解説

NP型問題の近似解法の可能性について

渡辺　治

1　はじめに

コンピュータの出現した当初からＮＰ問題は我々の悩みの種である．様々な分野で「これがうまく解ければいいのだが」と思うような問題が往々にしてＮＰ困難な問題だったりする．しかも「ＮＰ困難な問題を妥当な時間内で解く一般的手法はない」といわれている．しかし，たとえ“困難”といわれようが，それで諦めたら研究者／技術者として失格で，何とか解こうと試みるのが普通だろう．そうした試みの一つが近似解法である．つまり,「完全解や最適解は難しくても近似解ならばできるかもしれない」という考えのもと，“次善の策”を試みる方法である．本稿では，ＮＰ問題に対する様々な近似のアプローチを述べ，その可能性に関してこれまでにわかってきたことを解説する．

ＮＰ問題に関する研究では，個々の問題だけでなく，ＮＰ問題に共通する構造や性質に関する研究もある．そのような一般的な研究は“ＮＰ問題の構造的研究”と呼ばれているが，近似解法の分野でも，より一般的に「どのアプローチがどの程度見込みがありそうか？」という研究が行なわれている．ここでは，そのような一般的な研究で何がどこまで解明されているかを紹介する．

2　準備

議論を始める前に，まずこれから使う用語や概念などを整理しておこう．なお，本稿では，数学的な厳密さを多少犠牲にしても，できるだけ直観的に説明するよう試みた．細かい議論や厳密な証明については，この分野の入門的な教科書 [3, 7, 15] を参考にされたい．

わたなべ　おさむ
東京工業大学　工学部
〒152　目黒区大岡山 2-12-1

問題とは，計算量とは

我々の対象とする問題とは，かしこまって言えば，入力に対して妥当な出力とは何かを規定したものであり，問題を解くとは，与えられた入力例（input instance）に対して問題に規定された出力を求めることである．ただし，入力として与えられるのは数だけではない．文字列もあればグラフもある．また式も自身も入力として与えられる．出力も同様で，つねに数が出力されるとは限らない．その一例として巡回セールスマン問題を考えてみよう．これは，図１のように，都市間の輸送コストを表したグラフ G と総コストの上限 B が与えられたとき，総コスト B 以下ですべての都市を巡る順路を求める問題である．

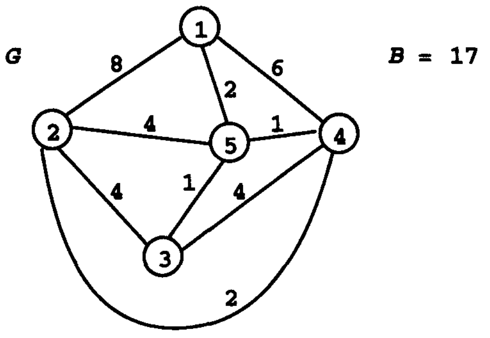

図１

たとえば上の例では，$1 \to 5 \to 3 \to 4 \to 2 \to 1$ が答え（の一つ）である．この巡回セールスマン問題では，都市間の輸送コストのグラフ G と総コストの上限 B が入力例で，条件を満たす巡回路が出力である．

さて問題を解く“アルゴリズム”の効率だが，本稿では，アルゴリズムの計算効率を（適当な機械の上で走らせたときの）計算時間で評価する．また，とくに断らない限り最悪時計算量を用い，アルゴリズム A の計算量

として次の関数 time_A を用いる．

$$\mathrm{time}_A(n) = \max_{\text{長さ } n \text{ の入力例 } x} \{ A(x) \text{ の計算時間} \}.$$

つまり，同じ長さ n の入力例の中でも最悪の入力に対する計算時間を $\mathrm{time}_A(n)$ とし，この関数 time_A の大小で A の効率を評価していく．なお，一般論を展開するときは，このように入力例の長さをパラメータ（問題のサイズ）として計算時間を表すのが普通だが，個々の問題に対しては入力例の複雑さを表す妥当な量 — たとえば巡回セールスマン問題の場合は都市数 — をパラメータとして議論した方がよいだろう．

