数理解析研究所講究録
第716巻 1990年 10-18

10

平均計算時間に基づく計算複雑さの研究について
（解説）

東工大・工学部　渡辺　治（Osamu Watanabe）

概要：　ＮＰ問題の平均計算時間の理論的研究について解説する．

1．問題の背景

計算の複雑さの理論では，“時間計算量”はおもに最悪計算時間に基づいて測られている．（以下では，これを簡単に最悪時間計算量と呼ぼう．）たとえば，「プログラムの計算時間が$3n^2$である」といえば，普通，「最悪の入力を考えても（その入力のサイズnにたいし）$3n^2$の計算時間でプログラムが答えを出す」ということを意味する．もちろん最悪時にも速いのはよいことだが，プログラムＡの最悪時間計算量が$10n^8$で，プログラムＢがn^2のとき，簡単にＢのほうがＡより優れているとは言い切れない．ほとんどの場合で，Ａの方がＢより速い，つまりＡがＢよりも平均的に速いこともありうるからだ．それなら平均の計算時間に基づいて計算時間を測ればよいではないか，それが“平均時間計算量”である．

ＮＰ問題，とくにその中でもＮＰ完全問題は，手に負えない問題と予想されている．つまり，それらを解く効率的アルゴリズムがなく，どんな速いコンピュータをもってしても天文学的（いやそれ以上，計算科学的！[Mi89]）計算時間がかかると予想されている．ところがこれは“最悪時”の話である．もしかすれば平均的には速く解けるのかもしれない．たとえばつぎのような例を考えてみよう（この話は [Jo84] より）．

ハミルトン閉路問題（与えられたグラフにハミルトン閉路があるか否かを判定する問題）は，ＮＰ完全問題としてよく知られている．ところが，ある妥当

な入力分布を仮定すると，この問題は平均時間 $O(n^2 \log n)$ で解くことができる．

定理：　つぎのようなアルゴリズムHが存在する．

（1）すべてのグラフにたいし，Hはそれがハミルトン閉路を持つか否かを正しく判定する，そして

（2）Aの計算時間は，

$$\exists c, \forall n, \quad \sum_{G \in \mathcal{G}_n} p_n(G) \times time_H(G) \leq c n^2 \log n + c.$$

ただし，$\mathcal{G}_n$ は n 頂点のグラフの全体．また，各 n ごとに p_n は，n 頂点のグラフ G が入力として与えられる確率で，つぎのように定義される：

$$p_n(G) = \left(\frac{1}{2}+\varepsilon\right)^e \times \left(\frac{1}{2}-\varepsilon\right)^{N-e}.$$

ただし，　$N = n(n-1)/2$ ，　e = 辺の数．

（簡単にいえば，各辺を $\frac{1}{2}+\varepsilon$ の確率でランダムに発生させたときにが生成される確率．）

すなわち，確率分布 ρ_H $(=\{p_n\})$ に従って入力が与えられるとすると，アルゴリズムHは $O(n^2 \log n)$ の平均時間計算量でハミルトン閉路問題を解いてしまうのである．この定理は，ハミルトン閉路問題が（最悪時間計算量的にみて）とうてい多項式時間内に解けないと予想されていることを考えると，驚異的な結果のように思える．しかし，そうでもない．実は，この確率分布で生成したグラフは，ほとんどの場合ハミルトン閉路を持ってしまうのだ．もう少し正確にいえば「ρ_H でグラフを生成すると，（ある方法で簡単に見つかる）ハミルトン閉路を持つグラフができる確率が $1-o(2^{-n})$ になる」ことが Chvátal により証明されている．したがって，ほとんどの入力でアッと言う間にハミルトン閉路が見つかる．アルゴリズムHは，非常に運の悪い場合だけ，まじめにハミルトン閉路を捜せばよいのである．だから $O(n^2 \log n)$ の平均計算時間ですむのだ．なんだかペテンにかけれらたような気持ちだろう；アルゴリズムHの速さはハミルトン閉路問題の“易しさ”とはあまり関係ないからだ．もちろん，Chvátal の定理は非常におもしろい．しかし，この定理はハミルトン閉路問題の本質的な“難しさ／易しさ”を解析しているわけではない．では，平均

時間という立場から見て，ハミルトン閉路問題の本質的な“難しさ／易しさ”を議論するにはどうすればよいのだろう?

