モデルの複雑さの問題点

伊理　正夫

はじめに

どのような科学あるいは技術の分野においても同じことではあるが，ことにORにおいては，対象とする現実の問題のモデル化がそれに続く解析の成否に決定的な影響を与える．どのようなモデルを作るかということは，即，問題をどのように把え理解するかということである．モデルで考えることは不得意で，データにもとづいて考えるという型のやり方もあるそうであるが[1]，そこでは“無構造モデル”というモデルが使われているとみることができる．(空集合も集合である！)

モデル化とは，現実に存在する対象から問題にとって本質的であると思われる要因・構造を選んで取り出すことであるから一つの抽象化であり，また，逆に見れば，残りのものを捨て去ることであるから一つの捨象過程でもある．そこで，モデルは必然的に“数学的”なものとなる．抽象化の効用は数々あるが，その一つに，異なる分野間の情報交換が容易になるということがあげられる．ORの基本的な数学的手法——数理計画，待行列，ネットワーク，ゲーム，信頼性，…——は，そのような意味でも大変有用なものである．異分野間の交流については，その重要性が叫ばれてから久しいが，現実には未だ他分野での成果を十分取り入れられずにいる所が少なくない．ORがそのような面からも社会科学，工学等の諸分野に貢献することが期待される．

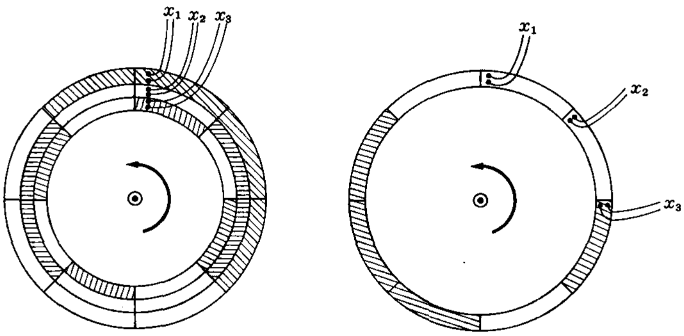
モデル作りは，まだ何が本当の問題であるのかすらはっきりしないような状態から出発するのが常であるから，もちろん，モデル作りにたずさわる人の腕と頭によって，できあがるものは千差万別であろう．それにたずさわる人は誰でも，しかし，“良い”モデルを作ろうと努めることであろう．ここでの“良さ”の規準が何であるかを明確に述べることは困難であるが，それが多面的であることだけは確かである．それら多くの大切な側面の中で最も重要なものの一つとして，モデルの“扱いやすさ”があげられよう．すなわち，作られたモデルを使って，どれだけ多くのことをどのくらい容易に知ることができるかということである．

本特集号の目的は，このモデルの扱いやすさを支配する一大要因である“計算複雑度”という概念を，OR関係者に広く知っていただくことにある．具体的な話題についてはそれぞれの解説記事をご覧いただくことにして，本稿では，モデルの複雑さという見方への動機づけを主にして述べてみたい．

1.　一つの例

数学モデルの作り方一つで扱いがやさしくもな

いり　まさお　東京大学



x_1	0	0	0	0	1	1	1	1
x_2	0	0	1	1	0	0	1	1
x_3	0	1	0	1	0	1	0	1

(a)

0 $\boxed{0\ 0\ 1}$ 0 1 1 1

(b)

