

〔論文〕

## 最適化ソフトの比較と最近の動向

江戸川大学 メディアコミュニケーション学部 情報文化学科 ザンピン

近年、コンピュータのデータ処理スピード、あるいは、スペックの向上は周知の通りである。筆者の専門分野である最適化に関連する専用ソフトも、ここ 10 年、コンピュータに勝る発展を遂げてきた。ここには、IBM ILOG OPL を中心に、いくつかの代表的なソフトとその特徴、および CPLEX の創始者 R.Bixby の講演を含む商用ソフトの発展をここでまとめて紹介する。

### 1. Excel のソルバー

最適化ソフト、あるいは、Optimizer といえ、Excel のソルバーが一番よく知られている。近年、PC の性能が飛躍的な向上につれ、数百個の変数、数千個の変数の問題でも、合理的な時間内で解けるようになり、その実用性が発揮されてきた。

しかし、この Excel のソルバーは整数、あるいは、0、1 のようなバイナリ変数を含む混合整数の問題に対し、極めて不安定である。例を見てみよう。

工場から市場が集中している都会に近い倉庫へ製品を輸送すると仮定する。ここには運送コストを最小にすると考えよう。また、各工場の供給能力を超えず、しかも都市部の市場需要を満たすという条件を付け加える。このような問題がいわゆる線形計画で、Excel のソルバーでも簡単に解を得られる。しかし、一つの市場がいくつかの工場から同時に仕入はやはり不便で、一つの市場が一つの工場からしか導入できないという制約条件を入れれば、これはいわゆる整数計画問題になる。このような整数問題について、わずか数個変数の小さい問題でも、ソルバーからには正しい解を得られなくなる。

図 1-1 は、運送コストについての典型的な例。B 列は各工場の生産能力、7 行目は市場の需要、セル C4~G6 は 1 単位あたりの郵送コスト、例えば、新潟から名古屋まで 1 単位の運送コストは 1000 円。

	A	B	C	D	E	F	G
1	例: 輸送コストが最小化						
2			名古屋	京都	大阪	仙台	東京
3	工場 ↓	供給能力 ↓	工場Xから倉庫Yへの運送費/単位				
4	新潟	410	1000	800	600	500	400
5	群馬	300	600	500	400	300	600
6	金沢	350	300	400	500	500	900
7	市場の需要 →		160	80	180	160	220

図 1-1 : 運送コストの入力データ

変化させる量は 0, 1 しか取らないバイナリ変数と設定。(新潟、名古屋) = 1 とは名古屋倉庫の補充は新潟から、(新潟、名古屋) = 0 は、新潟からは補充しないという意味。これらの変数はセル C10 から G12 に充てる。

他の式の入力は表 1-1 にまとめる。

セル	入力式	コピー
C13	=SUM(C10:C12)	C13~G13
C16	=C10*C\$7	C16~G18
B16	=SUM(C16:G16)	B16~B18
C19	=SUM(C16:C18)	C19~G19
B21	=SUMPRODUCT(C4:G6,C10:G12)	

表 1-1 式の入力

以上入力した結果は下の図 1-2 にまとめる。変化する量は初期値が与えられた。図 1-5 は初期値と結果の比較。

	A	B	C	D	E	F	G
1	例: 輸送コストが最小化						
2			名古屋	京都	大阪	仙台	東京
3	工場↓	供給能力↓	工場Xから倉庫Yへの運送費/単位				
4	新潟	410	1000	800	600	500	400
5	群馬	300	600	500	400	300	600
6	金沢	350	300	400	500	500	900
7	市場の需要→		160	80	180	160	220
8							
9			変化させる量↓				
10			1	0	0	0	0
11			0	1	0	0	0
12			0	0	1	0	0
13	1工場の条件→		1	1	1	0	0
14							
15	各倉庫からの合計↓		工場Xから倉庫Yへの出荷量↓				
16		160	160	0	0	0	0
17		80	0	80	0	0	0
18		180	0	0	180	0	0
19	市場への合計→		160	80	180	0	0
20							
21	総費用	290000					

図1-2 EXCELシートに入力した初期データ

「ソルバー：パラメータ設定」は図1-3の通り。制約の意味について表1-2にまとめられている。ソルバーによる計算結果は図1-4に表示される（ $3.79E-08$  は  $3.79 \times 10^{-8} \approx 0$ ）。



