

# タダより安い数理計画入門

久保 幹雄

## 登場人物紹介

**N教授:** 元女流物流コンサルタント。現在、大学教授。趣味「お昼寝」

**Rさん:** 某有名メーカー物流部長。儉約家。好きな言葉「タダより安いものはない。」

暖かい日差しが差し込む午後、いつものようにN教授は教官室でウトウト、もとい、熱心に研究をしていました。そこに、あるセミナーで知り合った某メーカー物流部長のRさんが物流費用削減の相談にやってきました。

「どうも、どうも、先生。お忙しいところ申しわけありません。早速ですが、先日電子メールでお話したように、我が社では、物流費用の大幅な削減のためにトラック輸送の見直しをしているのですが、どうも問題が複雑で...」

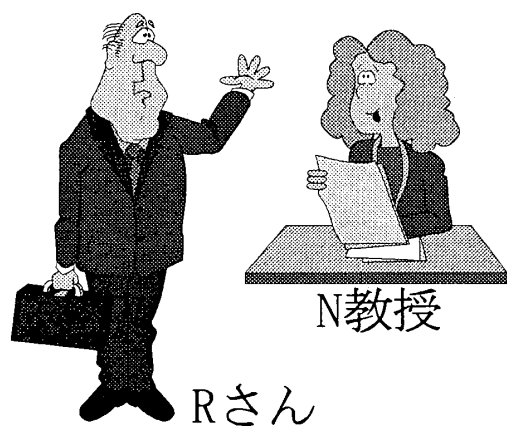


図 1: N教授とRさん

N教授は眠そうな目を擦りながら答えた。

「たしか配送計画の問題ね。この問題に関しては実にたくさんの研究がされているわ。たとえば、Elsevierという出版社から出ている Handbooks in Oper-

ations Research and Management Science シリーズの第8巻に“Network Routing”が最近登場したのだけど、800ページ近くあるこの大著のほとんどが運搬経路問題という一つの問題とその変形に対する解析に費やされているほどよ。」

「でも先生、私も会社にあった物流関連の本をいろいろ調べたのですが、だいたい哲学とか理念とか、もしくは新聞の切り貼りみたいな事例の紹介ばかりで...、それに先生のご専門の数理的なモデルといえはKantrovichの輸送モデルとか、IBMのVSPXとか、Weberの重心法などが載っていましたが、どうも我が社の問題とは若干異なっているようで... そう、なんと言ったらいいか... あの、大変申し上げ難いのですが... 使いものにならないというか...」

「たしかにそういった教科書レベルのモデルだけで実際の問題が解決できることは滅多にないわね。この分野の教科書に書かれているのは、ほとんど四半世紀前までの研究成果だから、Rさんが誤解されるのも無理はないわ。でもね、Kantrovichの論文は1939年、VSPXのもとになったClarkeとWrightの論文は比較的新しいといっても1964年、Weberの論文にいたっては1929年のものなのよ。」

「そんなに昔の話なんですか... ということは、この分野の研究はその後あまり進んでいないということなんですね。」

「いえいえ、とんでもない!この分野の研究はここ10年で飛躍的に進んだのよ。専門雑誌を見ていただければお分かりになると思うけど、ものすごい勢いで論文や事例が出てきていて、今では数千本の論文が出ていると言われているのよ。もちろん、その数千の論文を全部読んでから来てちょうだいと言っている訳じゃないわ。問題の概要を教えていただければ、我々専門家がその問題に近い論文を探し出して、それに若干のスパイスを与えてあげると言っているだけなのよ。」

「はあ、問題の概要ですか。それなら簡単です。数十台のトラックの一日のスケジュールを決めるだけなんですけど、勤務時間とか積載量の制約とかがついてい

くぼ みきお 東京商船大学 流通情報工学  
〒135 東京都江東区越中島 2-1-6  
e-mail: kubo@ship2.ipc.tosho-u.ac.jp

