「問題解決エンジン」群とモデリング

茨木　俊秀

1. はじめに

最適化モデリングというと，まずLP（線形計画）やIP（整数計画）による定式化が思い浮かぶ．LPは実用化されてすでに久しいし，IPも最近では優れた商用パッケージが出回るようになり，広く使われるようになっている．人工知能の分野ではCP（制約プログラミング）の枠組みが提唱され，実用化が進んでいる．

組合せ最適化に興味を持つ研究者として，これらの恩恵に大いに浴しているのであるが，しかし同時に，これらだけでは十分ではないという印象も払拭できないでいる．この事情をもう少し詳しく説明すると，NP完全性の理論によれば，ほとんどすべての組合せ問題は（つまりクラスNPの問題は）一つのNP困難問題（例えばIP）に定式化できることが分かっているが，定式化に際して多数の変数や制約条件を導入しなければならないとか，定式化自体は簡潔であっても解くのが容易でない，という理由で実用的には使えないことが結構あるからである．同様な感想は，応用の現場でORを利用しておられる実務家の方々からもしばしば発せられている．

広い範囲の組合せ問題を最適化手法によって解決するにはどうすればよいだろうか．少々面倒でも，次のようなアプローチ以外にないというのが，いろいろな試みを経て得た私の結論である[1]．すなわち，いくつかの標準問題を設定し，それらに対して「問題解決エンジン」を開発しておき，解くべき問題に適した標準問題のエンジンを利用するという図1のスキームである．この目的に，ここ10年近く京都大学の柳浦睦憲さん，野々部宏司さんらとともに，学生たちの協力を得て，問題解決エンジンを開発してきた．最近，ようやく実用的に使えるレベルに近づいてきたと思われるので，その全体像を紹介し，利用に当たっての問題点，特にモデリングの部分について述べてみたい．

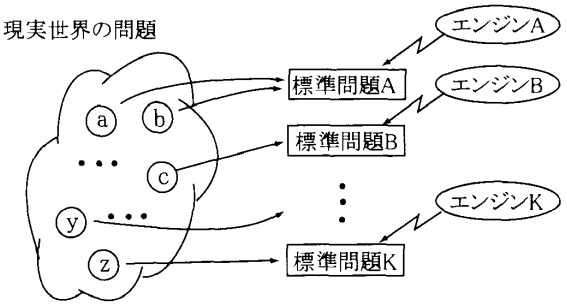

図1　標準問題による問題解決

2. 標準問題のリストとモデリング

まず，これまでに採用した標準問題のリストを与える．

1. 制約充足問題（CSP）
2. 資源制約プロジェクトスケジューリング問題（RCPSP）
3. 配送計画問題（VRP）
4. 2次元箱詰め問題（2PP）
5. 一般化割当問題（GAP）
6. 集合被覆問題（SCP）
7. 最大充足可能性問題（MAXSAT）

以上に加え，代表的な標準問題であるLPとIPについては，商用パッケージの利用を前提にしている．それぞれの標準問題は，多様な制約と目的関数を許容できるよう，柔軟な構造に設定されている．また，解を求めるアルゴリズムはすべてメタヒューリスティクスのアイデアに基づき，性能向上のために問題構造を利用したさまざまな工夫を加えて実現されている（これらの詳細は，科学研究費の報告書[2]，あるいはその中に引用されている文献をご覧いただきたい）．

上記の標準問題はいずれも理論的にはNP困難であって，厳密解を求めるのはきわめて困難と考えられている．そのため，問題解決エンジンはすべて近似解を求めることを目的に作られている．現実の場では，

いばらき　としひで
関西学院大学 理工学部
〒669-1337 三田市学園2-1

近似解で十分であることが多く，また，得られる近似解の精度はきわめて高いので，開発された問題解決エンジンを用いると，上記の標準問題はすべて現実的な意味で解けると言ってよい．この認識は重要である．

さて，問題解決エンジンが手に入ったとして，よくある質問（FAQ）は次のようなものであろう．いま会社で解決しなければならないある課題を抱えていて，組合せ最適化問題の一種ではないかと思っているが，それをどのように定式化してよいか分からない．定式化しなければ折角の問題解決エンジンを使うことができない．どうすればよいか．

つまり，モデリングをどうするかという質問であって，モデリング支援システムのようなものがないか，という質問もよく受ける．ある程度問題の範囲を限定して，例えば，スケジューリングで，さらに，例えば多段工程に限定して考える，というのであれば，その目的に役立つモデリングツールを作ることは可能であろうし，現にそのようなシステムも作られている．しかし，ばくぜんと組合せ最適化の範疇にあるといった問題を理解して，それに適した標準問題を選択し，定式化するというプロセスは，メタ・モデリングとでもいうべき高度な思考を必要とし，問題領域の専門家とアルゴリズムの専門家の両者が知識と経験を生かして協力しなければ成功しないだろう．