図1-3 パラメータの入力

図1-5から分かるようにバイナリの条件を満たしてないまま、計算が止まったり、例えば条件を満たされでも、最適値（328000

図1-2、最適値 324000）と最適解がばらばらである（この問題の最適値 314000 は第3節に与えられている）。

次の節は現在最適の分野に使われている専用ソフトを紹介する前に、20~30年前によく使われているソフト Lindo を簡単に触れる。

入力	意味
$\$C\$10:\$G\$12 = \text{バイナリ}$	変化する量は0, 1の値しか取らない
$\$C\$13:\$G\$13 = 1$	一倉庫は一工場からしか供給しない
$\$B\$16:\$B\$18 \leq \$B\$4:\$B\$6$	各工場の出荷量はその工場の供給能力を上回らない
$\$C\$19:\$G\$19 \geq \$C\$7:\$G\$7$	各倉庫への入荷量はその地域の需要以上

表1-2：制約条件の意味

1	例: 輸送コストが最小化					
2		名古屋	京都	大阪	仙台	東京
3	工場↓	供給能力↓	工場Xから倉庫Yへの運送費/単位			
4	新潟	410	1000	800	600	500
5	群馬	300	600	500	400	300
6	金沢	350	300	400	500	900
7	市場の需要→		160	80	180	160
8						
9			変化させる量↓			
10			0	0	0	1
11			0	1	1	0
12			1	0	2.11E-10	1
13	1工場の条件→		1	1	1	1
14						
15	各倉庫からの合計↓		工場Xから倉庫Yへの出荷量↓			
16		220	0	0	0	220
17		260	0	80	180	0
18		320	160	0	3.79E-08	160
19	市場への合計→		160	80	180	220
20						
21	総費用	328000				
22						

図1-4 ソルバーによる計算結果

変化させる量↓				
1	0	0	0	0
0	1	0	1	0
0	0	1	0	0
1	1	1	1	0

(a) 初期値

変化させる量↓				
1	0	0	0	0.34
0	1	0	0.95	0.31
0	0	1	0.05	0.34
1	1	1	1	1

(b) (a)の初期値に対しソルバーの計算結果

変化させる量↓				
1	0	0	0	0
0	1	0	1	0
0	0	1	0	1
1	1	1	1	1

(c) 初期値

変化させる量↓				
0	0	0	0	0.91
0.2785	1	0.09	1	0
0.7215	0	0.91	0	0.09
1	1	1	1	1

(d) (c)の初期値に対しソルバーの計算結果

変化させる量↓				
1	0	0	0	0
1	0	1	0	1
0	1	0	0	0
2	1	1	0	1

(e) 初期値

変化させる量↓				
0	0	1	0	1
0	-0	0	1	0
1	1	0	0	0
1	1	1	1	1

(f) (e)の初期値に対しソルバーの計算結果

図1-5 初期値とソルバーによる計算結果の比較

## 2. 数理計画ソフト Lindo

LindoはLindo社が1981年に公開され、長い間に多くの数理計画者に利用されてきたソリューションである。筆者も教育の現場から、このソフトをはじめて知り、教育に適するソフトと感じる。

上の節で述べた運送コスト最初化の線形計画問題（式を簡単にするため、1倉庫の提供を1工場の整数制約を除き）を数式で定式すると、次のようになる。

まず、記号を定義する。供給元の工場は  $m$  個あり、提供先の倉庫（市場）が  $n$  個ありと仮定する：

$x_{ij}$  は工場  $i$  から倉庫  $j$  への運送量、

$c_{ij}$  は工場  $i$  から倉庫  $j$  への単位あたりの運送コスト、

$a_i$  は市場  $i$  の需要、

$b_j$  は工場  $j$  の生産能力

以上の記号により、運送コスト最小化の問題が次のような線形問題にモデリングすることができる

$$\begin{aligned} \min \quad & \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \\ \text{s.t.} \quad & \sum_{j=1}^n x_{ij} \geq a_i \quad \text{for } i = 1, 2, \dots, m, \\ & \sum_{i=1}^m x_{ij} \leq b_j \quad \text{for } j = 1, 2, \dots, n, \\ & x_{ij} \geq 0. \end{aligned}$$

このような運送モデルに対し、Lindoの入力は図2-1のようになる（Lindoのバージョンが少し古いが）。見ている通り、多

くの数理モデルが付いているゼロ以上の制約条件が省略されている以外に、入力はモデルの数式そのものである。そのため、Lindoのソフトが学びやすい、特に教育に適する。この問題のLindoの解は図2-2に示されている。

