

T1＝72 インチパターンのカットされたフィート数

T2＝48 インチパターンのカットされたフィート数

T3＝36 インチパターンのカットされたフィート数

W1＝72 インチパターンからの無駄のインチ×フィート

W2＝48 インチパターンからの無駄のインチ×フィート

W3＝36 インチパターンからの無駄のインチ×フィート

X1＝60 インチ幅の過剰カットフィート数

X2＝56 インチ幅の過剰カットフィート数

・

・

・

X8＝10 インチ幅の過剰カットフィート数

目的関数を何にすべきか直接には分からないかも知れない．たとえば，各パターンに対するフィートあたりの無駄の費用を計算することを試み，全ての無駄の費用を最小化したいと思いがちである．すなわち，

Min=0.3888891*W1 + 0.395833*W2 + 0.416667*W3;

しかし，このような目的関数では，無駄は非常に少ないが費用がかかる解になる．これは特に平方インチあたりの費用が，全ての幅に対して同じでないときに起きる．より適当な目的関数は，次のように全費用を最小にすることである．

MIN = 28 * T1 + 19 * T2 + 15 * T3;

これをモデルに組み込むと次のモデルになる．

```
MODEL:
 SETS:
! Each raw material has a Raw material width, Total used,
   Waste total, Cost per unit, Waste cost, and Supply available;
 RM: RWDTH,T, W, C, WCOST, S;
! Each Finished good has a Width, units Required. eXtra produced;
 FG: FWDTH, REQ, X;
 PATTERN: USERM, WASTE, AMT;
 PXF( PATTERN, FG): NUM;
 ENDSETS
 DATA:
! The raw material widths;
   RM =    R72       R48       R36;
 RWDTH=     72        48        36;
    C =    .28       .19       .15;
 WCOST= .00388889 .00395833 .00416667;
    S =    1600     10000     10000;
! The finished good widths;
   FG = F60 F56 F42 F38 F34 F24 F15 F10;
 FWDTH=  60  56  42  38  34  24  15  10;
   REQ= 500 400 300 450 350 100 800 1000;
! Index of R.M. that each pattern uses;
 USERM = 1 1 1 1 1 1 1 1 1 1
         1 1 1 1 1 1 1 1 1 1
         1 1 1 1 1 1 1 1 1
         2 2 2 2 2 2 2 2 2 2
         3 3 3 3 3;
! How many of each F.G. are in each R.M. pattern;
 NUM=   1 0 0 0 0 0 0 1
        0 1 0 0 0 0 1 0
        0 1 0 0 0 0 0 1
        0 0 1 0 0 1 0 0
        0 0 1 0 0 0 2 0
        0 0 1 0 0 0 1 1
        0 0 1 0 0 0 0 3
        0 0 0 1 1 0 0 0
        0 0 0 1 0 1 0 1
        0 0 0 1 0 0 2 0
        0 0 0 1 0 0 1 1
        0 0 0 1 0 0 0 3
```

```
          0 0 0 0 2 0 0 0
          0 0 0 0 1 1 0 1
          0 0 0 0 1 0 2 0
          0 0 0 0 1 0 1 2
          0 0 0 0 1 0 0 3
          0 0 0 0 0 3 0 0
          0 0 0 0 0 2 1 0
          0 0 0 0 0 2 0 2
          0 0 0 0 0 1 3 0
          0 0 0 0 0 1 2 1
          0 0 0 0 0 1 1 3
          0 0 0 0 0 1 0 4
          0 0 0 0 0 0 4 1
          0 0 0 0 0 0 3 2
          0 0 0 0 0 0 2 4
          0 0 0 0 0 0 1 5
          0 0 0 0 0 0 0 7
          0 0 1 0 0 0 0 0
          0 0 0 1 0 0 0 1
          0 0 0 0 1 0 0 1
          0 0 0 0 0 2 0 0
          0 0 0 0 0 1 1 0
          0 0 0 0 0 1 0 2
          0 0 0 0 0 0 3 0
          0 0 0 0 0 0 2 1
          0 0 0 0 0 0 1 3
          0 0 0 0 0 0 0 4
          0 0 0 0 1 0 0 0
          0 0 0 0 0 1 0 1
          0 0 0 0 0 0 2 0
          0 0 0 0 0 0 1 2
          0 0 0 0 0 0 0 3;
ENDDATA
! Minimize cost of raw material used;
MIN = TCOST;
 TCOST = @SUM(RM(I): C(I)*T(I) );

@FOR( RM( I):
  T( I) = @SUM( PATTERN( K)| USERM(K) #EQ# I: AMT( K));
! Raw material supply constraints;
     T(I) <= S(I);
```

```
      );

 ! Must produce at least amount required of each F.G.;
 @FOR( FG(J):
    @SUM(PATTERN(K): NUM(K,J)*AMT(K)) = REQ(J) + X(J);
     );
  ! Turn this on to get integer solutions;
  !@FOR( PATTERN(K): @GIN(AMT(K)));
 ! Waste related computations;
 !  Compute waste associated with each pattern;
  @FOR( PATTERN(K):
    WASTE(K) = RWDTH(USERM(K)) - @SUM(FG(J): FWDTH(J)*NUM(K,J));););
 ! Waste for each R.M. in this solution;
  @FOR( RM( I):
     W(I) = @SUM( PATTERN( K)| USERM(K) #EQ# I: WASTE(K)*AMT( K)););
 ! Compute total cost of waste;
  TOTWASTE = @SUM( RM(I): WCOST(I)*W(I) );
 END
```