ますし、大きいトラックだと入れないお客様とか、何時から何時までに持って来いというお客様がたくさんいるために問題がとても複雑になっているんです。」

Rさんは黒板に図を書きながら説明をしていたが、途中で教授が質問をした。

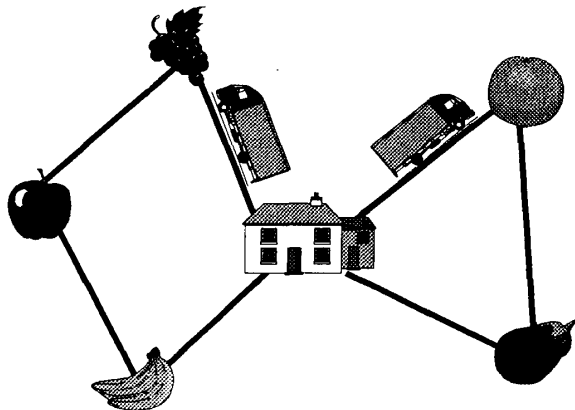


図 2: Rさんの会社の配送計画

「ちょっといいかしら、トラックが配送センターを出た後、何人くらいのお客さんを廻ってから帰ってくるかを教えていただけませんか? それによって問題の難しさ、言い換えれば使う解法が変わってくるのよ。」

Rさんは持ってきた資料を開いた。

「はい、この問題の場合には多くて4件です。我社ではかなりの重量物を運ぶので、4件のお客様を廻れば、再び配送センターに戻ってこなければならないからです。」

「4件! たった4件ですって? それならルート生成法とよばれる解法がいいわね。この解法は単純だし、既存の数理計画のソフトウェアを利用できるので開発も短時間でできるし... それで、この解法のためには、あらかじめルートを生成しておくことが必要ね。」

「えっ? ルートを生成しておくですって? 教授のおっしゃっているルートというのはトラックの一日の運行スケジュールのことなんですか?」

「ええそうよ。もっと正確に言うと、ルートというのは“配送センターを出発して、何人かのお客さんを訪問して、再び配送センターに戻ってくる巡回路”のことよ。」

「だったらおかしいじゃありませんか? だってルートを見つけることが相談に来た問題の目的なんですよ。それをあらかじめ与えておこなって、テストの前に解

答を配ってしまうようなものですよ。」

「あら、ごめんなさい。私が言ったのは実行可能なルートの候補のことなの。つまり、一日の運行スケジュールになりそうなものを列挙しておくということなの。もちろん全ての実行可能なルートを列挙する必要はなくて、だいたいこんなルートが必要かなと思うものを選ぶだけでいいのよ。」

Rさんは頷きながら言った。

「なるほど、それなら分かります。要は、あまり遠回りをせず、かつお客様の時間指定を守って、かつトラックの重量や容量の制約を守ったような運行スケジュールをたくさん並べておけばいいんですね。我社の場合は簡単です。なんといっても、わがままなお客様が多いため制限がたくさんついてしまうので、実行可能なルートの数はそれほどたくさんはありませんから。」

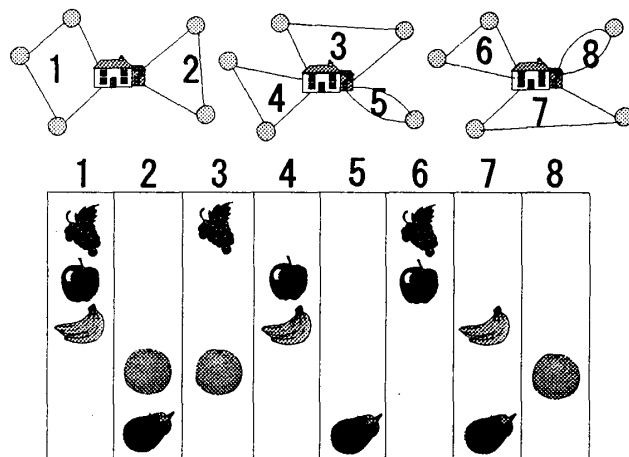


図 3: ルート生成法の参考図。ブドウ、リンゴ、オレンジ、バナナ、ナスをちょうど一回づつ含むように列(ルート)を選択する。

「そうね。ルート生成法というのは制限がつけばくほど有利になるのよ。で、実行可能なルートがたくさん生成されたとして、それぞれのルートのコストを計算する必要があるわね。本来ならルートに含まれるお客さんを与えたときに、どういう順番で廻ってくるかを決める問題は巡回セールスマン問題<sup>1</sup> といって難

