

数理計画法の実用モデリングについて

草刈 君子

1. はじめに

ずっと生きていくような気がする。それでいて、いつも知りたいことが溢れている。そんな気持ちのまま数理計画法とつきあっている。学生時代を除けば、研究機関に身をおいたことがないこともあり、いつも実用を追ってきた。この過程で何を知りたかったのか、今、何を求めているのかを、次の4点を軸に記述する。

- ・数理計画法のアルゴリズム
- ・制約プログラミングとの融合
- ・自動モデリング
- ・データ、シナリオとの関連

世の中の他の職業を知らないまま、数理計画法とそれを使うシステムに絡んだ職業生活を送ってきた。こんなに面白い仕事を自分がしたいと生きている人間の狭い世界の意見かと思いつつ書かせていただいた。文章の背景を整理するため略歴を書くと、現在の勤務先に至るまでの前職は次の通りである。

- ・富士通(株)の数理計画法エンジンの開発部門に在籍、混合整数計画法を担当、SE部門の要請があればコンサルティングも担当。
- ・アイログ (ILOG) (株)にて最適系のシニアコンサルタントとして勤務。

2. 数理計画法のアルゴリズム

アイログ時代、海外のメンバに数理計画法のエンジンの開発をしていたと自己紹介すると、「なぜ? CPLEXがあるのに?」と言われた。今となっては信じていただけないとも思うが、1990年代初めまで実用的に数理計画法のエンジンを実行しようとするならばメインフレームを使わざるを得ない状況が続いていた。例えば、CPLEX社の設立が1988年であり(混合整数計画法の提供はアイログ社のwebには1991年とあ

る)、Dash社の設立は1989年であることから推測いただけると思う。

1990年代初めまで、国産のメインフレームメーカーに勤務していることは、アルゴリズムを自分たちで考え、改良して、計算機を十分利用した数値実験ができる特権を意味していた。数理計画法の計算エンジンそのものを「より速く、より大きな問題を、より安定的に解く」ように作ることに最大の視点を置いていた。それが、数理計画法を利用する顧客の最大の要望であると信じていた。アルゴリズムの改良は、①線形計画単独、②線形計画と混合整数計画の融合、③混合整数計画単独の3通りで進んだ。①、③については論文も多いが、②線形計画と混合整数計画の融合はあまりなかったように思う。そうした中で、特定条件の場合の逆行列生成の抑制、基底情報の入れ替えなど、実データを調べ、改善しては効果を測定していた[2]。

当時は顧客側にモデルを作成する部門があり、お手伝いすることと言えば、顧客から語られるモデルの目的と、作成された数式を見て、数理計画のエンジンの内部ロジックとうまくマッチするようにエンジンのパラメータを紹介したり、場合によっては入力形式を変更して貰ったりという内容である。当時、モデルと密接に関係するアルゴリズムで一番感激したものは、SOS (special ordered sets) である[1]。混合整数計画問題を最適化する方法に分枝限定法がある。通常は、整数変数を対象とするこのアルゴリズムを(順序の概念を入れることで)、自然に有限の要素集合に拡張している。SOSは論理的な組み合わせの数を激減するが、かわりに計算コストが高くつく場合もある。面白いのは(アルゴリズム上は当然のことなのだが)、順序を定義する係数データに依存して動きが変わることである。適当に入れたデータではうまく動かないのに、実データではきれいに動くといったことを通して、数理モデルにおける、モデル構造とデータの関連性に傾いていた。

内点法が実用化され線形計画法が安定的に最適化を

くさかり きみこ

SAS Institute Japan (株)

〒104-0054 中央区勝どき 1-13-1

行うのに反して、混合整数計画法は下手なモデリングや、乱数生成の無機質なデータを使用すると致命的に性能が悪化する。全体を通しての最適性と、要求される細かさを数理モデルとしてどう実現するのか、SOSとは別の形の選択肢に列生成法があり、制約プログラミングとの融合があった。

3. 制約プログラミングとの融合

数理計画法と同じく宣言的な最適化手法に制約プログラミングがある。アイログ社がCPLEX社を買収し、Concert、OPL上で両手法の融合を行ったように、現在ではDash社、SAS社の製品でこれらの両手法を利用できる。

数理計画法は全体を、目的関数を活用して最適化する。制約プログラミングは原理的には（確かに小さな問題サイズであれば）全体をまとめて解くが、目的関数の扱いが数理計画法と異なる[9]。少し曖昧な書き方を許して貰えば、数理計画法は全体を見渡す力があるが、細かい条件が離散である場合には性能的に辛くなる。それに、整数変数の定式化は慣れるまでは面倒だとも言われる（本当は、ほんの少し慣れてしまえば楽しい世界が続くのだが）。反面、制約プログラミングは、モデル表現は比較的容易である。ただどうしても、大型の最適化には向かない。結局、数理計画法で全体を見た配分を行い、探索領域を適度に縮めておいてから、制約プログラミングで細かい配置を立案する形が一番だと考えている[3]。