平均時間計算量の方が最悪時間計算量より，より現実的であるにもかかわらず，その理論的研究があまりされていない．そのもっとも大きな理由は「妥当な入力（確率）分布を決めることができない」という点だろう．入力分布は，その問題が考えられている状況に応じて違うのだから，ひとつの入力分布にたいして理論を作り上げてもただの机上の空論になりかねない．一方，すべての入力分布にたいして計算時間を議論することは，最悪時間計算量を解析することに他ならない．また，哲学的（？）にいえば，ひとつの入力分布のもとでの平均時間計算量の解析が，その問題の平均的難しさの本質的解析になるかどうかが疑問である．たとえば，ハミルトン閉路問題は，入力分布 ρ_H では平均的に多項式時間で解けるけれども，その事実から「他の入力分布でもハミルトン閉路問題が易しい」ことは示せない．こういった問題点に決着をつけないことには平均時間計算量の理論的研究を始めることができないのである．

ところが最近（といっても1989年頃），Levin がこういった問題に決着をつけるひとつの糸口を示した．つまり平均時間計算量を議論する枠組みを提案し，ある問題にたいしては“妥当な入力分布”を示したのである．また，それに関していくつかの研究が報告されている．以下ではその概要を述べたい．なお，以下の議論は，計算の複雑さの理論で普段行われている議論の方法にのっとって行い，よく知られている用語・記号は定義せずに用いる（[HU79]などを参照されたい）．ここでのアルファベット Σ は $\{0,1\}$ とする．

2．Ｌｅｖｉｎ の枠組み

Levin の枠組みは“多項式時間計算可能性”を平均時間計算量の立場から議論するためのものである．平均時間計算量を一般的に考えるため，Levin の枠組みでは，問題と入力分布の組を扱う．入力分布は，Σ^* 上の分布関数で与える．ただし，つぎの条件を満たす関数 μ が Σ^* 上の分布関数と呼ばれる：

(i) $\forall x_1, x_2 \in \Sigma^*,\ x_1 \leq x_2 \rightarrow \mu(x_1) \leqq \mu(x_2)$,

(ii) $\lim_{x \to \infty} \mu(x) = 1$.

（注：Σ^* 上の順序としては辞書式順序を用いる）

すなわち，分布関数μのもとでは，入力xは $\hat{\mu}(x) = \mu(x) - \mu(\hat{x})$ という確率分布で与えられると考える．（ただし，$\hat{x}$ は Σ^* 上でxのひとつ手前の元．つまり，$\hat{\mu}$ はμの微分．）

入力分布のこの決め方には少し違和感を感じられる読者も多いだろう．たとえばグラフの入力分布を考えるとき，普通は ρ_H のように入力のサイズ（頂点数）ごとの確率分布を考える．それに対し，上の方法は，すべてのグラフに対する確率分布を考えているのだ．たしかに ρ_H のような入力分布の与え方は直感に合うが，その方法だとサイズの決め方やコーディングの方法に依存した定義になってしまう．一方，上の方法はそれらに影響されないので，一般論に向いている．また，入力のサイズごとの議論も簡単に上の枠組みにはめ込むことができる．たとえば，頂点nのグラフがすべて長さmの Σ^* の文字列でコーディングされるとすると，ρ_H に対応する分布関数 μ_H には，

$$\forall n, \forall G \in \mathcal{G}_n, \quad \hat{\mu}_H(G\text{のコード}) = \frac{6}{\pi^2 m^2} \rho_n(G).$$

（注：$\sum_{m=1}^{\infty} \frac{6}{\pi^2 m^2} = 1$ ）

を用いればよい．

さらに，Levin の枠組みでは，つぎのような条件を満たす分布関数　だけを考える：

(iii) μ は多項式時間内に計算可能．

“自然な”入力分布の多くははこの条件を満たしている．以下では，単に分布関数といったならば (i), (ii), (iii) の条件を満たす関数を指すことにしよう．Levin 流議論では，集合Dと分布関数μの組(D, μ)を対象とするのである．この組を分布付き問題と呼ぶことにしよう．

