3

AMPL：数理計画のためのモデリング言語

第 3 章

輸送モデルと割当モデル

発地から着地へ — 最小費用フローの基本形と、割当問題への応用

Fourer, Gay, Kernighan 著 ／ 日本語版

次 本章の内容

AGENDA

3-1 輸送問題の線形計画問題

圧延所（発地）から自動車工場（着地）への鋼コイル輸送を例に、最小費用輸送計画の問題を定式化する。

3-2 輸送問題の AMPL モデル

発地集合・着地集合を軸に、supply / demand / cost / Trans と目的関数・制約を一般化したモデルを組み立てる。

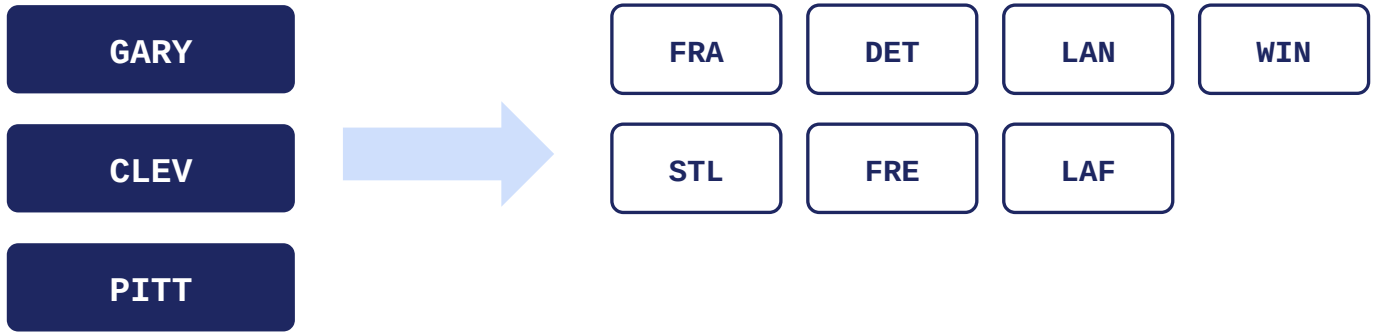
3-3 輸送モデルの他の解釈

「輸送」を伴わない対応づけ問題――割当問題――として同じモデルを再利用する方法を見る。

問 輸送問題とは

3-1　輸送問題の線形計画問題

3 か所の圧延所で生産した鋼コイルを、**7 か所の自動車工場**の注文どおりに配送する。輸送費用を最小にする配送計画は何か。



3
発地
（圧延所）

7
着地
（工場）

6,900 トン
合計輸送量
（供給＝需要）

問い：コイルを圧延所から工場へ輸送する最も安上がりな計画は何か。

GARY:FRA ＝ GARY から FRA へ輸送するトン数、というように 3 発地 × 7 着地 ＝ **21 個の変数** を用いて表す。

> 小さな輸送問題でさえ、圧延所と工場の組合せごとに 1 変数が必要になるため変数の数はすぐ増える。

表 供給地と需要地のデータ

3-1 輸送問題の線形計画問題

発地（圧延所）：週間生産量

略称	所在地	供給量（トン）
GARY	インディアナ州ゲーリー	1,400
CLEV	オハイオ州クリーブランド	2,600
PITT	ペンシルベニア州ピッツバーグ	2,900
合計		**6,900**

着地（自動車工場）：週間注文量

略称	所在地	需要量（トン）
FRA	マサチューセッツ州フレーミングハム	900
DET	ミシガン州デトロイト	1,200
LAN	ミシガン州ランシング	600
WIN	オンタリオ州ウィンザー	400
STL	ミズーリ州セントルイス	1,700
FRE	カリフォルニア州フリーモント	1,100
LAF	インディアナ州ラファイエット	1,000

