企業・産業の進化研究会

設計の複雑さのもうひとつのパラブル

2013.6.19 塩沢由典

（0）意図と失敗

タイトル：「設計の複雑さのもうひとつのパラブル」

要旨：「設計の複雑さは、アーキテクチャ論をはじめ、多くの分野で基本的な主題となっている。本報告では、設計の複雑さを、多数の制約条件をもった最適化問題(たとえば、作業順位つき最適スケジューリング問題)の計算複雑さと捉えることにより、いくつかの問題が整理できることを示す。」

0.1　生産とはなにか

藤本先生の設計情報の転写論

藤本隆宏・大隈慎吾(2007)、藤本隆宏(2009)、藤本隆宏編(2013)

0.2 技術とはなにか、生産性の差異はいかに生ずるか

「大量生産を行なう産業においては、生産コストの差は、どの国におけるどのような特定の生産においても同じ技術が用いられるので、一国の気候や好物資源といった天賦の条件による差をさほど反映しなしそうにないように思われる。」(Davidson, 2011, 第 7 章「国際貿易の改革」、p.121.)

生産コストがいかなる要因に影響をうけるのか。たとえば、おおきな労働生産性の差異を生み出すものはなにか。「同じ技術」とされているものが、じつは大きな生産性の差異をともなうことをどう説明し、(国際)経済学の標準的知識とするか。

0.3　Is 'evolution' compatible with 'design'?

'Design' is central to modern technology. How can that be reconciled with 'evolution' which both Darwin and Lamarck explained as a process through which complex adaptive systems emerge in the absence of design?　(Ziman, 2000, §1.3)

（1）公理系設計の枠組み

1.1 帰納・構造の対応関係を示すマトリックス

$$A = \begin{「} a_{11} \dots a_{1n} \\ \dots \\ a_{n1} \dots a_{nn} \end{」}$$

機能要素のベクトル

$f = (f_1, \ldots, f_n)^t$　　$(\,)^t$ は縦ベクトル表す。

構造要素のベクトル

$s = (s_1, \ldots, s_n)^t$

要件　$f = A\,s$

A の構造

設計の複雑さを表す。> モジュール型　インテグラル型

Cf. 藤本先生の多数の著書、論文　とくに藤本編『「人工物」複雑化の時代』

方程式　$f = A\,s$ の計算複雑さ

Cf. 藤本(2013)p.104 脚注 1 Jordan 法、Guass 法の計算量　O(n^3)

工夫したものでは、$O(n^{2.49+})$で可能。伊理・野崎・野下(1980) p.46.

☆計算複雑さの方面では、n^3 の時間複雑さは、ほとんど「複雑さ」と考えられていない。

1.2 線形計画法の場合

ほとんどの場合、$O(m^2\,n)$で解ける。これは、非常に高速な計算法と考えられていて、じっさい　m=1,000, n =1,000 程度の問題も数値的に解かれている。

☆公理系設計の機能・構造要件　$f = A\,s$ では、A の詳細が不明という制約を除けば、設計計算が「つねに進化しうる」ものであるとはいえない。

Cf. 2 段階設計モデル　藤本(2013)第 2 章第 5 節。

問題:設計計算のみで、設計は「つねに進化しうる」ものであることを示すような適切な parable(譬え話)はないのか。

（2）計算複雑さの好例としての線形計画法

2.1 問題

制約条件

$$A\,x \leqq b \qquad (2\text{-}1)$$

のもとに

$$c_1\,x_1 + \ldots + c_n\,x_n \quad (2\text{-}2)$$

を最大化せよ。

ただし、A は m 行 n 列の行列で m≧n。

①(2-1)を満たす解が存在するか。
存在すれば、多面集合(有界でないかもしれない、有界なら多面体)　P。

n 個の $\mathbf{a}^j$ について

$$a^j{}_1 x_1 + \ldots + a^j{}_n x_n = 0 \quad (2\text{-}3)$$

最初に勝手に(たとえば、上から n 個) $\mathbf{a}^j$ たちを選び、方程式を解いて、解が(2-1)を満たすかどうか確認する。どんな n 個をとっても解がなければ、P は空集合。