図 1 円板の回転位置(1/8回転ごと)を検出する装置

れば難しくもなるという例として，次の有名な問題をとりあげてみよう．

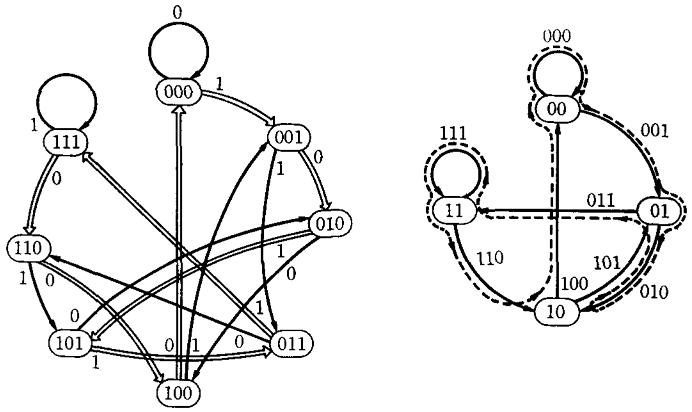
回転軸の位置を検出する装置を作りたい．軸は一回転の1/8ずつしか動かないので，円周を8等分してそのどこにあるかを区別しさえすればよい．そのような区別をするには，軸に絶縁体の円板をとりつけて，その縁に3重に環状の部分を設け，それぞれ円周の1/2，1/4，1/8ごとに交互に導体箔を貼りつけ，導体箔があるかないかを調べるための刷子を3組とりつければよいことは明らかであろう．できあがりは，たとえば，図1(a)のようなものになる．端子対 $x_i(i=1,2,3)$ が導通状態にあるか絶縁状態にあるかを1，0で表わせば，図の下に記してあるように，8個の位置が区別できる．これは，3桁の2進数で8個の数（10進表現で0，1，2，3，4，5，6，7）を表わせるという事実を素朴に利用した方法である．

ところで，環状の部分を1重で済ますことはできないであろうか．これが問題である．刷子はやはり3組用いなければならない（8個のものを区別するには3ビットの情報が要るから）．導体箔を円周8等分の上に上手に貼りつけて，相続く3部分に導体箔があるかどうかのパターンが8通り異なるようにできれば，そのような設計ができることになる．今の場合，図1(b)のようにすれば可能である（図1(b)の下に記してある0，1の系列（円順列とみなす）において，どの連続する3数字もみな異なること，すなわち000～111の8通りの2進数がみな現われていること，を確認されたい）．なぜこのようなことが可能なのであろうか．また，8等分を16等分，32等分，…と一般化してもこのようなことが常に可能であろうか．

一般的に述べると，「円周に沿って0と1を合わせて2^q個並べ，相続くq個の数字を2進数とみなしたとき0から2^q-1までがすべて現われるようにすることができるか」という問題になる．この問題を“グラフ理論的”にモデル化してみよう．とりあえず$q=3$としておく．

第1に考えられるのは，系列中のある位置にある3文字連続 uvw ($u, v, w=0$ または1）に注目し，その“次”に現われる3文字連続として可能なものは $vw0$ か $vw1$ のどちらかに限るという事実を利用することである．すなわち，000～111の8個の3桁の2進数に対応した8個の頂点をもつグラフを考え，2進数 uvw に対応する頂点から2進数 $vw0$ および $vw1$ に対応する頂点へ向かう弧を描く（弧の側にはそれぞれ0，1と付記しておく）．このようにすると図2(a)のようなグラフが得られる．もし求める性質を有する系列が存在するならば，このグラフの上で，すべての頂点をちょうど1回ずつ通ってもとにもどる有向閉路――すなわち，有向 Hamilton 閉路――が存在するはずである．逆に，このグラフに有向 Hamilton 閉路が存在すれば，その弧に付記されている0，1を並べれば求める系列が得られることも明らかであろう．つまり，問題は，有向グラフの上の有向 Hamilton 閉路の存在を調べることに帰着された．

第2の考え方は，系列中の2文字連続に注目することである．上と同様にして，2文字連続 uv の次に現われる可能性のある2文字連続は $v0$ と $v1$ である．ところが，uv の次に $v0$ が現われるときには，そこに3文字連続 $uv0$ があるはずであり，uv の次に $v1$ が現われるときには，$uv1$ があるはずである．そこで，2文字連続00，01，10，11に対応する4個の頂点をもつグラフを考え，uv に対応する頂点から $v0$，$v1$ に対応する頂点に弧を引き，それらの弧の側に，それぞれ，$uv0$，$uv1$ と付記する．このようにして作られたグラフ（図2(b)）には，$4 \times 2 = 8$ 本の弧があり，それらの弧には $0 \sim (2^3-1)$ の8個の2進数がちょうど一つずつ付記されている．もし求める性質を有する系列が存在するならば，このグラフの上で，すべての弧をちょうど1回ずつ通ってもとにもどる有向閉路——すなわち，有向 Euler 閉路——が存在するはずである．逆に，このグラフに有向 Euler 閉路が存在すれば，それにもとづいて求める性質を有する系列を作ることができることも明らかであろう．



(a) Hamilton閉路(⟹)　　(b) Euler閉路(--→)