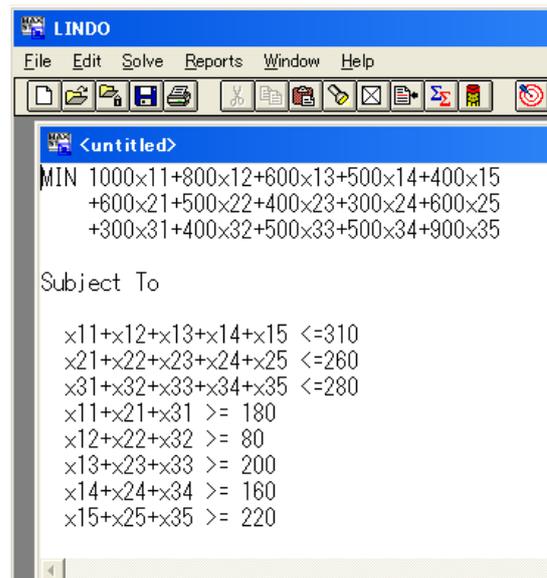


図2-1 Lindoの入力例

VARIABLE	VALUE	REDUCED COST
X11	0.000000	30600.000000
X12	0.000000	300.000000
X13	0.000000	0.000000
X14	80.000000	0.000000
X15	220.000000	0.000000
X21	0.000000	400.000000
X22	0.000000	200.000000
X23	180.000000	0.000000
X24	80.000000	0.000000
X25	0.000000	400.000000
X31	180.000000	0.000000
X32	80.000000	0.000000
X33	20.000000	0.000000
X34	0.000000	100.000000
X35	0.000000	600.000000

図2-2 Lindoによる実行結果

### 3. ILOG の OPL IDE

Lindo 社の教育向きと対照的に、IBM ILOG 社の OPL (Optimization Programming Language) は最適化プログラミング言語を利用し、企業の意思決定の手段の一つとして、技術者向きである。その IDE (Integrated Development Environment) ,つまり、統合開発環境 (図 1-1 を参照) は大変便利が、利用するには複雑すぎと思われるかもしれない。

しかし、筆者が数ヶ月前に、このソフトを始めてインストールし、そのときに、ドキュメントとマニュアルを多少読んだ。この紹介文を書くために、今度与えられた運

送コスト最小化問題に対し、初めて自分が一からモデルファイルを書いたが、まるでオンラインデバッグのように、ドキュメントを参照しながら、1 時間前後でモデルファイル (とデータファイル) を書き、実行し解が得られた。

そのため、OPL の IDE について、ここでは第一節の運送費用コスト最小化の例を通じて紹介する。ILOG の一般的な知識、また、他の環境との関連などについて、インターネットにすでに多数存在し、ここには割愛させてもらう。