<sup>1</sup>巡回セールスマン問題については、連載講座「巡回セールスマン問題への招待 I,II,III」オペレーションズリサーチ, Vol. 39, No.1,2,3 (1994) または筑波大の山本芳嗣教授と書いている「巡回セールスマン問題への招待」朝倉書店 (1996) を参照してください。

しいクラスに含まれる問題になってしまうんだけど、この場合は高々4件のお客さんを回るのだから、全ての回り方を調べても大した計算量じゃないわね。」

「我社の場合にはトラックは全て借り物の車なので、運行スケジュールが決まれば、料率表からすぐにコストは計算できます。」

「それじゃ話は簡単だわ、ルート候補から最も安いものを選択するところがこの解法のミソなんだけど、それはこういった問題を解けばいい訳ね。」

最小化: 選択されたルートのコストの和  
ただし: 全てのお客さんは何れかのトラック  
でちょうど一回訪問されなければならない。

「あれっ? でも教授、数式を使って定式化をしないんですか? 数理計画の論文誌を見るとそれはものすごい数式が載っていますよね。」

「ええ、でも本質は式を使わなくても同じよ、それに数式を使うと逆瀬川さん<sup>2</sup>に怒られるのよ。」

「なるほど、でも教授、それで、この問題はどのように解くのでしょうか? 何か問題を変形しただけのような気がします…」

「そうね、数理計画ソフトウェアを買い込んできて、それに式を入れて解かせるというのが常套手段だわね。そういったソフトウェアも最近はいくら安くなってきたみたいだし、パソコンでも十分高速に動くと思うわ。」

「いや教授、ご存じのように昨今の不況で物流部門の予算は大幅に削られてしまっていて、新しいソフトウェアを買う予算などとうていありません。」

「それならフリーのソフトウェアという手もあるけど…」

N教授が言いかけると、Rさんは興奮して遮るようにしてまくしたてた。

「フリーというのは無料、つまりタダってことですか? 私の大好きな言葉です。無料! タダ!! なんていい響きなんでしょう。で、どこにいけばもらえるんですか 行列でもなんでも並びますよ。」

「いえいえ、バーゲンの目玉商品じゃないんだから何も行列に並ぶ必要はありませんよ。たとえば、線形計画のツールなら双対シンプレックス法を使った

minit<sup>3</sup> や内点法を使ったLIPSOL<sup>4</sup> なんていうのがあるわ。でも、やっぱりVanderbeiのLOQO<sup>5</sup> が二次計画も解けるし、東京工業大学の小島さん達が発明した主双対内点法を使っているのととても高速だし、お薦めの一品だわね。あら、でもこれは企業で使うときは有料だわ。それから、読者の皆様のために付け加えておくと、教育用のツールとしてはAnima-LP<sup>6</sup> なんていうのもあるわね。これはJava言語で書かれているので、わざわざダウンロードしなくてもネットワーク経由で機種に依存せず動かすことができるの。これからの主流ね。それから、ツウの間で注目を浴びている半正定値計画 (semi-definite programming) のプログラムSDPAも東京工業大学のftpサイト<sup>7</sup>でもらえるわ。」

「でも、教授。そういったプログラムをダウンロードして我社の複雑な問題を解くためには、ある程度のメモリを装備したワークステーションを用意しないとイケないんですよ。ご存じのように昨今の不況で、我社は新しい計算機なんか購入できない状態なんです。せいぜい今あるパソコンを利用して何とかならないでしょうか?」