供給の合計と需要の合計は一致していなければならない：

1,400 + 2,600 + 2,900 = 900 + 1,200 + 600 + 400 + 1,700 + 1,100 + 1,000 = 6,900

Σ 目的関数：総輸送費用の最小化

3-1　輸送問題の線形計画問題

1 トン当たり輸送費用（ドル）

	GARY	CLEV	PITT
FRA	39	27	24
DET	14	9	14
LAN	11	12	17
WIN	14	9	13
STL	16	26	28
FRE	82	95	99
LAF	8	17	20

目的関数（プレーンテキスト表記）

```
Minimize
  39・GARY:FRA + 27・CLEV:FRA + 24・PITT:FRA
+ 14・GARY:DET +  9・CLEV:DET + 14・PITT:DET
+ 11・GARY:LAN + 12・CLEV:LAN + 17・PITT:LAN
+ 14・GARY:WIN +  9・CLEV:WIN + 13・PITT:WIN
+ 16・GARY:STL + 26・CLEV:STL + 28・PITT:STL
+ 82・GARY:FRE + 95・CLEV:FRE + 99・PITT:FRE
+  8・GARY:LAF + 17・CLEV:LAF + 20・PITT:LAF
```

変数は全部で **21 個**（発地 3 × 着地 7）。各項は「単価 × 輸送量」で、和をとると総費用になる。

約 制約：供給制約と需要制約

3-1 輸送問題の線形計画問題

発地側の制約（圧延所ごとに 1 本、計 3 本）

```
GARY:FRA + GARY:DET + GARY:LAN + GARY:WIN
  + GARY:STL + GARY:FRE + GARY:LAF  =  1400
```

（CLEV、PITT についても同様の等式が 1 本ずつ）

着地側の制約（工場ごとに 1 本、計 7 本）

```
GARY:FRA + CLEV:FRA + PITT:FRA  =  900    (FRA 向け)
```

（他 6 工場：DET, LAN, WIN, STL, FRE, LAF についても同様）

3 + 7
= 10 本の制約

すべて非負
Trans[i,j] ≥ 0 も必要

これをすべて手入力するのは非現実的 → 一般化されたモデルが必要

集 AMPL モデル：集合とパラメータ

3-2　輸送問題の AMPL モデル

輸送モデルの根底には 2 つの基本的な集合がある：
発地（ORIG） と **着地（DEST）**。この 2 集合を軸にモデル全体を組み立てる。

```
set ORIG;   # origins
set DEST;   # destinations

param supply {ORIG} >= 0;   # 発地の供給量
param demand {DEST} >= 0;   # 着地の需要量

check: sum {i in ORIG} supply[i]
     = sum {j in DEST} demand[j];
```

check 文：供給の和と需要の和が等しいかをデータ読込み後に検査し、違反していればエラーを出す。

変 AMPL モデル：変数と目的関数

3-2 輸送問題の AMPL モデル

```
param cost {ORIG,DEST} >= 0;  # 輸送単価
var Trans {ORIG,DEST} >= 0;   # 輸送量

minimize Total_Cost:
   sum {i in ORIG, j in DEST} cost[i,j] * Trans[i,j];

# 発地 i から着地 j へ Trans[i,j] 単位を、
# 単価 cost[i,j] で輸送する。総費用は
#   cost[i,j] * Trans[i,j]  を全組合せで合計。
```

同値な書き方

- sum {j in DEST, i in ORIG} cost[i,j] * Trans[i,j];
- sum {i in ORIG} sum {j in DEST} cost[i,j] * Trans[i,j];

数学的に同じ意味であれば、書き方が違っても AMPL は同一の式として扱う。

AMPL モデル：Supply / Demand 制約

3-2 輸送問題の AMPL モデル

```
subject to Supply {i in ORIG}:
   sum {j in DEST} Trans[i,j] = supply[i];

subject to Demand {j in DEST}:
   sum {i in ORIG} Trans[i,j] = demand[j];
```

添字づけの慣習と有効範囲

- ORIG を走る添字には i を、DEST を走る添字には j を一貫して使用（AMPL の要件ではなく可読性のための慣習）。
- 添字の有効範囲（同じ意味を持つ範囲）は、その添字を定義する式の終わりまで。Demand 制約では j の有効範囲は文末のセミコロンまで、i の有効範囲は Trans[i,j] まで。
- supply（パラメータ）と Supply（制約名）は無関係 — AMPL は大文字・小文字を区別する。