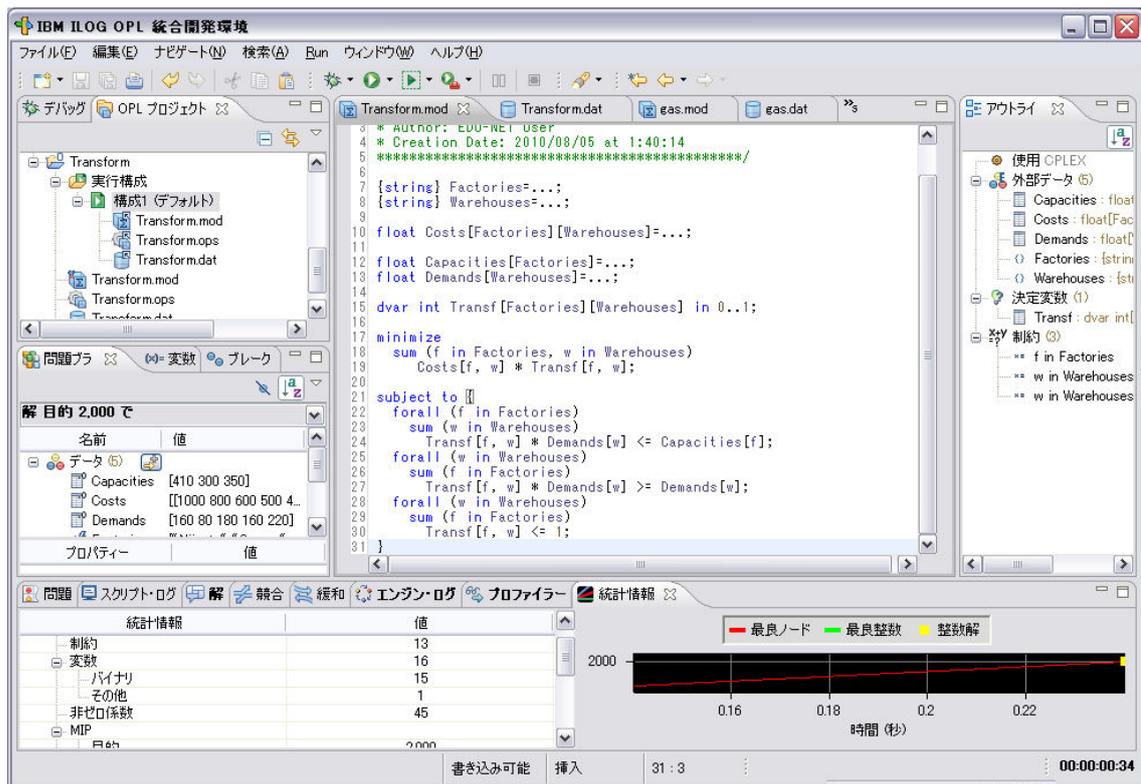


図 3-1 ILOG の OPL の統合開発環境

3-1 プロジェクトの新規作成  
Excel、また、Lindo と違い、OPL は単

一のファイルではなく、プロジェクトで動作する。プロジェクトの新規作成する際に、

「オプション」(図3-2)のところにチェックを入れるのは一番簡単である。

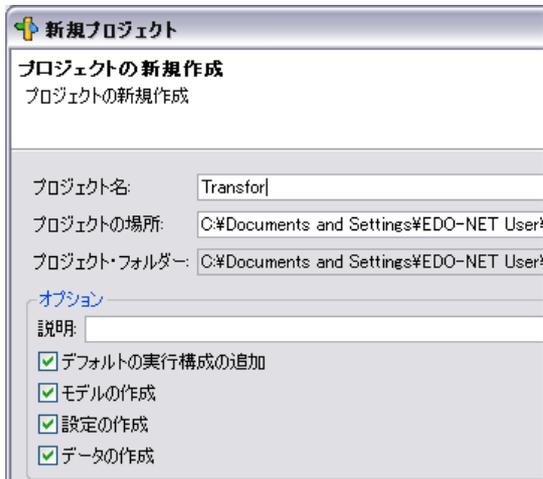


図3-2 新規プロジェクト

上の操作によって、モデルファイル、データファイル、パラメータなどの設定用のオプションファイルと実行構成は次の図3-3のように自動生成される。

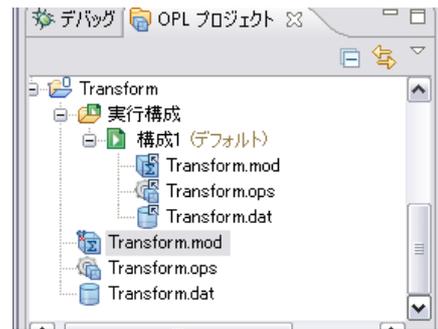


図3-3 プロジェクトの構成

### 3-2. モデルファイル

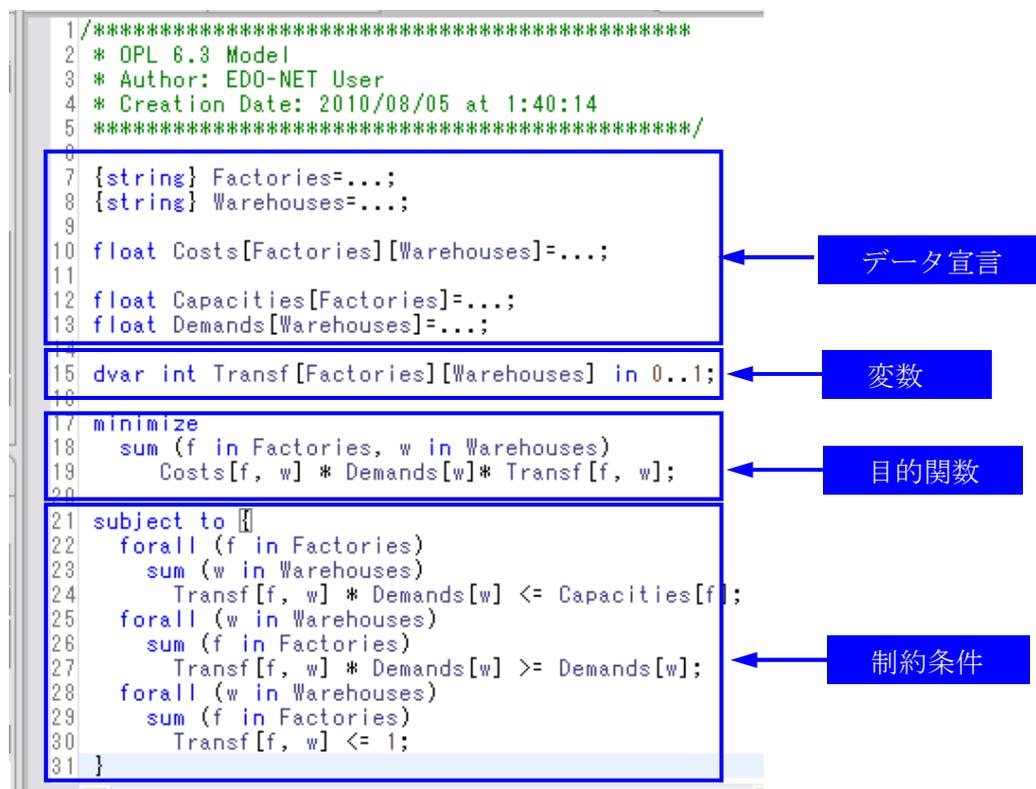


図3-4 モデルファイル

モデルファイルは ILOG OPL IDE の一番基本的なコンポーネント。最適化プログラミング言語を利用し、記述する。図 3-4 には運送コスト最小化モデルファイルが与えられている。文法的な詳細の説明はここで省略する。

モデルファイルから分かると思うが、データ宣言、目的関数と制約条件の和、集合の表現で、データの長さ、つまり、サイズさえ限定されていない、この点でも、Excel と Lindo より明らかに汎用性が拡張されている。

### 3-3 データファイル

具体的な問題のデータは直接にモデルファイルのデータ宣言の直後に代入するのも可能だが、汎用性を考えれば、データファイルにまとめたほうがよい。図 3-5 は運送問題のデータファイル。モデルファイルとともに、Excel の場合と比較すれば大体理解できるかと思う。また、データファイルが Excel、Access などからもインポートできる。

```

1/*****
2 * OPL 6.3 Data
3 * Author: EDO-NET User
4 * Creation Date: 2010/08/05 at 1:40:14
5 *****/
6
7 Factories = { "Niigata" "Gunma" "Kanazawa" };
8 Warehouses = { "Nagoya" "Kyouto" "Osaka" "Sendai"
9               "Tokyo" };
10
11 Costs = [ [1000 800 600 500 400]
12           [600 500 400 300 600]