「それならNEOS<sup>8</sup>がいいわね。ここではネットワーク上で問題を解いてくれるというサービスをタダで行っているのよ。電子メールやWWWブラウザで問題を送ってあげると自動的にあちらの計算機で解いてくれて、答えだけを返してくれるわ。線形計画の他にも確率計画、ネットワーク関連の問題や非線形計画のサービスもしてるみたいね。」

「うーん、何度聞いても“タダ”というのはいい響きですね。あれっ? でも教授、トラックの台数を決めるのに線形計画法ではまずいんじゃないでしょうか? だって半分のトラックで運ぶなんて真似はできませんよ。」

「うーん、確かにそうね。でも半分の積載量を持つトラックを借りてくればいいんじゃないかしら? たとえば、10トン車が半分使うとかいう答えがでたら5トン車を調達するとか…」

「いやいや教授、日本の場合は費用のほとんどが人

<sup>2</sup>ダンディーなお髭がトレードマークの編集長。本特集は「分かり易くするためになり」かつ「数式・プログラミングはダメ」という難問が付いている。

<sup>3</sup><ftp://ftp.uu.net/usenet/comp.sources.misc/volume7>

<sup>4</sup><http://math.umbc.edu/~yzhang/lipsol/>

<sup>5</sup><ftp://elib.zib-berlin.de/pub/opt-net/software/loqo/1.08>

<sup>6</sup><http://gpu.srv.ualberta.ca/~amonga/informs/taids.html>

<sup>7</sup><ftp://ftp.is.titech.ac.jp/pub/OpRes/software>

<sup>8</sup><http://www.mcs.anl.gov/home/otc/Server/neos.html>

件費ですから、トラックだけを半分にしたんじゃない駄目ですよ。人間も半分に切らなければ...」

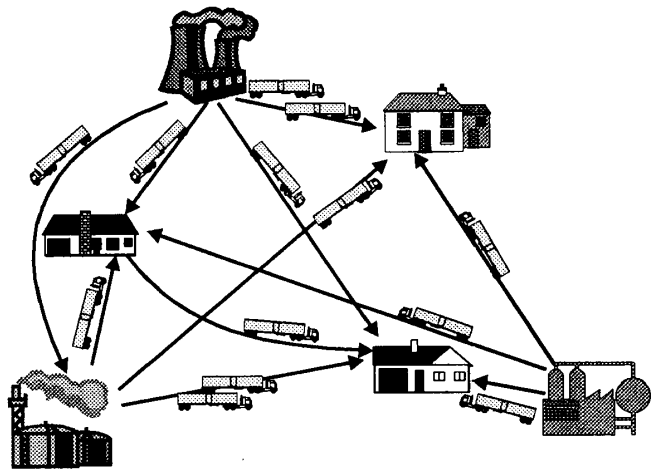
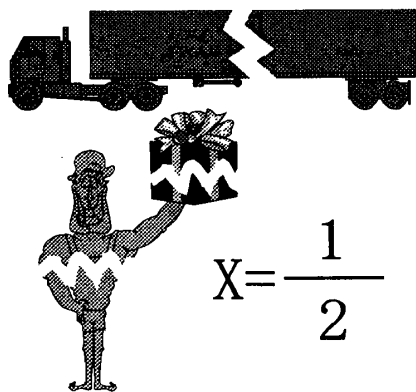


図 4: 線形計画法による解  $X = \frac{1}{2}$ . 運転手半分, トラック半分, 荷物も半分.