２つの異なった目的関数の異なった解は，以下の表で比較できる．

分割問題の解

ゼロでないパターン	無駄を最小にした解（フィート）	総費用を最小にした解（フィート）
A_1	500	500
A_2	400	400
A_5	200	171.4286
A_7	100	128.5714
A_8	350	350
A_9	50	3.571429
B_8	0	32.14286
C_9	0	14.28571
D_1	150	96.42857
D_3	25	0
無駄費用	$5.44	$5.55
総費用	$2348.00	$466.32
X_4	100.000	0
X_6	19650	0
T_1	1600	1600
T_2	10000	96.429
T_3	0	0

この解の鍵となる相違は，「無駄を最小にする解」は 48 インチの材料（T2)をより多く使用し，端の無駄が最小になる方法で分割している．さらに 38 インチ幅と 24 インチ幅の材料を必要以上に作っている．しかしながら，目的関数はそれを無駄としては数えていない．

どちらの解も，その数値が整数でない．実際問題で多くの人は最も近い整数に丸める．ここで＠GIN で整数指定すると，費用最小化の解は$466.34 に増える．

7.2.3　分割問題の一般化

巨大な分割問題で，全ての可能性のあるパターンを生成することは非現実的である．最適解で，非常に高い確率で現れるパターンのみを生成する効率的な方法が存在する．この手続きを説明することは，この節の範囲を越える．しかし巨大な問題で，それが重要になる．Dyckhoff(1981)による，この問題の他の定式化を 18 章で紹介している．しかし，制約式は非常に巨大になる．

複雑な分割問題で，さらに以下の費用の考察が重要になる．

① **特定のパターンを生成するための固定費用**：これは，機械使用時間，労働力，その他からなる．これは，より少ないパターンの解を魅力あるものとする．

② **余剰または最終的な無駄が有する価値**：例として，当該期間における無駄は，

次の期になんらかの需要があるかもしれない．

③**機械使用費用**：機械を操作する費用は，普通は材料の状態と独立である．これは，幅広の材料を切断する解に有利である．

④**規定の量に達しない費用**：ある産業で，例えば±5%の供給が許されるかもしれない．最少限の許される量で生産する費用は，既定利益になる．

⑤**特別の生産をする材料**：もし異なった材料（例えば異なった厚さ，品質，表面の仕上げ，型など）を要求しても，2つの異なった製品を同じパターンで作ることは不可能であるかもしれない．

⑥**グレードアップの費用**：特定需要幅に対し，要求されるものよりグレードの高い材料に変えることで，セットアップ，縁または先端の無駄を減少させることが可能なこともある．

⑦**注文による分割費用**：もし要求される幅がいくつかのパターンから生成されるならば，出荷に対し，同時に出荷する異なったロットを運ぶための統一費用が存在するかもしれない．

⑧**材料幅変更費用**：パターンの変更のみを含むセットアップは，普通パターンの変更と材料幅の変更の両方を含むセットアップに比べて少ない時間で済む．これは，少ない材料幅を使用する解を魅力あるものにする．