多項式時間計算可能性

アルゴリズム A の時間計算量 $\mathrm{time}_A(n)$ が n の多項式で押えられるとき，A を多項式時間アルゴリズムという．そして多項式時間アルゴリズムで解ける問題を多項式時間計算可能という．計算の複雑さの理論における最大の関心事は「多項式時間計算不可能なＮＰ問題がある」という予想 — $\mathrm{P} \neq \mathrm{NP}$ 予想 — の証明である．ところでなぜ多項式時間計算可能性を問題にしているのだろうか？よく「多項式時間が現実的な計算時間か否かの境目だから」といった説明を耳にする．しかし，これは非常に大雑把な言い方なので以下の点に十分注意して使わなくてはならない．まず第一に，“手に負えない”の十分条件として多項式時間計算不可能を用いた方がよい．というのも多項式時間計算可能だからといっても，たとえば n^{100} 時間かかるようなアルゴリズムは，とても現実的ではない場合が多いからである．第二に，これは一般的な尺度であり，問題によっては計算量が指数関数時間になっても，実際にはある程度まで“手に負える”場合もある，という点．たとえば，ある問題を $2^{n/10000}$ 時間で解くアルゴリズムがあったとしよう．そうすれば，数十万のサイズ（n）までは十分解くことができるし，もし現実的にはその程度のサイズの入力例しか考える必要がないのならば，その問題は十分手に負えると考えてよい．したがって，非多項式時間だからといって，一概に非現実的と決めつけるのは危険である．ただし，多くの場合「非多項式時間 $\Longrightarrow$ 手に負えない」という図式は成り立つので，多項式時間計算可能性を使って一般論を展開しているのである．

ＮＰ型問題

世間では“ＮＰ問題”という用語は，かなり広い意味で使われている．本質的には同じことなので，それ自体，目くじらを立てることではないのだが，本稿での議論のように可能性の限界ぎりぎりの話しになってくると，微妙な違いでも重要になってくる．そこで，以下では広い意味での“ＮＰ問題”をＮＰ型問題と呼び，本来のＮＰ問題とは区別することにする．

さて，ＮＰ型問題とは何か？ここではまず，ＮＰ型問題のひとつ— ＮＰ探索問題 — から考えよう．簡単にいえば，ＮＰ探索問題とは，与えられた入力例に対し，その解をもらえば，それが正しい解かどうかを容易に判定できる問題である．（より正確な定義は [15] を参照されたい．）たとえば，先の巡回セールスマン問題も，与えられた入力例に対し答えの巡回路を見せられれば，それが条件を満たすかどうかが容易に判定できる．したがってＮＰ探索問題の一つである．

一般にいわれているＮＰ型問題には，この他に次の二種類がある．

ＮＰ（判定）問題：与えられた入力例に対し，条件を満たす解が存在するか否かを判定する問題．これが本来のＮＰ問題だ．たとえば，巡回セールス問題の例でいえば，入力例のグラフ G と総コスト上限 B に対し，「G にコスト B 以下の巡回路があるか？」を答える問題である．

ＮＰ最適化問題：与えられた入力例に対し，条件を満たし，しかも（問題に定められた基準で）最適な解を見つける問題．たとえば巡回セールスマン問題では，与えられた G に対し，コストが最小となる巡回路を求める問題である．この種の問題では当然，問題の規定の一部として最適性の基準が与えられている．つまりそれぞれの最適化問題 X ごとに，入力例 x に対する解 y のコスト $\mathrm{cost}_X(x,y)$ を与える関数（コスト関数）が決められている．与えられた x に対し，この $\mathrm{cost}_X(x,y)$ を最小（最大）にする解 y を求めるのが最適化問題である．なお，$\mathrm{opt}_X(x)$ で入力例 x に対する最適解のコストを表すことにする．

単に NP といったときは，ＮＰ判定問題の全体を意味する．一方，P は多項式時間計算可能な判定問題の全体．したがって，$\mathrm{P} \neq \mathrm{NP}$ 予想とは「ＮＰ判定問題の中に多項式時間で計算不可能なものが存在する」という意味である．ただしＮＰ探索問題あるいはＮＰ最適化問題の多項式時間計算可能性は，次に示すように $\mathrm{P} = \mathrm{NP}$ と同値である．

定理 2.1. (たとえば [3, §5.1] 参照)
次の (1), (2), (3) は同値:
(1) $\mathrm{P} = \mathrm{NP}$,
(2) すべてのNP探索問題は多項式時間計算可能,
(3) すべてのNP最適化問題は多項式時間計算可能.