「うーん, そんなことをしたら人事部から文句を言われそうね. となると, やはり変数に整数条件をつけなければいけないわね. こうなると問題は急に解きにくくなってしまいうんだけど, やっぱりフリーのソフトウェアがあるわ. たとえば, 陰的列挙法もどきを使った opbdp<sup>9</sup> とか lp\_solve<sup>10</sup> なんていうのがポピュラーね. lp\_solve は名前とは裏腹に混合整数計画問題も解けるし, ソースコードで提供されているので結構便利なツールね.」

「分かりました. これで配送センターからお客さんへの運行スケジュールは解決しました. 早速それを使ってシステム開発にとりかかります. で, ついでといっちはなんですが, 今度は工場から配送センターの問題についてのお伺いしたいのですがよろしいでしょうか?」

「ええ, OK よ. それじゃ問題の概要からお聞かせ願えるかしら.」

Rさんは再び黒板に図を書きながら説明をはじめた.

「エーと, 問題はだいたいこんな感じです. 工場から配送センターへは大型トレーラーで輸送をしますが, 一日に何台分といった輸送指示が来ます. それから配送センターから工場や配送センター同士でも, 一日に何台分といった輸送指示が来るときがあります. さらに...」

図 5: ついでに聞いた問題の説明図

途中で教授が遮った.

「要は荷物を積まずに移動する空輸送の距離を最小にしたい訳ですね.」

「その通りです. もちろん運転手さんの稼働時間の制約なども付加されますが...」

「それじゃ, まずは基本的な部分から解きははじめましょう. 要は, 配送センターや工場に入ってくるトラックの台数と出ていくトラックの台数が不均衡だからいけないのね.」

「そうなんです, 教授. 台数の不均衡を最小費用で是正するような方法が見つかればいいんですが, 我社のベテランスケジューラーでも複雑すぎてよくわからんということです.」

「まず問題を整理しましょう. 入ってくるトラックの台数が出ていく台数より多い地点を左側にまとめて書いて, 逆に出ていくトラックの台数が入ってくる台数より多い地点を右側に書きましょう. すると問題は左側の地点から右側の地点を最小費用で結ぶ問題に帰着されるわね. こういった問題をご存じないかしら?」

「輸送問題ですね! 教授」

Rさんは興奮していった.

「そう! 先ほどRさんが使いものにならないとおっしゃった輸送問題よ. 現実の問題がそのままの形で教科書に載っていることは少ないけれど, こうやって問題をすこし変形させてやることによって標準的な問題に帰着されることが多いのよ.」

「いやー, 申しわけありませんでした. 先ほどの発

<sup>9</sup><http://www.mpi-sb.mpg.de/guide/staff/barth>

<sup>10</sup>[ftp://ftp.es.ele.tue.nl/pub/lp\\_solve](ftp://ftp.es.ele.tue.nl/pub/lp_solve)

言は取り消します。で、この輸送問題を解くためのタダのツールとかはあるんですか?」

「そうね。輸送問題だけに限定されたツールとしては、大型計算機用のFORTRANルーチンくらいしかないかもしれないけど、再びちょっと変形してフリーソフトがたくさんある問題に帰着させてあげればいいのよ。たとえば最小費用流問題というのをご存じないかしら?」

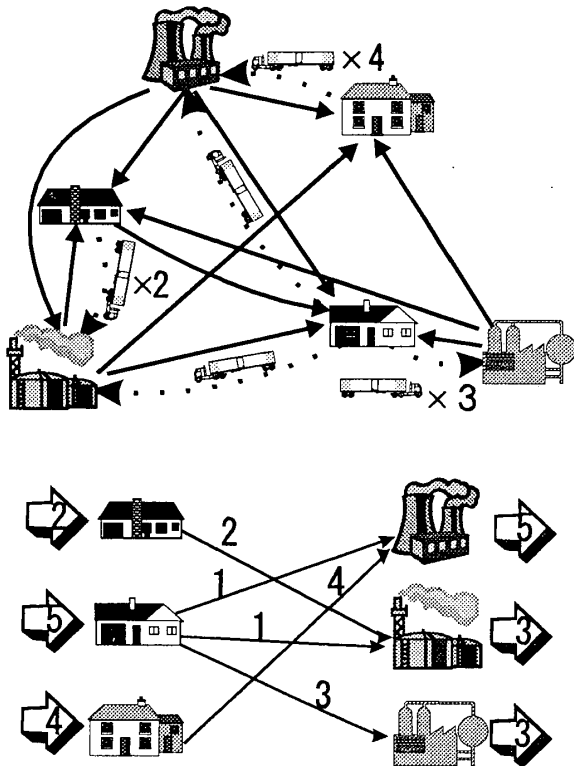


図 6: 輸送問題を解いて空輸送を最小化する!

