

■研究最前線

■実社会に通じる最適化法の研究

最適化法を用いた問題解決策の追求

数学とコンピュータの力で最適な状態を求める

●環境都市工学部
檀 寛成 助教

最適化とは、数学的に表現される目的関数と制約条件があるとき、その制約を満たしながら目的関数の値を最小、あるいは最大にするというものであり、数学とコンピュータの力により解を求める。では、最適化法は社会のどのような場面に活用されているのだろうか。最適化法の実問題への適用を研究している檀助教に話を聞いた。

■問題を定式化し、解決する

—どのような学問なのですか？

わかりやすく言うと、数学とコンピュータを利用して、制約を考慮しながら最もよい状態を探し出すという学問分野です。例えば、工場が原料を調達して配送するまでの物流合理化問題を考えると、価格も違い何か所もある原料の購入元、各工場の作る製品の種別や出荷可能量などが制約。これらを組み合わせ、最もよい解…この場合はどこで原料を購入し、どの工場、どれだけの量を生産すれば最もコストが安くなるか、あるいは利益が多くなるかという最適解を求めます。

—現実問題への適用のプロセスは？

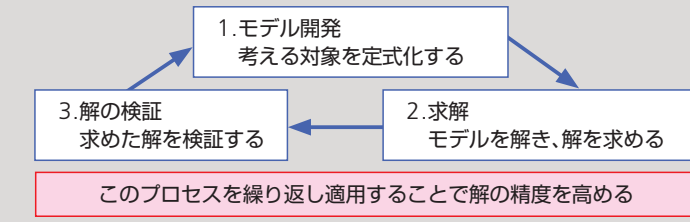
最適化法の適用プロセスには、①問題のモデル化 ②求解 ③解の検証 という、3つの段階があります。まず①では、実際の問題を抽象的に捉え、数学的に表現します。数学とコンピュータは抽象的にものを扱うことができるため、汎用性があります。上の例でいう工場の生産量など、決定すべき量は変数Xなどします。その際、「X」は制約と目的を数学的に表現できるものであれば何でもよいのです。制約として、等しくなければならない場合は「=」。多い少ないは「≤」「≥」で表します。次に②では、定式化した問題を解きます。ソルバというソフトウェアで解くのが一般的であり、問題に適したアルゴリズム(解法)を選択し、解を求めます。または新しくアルゴリズムを作って解を求める場合もあります。最後に③では、②の解が正しいかどうかを検証します。①の時点での間違いや、入力データのケアレスミスもあり得ます。実際の状況に合致しているかを調べ、間違っていれば①②に戻ってモデルを修正し、解を求めます。それを繰り返すことで、満足のいく解を求めます。仕事と同じで、最終的には現場の人間の判断が重要です。

■多様なジャンルで応用されている最適化法

—輸送問題のほか、どのような場面で応用されているのですか？



▶最適化法の適用プロセス



面白いものでは「最長片道切符」があります。JRの片道切符で乗れる最も距離の長い路線ルートを最適化により導き出し、募金を集めて実際に最長片道切符を購入し、電車に乗って話題となりました。また、Jリーグの対戦順を決める際、なるべく均等な条件でスケジュールを組むために利用されています。ほかにも、身近なところでは最短ルートを求める必要のあるカーナビなどにも応用されています。最適化という言葉の認知度は低いですが、実は日常に深く繋がっているのです。

—ご自身ではどのような研究をされているのですか？

まだ運用されていませんが、大学における時間割作成問題を整数計画問題として定式化する研究などを手掛けています。現状、時間割は担当者が時間をかけて手作業で作成していますが、それを定式化し、問題を設定します。そこさえ作ってしまえばあとはコンピュータが作成してくれるため、実際の運用に至れば作業時間は短縮され、負担もかなり軽減できるはずです。人間が作る場合は毎年一からの作業ですが、定式化した問題には汎用性があるので、一旦定式化してしまえば条件が変わらない限り毎年同じものが使え、手間は一回でよくなります。

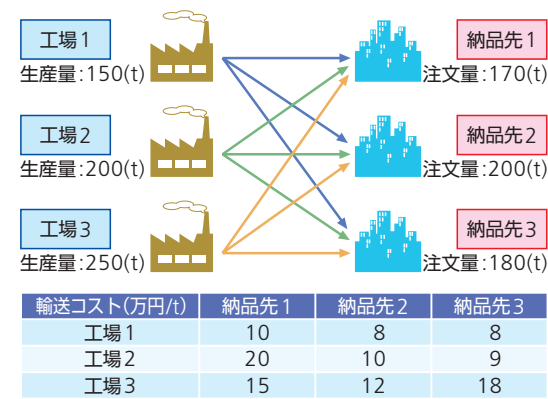
—最適化法を運用する会社での社員経験もあるそうですが。

ソフトウェアベンダーで、実際にお客様から持ち込まれた問題を定式化し、解を求め、問題解決してソリューションを提供するという、最適化に関する仕事をしていました。また、ソルバの実装もしていました。

より研究にシフトするため在社中に学位を取得し縁あって関西大学に着任しましたが、本学に着任してからも、総合ビル管理会社との共同研究で「エレベータの保守点検作業の効率化」という共同研究を行いました。これは、安全性が求められるエレ

▶最適化問題の具体例(輸送問題)