つまり「多項式時間計算可能か?」という立場で見ると,この三種類の問題は同じ複雑さなのである.その意味では,これらすべてを“NP問題”とみなしても間違いではない.

ところで,ひとくちにNP型問題といっても全部が全部難しいわけでもない.NP型問題の中にはいとも簡単に解けてしまうものもある.一般的に議論する際には,そうした簡単な問題ではなく難しそうなNP型問題を対象にしたいので,普通はその代表格であるNP困難な問題について考える.簡単にいえば,NP困難な問題とは,その問題が多項式時間で解けてしまうと,他のNP型問題がすべて多項式時間で計算可能になるような問題のこと.たとえば先の巡回セールスマン問題は,判定,探索,最適化のどの形でもNP困難である.以下では,「××近似で多項式時間に解けないNP型問題がある」といった結果がでてくるが,そうした否定的な結果はすべてのNP困難な問題に当てはまる.(注:よく“NP完全”という用語が使われるが,NP完全問題とはNP困難なNP判定問題のこと.それ以外のNP型問題には“NP困難”といっておいた方が無難だろう.)

3 様々な近似へのアプローチ

ここでは“次善の策”をすべて近似的手法と考え,広い意味での近似可能性について述べていこう.具体的には,(i) 近似問題,(ii) 疑似多項式時間アルゴリズム,(iii) 平均的な解析,の三種類の近似アプローチについて紹介する.

3.1 近似問題

次善の策として,まず「問題の目標を緩める」という方針がある.つまり「完全に正解を出すのが難しいのならば,ある程度不正確な答えでもよい」という考え方だ.このように問題の目標を緩めたものを一般に近似問題という.ただし目標の緩め方も問題の種類に応じて異なってくるので,以下では判定問題と最適化問題に分けて説明していこう.(探索問題に関してはあまり研究が進んでいないので今回は省略する.)

NP判定問題の近似問題
「間違って答えてしまう入力例があってもよい」というのが近似判定問題である.ではどの程度間違ってよいか?ということで近似の程度が決まってくる.ここでは間違ってよい入力例の個数を制限した場合を考えてみよう.

サイズ n の入力例は一般に n の指数関数程度ある.たとえば入力例はすべて 0,1 列に符号化されているものとし,そのビット長を入力のサイズとした場合には,入力例の数は(ゴミのような入力例も入れて)2^n 個である.「そのうち多項式個は間違ってもよい」というのが以下に定義する P-close 近似である.

定義 3.1. X を判定問題,A を判定アルゴリズム(Yes/No と答えるアルゴリズム)とする.このとき,適当な多項式 p に対して(ほとんど)すべての n で,

$$A\text{ が間違って答えるサイズ } n \text{ の入力例の数} \le p(n)$$

ならば,A は X の P-close 近似アルゴリズムという.また,A のような多項式時間アルゴリズムがある場合に,X は P-close 近似可能という.

この概念は Yesha (1983), Schöning (1986) により提案され,NP問題の P-close 近似可能性が議論されてきたが,結局,否定的な結果が示されている.

定理 3.2. [9] $\mathrm{P} \neq \mathrm{NP} \Longrightarrow$ P-close 近似不可能なNP問題が存在する.(とくにNP困難な問題はすべて P-close 近似不可能.)

つまり多項式個の間違いを許したくらいではNP問題はやさしくならない.しかし“多項式個”というのは 2^n に比べるとかなり小さい.では,もっと間違いを許したらどうか?たとえば「2^n 個のうちの 1% は間違えてよい」としたらどうだろう.このように入力例全体に対する割合を考えるようになると,これは平均的解析に他ならない.これについては 3.3 節で述べる.

NP最適化問題の近似問題
最適化問題の場合には「最適の $(1+\varepsilon)$ 倍程度のコスト以内の解でもよい」といった近似問題が妥当だろう.その方針に従ったのが以下の定義である.