逆に，だからモデリングは面白いともいえる．ある先生が次のようにおっしゃっていたことを思い出す．「もし，世の中すべてが線形システムででき上がっているのであれば，それらを数学的に正確に記述し解明することができるが，そのかわり単純で予見できる現象しか生じないので，面白くも何ともない．幸いなことにこの世の中は大変非線形にできていて，いつまでたっても未知な現象が残っている，だから退屈しないのである．」

組合せ数学の対象は，非線形の極限とでもいうべきものである．NP完全とかNP困難という概念が，これらの問題は一筋縄ではいかないということを述べていると理解すれば，だから面白いのである．これら面白い標準問題を相手にモデリングを考えるのは，当然もっと面白いにちがいない．

しかし，私にはモデリングの面白さについて総合的に述べる力はないので，次では，実際に標準問題へのモデリングを行った経験のなかから，定式化に少々工夫を要したという話題を二つ，簡単に紹介して，お茶を濁させていただく．

もう一度強調しておくと，ここではモデル化すべき標準問題があらかじめ決まっているわけではない．つまり，どの標準問題を選ぶかという問題と，その標準問題へどのように記述するかという二つの問題を解決しなければならない．この二つは独立ではなく，相互にフィードバックを重ねつつ次第に固めていくという面倒なプロセスである．

3. 標準問題へのモデリング

3.1 ルート決定問題—VRPとSCP

飛行機（列車，バスなど他の交通でもよいが）の操縦士の勤務スケジュールを考える．一人の操縦士に対し，A空港を出発する便をB空港まで操縦した後，B空港からC空港までの便を操縦し，さらに…という具合に1勤務のスケジュールが決まる．このとき，連続する二つの便は，同じ空港に着発するものでなければならないとか，着発の時間の先行関係を満たすものでなければならないという条件，さらに，安全上や勤務条件の考慮から，連続する便の時間間隔，1勤務の総時間の制約などさまざまな条件が入る．さて，この航空会社が保有しているk人の操縦士ですべての便を飛ばすためのスケジュールを組みたい．

それぞれの便を点で表すと，一人の操縦士の勤務は，これらの点をつなぐ1本のルートと考えることができる（図2）．簡単のため，すべての操縦士はデポと呼ばれる特別な点から出発し，同じデポに戻るとする．さて，どの便にも少なくとも一人の操縦士が必要だから，この問題は，すべての点をカバーするように，k本のルートを構成する問題である．ただし，ルートは自由に作れるわけではなく，ある点（便）から次に訪問する点（便）は，先に述べたさまざまな制約を満たすものでなければならない．

この問題を解くために，二つのアプローチが考えられる．その一つは，一人の操縦士の勤務ルートとして可能なものを，勤務に関する条件を考慮して，あらか

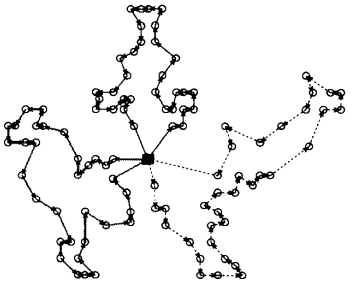

図2　全点をカバーするルート集合

じめすべて列挙しておき，それらの中からk個を選ぶという方法である．もちろん，選ばれたルートを合せるとすべての便に操縦士が搭乗していなければならない．これは，各ルートをそれが訪問する点の集合と捉えると，選ばれたルート集合の和集合が全集合になるという条件である．すなわち，標準問題の中でSCP（集合被覆問題）に定式化できて，SCPエンジンを用いて解くことになる．

もう一つのアプローチは，k本のルートを，勤務条件を考慮しつつ直接構成するものである．これは，平面上に置かれた顧客を点で表し，k台のトラックですべての顧客へ荷物を配送するという配送計画問題(VRP)と同じタイプであって，やはり，VRPエンジンを適用することができる．ただ，各ルートの構成は，先に述べたように勤務条件からくる制約を考慮しなければならないため，ルート探索に工夫が必要である．

この二つの標準問題のどちらを使うかは微妙なところであり，その判断によって大きな違いが生じる可能性がある．一般的に述べれば，ルート構成の際の制約が厳しく，実行可能なルートの個数がそれほど多くなければ，SCPに定式化したときの規模が大きくないので，SCPの利用が有利であり，逆の場合は，VRPが有利になる．これは，問題のサイズからの考察とともに，VRPのアルゴリズムが，ルートを少しずつ修正して改良していく局所探索のアイデアに基づいているため，ルートの制約条件が厳しければ，近傍解の中に改良解を見つけることが困難になって，探索性能が落ちるという理由もある．

交通の勤務スケジュールについて，過去の例を見ると，SCPによるアプローチが多いようである．この場合，SCPエンジンの適用の前に，実行可能ルートの生成を行うルーチンを用意しなければならない．ルート数があまりに多くなると，その中で重要なものを選択するための補助ルールの導入も必要である．簡便法として，ランダムに適当数のルートを選ぶという方法も用いられるが，近似精度が落ちることは避けられない．この部分をどのように作るかでSCPアプローチの成否が決まってくる．VRPのアプローチでは，局所探索にルートの制約条件をどのように組み込むかがポイントになる．通常は，制約の違反度をペナルティの形で組み込みつつ探索を進めることになるが，探索性能が落ちないように，具体的なペナルティの形を慎重に決定しなければならない．