- 工場から納品先に商品を輸送する
- 工場の輸送量：生産量以下、納品先への輸送量：注文量
- 輸送にはコストが発生する
- コストを最小にするにはどのように輸送すればよい？



▶輸送問題の定式化

●工場を i 、納品先を j で表す。

変数 x_{ij} : 工場 i から納品先 j への輸送量 (量を表すので0以上)

パラメータ(データ)

s_i : 工場 i の生産量

d_j : 納品先 j の注文量

c_{ij} : 工場 i から納品先 j への輸送コスト

定式化した最適化問題

目的関数: $\sum_{i,j} c_{ij}x_{ij} \rightarrow \text{最小}$
・輸送コストの合計を最小にする

制約条件1: $\sum_j x_{ij} \leq s_i$
・工場 i から各納品先 j への輸送量の合計が生産量 s_i を上回らない

制約条件2: $\sum_i x_{ij} = d_j$
・各工場 i から納品先 j への輸送量の合計が注文量 d_j と一致する

▶求解結果

●最適解(最小輸送コスト: 5780万円)

最適輸送量(t)	納品先1	納品先2	納品先3
工場1	150	0	0
工場2	0	20	180
工場3	20	180	0

- ・各工場からの輸送量：工場の生産量を超えていない
- ・各納品先への輸送量：注文量と一致

工場/納品先の数1000程度になっても、普通のコンピュータで数秒程度で解を求めることが可能!

ベータの保守作業において、人間が考えた従来の手順が本当に最適かどうかを検証することが目的です。エレベータのピット(地下)に入り作業するためには安全性を確保するための一連の手順があります。この作業手順が最適であるかを調べるため、ルールを定式化し、実際のデータと合わせてソルバで解き、最適手順を求めたのです。結論としては、従来の手順が最適でした。また、作業手順が漏れた際、その後の作業のどこかに安全な状態で挟みこめるタイミングを見つける、なるべく手戻りの少ない状況を見つけて作業するなど方法も提案しました。そのほか、消防力の最適配置システムの開発などにも携わったこともあります。

Research Front Line

▶問題をソルバがわかるように記述したもの

```
set I; # 工場の集合
set J; # 納品先の集合

var x{I, J} >= 0; # 輸送量(0以上)
param s{I}; # 各工場の生産量
param d{J}; # 納品先の注文量
param c{I, J}; # 輸送コスト

minimize f: sum {i in I, j in J} (c{i, j} * x{i, j});
# 輸送コストの和を最小に

s.t. C1{i in I}: sum {j in J} x{i, j} <= s{i};
# 制約1: 各工場からの輸送量が生産量を超えない

s.t. C2{j in J}: sum {i in I} x{i, j} == d{j};
# 制約2: 各納品先には注文量を輸送する
```

▶データ(各工場の生産量、各納品先の注文量、工場・納品先間の輸送コスト)

```
set I := 1 2 3;
set J := 1 2 3;

param s := [1] 150 [2] 200 [3] 250;
param d := [1] 170 [2] 200 [3] 180;

param c :=
[1, 1] 10 [1, 2] 8 [1, 3] 8
[2, 1] 20 [2, 2] 10 [2, 3] 9
[3, 1] 15 [3, 2] 12 [3, 3] 18;
```

ソルバに入力

▶ソルバの出力—最適解に関する情報が記載されている

```
Problem: sample
Rows: 7
Columns: 9
Non-zeros: 27
Status: OPTIMAL
Objective: f = 5780 (MINimum)
```

No.	Row name	St	Activity	Lower bound	Upper bound	Marginal
1	f	B	5780			
2	C1[1]	NU	150		150	-5
3	C1[2]	NU	200		200	-2
4	C1[3]	B	200		250	
5	C2[1]	NS	170	170		15
6	C2[2]	NS	200	200		12
7	C2[3]	NS	180	180		11

No.	Column name	St	Activity	Lower bound	Upper bound	Marginal
1	x[1,1]	B	150	0		
2	x[1,2]	NL	0	0		1
3	x[1,3]	NL	0	0		2
4	x[2,1]	NL	0	0		7
5	x[2,2]	B	20	0		
6	x[2,3]	B	180	0		
7	x[3,1]	B	20	0		
8	x[3,2]	B	180	0		
9	x[3,3]	NL	0	0		7

Karush-Kuhn-Tucker optimality conditions:

KKT.PE: max.abs.err. = 0.00e+00 on row 0
max.rel.err. = 0.00e+00 on row 0
High quality

KKT.PB: max.abs.err. = 0.00e+00 on row 0
max.rel.err. = 0.00e+00 on row 0
High quality

KKT.DE: max.abs.err. = 0.00e+00 on column 0
max.rel.err. = 0.00e+00 on column 0
High quality

KKT.DB: max.abs.err. = 0.00e+00 on row 0
max.rel.err. = 0.00e+00 on row 0
High quality

End of output

■一般ユーザー向けソフトウェアの開発

—今後の課題をお聞かせ下さい。

現在、最適化法における3つのプロセスの運用全体を効率化するソフトウェアを作成中です。具体的には、モデルの定式化をサポートし、ソルバを用いて解を求め、解の検証にも役立つソフトウェアであり、大学院生・卒業研究生たちと一緒に作成しています。汎用性を持たせることを目指してはいますが、ユーザーの使いやすさを考えるとその分野に特化した仕組みが入っていた方がよいため、カスタマイズもできるように考えています。