図 2　系列 00010111 を作り出す二つの方法

グラフに関する各種の問題が，それぞれ，どのくらい“扱いやすい”あるいは“扱いにくい”ものであるかは，ここ10余年の間に発達した“計算複雑度 (computational complexity)”の理論によってかなり良く解明されている．それによると，与えられた任意の有向グラフ上に有向 Hamilton 閉路が存在するかどうかを判定する問題は，いわゆるＮＰ完全問題の一つであって，今のところ“しらみつぶし”式に調べてみるより致し方ない，したがって，グラフが少し大きくなると，かかる手間が急に大きくなる，ような問題であることが知られている．一方，与えられた有向グラフ上に有向 Euler 閉路が存在するかどうかを判定（し，もし存在すればその一つを作製）する問題は，グラフの頂点や弧の数に比例する程度の時間と領域を用いて解くことができる，すなわち，非常に大きなグラフに対しても効率よく解くことのできる，という，“最も扱いやすい”問題であることが知られている．このことは，グラフ理論の既製の理論や手法をそっくりそのまま利用しようとするなら，われわれの問題は第2の考え方にしたがってモデル化するほうが絶対に有利であるということを意味している．そして，このように“上手なモデル作り”をするには，各種の数学モデルの複雑さについての知識が基本的に重要であるということが理解されよう．

ちなみに，本節でとりあげた問題についていえば，図2(b)のようなグラフは，q が2以上のどんな値であっても，2^{q-1} 個の頂点のどれについても，それに入る弧が2本，そこから出る弧が2本という形をしており，また，グラフは連結である．有名な Euler の定理によると，連結グラフの各頂点に入る弧の数と出る弧の数が等しければ有向 Euler 閉路が存在する．したがって，円周を 2^q 等分 $(q \geqq 2)$ した位置を区別するには，その縁に環状の部分を一つだけ設け，刷子を q 組用意すれば十分である．どのように導体箔を貼ればよいかも，上に述べたように，線形時間・線形領域の算法で定めることができる．この問題はさらに

「p 種類の文字を(重複を許して)p^q 個並べた円順列を作り，相続く q 文字連続 p^q 個の集合が p 種類の文字から作られる可能な q 文字連続のすべての集合と一致するようにすることができる」という“de Bruijn の定理”(たとえば[2]参照)に一般化することができる(これは上記の特殊な場合に対するものとまったく同様にして証明される).また，Hamilton(閉)路の問題は“一般には”非常に難しい問題であるが，特殊な形のグラフに対しては存在・非存在条件が知られている場合があり，それに関する論文も数えきれないくらいある(その“専門家”でない，ご用とお急ぎの方のお役には立たないくらい数が多い).

2. 計算複雑度の理論

大規模な問題を手早く解きたいという願望をきちんと数学的に定式化して，どのような型の問題は大規模なものまで比較的やさしく解け，どのような型の問題は少し大きくなると大型計算機の助けを借りてもとても手に負えないようなものであるか，というようなことを明らかにすることを目的とした“計算複雑度の理論”が，ここ10年余りの間に，確立された．それがどんな理論であるかについては成書(たとえば，代表的なものとしての[3]，より通俗的な入門解説書としての[4]，など)を，また，ORとの関連においてぜひ心得ておくべき典型的な話題については本特集号の他の記事を，それぞれ，参考にしていただくことにして，ここでは，そのような理論的な進歩のもたらした実用への貢献，理論の進歩と並んで進行した技術の進歩，等を概括しよう．

過去10年余りの間に計算機の金物(および基本ソフト)の進歩は目ざましかったが，それとともに，大規模な問題を効率よく(計算時間の意味でも必要とされる記憶容量の意味でも)扱うためのデータ構造と算法も長足の進歩をとげ，多くの“定石”が専門技術者の間に定着してきた．15年前には「とにかく，計算機に載せた」というだけで一つの成果あるいは成功であったことが現在では「どのくらい効率よく処理できるようにしたか」が勝負のしどころになっているという例は枚挙にいとまがないくらいある．ことに，組合せ論的側面を有する問題を扱うときにはそのような傾向が強い．ソーティングとか各種のグラフ的な問題とかシミュレーションにおける将来事象の管理法とかは，その代表的なものであろう．数百，数千のノード(頂点)やリンク(弧)からなる道路網上で，最短経路を求めたり最大可能流量を求めたりするとき，素朴なＤＰ式のやり方と現在最もよいとされているやり方とでは，おそらく，少なくて数十倍，ひょっとすると数百倍もの所要時間の違いが出るであろう．それなのに，実際には，まだ旧式のやり方しか知らない，あるいは知っていても使っていない向きがまゝあるやに聞く．(この辺の事情については本特集号の真鍋氏の記事や，少々古いが[5]などが参考になろう.)