4. 自動モデリング

モデル構造をエンジン側が把握できれば、より高速に適切にエンジンが動くことができるかもしれないという思いをもって1990年代に入った。また、ワークステーションの台頭により、数理計画法のエンジンはすでにメインフレームから離れ始めワークステーションに軸を移していた。そうなれば、メインフレーム時代のように毎月レンタル料金が入ることを前提とした顧客コンサルティング（無償、無制限）は続けられない。

今まで使わなかった層に数理計画法を広めること、より手離れの良い提供が課題となった。慣れてしまえば何のこともない数式モデルの構築がもしかしら分かりにくいのではないかと、構築に時間がかかることが原因か、の判断のもとに、次の視点は、「より簡単に、より分かりやすく表現すること」になった。オブジェ

クト指向、ER分析が定着していたこともあり、装置産業向けと物流ネットワークの2通りに展開した。GUIも付けた[5]。この背景には、数理計画法の中でも線形計画法については、ある程度冗長なモデリングをしても、必要程度の高速に最適化が行える段階にあるという背景がある（数理計画法の前処理でマトリクス解析がこの手の冗長性を除去するという自信もある）。

4.1 物流ネットワーク問題

生産から販売までにわたって考慮する問題を考える（図1）。この問題では複数の生産ラインを持つ複数の工場と、生産を委託している委託工場が複数存在し、それらの商品は、複数の倉庫で保管し、マーケットに出荷する。

きちんと教科書的にモデル表現するならば、解を得る必要がある変数を最低限度定義し、それらの関係を数式で表現していく。一方、プログラムで自動的に処理をする場合は多少の冗長性（数理計画法の前処理で除去される）を覚悟し汎用的に表現する。例えば、このモデルであれば、モデル構成要素として、ノードを表現するための要素とアークを表現するための要素の2通りに区分し、それぞれ、ノードについては図2に示す形の4変数（搬入を X_{in} 、搬出を X_{out} 、期首在庫を X_{start} 、期末在庫を X_{end} として表現する）を用意し、関連を定義すること、アークについては図3に示す形で図2に示す X_{out} または X_{in} と結びつける変数を各アーク上に用意することで、自動的にモデルを生成する。

つまり、物流ネットワークを構成するノードを次の3通りに区分する。

- ・Supplyノード（その拠点からの「出」のみ注目）
- ・Demandノード（その拠点への「入」のみ注目）

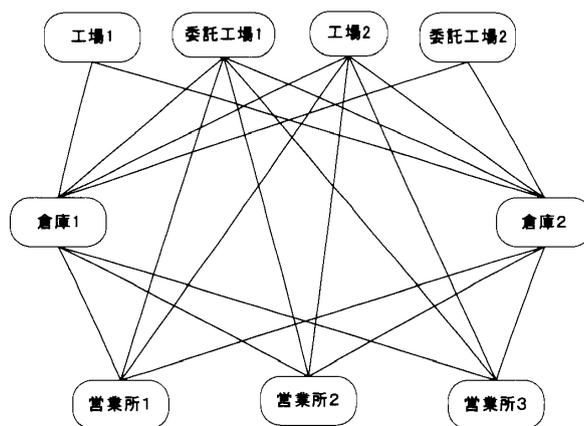
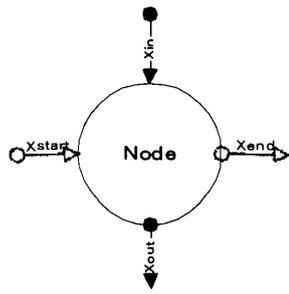


図1

ノード



例えば、一期間かつノードでの形態変化がない場合で考えると、

$$Xin + Xstart = Xout + Xend$$

ここで、

Xin = 入量

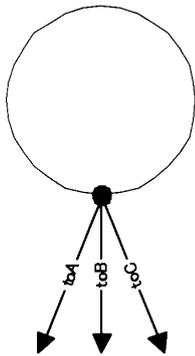
$Xstart$ = 期首在庫

$Xout$ = 出量

$Xend$ = 期末在庫

図 2

アーク



$$Xout = XtoA + XtoB + XtoC$$

ここで、

$Xout$ = 出量

$XtoA$ = ノードAへの輸送量

$XtoB$ = ノードBへの輸送量

$XtoC$ = ノードCへの輸送量

図 3

- ・ Make ノード (その拠点への「入」「出」とも注目)