結局，どちらの標準問題を採用するかの判断には，一方では解くべき問題の詳しい知識がまず必要であり，他方，定式化された標準問題のエンジンの動作を理解しておかなければならない．そのためには，現場で解決すべき問題を正確に把握している実務家と，アルゴリズムの動作をよく理解しているエンジン側の専門家が協力して，知識と知恵を出し合う，という共同作業が不可欠である．

3.2 巨大構造物の組立—RCPSPの意外な利用法

RCPSPはn個のジョブを，それぞれのジョブが要求する資源量（機械，作業員，材料，電力，経費など）の制約を満たすように，時間軸上に配置する問題である．多くのスケジューリング問題は，この形式に記述できるので，きわめて利用価値が高い．ここでは，標準的なRCPSPの利用例ではなく，定式化を工夫するとより広い応用があるという観点から，次の例を紹介する．

細長い工場内にいくつかの構造物（ブロック）を一列に配置して，それらを完成させるという作業を行っている．各ブロックの長さ（工場は細長いので，1次元の問題と考えられる），完成までの作業日数と必要資源量はブロックによって異なるが，あらかじめデータとして与えられている．ブロックは大きく重いので，一度ある場所に置くと同じ位置に留まるが，完成すると工場から出て行く．さて，すべてのブロックを最少日数で完成するには，どのブロックをいつどの位置に置いて作業すればよいだろうか，そのスケジュールを作れ．

この問題は，ブロックをジョブと見なせば，標準問題RCPSPにほぼそのまま定式化できる．しかし，RCPSPにはブロックの位置を決めるという機能は含まれていない．そこで，他の資源に加え，長さも資源の一つと考えて，工場の全長Lを消費するという考えで，RCPSPにかける．ただし，ブロックの配置にはある程度のすき間が避けられないことを考慮して，$0.9L$程度に割り引いた長さを実際の全長と考える．この計算で得られたスケジュールでは，長さ資源の和は$0.9L$以内に入っているが，ブロックを置く位置はまだ決まっていない．そこで，もう一度RCPSPを使うが，今度は1次元工場における配置位置を示す座標を，スケジュールの時間軸と考えて，この時間軸に沿って再スケジュールするのである．この時，各ジョブの元の時間軸での位置が動いてはならないが，各位置とジョブに固有の資源を導入することによって，この

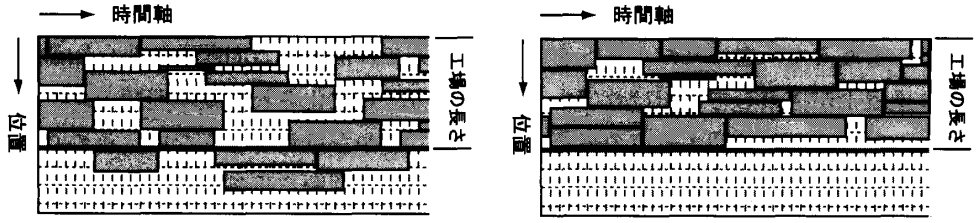

図3　RCPSPによるスケジュール

制約をRCPSPの枠内で実現できるというところがみそである．

図3の左に1回目のRCPSPの解を，右に2回目のそれを示す．1回目の解では，ブロックの上下位置は決めていないので，工場の全長からはみ出ているものもあるが，2回目の計算でブロックを上下に調節して(左右には動かさない) うまく収めているところが見てとれる．すなわち，RCPSPの適用を，変数の取り方を工夫しつつ2度行うことによって，直接的には扱えない問題を解いたという例である．

4. むすび

我々の問題解決エンジン群も，アルゴリズムの改良の結果，実用的に結構使えるということが認識されてきたせいか，すでに何人かの研究者の方々，数社の企業で利用いただいている．なかでもCSPとRCPSPは，汎用性が高いこともあって，利用例が多い．しかし，そのすべてが成功しているわけではなく，やはり，モデリングがうまく行かないと駄目である．うまく行くというのは，問題解決エンジンが高性能をだせるようにモデル化するという意味でもある．改めてモデリングの重要さと難しさを実感している．

ここでは，紙数の都合で二つの例しか書けなかったが，問題解決エンジンにおけるモデリングの役割を幾分でも伝えることができただろうか．究極的には，モデリング作業の支援システムを作って，このプロセスの簡略化に役立てたいものであるが，すでに述べたように，今直ちに構築可能とは考えていない．しばらくは，典型的なモデリング例を積み上げて，理解を深めるという努力を続けるつもりである．

参考文献

[1] 茨木俊秀:「問題解決エンジン」への道，応用数理，Vol. 14, No. 1, pp. 67-70, 平成16年3月．

[2] 茨木俊秀: メタヒューリスティクスによる汎用問題解決システムの構築，科学研究費基盤研究(B)(2)報告書，平成16年5月．