AMPL SERIES · 全 22 章

AMPL：数理計画のためのモデリング言語

Σ

はじめに

数理計画法と *AMPL* モデリング言語への招待

Robert Fourer ／ David M. Gay ／ Brian W. Kernighan（日本語版）

本セクションの内容

1. **数理計画法とは何か**
   制約のもとで目的関数を最適化するという考え方の起こり
2. **数理計画法の広がり**
   線形計画法・非線形計画法・整数計画問題という 3 つの姿
3. **AMPL モデリング言語の役割**
   モデル作成者の形式をそのまま計算機に渡す代数的言語
4. **モデルとデータを分離する設計思想**
   大規模かつ保守しやすい最適化を可能にする発想
5. **本書全 22 章の構成と入手方法**
   4 部構成の全体像と AMPL ソフトウェアの試し方

数理計画法とは何か

本書の主題には二つの側面がある。第一は**数理計画法**（mathematical programming）——制約条件のもとで多変数関数を最適化することである。第二は、その開発・適用を助ける **AMPL モデリング言語** である。

「計画法（programming）」の語源

1940 年頃、大組織における活動の立案・日程計画を指す語として使われ始めた。「プランナー」は各活動の水準を**変数**として表現し、立案上の制限を等式・不等式の集まりとして数学的に記述した。

数理計画法の定式化

1. **変数**
   活動の量・水準を表す未知数
2. **制約**
   変数に関する等式・不等式の集まり
3. **目的**
   解の良し悪しを測る費用・利益などの関数
4. **最適解**
   目的を最小化／最大化する受理可能解

なぜ「目的」が鍵となる洞察か

制約を指定するだけでは複雑な業務のモデル化は難しい――制約が少なすぎれば劣った解が数多く残り、多すぎれば望ましい解が排除され、最悪の場合は解が存在しなくなる。

! 制約だけで表現すると

- 制約が少なすぎる → 劣った解が数多く条件を満たす
- 制約が多すぎる → 望ましい解が排除される
- 最悪の場合、条件を満たす解が一つも存在しなくなる

✓ 目的を追加すると

- 費用・利益など、解の良し悪しを測る目的関数を導入する
- 多くの解が制約を満たしても問題にならない
- 目的を最小化・最大化する解を一つ見つければ十分

数理計画法 = 制約のもとで目的関数を最小化・最大化すること

数理計画法の広がり

目的・制約の性質によって、数理計画問題はいくつかの特別な場合に分けられる。

L

線形計画問題

Linear Program

費用・必要量などが活動水準に厳密に比例――目的も制約もすべて線形。

数千変数でも高速・確実に解ける単体法（1940 年代後半〜）が発展の礎。

N

非線形計画問題

Nonlinear Program

目的や制約に滑らかな非線形関数が現れる場合。「大規模最適化」とも呼ばれる。

解くのはより困難だが、線形計画法の成功を受け近年になり実用計算手法が発展。

I

整数計画問題

Integer Program

一部の変数が整数値をとらなければならない場合。一般にはるかに難しい。

計算機の高速化と手法の洗練により、大規模問題も年々扱いやすくなっている。

AMPL とは何か

AMPL は数理計画法のための**代数的モデリング言語**（algebraic modeling language）である。1985 年頃に著者らによって設計・実装され、以来進化を続けてきた。

算術式が代数記法に酷似

x + y、Σ a[i,j]・x[j]、x ≥ 0 のような見慣れた数学記法をそのまま計算機で扱える

集合と添字が汎用的で強力

問題の構造に応じて柔軟に変数・制約の集まりを表現できる

数理計画特有の構造も表現

ネットワークフロー制約や区分的線形性なども言語として拡張

数式 → AMPL コード

伝統的な数学記法

```
最大化: Σj cj xj
条件:   Σj aij xj ≤ bi    （すべての i）
        xj ≥ 0
```

AMPL 記述

```
maximize Profit:
   sum {j in J} c[j] * x[j];
subject to Limit {i in I}:
   sum {j in J} a[i,j] * x[j] <= b[i];
var x {j in J} >= 0;
```