計算複雑度の理論では，算法の複雑さ(あるいは問題そのものの複雑さ)を「問題の規模 n が大きくなるにつれて，それを解くのに要する時間が n のどのような関数として増大する可能性があるか」によって測ることにしている．その関数 $T(n)$ が $\limsup_{n\to\infty} T(n)/n^k<\infty$ を満たすとき，すなわち，$T(n)$ が高々 n^k 程度でしか大きくならないとき，$T(n)=O(n^k)$ と書く．このような $T(n)$ をもつ算法(あるいは問題)は“多項式オーダー”の複雑度であるという．多項式オーダーの複雑度というのは，問題の規模 n が大きくなっても実用的にかなりの所までは扱うことができるという種類のものであるとみなされる．これに対して，“非多項式オーダー”のものは，どのような k を選んでも $O(n^k)$ と表わせない，すなわち $O(2^n)$ とか $O(n^n)$ とかいうような複雑度である．あらゆる可能な場合を“しらみつぶし式”に調べてゆく方法はたいていこのような複雑度をもつ．非多項式オーダーの複雑度をもつ方法を用いたのでは，ちょっと大きな問題(たとえば $n=10$ とか15とかいう程

度）でも計算時間がかかりすぎて扱うことができなくなる．

“理論”では，このように，多項式オーダーと非多項式オーダーをはっきりと区別し，前者を“実用的”，後者を“非実用的”なものの数学的な定義としている．さらに，同じ多項式オーダーの中でも$O(n^k)$のkの小さいものほど“より効率がよい”とみなしている．しかし，この“実用性”，“効率性”の理論的定義が必ずしも現実的環境における実用性，効率性と一致するとは限らないことにも注意すべきである．たとえば，最近のトップニュースの一つであるKhachianによる線形計画問題の多項式オーダーの解法（解説［6］参照）よりは最悪の場合非多項式オーダーになる単体法のほうが，現実的環境のもとでは遙かにすぐれている．また大きな行列（そのような行列には零要素が多いのが普通である）の積を作るのには，四則演算の総数が$O(n^{2.8\cdots})$であるStrassenの方法よりは，それが$O(n^3)$である通常の方法のほうがよいことも確かめられている（nは行列の次数）．ネットワーク関係の算法についても，理論的な複雑度が小さいものより，主単体法系統の算法を“磨き上げた”もののほうが実際には性能がよいと，実績にもとづいて主張しているグループがある［7］．さらに，“しらみつぶし法”そのものも，分枝限定法等にいろいろな工夫を凝らして，少しでも実用性を向上させようとの努力が続けられている．“理論”のほうも，少しでも現実に近づこうとして，厳密解でなく近似解を求める算法の複雑度とか，最悪の場合でなく平均的な場合の計算時間とかいうようなものの研究へと進んできている（が，果たして本当に現実に近づいているであろうか）．

計算複雑度の理論の中での大傑作はS. Cookによる“NP完全性”の概念である．世の中には，「多項式オーダーの解法がまだ知られてはいないが，そのような解法が存在しないということが証明されているわけでもない」という問題が数多くある．そのような問題の相当部分が「そのうちのどれか一つの問題に対して多項式オーダーの解法が存在すれば他のすべての問題に対しても多項式オーダーの解法が存在する」という関係でお互いに結ばれている．この事実を指摘し，そのようなグループに属する問題を“NP完全問題”と呼んだのがCookである．前節で触れたHamilton閉路の問題もNP完全問題の一つである．現在のところ，NP完全問題を解く多項式オーダーの算法は存在しないであろうと思われている．