⑨**縁の無駄についての許容値**：ある材料は，幅の非常に狭い縁の無駄の扱いが非常にむずかしい．従って，縁の無駄のない，または最小値（例えば2センチメートル）を越えるパターンに集中することが好まれる．

⑩***納期と順序***：ある種の需要は，他が緊急性の少ない場合，すぐに対応が必要である．緊急性が高いか優先度の高いパターンは，すばやく対応する必要がある．緊急性の高い需要が優先度の低いパターンと同じなら，これにすぐに対応するのは難しくなる．

⑪在庫制約： 普通顧客の発注は，顧客の全ての需要が出荷可能になるまで出荷されない．このように，特定顧客の需要が，できるだけ少ないパターンであることを希望する．顧客が全てのパターンの製品を必要なら，顧客の注文はあらゆるパターンが動くまで出荷されない．このように，全てのパターンが動くまで，相当な仕掛品がでてくる．

⑫1セット・パターンに対する制限： 製紙のような産業では，パターンを準備することと関連する明示的な費用はないが，パターン変化が起きると対応できる率に制限がある．パターンの変更におよそ 15 分かかるかもしれないが，この仕事の多くが主要機械を止めてされる．1本のロールの生産に，10 分かかるかもしれない．あまりに多くの 1 セットのパターンが稼動するなら，主要機械はパターンの変更が完了されるのを待たなければならない．

⑬パターン規制： ある応用では，最終製品の幅の合計数やパターンの小さな幅の数に制限があるかもしれない．限られた数の受け取りリールがスリット状の商品を巻き上げるために用いられる場合，最初の規制にあてはまる．第二の規制は，ぎりぎりの製品幅のロールが倒れる傾向を持つ紙業界で起こるかもしれない，一つのパターンで，あまりに多くのものを扱いたくないである．一部の顧客は，材料の品質がある場所でより高いと感じるので，製品をその特定の場所（例えばセ

ンター）から切りとることを依頼するかもしれない．

⑭対になっているパターン：プラスチックのラップ製造で，生産プロセスの特性から，同時に上と下の出力から2つの幅をもつ原料を生じる．例えば，下の出力と同じだけ，上で出力しなければならないことは，本質的に避けられない．類似した状況は，時々紙製造において偶然に起こる．もし欠陥が生産ラインのベルト上で発生すると，その幅の内部にある小幅な紙は使えない．こうして機械は，欠陥の右と左に2つの出力幅のものを生じる．

最も手に負えない複雑さは，高く固定された段取り費用，注文分割費用，材料幅変更費用である．もしこれらが無視できないほど重要であるなら，人は手動による解を強いられる．LP 解は良い解の洞察を与えるが，他の方法で最終的に実行し得る解を決定しなければならない．

7.2.4　2次元分割問題

1次元の分割問題は，コイル状の原料を分割することに関係している．基本的な考えは，原料がシート状のものに応用でき，問題はこれをより小さなシートに分割することになる．例えば，合板が 48×96 インチの矩形で供給され，最終的に要求される製品は，36×50，24×36，20×60，18×30 インチのシートとする．一度あなたが 48×96 のシートを分割する4つのより小さなシートの組合せの全ての可能なパターンを数えたなら，問題は前とまったく同じになる．

すべての可能な二次元パターンを列挙することは，非常に困難な課題である．実用的な二次元の切断問題の2つの特徴は，この仕事のサイズを次のように減らすことである：(a)方向性の要求と，(b)「ギロチン切断」要求である．(a)が重要な応用分野は，木や生地の切断である．強度や見た目の理由から，要求の単位は原料で限られた位置にあるかもしれない．例えば，服の製造業者が方向のある格子縞のスーツを作ることを想像しなさい．よい野球選手は，ヒットするときバットの木目が球に合致していることを知っている．木製品が構造や審美的な目的で使用される場合，木の木目に注意を払われなければならない．ガラスは，方向が重要でない均質な原料の例である．ギロチンカットは，シート材の端から端まで切り取る方法である．

7.3　乗員のスケジューリング問題

航空会社の運営経費の主なものは，乗員の人件費である．大手航空会社の航空機と乗員の管理は複雑な問題である．これらに注目することは価値がある．乗員の年間費用は，典型的なコンピュータの1次的費用を上回るので，より効果的に乗員と飛行機を使用するためにコンピュータ資源をつぎこむことは魅力的である．次の小さな問題を以下で議論する．