モデルとアルゴリズムをつなぐ実務の流れ

実務の数理計画法は、単にアルゴリズムを走らせて終わりではない。次のサイクルを繰り返す。



課題：モデル作成者の形式とアルゴリズムの形式の間には大きな隔たりがある

線形計画法では、制約係数行列（全制約×全変数の数表）が形式の大部分を占める。行数・列数が数百〜数十万に及ぶ疎行列を生成する「行列生成器」を書く方法は、変換作業を自動化できる一方、デバッグと保守が難しいままである。

モデリング言語という解決策

モデリング言語は、モデル作成者の形式を計算機システムへの直接の入力として表現できるように設計されている。翻訳という中間段階をすべて計算機に任せられるため、開発と保守がより経済的・信頼できるものになる。

馴染み深い表記

x+y、Σ aij xj、xj≥0 など、代数や解析を学んだ人になじみの記法をそのまま使える

多様なモデルに対応

線形・非線形・整数計画のきわめて多様なモデルに適用できる

大規模問題にも通用

数百〜数千、さらには数十万規模の変数・制約でも同じ手法が通用する

代数的モデリング言語の歩み

1950s

数理計画法の実用的なアルゴリズムが初めて使われる

1970s〜

代数的モデリング言語の開発・普及が始まる

1985 頃〜

AMPL が設計・実装され、以来進化を続けている

§ 本書の構成（全 22 章・4 部構成）

1

第 1 部　1〜4 章

線形計画法への入門

生産モデル／配合モデル／輸送・割当モデル／より大きなモデルの構築。できるだけ早く AMPL を使い始めるためのチュートリアル。

2

第 2 部　5〜8 章

AMPL の基本構成要素

単純な集合と添字／複合集合と添字／パラメータと式／変数・目的・制約。一般的なユーザーズガイドとしても使える。

3

第 3 部　9〜14 章

AMPL の高度な使い方

データの指定／データベースアクセス／モデリングコマンド／表示コマンド／コマンドスクリプト／ソルバーとのやりとり。

4

第 4 部　15〜20 章

特別な場合と一般化

ネットワーク線形計画／列単位の定式化／区分的線形計画／非線形計画／相補性問題／整数線形計画。付録 A に AMPL リファレンスマニュアル。

第 2 版と AMPL の入手方法

第 2 版について

- AMPL の中核は 10 年間で本質的に変わっておらず、第 1 版のモデルはほぼすべて現行版で動作する。
- 全面的に実質改訂したが、新規内容の大半は第 3 部に集中。
- コマンド環境の章が 1 章 → 5 章に拡張。
- データベースアクセス・スクリプト・プログラミング構文はすべて新規。

無償

www.ampl.com から「学生版」AMPL と代表的なソルバーを Windows／Unix・Linux／Mac OS X 向けに無償ダウンロード可能。数百個程度の変数・制約なら十分扱える。

`www.ampl.com/TRYAMPL`

ブラウザからダウンロードせずに AMPL を試せる

`neos.mcs.anl.gov`

NEOS サーバーのブラウザインタフェースを利用可能

★ まとめ

1	**数理計画法とは**	制約のもとで多変数の目的関数を最小化・最大化する数学的枠組み。
2	**成功の鍵は「目的」**	制約だけでなく目的関数を導入したことが、実用化を可能にした。
3	**AMPL は代数的モデリング言語**	モデル作成者の形式をそのまま計算機への直接入力にできる。
4	**モデルとデータの分離**	同じモデルを異なるデータ集合に適用でき、大規模問題にも通用する。
5	**本書は全 22 章 4 部構成**	入門チュートリアルから高度な機能・特別な場合までを体系的に解説。

次章から、*AMPL* による最初の線形計画モデル（生産モデル）を扱う。