Supply ノードは、図 2 の $Xin=0$ で表現され、何も入って来ない状態になる。Demand ノードは、図 2 の $Xout=0$ で表現され、何も出て行かないことになる。Make ノードは、入ってくる量も出て行く量も考えるので Xin も $Xout$ も値を取る。各ノードで在庫を考える必要があれば $Xstart$ と $Xend$ も必要になり、在庫が必要なければどちらも 0 となる。

ノードにおける原料や BOM など生産条件を表現したい場合は、図 2 の関係式を拡張する。複数の輸送手段があればアークを複数にするといった拡張が必要になる。

4.2 プロセスフロー問題

石油精製に代表されるプロセスフロー問題の場合、物流ネットワークより少しだけノードとアークの表現は複雑になる。物流ネットワークではアーク上に入った物はそのままアークから出た。プロセスフローでは、アーク上で分離や合流が行われるため、マテリアルバランス式が必要になる。このために物流ネットワークで必要なノードとアークの他に、コネクタが表面化す

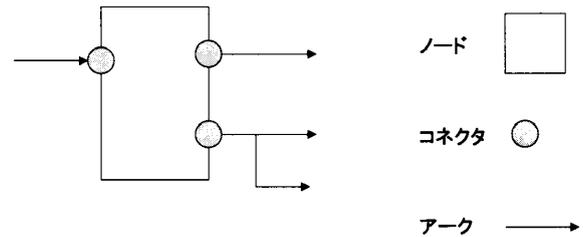


図 4

る (図 4)。

ここで表面化すると書いたのは、物流ネットワークではノードタイプによりコネクタ数固定であり、マテリアルバランス式が単純だと考えれば表現上の両者の区分はいらぬことになるからである。もちろんノード内部の関係式はそれなりに異なる。

5. データ、シナリオとの関連

今一番の視点は、データにある。「あなたのわがまま運びます」という古い言葉にひっかけて愚痴ったことがある。「エンジンを速くして、と言われて速くした。モデリングが難しい、と言われて自動モデリングも GUI も作った。今度は、入れるべきデータがないから使えない、と言われた」と。エンジンの性能競争が続いていることを理解しているし、それがとても大きな意味を持つことも知っている。自動モデリングのオブジェクト指向のクラス表現は、モデルの視覚化のクラス表現とシームレスに結びつけられる。モデルを維持する顧客層にとって、問題の理解しやすさ、結果の理解しやすさ、用意された GUI の範囲でのモデル改造の容易さの面で大きな利点となることも実感している。それでも、どんなに速いエンジンに、どんなに素敵な GUI をつけても「GIGO (garbage in, garbage out)」の言葉からは逃げられない。

顧客が問題を熟知しているかという話がある。もちろん現実の中で顧客が問題を一番よく知っている。それでいて、商談の初めに要件定義を (数理モデルとして矛盾のない形で) 記述してくれるわけではない。顧客が問題をととても難しいと感じている場合に次の状況が起こる。①制約条件が小出しされる場合「こんなに条件を言ったら、とても無理だから言わないでおこう」と、②思いっきり制約条件が入る場合「色々なケースがあるから全部包含して入れておこう」である。前者の場合、構築したモデルがうまく行き相互に信頼ができたころ、顧客が「本当はこれもある」と追加した、もしかするとその中のたった一つの、顧客は難し

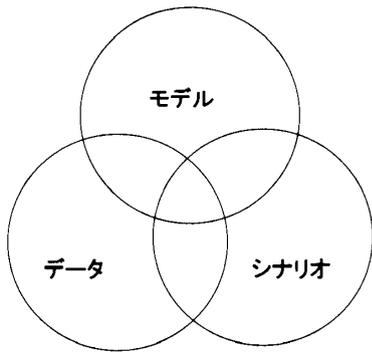


図5

いとも思わなかった制約のために、がらりと異なるモデルへの移行が必要になる場合がある（そうならなければ幸いである）。後者の場合、必要以上の難しさに下手をすれば全体が失われていく。それでも、要件が矛盾している場合はヒアリングで確認できるが、要件とデータが矛盾してくると中々面倒になる。

結局、数理計画法の実用モデリングに必要な項目は、モデル、データ、そしてどう利用したいかのシナリオだと今は考えている。数理計画法のエンジンも、制約プログラミングとの融合モデルも、自動モデリングも図5の「モデル」を支える技術だと思う。