定義 3.3. (以下の定義は [7] に従ったもの)
X を最適化問題,A をアルゴリズムとする.
(1) 定数 $\delta > 0$ に対して,すべての入力例 x で

$$\frac{|\mathrm{cost}_X(x, A(x)) - \mathrm{opt}_X(x)|}{\mathrm{opt}_X(x)} \le \delta$$

となるとき，A は X の δ-近似アルゴリズムという．なお，上の条件を満たす解を δ-近似解と呼ぶ．

(2) X に対して δ-近似となる多項式時間アルゴリズムが存在するとき，X は δ-近似可能という．さらに，任意の $\delta > 0$ に対して δ-近似可能な場合には，弱近似可能という．

(3) 入力例 x の他に近似パラメータ δ も入力とし，x のサイズと $1/\delta$ の多項式時間以内で，x に対する δ-近似解を出力するアルゴリズムを強近似アルゴリズムといい，強近似アルゴリズムを持つ問題を強近似可能という.（注：本稿で用いる“弱近似”や“強近似”といった用語は，まだ定着したものではないので別の用語が使われている場合もある.）

ここに定義した近似可能性の概念はかなり早くから研究されているようで，たとえばナップサック問題の強近似可能性は Ibarra-Kim (1975) によって示されている.（具体的な近似アルゴリズムについては [7] を参照.）一方，近似不可能性が簡単に示される場合もある．たとえば（もし $\mathrm{P} \neq \mathrm{NP}$ ならば）どんな大きな δ を考えても一般の巡回セールスマン問題が δ-近似不可能なこと，あるいはグラフ頂点彩色問題が弱近似不可能なことなどは，比較的簡単に示せる [3]．このようにＮＰ困難な最適化問題のいくつかについては，その近似可能性や不可能性が早いうちから知られていたが，不思議なことに，これらの結果は他のＮＰ困難な最適化問題へは簡単に移行できなかった．ＮＰ困難な最適化問題は，それぞれ互いに還元可能であり，多項式時間計算可能性の意味では同等の複雑さを持つのだが，近似可能性を議論するには，普通の還元では弱過ぎて関係がうまく示せないからである．そのため一般論がなかなか展開できなかったのだが，Papadimitriou-Yanakakis [10] がより制限された還元（Linear-reduction）を導入し，一般論のための枠組を作った．とくに彼らの定義した MAX SNP というクラスは，重要な最適化問題のほとんどが入るクラスで，このクラスに属しているＮＰ困難な問題の弱近似可能性が重要であることがわかった．そして昨年，ＮＰの性質に関する非常に面白い結果をもとに次の事実が証明された.（この定理の証明に使われた技法はそれ自身，非常におもしろいもので，ニューヨーク・タイムズ紙 (April 7, 1992) にも大きく紹介された.）

定理 3.4. [1] $\mathrm{P} \neq \mathrm{NP} \Longrightarrow$ クラス MAX SNP の最適化問題のうち弱近似不可能なものがある．

3.2 疑似多項式時間アルゴリズム

「多項式時間アルゴリズムが難しそうならば，もう少し強力なアルゴリズムで解くことができないか?」という方向もある．つまり，多項式時間計算可能性を拡大解釈するという方針だ．“強力なアルゴリズム”にも多種多様なものが考えられるが，(i) 計算時間を長くしたものと，(ii) 従来の計算モデルと異なったモデルの上でのアルゴリズム，の二つに大別される．ただし，残念ながら前者に関しては適当な題材が見当たらなかったので省略し，ここでは後者についてのみ説明する．なお，以下の議論では，ＮＰ型問題の種類の違いはあまり問題にならない．そこで簡単のためＮＰ判定問題（以下，ＮＰ問題と呼ぶ）を例に考えていくことにする．

従来のアルゴリズムと異なったアルゴリズム，あるいは従来の計算モデルと異なったモデルとして，ここでは (i) ランダム性アルゴリズム，(ii) 論理回路族，(iii) ニューラル・ネットワークなどに代表されるアナログ計算モデル，の三つを考える．

ランダム性アルゴリズム

ランダム性アルゴリズム（確率的アルゴリズムともいう）とは，計算の途中で乱数を使い，適当に計算を進めていくアルゴリズムのこと．したがって，乱数の出方次第では間違った答えを出すこともある.「たまには間違えるけれども大抵の場合は正しく解いてくれればよい」という方針で作られたアルゴリズムである．もう少し正確にいうと，問題 X を解くランダム性アルゴリズム (randomized algorithm) とは，適当な $\varepsilon < 1/2$ に対して，誤り率がつねに ε 未満となるようなアルゴリズムのことをいう．ただし，ここで誤り率が出てきているが，これは個々の入力例に対する誤り率で，後で述べる平均的な議論での誤り率（入力全体の中で何 % 間違えるか）とは異なることに注意しておこう．つまり X を解くランダム性アルゴリズムは，すべての入力例に対して誤り率が ε 以下でなければならない．なお,「適当な定数 $\varepsilon < 1/2$ に対して」でよいのは，誤り率が 1/2 よりも少しでも低いアルゴリズムがあれば，あまり計算時間を増やさないでも誤り率を非常に小さくできるからである．