「そういえば昔ネットワーク理論の授業で習ったような気がします。たしか、始点と呼ばれる地点から、終点と呼ばれる地点へ物を流すときに、枝の容量を超えず、かつ流すときに発生する費用の合計を最小にする問題ですよ。でも、私はパイプに物を流すようなときにしか使えないと思っていました。」

「そのへんは想像力の問題ね。今回の問題の場合には、荷物を積まずに移動しているトラックをネットワーク上を流れる“物”と思えばいいのよ。」

「なるほど。それでしたら最小費用流問題に帰着させる方法がわかりました。」

Rさんは立ち上がって黒板に図を書きながら続けた。

「こうやって仮定の始点と終点を作ってやって、空輸送の必要な台数分の“物”を流すものとします。それから、仮定の始点から左側の地点に向かう枝にその地点から出る必要のあるトラック台数分の容量を付けます。同じように、右側の地点から仮定の終点に向かう枝にもその地点に入る必要のあるトラック台数分の容量を付けます。最後に左側と右側の地点の間に移動に伴う費用を付ければ出来上がりです。」

「だんだん問題を標準的な問題に帰着させるコツが分かってきたみたいですね、Rさん」

「まあ、このくらいならなんとか。それで、最小費用流のコードはどこへ行けばもらえるんでしょうか? もちろんタダのものですが。」

「そうね。たくさん出ているみたいだけど、やはり有名な研究者が作ったものが安定していて使いやすいみたいね。たとえば、MITの Bertsekas が作った RELAX は彼のホームページ<sup>11</sup> に行けばもらえるし、Stanford大学の Goldberg が作ったコード<sup>12</sup> も定評があるわ。それから、もっと複雑な非線形費用の多品種流問題なんかを解くツール<sup>13</sup> も出ているけど、この場合には必要なさそうね。」

「なるほど、空輸送をどの地点間で行えば費用が最小になるかはわかりました。でも教授、運行スケジュールを組むためには、前の問題と同じように複雑な条件をつけなければいけないんですよ。」

「そうね。前と同じように制限がたくさんついているのならルート生成法で大丈夫ね。」

「残念ですけど今度の場合は制約がそれほどきつくないんです。それに配送センターの数や工場の数も大分多いし...」

「そうなるちょっと無理ね。汎用の数理計画ソフトも使い方によっては便利なんだけど、なんでもそれに入れば解けるってもんじゃないし...。そもそも、こういった難しいタイプの組合せ最適化問題というのは、問題に依存した工夫をいかにうまく使うかが解決の鍵になるのよ。たとえば、時間枠が極端にきつい場合には動的計画法とか、基本構造にちょっとイヤな条

<sup>11</sup> <http://web.mit.edu/afs/athena.mit.edu/user/d/i/dimitrib/www/home.html>

<sup>12</sup> <ftp://theory.stanford.edu/pub/goldberg/opt-code-info>

<sup>13</sup> PPRN(<ftp://ftp-eio.upc.es/pub/onl/codes/pprn>) または LSNNO(<ftp://ftp.bilkent.edu.tr/pub/IEOR/Opt/Network>)

件がついているときにはラグランジュ緩和とかいう案配にね。でも、こういった問題の場合は、現場で使うとすると次から次へと付加条件が増えていくので、後で苦労しないためにも専用のメタ解法を開発した方がいいかもしれないわね。」