13           [300 400 500 500 900] ];
14
15 Capacities = [410 300 350];
16
17 Demands = [160 80 180 160 220];
18

```

図 3-5 データファイル

### 3-4 実行

モデルファイルなどについて間違いがなければ、実行し、解が得られる。前にも述べたが、IDE は優れたデバックの機能が備えている。文法的なミスは実行する前に、コーディングの段階でほとんど示され、大変便利である。

例の運送問題の実行結果、最適値および最適解は図 3-6 に示されている (Excel の実行結果と比較してください)。

図 3-6 運送問題の実行結果 (解)

この節の最後に、一筆付け加える。この OPL の IDE の紹介は、あくまでも、ILOG OPL のイメージであり、Beginning、つまり、第一歩である。OPL の IDE はたくさんの大変優れた機能が備えている、これらを使いこなせるために、大量の学習時間が不可欠である。

## 4. 最適化ソフトの動向

最適化の商用ソフトは 1970-71 年ごろ開発され、販売された。しかし、1990 台までの 20 数年の間に、理論的な研究成果がソフトの実装と改良にほとんど貢献されていない。1990 年代の後半から、状況が一変し、ここ 10 年間は平均で毎年 2 倍の速度 (成功的な双対理論の実装、Gomory cut の場合に、年 5~8 倍)、これらの指数増加の効果により、10 年で 3400 倍のスピードで改良されてきた。これは、コンピュータハードのスピード 1600 倍より早い。これらは、R.Bixby のような一流の最適化理論の研究者が陣頭に直接に汎用ソフトの開発に携わることと大いに関係がある。

近年、ビジネス世界の情報化、競争化、グローバル化、サプライチェーンによる協働化により、最適化による意思決定がますます重要となり、最適化ソフトの更なる発展および他のソリューションとの統合、一層目が離せない。

### 謝辞 :

筆者は IBM の ILOG 社の大学に対し、ソフトの無料提供に心から感謝している。

**参考文献：**

- 「1」 IBM、 Optimization modeling with  
IBM ILOG OPL、 P1~354、 2009
- 「2」 S.Chopra,P.Meindl、 Supply Chain  
Management、 3th、 Pearson  
International Edition、 2007
- 「3」 R.Bixby、 A lecture in  
Tsukuba-University、 2010.5