もし P が空でないなら、上のような組合せ $\{\mathbf{a}^j\}$ はかならず存在する。

②点 X が P の頂点か。
(2-3)を満たす n 個の$\{\mathbf{a}^j\}$たちが一次独立ならば、$X = (x_1, \ldots, x_n)$は P の頂点。

③頂点 X が(2-2)を最大化する点か。
もし $\mathbf{a}^j$ たちが一次独立でかつ

$$a^j{}_1 x_1 + \ldots + a^j{}_n x_n \leqq 0$$

ならば、頂点 X は(2-1)を満たし、関数(2-2)を最大化する。

2.2　単体法(Dantzich, 1963)

最初に任意に n 個の $\mathbf{a}^j$ を選らび、①の解 X が存在したら、次には、X の隣りあう頂点で(2-2)がより大きくなるものがあるか、確認する。これは、掃き出し法とほぼ同じ。

より大きくなるもの X'があれば、その X'が③を満たすか確認する。③を満たせば、X'において(2-2)は最大。そうでなければ、X'について X と同じ操作を繰り返す。

単体法の良いところ
　すべての頂点について(2-2)を計算しなくても、最大かどうか確認できる(③)。
　経験的には、②の頂点は容易にもとまり、頂点の取替えも 3m 個程度の取替えで最大点が求まる。
　よって、ふつうの場合、$O(m^2 n)$ていどの計算の手間で解ける。

しかし、最悪の場合、たとえば m=2n で、すべての頂点を確認しなければならない場合、4^m もの時間がかかる(指数関数時間)。

☆計算時間は、あるプログラム(アルゴリスム)について、平均的な計算時間、最悪の計算時間、最小の計算時間などを区別する必要がある。
☆おなじ問題、与件でも、アルゴリスムが異なれば、計算時間も異なる。
　実際の計算においては、計算時間のほかに、必要な記憶容量の大きさも関係する。しかし、ここでは、必要記憶容量の問題にはいっさい触れない。

（3）NP 困難、NP 完全

3.1 定義　P 問題(Polynomial-time problem)

　多項式時間以内に解ける問題。たとえば、線形方程式、行列の掛算、ソーティングなどは P^問題である。このうち、判定問題を P^と書く。

判定問題と最適問題とは、しばしば関係している。
　　k を最適問題 P(w, max)の最適値とするとき、k を満たす解 w があるかどうかを判定する問題が考えられる。逆に最適問題の値の候補の数が w の多項式で押さえられるならば、最適問題は、多項式時間で判定問題に帰着できる。

☆これは問題のクラスに関する概念であり、その問題を解く(ある)アルゴリスムについての概念ではない。線形計画問題は、最悪の場合、指数関数時間かかるので P^問題ではない。

3.2 定義　NP 問題(Nondeterministic Polynomial time Problem)

　候補 c が問題 P(w)の解であることを確認する w のサイズの多項式時間アルゴリスムが存在する。

[定理] 問題 P が NP 問題であることと、P(w)が多項式時間の非決定性チューリング機械で判定されることとは同等である。あるいは、P(w)の非決性時間計算量がある k につき NTIME(n^k)に属する。ただし、n は w のサイズ。

☆線形計画問題は、NP 問題である。
P ⊆ NP であるが、P＝NP であるかどうかは、知られていない(未解決)

基本的予想は、P≠NP

3.3 定義　多項式時間帰着可能

ある問題のクラス A と別の問題のクラス B とがあって、A の問題 w は多項式時間内に B の問題は f(w)に変換可能であるとき、クラス A はクラス B に多項式時間帰着可能(還元可能)という。

3.4 定義　NP 完全

問題 B が NP であり、かつ NP に属する任意の問題 A が問題 B に多項式時間帰着可能なとき、問題 B は NP 完全という。

定理(Cook-Levin の定理)