最近では，このランダム性アルゴリズムを“次善の策”ととらえずに，多項式時間計算可能性の概念をラン

ダム性アルゴリズムによる多項式時間計算可能性まで広げて議論する傾向が強まっている．実際，適当な乱数発生器があれば，ランダム性アルゴリズムを実現するのは簡単なので，この拡張は自然だろう．それではランダム性アルゴリズムまで考えることによって，ＮＰ問題は多項式時間で解けるようになるだろうか?残念ながらその見込みはほとんどない．確かにランダム性アルゴリズムまで考えることによって，技術的にはおもしろい手法がいろいろと使えるようになったし，それによって解けるようになったＮＰ問題もある．(具体的な問題に対するランダム性アルゴリズムついては [7, 13] などを参照されたい.) しかしそういった問題はＮＰ問題の中でも限られており，一般のＮＰ問題，とくにＮＰ完全問題がランダム性アルゴリズムで容易に解けるとは信じられていない．

多項式サイズ論理回路族

論理回路とは，AND, OR, NOT ゲートと入出力ゲートからなる組合せ回路（回路中に閉路を持たない回路）のこと．以下ではこの論理回路を単に“回路”と呼ぶことにする．普通，回路の入力ゲート数は可変ではないので，入力例の長さ（ビット長）ごとに回路が必要になる．そこで問題 X を解く回路といった場合には，各 n ごとに定義された回路族 $\{C_n\}_{n\geq 1}$ を意味する．ただし，それぞれの C_n がサイズ n の入力例すべてに対して正しい答えを返す回路である．また，回路の複雑さは回路中のゲート数で評価される．適当な多項式 p に対し，各 C_n のゲート数が $p(n)$ で押えられるとき，$\{C_n\}_{n\geq 1}$ を多項式サイズ（論理）回路族という．

さて，多項式サイズ回路族による計算可能性は，“多項式時間計算可能性”の自然な拡張になっている．実際，次の関係が比較的容易に示せる（[2, §5] 参照）．

定理 3.5. X が（ランダム性アルゴリズムで）多項式時間計算可能 $\Longrightarrow X$ は多項式サイズ回路族を持つ．

ところが，多項式サイズ回路族は多項式時間アルゴリズムより計算能力が高い．これは回路族の非一様性に起因している．回路族 $\{C_n\}_{n\geq 1}$ を考えた場合，$C_1, C_2, C_3, \ldots$ に何らかの関連があるのが普通だろう．たとえば，パラメータ n を与えると C_n が簡単に作れるようになっている．ところが我々の多項式サイズ回路族の定義では，この種の一様性を求めてはいない．つまり各 n ごとに C_n がまったく違った回路であっても構わない．この部分が多項式時間アルゴリズムとの違い（拡張されている点）になっている．しかし,「非一様な回路族を考えて何の意味がある．n から C_n が簡単に作れないようじゃ意味がないではないか」と思われる方も多いだろう．たしかに非構成的な多項式サイズ回路族でＮＰ問題が解けてもあまりうれしくない．だが逆に「非一様性を許した多項式サイズ回路族でもＮＰ問題は解けない」ことが示せれば，$\mathrm{P}\neq\mathrm{NP}$ よりかなり強い計算不可能性がいえたことになる．このように強い意味での多項式時間計算不可能性を言うために“多項式サイズ回路族”という概念が重要なのである．実際，次のような関係が予想されており，その有力な状況証拠も示されている．

予想. $\mathrm{P}\neq\mathrm{NP} \Longrightarrow$ 多項式サイズ回路族でも解けないＮＰ問題が存在する．

ニューラル・ネットワーク

アナログ計算モデルの一例として，ニューラル・ネットワークの能力に関する Siegelmann-Sontag の最近の結果を紹介する．要点をひとことでいえば「(多項式時間内計算では) ニューラル・ネットワークと多項式サイズ回路族の能力に大差はない」という結果である．