「“計算機科学科”の卒業生は企業で評判がよくない」という噂が，ある国では，流されているという．その根拠は，何か現実的な問題を与えて解決するよう頼むと，「その問題は，理論的には有名なNP完全問題であるこれこれしかじかの問題となりますので，解くのは諦めるべきであります」という返事をすぐするからであるという．これは，本特集の主題の核心に触れることである．計算複雑度の知識をこのように消極的に利用するのは，決してOR的な態度ではない．与えられた現実の問題がそのような難しい理論的問題になることを知ったときに選ぶべき道は，上のような返事をすることではなく，次の二つでなければならない．

①モデル化を変えてみる：——現実の問題の理論モデルが一意的に定まるのでないことはすでに強調したとおりである．もっとうまいモデル，すなわち，考え方，があるかも知れない．

②理論的に難しい問題であることがわかったら，困惑するのではなくむしろ喜ぶべきであると悟る：——理論的によく調べ尽されている問題に自分の問題が当てはまるのであれば，先人の成果を十分に勉強してそれを役立てないと後で人に笑われるかも知れない．しかし，それについての有用な成果がないことが知られているような“難しい”問題であれば，どうせ調べても大して得るところはない

であろう．したがって，一般理論の助けを借りることはあきらめて，自分に与えられた“具体的な特定の”問題を曲がりなりにも——いくら泥臭い方法によっても，また，少々解の質が悪くとも——解決するよう努力しさえすればよい．

しかし，自信をもってこのような選択をなすことができるためには，計算複雑度についてそれなりの勉強をし，十分な知識をもっていなければならないことはもちろんである．

3. “複雑さ”以外にも大切なことはある

本稿のはじめにも述べたことではあるが，“複雑さ”はモデル作りの際に考慮すべき大切なことの一つであるが，それがすべてではない．たとえば，本質を損わずに複雑さを減らそうとして多大の頭脳力を投入することが，より広い視点からみて引き合うことであるかどうか．それは，そのようにして作られたモデルが後でどのくらい頻繁に利用されるかにもよるが，それ自身まさにOR的な問題である．(実務家の立場と研究者の立場との違いも出てこよう．)また，モデルには，もっと基本的な要請があることも忘れてはならない．それは，素朴な表現ではあるが“適切さ(adequacy)”とよばれるものである．いくら簡単で現実によく合うモデルであっても，物理的に考えてつじつまが合わないものであってはならないし，そのモデルの正当性を他人に説明できるようなものでなければならない．

“良い”モデル作りというもっと広い視野からみて，“複雑さ”という観点の果たすべき役割にORの実務家の方々が関心をもち，そして，計算複雑度の研究者達へ注文をつけていただけるなら，それはORにとっても計算複雑度の研究の将来にとっても，大変幸いなことと思う．

参 考 文 献

[1] 森口繁一：モデルとデータ，経営科学，Vol. 17, No. 4(1973年7月)，pp. 191～204.

[2] R. G. Busacker and T. L. Saaty : *Finite Graphs and Networks——An Introduction with Applications*. McGraw-Hill, 1965. (矢野，伊理訳：『グラフ理論とネットワーク——基礎と応用』培風館，1970.)

[3] A. V. Aho, J. E. Hopcroft and J. D. Ullman: *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974. (野崎，野下 他訳：『アルゴリズムの設計と解析I，II』サイエンス社，1977.)

[4] 伊理正夫，野崎昭弘，野下浩平(編著)：『計算の効率化とその限界』入門・現代の数学，13，日本評論社，1980.

[5] 『ネットワーク構造を有するオペレーションズ・リサーチ問題の電算機処理に関する基礎研究』日本オペレーションズ・リサーチ学会，報文シリーズ T-73-1，1973.

[6] 伊理正夫：線形計画法に画期的な新解法現わる？ オペレーションズ・リサーチ，Vol. 25, No. 3(1980年3月)，pp. 187～193.

伊理正夫：線形計画法の計算複雑度-Khachianの理論とその周辺．第1回数理計画シンポジウム論文集(1980年11月)，pp. 29～41.

[7] F. Glover, D. Klingman, J. Mote and D. Whitman : An extended abstract of an in depth algorithmic and computational study for maximum flow problems. *Discrete AppliedMathematics*, Vol. 2, No. 3 (1980), pp. 251～254.

F. Glover and D. Klingman : Precis of computational analysis of shortest path algorithms. *COAL Newsletter*, Committee on Algorithms, Mathematical Programming Society, August 1980, pp. 2～5.