充足可能問題 SAT (あるいは 3SAT)は、NP 完全である。

参考　定義　NP 困難

NP 完全問題に多項式時間帰着できる問題は、NP 困難という。NP 困難な NP 問題を完全という。

現在では、多数の NP 完全問題が知られている。

（4）スケジューリング問題 parable

4.1　多くのスケジューリング問題が NP 完全であることが知られている。

仕事$\{J_1, J_2, \ldots, J_n\}$があり、それらを処理する機械が一台ある。すべての仕事を一定の順番で処理することを考える。(メインフレームのタイムシェアリングのようなものを考えればよい)

4.2　想定

○各仕事 J_i は、機械によって処理される。

○同時に異なる二つの仕事をさせることはできない。

○仕事の間には、前後関係(先行関係)がありうる。それは半順序で表される。

○各仕事 J_i には、作業時間 p_i が決まっている。途中で、別の仕事に移ることはできない。

○各仕事 J_i には、納期 d_i が決まっている場合がある。

○各仕事 J_i には、準備時刻 r_i が決まっている(r_i 以降でないと仕事を始められない)。

スタート時から測って J_i の終了時刻を c_i とする。このとき、$T_i = \max\{d_i - c_i, 0\}$が遅れ時間である。

4.3　以下は、NP 完全問題の例

先行関係がある仕事の順序配分で準備時刻のない問題で、目標関数を

①Σc_i (総終了時間)

②$\Sigma w_i c_i$ (重みつき総終了時間)　　ただし、$\Sigma_i w_i=1$ とする。

③ΣT_i (総遅れ時間)

と最小とする問題、

④ $\Sigma w_i T_i$ (重みつき総遅れ時間)　先行関係を問わない。

準備時刻のある場合で、
⑤ Σc_i (総終了時間)
⑥ L_{max} (最大遅れ、$\max\{0, d_1-c_1, \ldots, d_n-c_n\}$
⑦ ΣT_i (総遅れ時間)
⑧ $\Sigma \delta_i$ (納期遅れ仕事数、$\delta_i = 1$ if $d_i > c_i$, 0 if not)

はすべて NP 完全問題となる。

注意 : これはスケジューリング問題がすべて NP 完全になるという意味ではない。与件に特別な条件があるとか、目的関数が別のものであるととき、$O(n^3)$とか $O(n \log n)$のように、高速に解ける問題もある。
以上は、伊理・野崎・野下(1980)p.165[茨木俊秀]による。

4.4　考察
これは、設計過程の流れ設計の難しさに関係しているかもしれないが、あまりうまい parable とはいえない。むしろ、生産過程の複雑さをかんがえるヒントになるかもしれない。

とくに、自動車のような大量生産型の生産と、受注生産で多くの仕事が流れていると生産との生産過程の最適化問題などには、示唆が得られるかもしれない。

これらは、「離散事象最適化」と呼ばれる、より広いクラスの一部である。離散事象最適化の方法の一つとして、ある種の問題(max と+のみで記述できるもの)は、非線形問題ではあるが、本質的には線形問題の類比で解けるものが多い。

（５）多面体 parable

5-1　凸多面体
空間 $\mathbb{R}^d$ において、凸多面体(凸多面集合で有界なもの)を考える。

凸多面体 P は次の二つの表現をもつ。

H 表現　　m 個の一次不等式の共通部分　(m 個の半空間の共通部分)
　　これだけだと、有界集合でないかもしれない。
V 表現　　n 個の頂点から生成される凸集合　(n 個の端点から生成される凸集合)

5.2 H 表現(面側による表現)

$\mathbf{a}^1 = (a_{11}, a_{12}, \ldots, a_{1d})$

$\mathbf{a}^2 = (a_{21}, a_{22}, \ldots, a_{2d})$

...