彼らのニューラル・ネットワークがどのような計算モデルであるかを簡単に述べてみよう．ただし，スペースが限られているのでニューラル・ネットワークについては他の文献に譲ることにして，ここでは彼らのモデルの特徴を述べるだけにする．まず，ネットワークの構造だが，有限個のニューロンの相互結合（再帰結合も許す）からなる回路を考える．つまり“ホップフィールド型ネットワーク”と呼ばれているものである．ただし，ニューロンの個数，結合方法などは問題によって固定で，前述の論理回路族などと異なり入力長によっても変わらない．したがって長い入力例の各ビットを一度に並列には受けられないので「適当な入力素子から逐次的に入力する」という方式をとる．次に，重みや閾値だが，これには任意の実数を用いてよい．ただし，この値はネットワークごとに固定で，入力例ごとに，あるいは計算途中で変えることはできない．最後に各ニューロンの出力値を決めるシグモイド関数だが，微分可能関数ならば何でも使ってよいことにする．

彼らの結果は，以上のニューラル・ネットワーク上での計算時間（ニューロン値の更新回数）に応じた計算能力に関するもので，その中でも多項式時間内での計算能力に関しては次の定理が示されている．

定理 3.6. [12] X があるニューラル・ネットワークで多項式時間計算可能 $\Longleftrightarrow X$ は多項式サイズ回路族を持つ．

したがって，ニューラル・ネットワークを使っても多項式時間内では状況が大幅に改善されることはない．たとえば，もし多項式サイズ回路族に関する先の予想が正しいとすると，(P $\neq$ NP な限り) ニューラル・ネットワークを使ってもＮＰ完全問題は多項式時間では解けないことになる．

3.3 平均的によいアルゴリズムの可能性

従来の計算の複雑さの解析は，最悪の入力例に対する計算量 — 最悪時計算量 — などの解析が主体であった．しかしそれでは，ほとんどの場合にうまく動くアルゴリズムも悪く評価されてしまう．たとえば，悪い計算量持つアルゴリズムでも，ほとんどの入力例に対して効率よく動くこともあり得るし，また，ある入力例に対しては間違えるアルゴリズムも，ほとんどの入力例に対して正しく答えることもある．こうした点を考察するのが平均的な解析である．

今までに，いくつかの具体的なＮＰ型問題（とくにＮＰ困難な問題やその部分問題）に対して，平均的に効率良く解くアルゴリズムが提案されている（これらについての解説は [6] を参照）．ただし，こうした結果の評価はかなり難しい．ＮＰ困難な問題の場合，逆にＮＰ困難性ゆえに簡単な部分も多く含んでおり，そうした簡単な部分が入力例の大半を占めてしまう場合も考えられるからだ．(なぜＮＰ困難性ゆえに簡単な部分を含むのか？これに関しては [15, §6.3] をご覧頂きたい．観点は少し異なるが直観は得られると思う．) 実際，提案されているアルゴリズムが驚くほど単純で，議論の中心は「それでも大抵は正しく解ける」という点になっている場合も多い．こうした結果を「大抵はうまく解けるんだ！」と喜ぶか，「本当に難しい部分は何ら解決されていない」と思うのべきかは，対象となっている問題や，それが生じてきた背景などによって異なるので一概に断定できないだろう．

一方，もっと一般的にＮＰ問題の平均的な難しさを研究するための理論が Levin [8] により導入され，その方面の研究も少しずつ進んでいる．たとえば，平均的ＮＰ完全問題もいくつか見つかっているので（詳しくは [4] 参照），それらに挑戦してみるのもおもしろいだろう．もし平均的ＮＰ完全問題を平均的に効率良く解くアルゴリズムがあれば，非常に一般的な意味で「平均的にＰ＝NP」がいえることになる．しかし，残念ながら「平均的にもＰ$\neq$NP」が予想されており，その状況証拠（？）が計算論的暗号理論の研究から得られている．現代の暗号研究では，公開暗号系などＮＰ型問題を使った暗号の研究が盛んである．その重要な鍵となるのが暗号的一方向関数（cryptographic one-way function）だ．一般に，$f(x)$ を計算するのは簡単だが，$f^{-1}(y)$ の計算は手に負えない関数を一方向関数というのだが，ほとんどすべての y に対して $f^{-1}(y)$ の計算が難しい（多項式時間でできない）関数をとくに暗号的一方向関数という．計算論的暗号理論では，こうした関数の構成法に関する研究が詳しく行なわれており，いくつか有力な候補も得られている（詳しくは [14, 16] を参照）．ところで逆関数の計算はＮＰ探索問題なので，暗号的一方向関数の存在は平均的なＮＰ型問題の計算不可能性につながりがあることが，容易に予想できるだろう．実際，次の事実が示されている．

定理 3.7. [5] 暗号的一方向関数が存在する $\Longrightarrow$ 平均的にＰ$\neq$NP.