大手航空会社は，乗員スケジュール問題と呼ばれる要因計画問題に直面する．カバーされる必要条件は，航空会社がある期間（例えば1ヵ月）予定しているフライトの乗員の要求を満たすことである．出勤日，特定の乗員は概して同じ航空機に勤務する．問題は，どの飛行機を乗員に割り振るか決定することである．

多くの航空会社でとられる方法は，一般的な要因計画である：①需要要求の確認（すなわちカバーされる飛行）．②1人の乗員が仕事期間にカバーできる

飛行の多数の可能な順序の生成．③②で生成された中から，飛行機と乗員も組み合わせを正確に 1 つ選んで，費用を最小にする．

ステップ③に IP が使われる．1985 まで大きな IP を解くのが難しいかったので，大部分の大手航空会社は③を解くため発見的な手続きを使用していた．しかし Marsten, Muller & Killion (1979)は，タイガー航空で非常にうまい IP モデルを記述した．タイガー航空は，乗客が少ないので，結果として IP は経済的に解くことができて，発見的な方法より著しく低い費用の解をえた．この最適化法は，現在大手の航空会社で利用されている．

次は，非常に単純化した例である．この例は 10 機の便だけを持つが，主要な航空会社は日に 2000 機以上の便を持つ．

7.3.1 例題

乗員スケジューリング問題の簡単な例として，以下の例を考える．Sayre-priors 航空会社は，以下のフライトを運行している．

フライト

フライト番号	出発地	目的地	時間
101	Chicago	Los Angeles	午後
410	New York	Chicago	午後
220	New York	Miami	夜
17	Miami	Chicago	朝
7	Los Angeles	Chicago	午後
13	Chicago	New York	夜
11	Miami	New York	朝
19	Chicago	Miami	夜
23	Los Angeles	Miami	夜
3	Miami	Los Angeles	午後

フライトのスケジュールは，次の図 7.2 に示す．

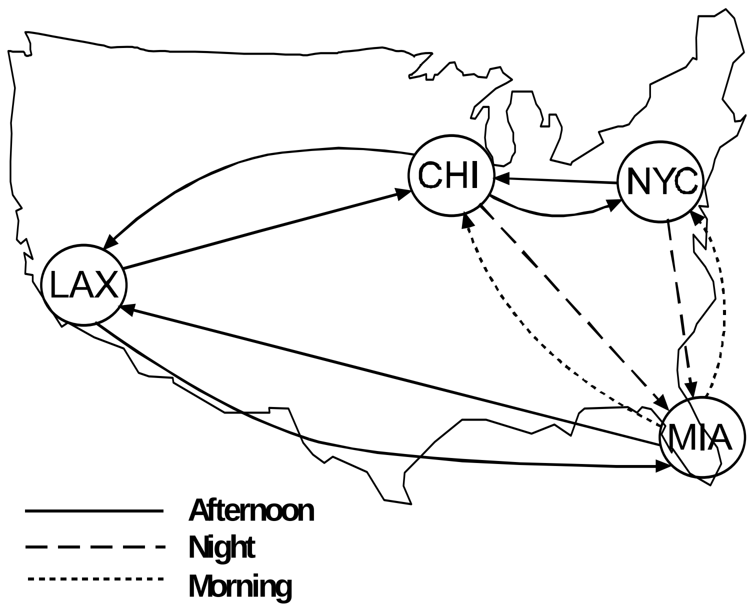

図 7.2 フライト・スケジュール

フライトを運用するスタッフは，低費用で乗員を割り当てたいと思っている．基本的な問題は，一人の乗員が一つのフライトを終えた後に彼が乗り込む次のフライトを決定することである．この問題を理解するのに必要な基本概念は，ツアーの概念である．ツアーの特徴は，次の通りである．

①ツアーは 1 つから 3 つの接続されたフライトからなる．
②ツアーは出発地点で終わると，2000 ドルの費用がかかる．
③出発地点以外の場所で終わる行止りツアーは，3000 ドルの費用がかかる．