$\mathbf{a}^m = (a_{m1}, a_{m2}, \ldots, a_{md})$

という m 個の横ベクトルを並べた行列を A とするとき、

$P = \{ \mathbf{x} = (x1, \ldots, xd)^t \mid \ <\mathbf{a}^j, \mathbf{x}> \ \leqq \ 1, j=1, \ldots, m\}$

$= \{ \mathbf{x} \mid A\mathbf{x} \ \leqq \ 1\}$

と表現できる。有界であるという条件については、いちいち触れない。

5.3 V 表現(頂点による表現、端点)

$\mathbf{v}^1 = (v_{11}, v_{12}, \ldots, v_{1d})$

$\mathbf{v}^2 = (v_{21}, v_{22}, \ldots, v_{2d})$

...

$\mathbf{v}^n = (v_{n1}, v_{n2}, \ldots, v_{nd})$

とするとき

$P = \{ \mathbf{x} \mid \ \exists \lambda_1, \ldots, \ \lambda_d \ \geqq \ 0, \ \lambda_1 + \ldots + \lambda_n = 1, \ \lambda_1 v^j + \ldots + \lambda_n v^n\}$

と表現できる。

5.4 凸多面体の両表現間の転換の計算時間

空間 $\mathbf{R}^d$ において、凸多面体 P を考える。

充足必要条件(一次不等式)の集合　H 表現

要素の凸配置(点)の集合　　V 表現

と考える。

これらは、幾何学的には同一物であるが、それを転換する計算量は、多大である。

①H 表現から V 表現

もし H の大きさのみを問題にするなら、これは(最悪の場合)指数関数時間かかる。

例 1.　二重確率行列(N×N)の頂点数は　N!

よって、書き下すための時間だけで、指数時間かかる。

つまり N^2+2N 個の不等式

$x_{ij} \ \geqq \ 0, \quad \Sigma_i x_{ij} \ \geqq 1 \ , \ \Sigma_i x_{ij} \ \leqq 1, \ \Sigma_j x_{ij} \ \geqq 1 \ , \ \Sigma_j x_{ij} \ \leqq 1$

の端点の集合は、N 個の元の置換の集合(全部で N! 個ある)に一致する。

この計算時間は、解を打ち出す時間だけでも O(N!)。

例 2. 超立方体
面の数　　$-1 \leqq x_i \leqq 1$　　$m=2d$
頂点の数　　$2^d = 2^{m/2}$

②V 表現から H 表現
極表現をとれば、これは①に帰着する。

例 1.　巡回多面体
$v(1) = (x_1, x_1^2, \ldots, x_1^d)$
$v(2) = (x_2, x_2^2, \ldots, x_2^d)$
...
$v(n) = (x_n, x_n^2, \ldots, x_n^d)$
ただし、$x_1, x_2, \ldots, x_n$ はたがいにあい異なるものとする。

これは n 個の頂点をもつに過ぎないが、面側(ファセット)は、{1, 2, ... , n}から任意の s 個(ただし、s = [d/2 を超えない最大の整数])選ぶとき、
$v(j(1)), v(j(2)), \ldots, v(j(s))$
はすべて面側 facet(となる。Cf. van der Monde の行列式

したがって、面側は、${}_nC_d$ 個ある。
n=2d で n を大きくするとき、面側の個数は 2^d 以上の速さで増大する。

③H 表現・V 表現の大きさを指定するとき
size(P) = |V|+|H|+d = m + n + d
と定義するとき、(これまで知られている)すべての転換式(conversion program)に必要な計算時間は、size(P)の superpolynomial である。(Avis, Bremmer and Seidel 1997)

5.5. 両表現を制約条件と機能条件と見立てる
$\mathbf{R}^d$ をどう解釈するか。
　機能空間か？　その一点は、なぜある構造要素で実現できるのか。

機能要素間関係　> H 表現(面側(facet)の集合と隣合う面の情報(稜 ridge))
　これは、m 個の一次不等式で書かれる。
　不等式を付け加える　> 制約条件の強化