ところで，先に述べた二つの近似アプローチに対しても平均的な解析が重要である．たとえば，「平均的に見てＮＰ最適化問題が近似可能か？」とか，「ニューラル・ネットワークの平均的な効率はどうか？」などといった問題もこれからの重要な課題になっていくだろう．

4 おわりに：理論からのメッセージ

以上，ＮＰ型問題の近似解法の可能性について急ぎ足で紹介してきたが，これらをまとめると，「様々な近似アプローチがあるが，どのアプローチでもすべてのＮＰ型問題を解く一般的で強力な方法はない」ということになるだろう．残念ながら否定的な結果ではあるが，こうした理論的結果はある意味で我々に対する重要なメッセージでもある．つまり，「ＮＰ型問題すべてに効く特効薬的な計算方法はないことがわかってきた．だからこそ個々のＮＰ型問題に対する丁寧なアルゴリズム作りが大切だ」というメッセージである．以上見てきた理論的結果は，単純な方法でアルゴリズムを大量生産するわけにはいかないことを示している．したがって，各々の問題の特徴を分析し，その問題の背景に合ったアルゴリズムを丁寧に作っていかなければな

らない．そのためには，アルゴリズム開発の技術力向上について真剣に考えていく必要があると思う．

本稿は第 5 回 RAMP シンポジウムでの発表原稿をもとに書いた．シンポジウム発表の機会を与えて下さり，本執筆を勧めて下さった，東京工業大学の小島政和教授ならびに森雅夫教授に深く感謝いたします．またシンポジウムの発表にご意見，ご批判を下さった方々にも感謝いたします．

参考文献

[1] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy, Proof verification and intractability of approximation problems, in *Proc. 33rd Annual Sympos. on Foundations of Computer Science*, IEEE (1992), 14–23.

[2] J. Balcázar, J. Díaz, and J. Gabarró, *Structural Complexity I*, EATCS Monographs on Theoretical Computer Science, Springer-Verlag, 1988.

[3] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Co., 1979.

[4] Y. Gurevich, Average case completeness, *Journal of Computer Systems and Science* 42 (1991), 346–398.

[5] R. Impagliazzo and L. Levin, No better ways to generate hard NP instances than picking uniformly at random, in *Proc. 31st Annual Sympos. on Foundations of Computer Science*, IEEE (1990), 812–821.

[6] D.S. Johhson, The NP-completeness column, *Journal of Algorithms* 5 (1984), 284–299.

[7] 笠井琢美，戸田誠之助，計算の理論，共立出版（株），1993.

[8] L. Levin, Average case complete problems, *SIAM Journal on Computing* 15 (1986), 285–286.

[9] M. Ogiwara and O. Watanabe, On polynomial-time bounded truth-table reducibility of NP sets to sparse sets, *SIAM Journal on Computing* 20 (1991), 471–483.

[10] C.H. Papadimitriou and M. Yannakakis, Optimization, approximation, and complexity classes, *Journal of Computer Systems and Science* 43 (1991), 425–440.

[11] M. Sipser, The history and status of the P versus NP questions, in *Proc. 24th Annual ACM Sympos. on Theory of Computing*, ACM (1992), 603–618.

[12] H.T. Siegelmann and E.D. Sontag, Neural networks with real weights: analog computational complexity, Technical Report, Dept. of Computer Science, Rutgers Univ. (1992).

[13] 戸田誠之助，実際的計算可能性の拡張について，情報処理 31 (4) (1990), 518–524.

[14] 渡辺 治，一方向関数のお話し，情報処理 32 (6) (1991), 704–713.

[15] 渡辺治，計算可能性・計算の複雑さ入門，近代科学社，1992.

[16] 渡辺治，一方向関数について，“離散数学とアルゴリズム III（室田一雄編）”，近代科学社，出版予定.