航空会社の用語では，ツアーはしばしば「ペアリング「または「ローテーション「と呼ばれる．以下は，受け入れられるツアーの例である．

ツアー	費用
17，101，23	$2,000
220，17，101	$3,000
410，13	$2,000

実際のツアー費用の計算は，上より複雑である．例えば，パイロットの人件費は，飛行中と，フライト間の待機期間と，自宅から離れていることに対する日当などさまざまである．

7.3.2 乗員スケジューリング問題に対する解

この小さな問題で最初に行うことは，全ての可能なツアーを数えあげることである．10 個の 1 回フライト，14 個の 2 回フライトのツアーがある．さらに，出発地点で終わるツアーは，出発地点を区別するかしないかに依存して 37 または 41 個の 3 回フライトのツアーがある．これらのツアーは以下に示される．

ツアーのリスト

1 回フライト	費用	2 回フライト	費用	3 回フライト	費用
1.　101	$3,000	11.　101，23	$3,000	25.　101，23，17	$2,000
2.　410	$3,000	12.　410，13	$2,000	26.　101，23，11	$3,000
3.　220	$3,000	13.　410，19	$3,000	27.　410，19，17	$3,000
4.　17	$3,000	14.　220，17	$3,000	28.　410，19，11	$2,000
5.　7	$3,000	15.　220，11	$2,000	29.220，17，101	$3,000
6.　13	$3,000	16.　17，101	$3,000	30.220，11，410	$3,000
7.　11	$3,000	17.　7，13	$3,000	25.　17，101，23	$2,000
8.　19	$3,000	18.　7，19	$3,000	31.　7，19，17	$3,000
9.　23	$3,000	19.　11，410	$3,000	32.　7，19，11	$3,000
10.　3	$3,000	20.　19，17	$2,000	33.　11，410，13	$3,000
		21.　19，11	$3,000	28.　11，410，19	$2,000
		22.　23，17	$3,000	34.　19，17，101	$3,000
		23.　23，11	$3,000	28.　19，11，410	$2,000
		24.　3，23	$2,000	25.　23，17，101	$2,000
				35.　23，11，410	$3,000

36.　3, 23, 17　$3,000
37.　3, 23, 11　$3,000

次の決定変数を定義する．

T_i ＝１（ツアー i が採用されたとき），

　　０（ツアー i が使用されないとき，　i ＝1,2,…,37）

出発地点で終わる３回フライトは，出発地を区別しないことにする．定式化は以下のようになる（費用は 1000 ドル単位）

選択したツアーの費用を最小化する；

条件：選んだツアーは正確に１つの飛行をカバーしている；

```
MODEL:
  [_1] MIN=  3 * T_1 + 3 * T_2 + 3 * T_3 + 3 * T_4 + 3 * T_5
     + 3 * T_6 + 3 * T_7 + 3 * T_8 + 3 * T_9 + 3 * T_10
     + 3 * T_11 + 2 * T_12 + 3 * T_13 + 3 * T_14 + 2 * T_15
     + 3 * T_16 + 3 * T_17 + 3 * T_18 + 3 * T_19 + 2 * T_20
     + 3 * T_21 + 3 * T_22 + 3 * T_23 + 2 * T_24 + 2 * T_25
     + 3 * T_26 + 3 * T_27 + 2 * T_28 + 3 * T_29 + 3 * T_30
     + 3 * T_31 + 3 * T_32 + 3 * T_33 + 3 * T_34 + 3 * T_35
     + 3 * T_36 + 3 * T_37 ;
  [COV_F101]  T_1 + T_11 + T_16 + T_25 + T_26 + T_29 + T_34 = 1 ;
  [COV_F410] T_2 + T_12 + T_13 + T_19 + T_27 + T_28 +  T_30
         + T_33 + T_35 = 1 ;
  [COV_F220]T_3 + T_14 + T_15 + T_29 + T_30 = 1 ;
  [COV_F17] T_4 + T_14 + T_16 + T_20 + T_22 + T_25 + T_27
         + T_29 + T_31 + T_34 + T_36 = 1 ;
  [COV_F7] T_5 + T_17 + T_18 + T_31 + T_32 = 1 ;
  [COV_F13] T_6 + T_12 + T_17 + T_26 = 1 ;
  [COV_F11] T_7 + T_15 + T_19 + T_21 + T_23 + T_26 + T_28
         + T_30 + T_32 + T_33 + T_35 + T_37 = 1 ;
  [COV_F19] T_8 + T_13 + T_18 + T_20 + T_21 + T_27 + T_28
         + T_31 + T_32 + T_34 = 1 ;
  [COV_F23] T_9 + T_11 + T_22 + T_23 + T_24 + T_25 + T_35
         + T_36 + T_37 = 1 ;
  [COV_F3] T_10 + T_24 + T_36 + T_37 = 1 ;
   @BIN( T_1);  @BIN( T_2);  @BIN( T_3);  @BIN( T_4);  @BIN( T_5); @BIN( T_6);
@BIN( T_7);  @BIN( T_8);  @BIN( T_9);  @BIN( T_10); @BIN( T_11); @BIN( T_12);
@BIN( T_13); @BIN( T_14); @BIN( T_15)  @BIN( T_16); @BIN( T_17); @BIN( T_18);
@BIN( T_19); @BIN( T_20)  @BIN( T_21); @BIN( T_22); @BIN( T_23); @BIN( T_24);
@BIN( T_25); @BIN( T_26); @BIN( T_27); @BIN( T_28); @BIN( T_29); @BIN( T_30);
@BIN( T_31); @BIN( T_32); @BIN( T_33); @BIN( T_34); @BIN( T_35);  @BIN( T_36);
@BIN( T_37);
 END
```