解 データを与えて solve する

3-2 輸送問題の AMPL モデル

transp.dat（抜粋）

```
param: ORIG: supply :=
  GARY 1400  CLEV 2600  PITT 2900 ;

param: DEST: demand :=
  FRA 900  DET 1200  LAN 600 ...

param cost:
     FRA DET LAN ... :=
  GARY 39  14  11 ...
```

ampl セッション

```
ampl: model transp.mod;
ampl: data transp.dat;
ampl: solve;
CPLEX: optimal solution;
  objective 196200

ampl: display Trans;
  DET → GARY:0 CLEV:1200 PITT:0
  FRA → PITT:900 ...
```

196,200
最小総輸送費用（ドル）

display Trans を見ると、大半の着地は単一の圧延所から供給されているが、CLEV・GARY・PITT の 3 か所すべてが LAF へ輸送している。

較 複数の最適解が存在しうる

同じ最小費用 196,200 に対し、ソルバーによって異なる輸送計画が得られることがある（LAF 向けの内訳を比較）。

CPLEX の解	
LAF ← CLEV	400
LAF ← GARY	300
LAF ← PITT	300
目的関数値	**196,200**

SNOPT の解	
LAF ← CLEV	400
LAF ← GARY	0
LAF ← PITT	600
目的関数値	**196,200**

- このような別解の存在は輸送問題ではしばしば見られ、とりわけ目的関数の係数が切りのよい数のとき顕著。
- すべての最適解を特徴づける簡単な方法はないが、複数目的の扱い（8-3 節）でより良い解を選べる場合がある。

輸送モデルの他の解釈：割当問題

3-3　輸送モデルの他の解釈

輸送モデルは「輸送」を伴わなくてもよい。発地の量が着地と対応づけられ、目的が対応づけの価値を測るだけであれば **割当問題** になる。

例：同数の人と部屋の割当

- 発地 ＝ 個々の人、着地 ＝ 個々の部屋
- supply[i] ＝ demand[j] ＝ 1（各人は 1 部屋、各部屋は 1 人）
- Trans[i,j] ＝ 1 なら人 i は部屋 j を使用、0 なら使用しない
- cost[i,j] ＝ 人 i が部屋 j に付けた希望順位
- 目的：割り当てられた部屋の順位の合計を最小化 → 多くの人を満足させる割当を探す

目的関数の各項

```
cost[i,j] * Trans[i,j]

＝ 人 i が部屋 j に割当済み
   （Trans[i,j]=1）なら選好順位
＝ 未割当（Trans[i,j]=0）なら 0
```

総和 ＝ 0 でない項の総和
＝ 各人の割当順位の合計

例 割当問題の例：11 人を 11 部屋へ

3-3 輸送モデルの他の解釈

assign.dat（抜粋）

```
set ORIG := Coullard Daskin Hazen ... ;
set DEST := C118 C138 C140 ... ;

param supply default 1 ;
param demand default 1 ;

param cost:  C118 C138 ... :=
  Coullard  6   9  ...
```

結果（一部）

人	割当部屋	希望順位
Coullard	C118	1
Iravani	C138	2
Hazen	C246	1
Hopp	D237	1
Nelson	C140	4
… 他 6 名		

28
目的関数値
（希望順位の合計）

0 と 1
のみ Trans[i,j] が取る値

supply・demand・上下限がすべて整数 → 最適解は自動的に整数値になる（輸送モデルの特別な性質）

注意：この整数性は輸送モデル特有の性質であり、他の問題では一般に保証されない（整数計画は第 20 章）。

要 まとめ

1. 輸送問題は、発地の供給と着地の需要を最小費用で結ぶ最小費用フロー問題の最も基本的な形である。

2. AMPL モデルは 2 つの集合（ORIG, DEST）を軸に、supply / demand / cost / Trans と目的関数・Supply/Demand 制約だけで一般化できる。

3. sum{} を使えば目的関数・制約を簡潔に書け、データを差し替えるだけで規模の異なる問題にも対応できる。

4. 同じ最小費用でも複数の最適解が存在しうる（ソルバーにより輸送計画が異なる場合がある）。

5. 割当問題（supply=demand=1）は輸送モデルの特殊ケースで、整数の供給・需要のもとでは最適解が自動的に整数になる。

次章：第 4 章 へ続く