現実の問題に向き合うとき、モデルとシナリオは顧客からヒアリングして作成し、データは顧客から提示されるものを使う立場もあると考える。それでも、自分が一番気に入っている方法は、これらを顧客と共に考える立場である。すなわち、顧客の要件を聞き、データも見て、どう使うかを確認して、モデリングをすることである（図6）。

さて、データとシナリオをどう扱うか、世の中にある問題に対するアプローチの仕方には色々ある。個人によっても異なるし、手法によっても異なる。例えば、データ解析手法の方がデータへのウエイトは大きいようだ[6]。

ビジネスインテリジェンスの世界は、データからインテリジェンスへの流れ（図7）と、データ解析と最適化のシームレスな連携と密接に関連して進展してきた。データ解析側からも最適化とシナリオに対して方向性がついてきた。

数理計画法が対象とするような長期計画を立案するための入力データをどのように生成するべきか、数理計画法のモデルに入力するデータの精度をどう制御し得るのかに強い興味がある。前述の物流ネットワークであれば、デマンドサイドの需要予測、安全在庫は実績データから生成できる。サプライサイドの評価には

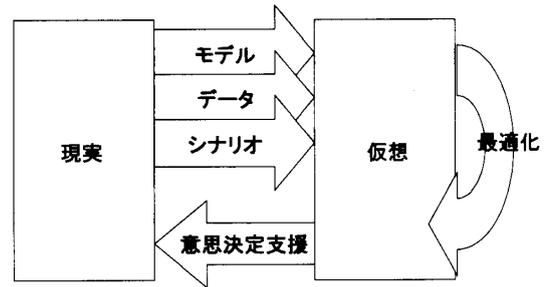


図6

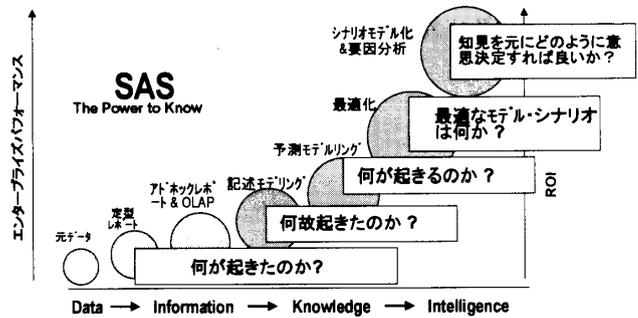


図7

すでにDEAが活用され[7]、従来の解析方法に加えて利用されている。

結果を見て解析する、シナリオを立てるといった場合、様々な視点からの分析が必要となり、初めに形式を決めてしまうことは難しい。データやシナリオとうまく連携するようにモデルを作るにはどうするのか、新しい課題がある。物流ネットワークの問題でこれにトライしたとき、入力データから欠損値を許した汎用形式（冗長性はあるがノードとアークを区別しないで扱う方式）に変換すること、入力と解を同一形式にマージすることを試みた[4]。こうして、最適化とデータ解析がシームレスに関連し、モデル、データ、シナリオを活性化する。より効果的な実用化への段階だと感じている。

6. おわりに

「学際 (interdisciplinary)」という言葉がある[8]。数理計画法のアルゴリズムから始めた世界が、どこまで広がれば実用的なモデリングになるのか、尽きない世界だと今も思う。

参考文献

- [1] Beale, E. M. L. and Forrest, J. J. H.: "Global optimization using special ordered sets", Mathematical Programming 10, pp. 52-69, 1976.

- [2] 草刈君子, 宮崎知明, 金指哲也: “富士通 AMPS からみた最近の混合整数計画法—その性能と適用例—”, 第2回 RAMP シンポジウム論文集, pp. 11-20, 1990.
- [3] 草刈君子, 陳界熹, KATAI Ferenc, GOSSELIN Vincent: “最適化の基盤としての制約伝播技術の利用について”, 第12回 RAMP シンポジウム論文集, pp. 53-60, 2000.
- [4] 草刈君子, 能島俊夫: “サプライチェーンネットワークの設計”, OR 学会春季研究発表会, 2005.
- [5] 宮崎知明, 中根智美, 草刈君子: “WS 版 AMPS による物流最適化及び装置産業用生産計画—数理計画法の自動モデリングアプローチ”, 統計数理研究所共同研究レポート 45 最適化: モデルとアルゴリズム, pp. 136-142, 1993.3.
- [6] 新村秀一: “SAS 言語入門”, 1994.
- [7] SAS Institute Inc.: “Optimization with the SAS System, A SAS White Paper”, 2003.
- [8] 刀根薫: “オペレーションズ・リサーチ読本”, 日本評論社, 2003.
- [9] 吉川昌澄: “制約最適化技術のスケジューリング問題への応用”, 人工知能学会誌, No. 3, pp. 379-386, 1998.