PICTURE は，問題の構造を明らかにする．例えば，最初の制約（２行目）は，フライト 101 を含むツアーを選ばなければならないことを示している．

```
                        T T T T T T T T T T T T T T T T T T T T T T T T T T T T
      T T T T T T T T T 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3
      1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7

  1:  3 3 3 3 3 3 3 3 3 3 3 2 3 3 2 3 3 3 3 2 3 3 3 2 2 3 3 2 3 3 3 3 3 3 3 3 3 MIN
  2:  1                   1         1                 1 1     1         1       = 1
  3:    1                   1 1           1               1 1   1     1   1     = 1
  4:      1                     1 1                           1 1               = 1
  5:        1                       1       1   1     1   1   1   1     1   1   = 1
  6:          1                       1 1                         1 1           = 1
  7:            1           1         1                               1         = 1
  8:              1               1       1   1   1     1   1   1   1 1   1   1 = 1
  9:                1         1         1   1 1           1 1     1 1   1       = 1
 10:                  1   1                     1 1 1 1 1                 1 1 1 = 1
 11:                    1                           1                       1 1 = 1
```

LP として解いたとき，解は都合の良いことに整数となり，次のものが選ばれた．

ツアー	フライト
T17	7， 13
T24	3， 23
T28	410， 19， 11
T29	220， 17， 101

この解の費用は 10,000 ドルである．T_17 + T_24 + T_28 + T_29 <= 3 という制約を加えると，同じ費用で，他の最適解がある．

ツアー	フライト
T12	410， 13
T24	3， 23
T29	220， 17， 101
T32	7， 11， 19

500 以上の制約を伴う乗員スケジューリング問題の IP は，解くのが驚くほど難しい．この問題の一般的な定式化は，7.4 で述べる．

7.3.3　付加的な事項

実用上，上の定式化に加える点は，乗員の生活場所である．ホームベースは，各旅行と関係している．各ホームベースで生活するパイロット数が分かれば，そのホームベースからの旅行数に上限を付けた制約を加えたいかもしれない．

また実際問題として，パイロットは概して 1 種類の航空機だけに適任である．そこで，各機種毎（例えばボーイング 747，エアバス 320，その他）に，乗員配置問題を解くことである．同じように，客室乗務員はパイロットと独立してスケジュールされる．

客室乗務員が選ばれたあと，客室乗務員をパイロットに割り当てる面倒な問題がある．組合との合意によりアメリカでは，主要な航空会社の多くは，パイロットの希望で予定を選ぶのを許す．より小さい航空会社や米国以外の会社では，全

社的な立場でパイロットの割り当てを決めているかもしれない．このような中央管理システムは能率の悪さを除去するが，少なくとも高い年功をもつパイロットの希望で決めることは，彼らにより幸せ感を与えるかもしれない．