「メタ解法? 何ですかそれは。普通の解法とはどう違うのでしょうか?」

「まあ、基本的には何の違いもないんだけど、最近の流行物の近似解法を総称してメタ解法とかメタヒューリスティックとか呼ぶのよ。例えば、Genetic AlgorithmとかSimulated Annealing法とかが代表例ね。でも私のお勧めはアダプティブ・メモリー・プログラミングかしら。」

「はっ? 何か難しそうな名前ですね。」

Rさんは内心、また難しそうな名前でごまかしに來たぞ! と思いながら尋ねた。

「で、その“アタクシハ・メアリー・プロモデル”というのはいったい何なんですか?」

「ちょっと違うわね。英語で言うと“adaptive memory programming”で、日本語に直訳すれば適応型記憶計画というところかしら。まあ、簡単にいうと探索の過去の履歴を積極的に使って探索をすすめるローカルサーチの変形のことよ。探索を知的なものにするための工夫をたくさん取り入れたメタ解法のフレームワークとでもいうのかしら。」

「要は“何でもあり”の解法ということですね。」

「まあ、そうともいうわね。昔はタブーサーチとも呼ばれていたの。こっちの方がポピュラーかしら。でも、Glover<sup>14</sup>がこっちの名前に変えようと言っているし...」

「タブーサーチ... すなわち禁断の探索ですか。ますます怪しい名前ですね。」

「名前は怪しいけど、実際問題を解くときにはとても便利な方法よ。運搬経路問題みたいな難しい問題にはタブーサーチが極めて有効なの。運搬経路問題に対する標準的ベンチマーク問題<sup>15</sup>に対する系統的なテストから判断すると、近似解法のベスト5は全てタブーサーチをベースに作られたものになるって言えば、その凄さが分かってもらえるかしら。それにローカル

サーチと同じで付加条件には極めて柔軟に対応ができるから、複雑な実際問題を解くときにはうってつけよ。ちなみに、タブーサーチや関連するメタ解法の詳細については“離散構造とアルゴリズムIV (近代科学社)”を参照して下さると有り難いわ。」

「また宣伝ですか? 度が過ぎると読者の皆様に怒られますよ。」

「ごめんなさい。でも、この本をあちこちで宣伝しないと室田さん<sup>16</sup>に怒られるのよ。」

「閑話休題。でも教授。専用のプログラムを開発するという事は、膨大な費用がかかるんじゃないですか? 何度も申し上げましたが、我社は今、非常に切迫した状態で...」

「でも、多少の出費をしてもそれに見合うだけの費用削減があればいいんじゃないかしら? 私の長年のコンサルティング経験から、この手のものをきちんとシステム化すれば、少なくとも10%くらいの費用は削減できると思うわ。それに、Rさんの会社の物流費用といたら膨大な金額だと思うんだけど...」

「なるほど、損して得とれですね。分かりました。なんとか社長を説得してみましょ。最終的に費用削減ができればタダより安くなる訳ですからね。うーん、タダより安いか! いい台詞だ。これから私のモットーは“タダより安い数理計画”にします!」

教授はRさんにウインクをしながら囁いた。

「私への報酬も忘れずにね!」

## 謝辞

いろいろなソフトを検索する際に東京大学の松井知己さんのホームページ<sup>17</sup>を参考にさせていただきました。ここには最適化関連の他の便利なサイトへのリンクが張られているので、国内からはじめて最適化関連の捜し物をされる方は、この辺から探し始めるのがいいと思います。それから、松井さんはこの原稿の初稿にもたくさん修正を入れてくれました。あわせて感謝いたします。最後に、綺麗な図を描いて下さった東京商船大学流通システム研究室の内田麻貴さんにも感謝します。

<sup>14</sup>Colorado大学のFred Glover, タブーサーチ教の宣伝係兼総帥。

<sup>15</sup><http://mscmga.ms.ic.ac.uk/info.html>

<sup>16</sup>室田一雄京都大学教授, 離散構造とアルゴリズムIVの編者兼宣伝部長。

<sup>17</sup><http://misojiro.t.u-tokyo.ac.jp/~tomomi>