冗長な不等式があるかもしれない。
H 表現がきつすぎれば、解がないかもしれない。

☆機能要件を制約条件と考えれば、これはかなりうまい比喩といえる。

構造要素間関係　＞V 表現(端点=頂点り集合と、隣り合う頂点の情報(辺 edge))
構造要素のひとつひとつを頂点と見なす。
冗長な頂点(つまり端点でないもの)があるかもしれない。
V 表現が与えられるとき、かならずそれを実現するものがある？

☆構造要素が端点で表されるという譬えがあまり納得的なものではない。

5.6　機能・構造関係の読替え
H → V,　V → H ともに超多項式時間がかかる。
いつくか得られる観察

帰結(1) 満足化原理
大規模なシステム(size(P)の大きい P)では､H 表現と V 表現の転換に超多項式時間かかるため、かならずどこかで停止しなければならない。
帰結(2)　局所化関係
機能要素・構造要素の変更は、うまい場合には、局所化できる(されている)。
最悪の場合には、すべでがすべてに関係する。
帰結(3)　閉写像関係
H → V,　V → H のやや微妙な"連続性"
ときに、新しい頂点、新しい面側が生まれ、消滅する。
帰結(4) 原価条件
原価に対する制約も、他の機能制約と同じ用に入れることができる。

（６）構造要素の実現問題

機能空間 $\mathbf{R}^d$　の任意の点が、機械的な構成要素になるだろうか。もしこれが一般には無理だとすると、機能空間 $\mathbf{R}^d$ は、ある構成要素空間に埋め込む必要がある。たとえば、ある機能はある一つの機械要素によってのみ実現できると考えるとき、機能空間 $\mathbf{R}^d$ の点

(1, 1, 2, 0, ... , 0)

は、

(1, 0, 0, 0, ... , 0)、(0, 1, 0, 0, ... , 0)、2 (0, 0, 1, 0, ... , 0)

という延べ 4 つの機械要素によってのみ実現可能かもしれない。

こう考えると、構成要素空間で考えたとき、要求仕様 H による多面体が P だとしても、P を実現するためには、それを構成要素空間に埋め込んで、さらに機械要素の配列 A により被覆しなければならないかもしれない。このとき、

$$P \subseteq A$$

という形となり、A と P との間に、かなりの過剰機能が付加されることになる。

言い換えれば、要求仕様に合うような機械設計は、つねに過剰機能をもつことを運命付けられており、それをいかに削減していくかも、設計上の工夫の一つであるかもしれない。

[参考文献]

伊理正夫・野崎昭弘・野下浩平編 1980 『計算の効率化とその限界』(数学セミナー増刊) 日本評論社

奥野正寛・瀧澤弘和 2006 「人工物の複雑化と製品アーキテクチャ」MMRC-J-81

瀧澤弘和 2001 「モジュール化の非インセンティブ理論」奥野正寛・池田信夫編『情報化と経済システムの転換』東洋経済新報社

藤本隆宏 2009「アーキテクチャとコーディネーションの経済分析に関する試論」『経済学論集』75(3):2-39.

藤本隆宏・大隈慎吾 2007 「設計立地の比較優位に関する試論」RIETI DP S07-J-025

藤本隆宏編 2013 『「人工物」複雑化の時代』有斐閣

電気学会進化技術応用調査専門委員会編 2010『進化技術ハンドブック』基礎編/応用編(情報・通信システム、生産・物流システム)、近代科学社

Davidson, P. 2011『ケインズ・ソリューション』日本経済評論社。

Sipser, M 2000 『計算理論の基礎』共立出版(英語版は新版あり)。

Avis, D., D. Bremmer and R. Seidel 1997 How good are convex hull algorithms? Computational Geometry 7: 265-301.

Fujimoto, T., and T. Oshika 2006 Empirical Analysis of the Hypothesis of Architechture-based Competitive Advantage and International Trade Theory, MMRC DP No.71.

Ziman, J. (Ed.) 2000 Evolutionary models for technological change, Technology innovation as an evolutionary process, Cambridge University Press.