悪天候と器材故障による不確実性は，定期航空路線にとって残念な事実である．そこで，混乱の影響を受けない乗員予定計画ができればありがたい．予定計画がしばしば飛行機を乗り換えることを乗員に要求し，これらの接続時間が短いならば，この計画は敏感で混乱の影響を受ける傾向がある．飛行機が半時間ほど若干の器材の修繕で遅延し，この飛行機の乗組員が次の空港で飛行機を乗り変える予定であるならば，この空港で次の最高 3 台の便の遅れが発生する：①遅れた飛行機の次の予定，② 遅れた飛行機の乗員を使う予定の次の飛行，そして③この飛行機の乗客が乗り換える飛行機の次の予定．

この小さい例で，代替案があることを知っている．例えば，総乗員費用を最小にすることに加えて，飛行機を乗り換える乗員を含むツアーを避けることで 2 次的な目的関数として遅れの期待値を最小化できるかもしれない．このために．次の方策がある：①短い乗換えを避ける②多くの乗客が乗り換える便を避ける③遅れた便の代わりをする乗員を空港に待機させる．

7.4 一般的なカバーリング/分割/梱包モデル

この乗員計画モデルは，特殊な問題である．以下は，一般的なカバーリング/分割/梱包モデルである．このモデルは，施設と顧客の視点で考える．各施設（またはパターン）を開くと，特定の顧客または需要に貢献する．この要因計画を特殊化すると，旅行は施設であり，飛行機は顧客である．入力する主なデータは，どの施設がどの顧客をサービスするかを表すことである．このモデルは，顧客が十分にサービスを受けれるか受けられないかにかなり柔軟である．パラメータ BUDU が 0 に設定されるならば，これはあらゆる顧客（または飛行）が少なくとも 1 つの開いた施設（または旅行）でサービスを受けることを意味する．あるいは，BUDV が 0 に設定されるならば，各飛行は少なくとも 1 つの選ばれた旅行に現れる．これは，時々パッキング問題と呼ばれている．BUDV と BUDU がゼロに設定されるならば，各飛行は正確に 1 つの選ばれた旅行に現れなければならない．これは，分割問題と呼ばれている．どんな可能な解でも顧客の集合を部分集合（各々の選ばれた開いた施設（または旅行））に分割するので，それは「集合分割」と呼ばれている．

```
     0 makes it a packing or partitioning problem;
  BNDV = 0;
! Both = 0 makes it a partitioning problem;
  PXD =
  1,F101 2,F410  3,F220  4,F17  5,F7 6,F13  7,F11  8,F19 9,F23 10,F3
 11,F101 11,F23 12,F410 12,F13 13,F410 13,F19 14,F220 14,F17
 15,F220 15,F11  16,F17 16,F101 17,F7 17,F13 18,F7 18,F19
 19,F11  19,F410 20,F19 20,F17 21,F19  21,F11 22,F23 22,F17
 23,F23  23,F11  24,F3  24,F23 25,F101 25,F23 25,F17
```

```
 26,F101 26,F13  26,F11 27,F410 27,F19 27,F17
 28,F410 28,F19 28,F11 29,F220 29,F17 29,F101 30,F220 30,F11 30,F410
 31,F7   31,F19 31,F17 32,F7 32,F19 32,F11 33,F11 33,F410
 34,F19  34,F17 34,F101 35,F23 35,F11 35,F410 36,F3 36,F23 36,F17
 37,F3   37,F23 37,F11;
ENDDATA
!-------------------------------------------------------;
! Minimize cost of facilities opened, demands poorly served;
   MIN = @SUM( PATTERN( I): COST( I) * T(I))
       + @SUM( DMND( J): CU( J) * U( J) + CV( J) * V( J));

 ! For each demand,
    sum of patterns serving it + under variable - over variable= 1;
   @FOR( DMND( J):
     @SUM( PXD( I, J): T( I)) + U( J) - V( J) = 1;
       );

! Stay within budget on facilities cost;
   @SUM( PATTERN: COST * T) <= BUDGET;

! and demand service costs;
   @SUM( DMND(J): CU(J) * U(J)) <= BNDU;
   @SUM( DMND(J): CV(J) * V(J)) <= BNDV;

! A PATTERN is either open or it is not, no halfsies;
   @FOR( PATTERN( I): @BIN( T( I)););
```

解は次の通りである：

Variable	Value
Y(12)	1.000000
Y(24)	1.000000
Y(29)	1.000000
Y(32)	1.000000

前の解と違っているが，最適解は同じなどで，代替案であることが分かる．