分布付き問題(D, μ)をひとつ考えよう．我々は，その問題が平均的にみて多項式時間で解けるか，という点を議論したい．そのために“平均計算時間が多項式以下”という概念を定義しよう．あるアルゴリズムＡと$k > 0$が存在して，

$$\exists C, \overset{\infty}{\forall} n, \quad \sum_{x \in \Sigma^n} \hat{\mu}(x) \times time_A(x) \leq C \cdot n^k,$$

となれば平均計算時間が多項式以下だ，というのが自然な考え方だろう．ところがこれはあまりうまくない．アルゴリズムＡが上の条件を満たしていても，その高々２乗の計算時間を要するアルゴリズムＢで，

$$\forall k, \forall C, \overset{\infty}{\exists} n, \sum_{x \in \Sigma^n} \hat{\mu}(x) \times time_B(x) > C \cdot n^k,$$

となることがある．これでは「どんな計算モデルでも多項式時間計算可能性という概念は一定である」という点が崩れてしまう．これに対し，もう少し弱い基準を考える．

$$\exists C, \overset{\infty}{\forall} n, \sum_{x \in \Sigma^n} \hat{\mu}(x) \times \left(time_A(x)\right)^{1/k} \leq C \cdot n.$$

これでもＡの平均計算時間が多項式以下といってよい（ような気がする）．しかも，もしＡが上の条件を満たすならば，それと高々多項式程度でしか計算時間がちがわないアルゴリズムはすべて上の条件を満たすことが示せる．そこでつぎのような定義が妥当だろう．

定義：　分布付き問題 (D,μ) にたいし，つぎのようなアルゴリズムＡが存在するとき，それを多項式平均時間計算可能という．

（１）　Ａは D を認識する決定性アルゴリズム，そして

（２）　$\exists k, \sum_{x \in \Sigma^*} \hat{\mu}(x) \times \left(time_A(x)\right)^{1/k} / |x| < +\infty.$

（注：この式は上の式をいいかえたものに過ぎない）

たとえば，(HAM, μ_H) はアルゴリズムＨのおかげで，多項式平均時間計算可能となる．

Ｐ，ＮＰに対応して，つぎの計算量のクラスが考えられている：

ave-Ｐ　＝　$\{(D,\mu) : (D,\mu)$ は，多項式平均時間計算可能$\}$，

dist-ＮＰ　＝　$\{(D,\mu) : \quad D \in$ ＮＰ$\}$．

Ｐ $\subseteq$ ＮＰとことなり，ave-Ｐ $\subseteq$ dist-ＮＰであるとは限らない点に注意．（なぜ，ave-ＮＰを考えないのだろう？）我々の最大の関心事はつぎの疑問だ：

疑問： dist-NP $\subseteq$ ave-P ?

さて，NP完全問題のようにdist-NP完全問題というのを定義できないだろうか？そのためには“多項式時間還元性”なるものをここでも定義する必要がある．そのためにいくつかの概念を定義しよう．

定義：

（1）分布関数 μ_1 と μ_2 にたいし，もしある多項式 p で，

$$\forall x \in \Sigma^*, \quad \mu_1(x) \leq p(|x|) \cdot \mu_2(x)$$

ならば，μ_1 は μ_2 に支配されるといい，$\mu_1 \sqsubseteq \mu_2$ と記述する．

（2）Σ^* から Σ^* への関数 f と分布関数 μ にたいし，$f(\mu)$ をつぎのような分布関数と定義する：

$$\forall y, \quad f(\mu)(y) = \sum_{f(x)=y} \mu(x).$$

定義： 任意の分布付き問題 (D_1, μ_1) と (D_2, μ_2) にたいし，つぎのような関数 f が存在するとき，(D_1, μ_1) は (D_2, μ_2) へ還元可能といい，$(D_1, \mu_1) \propto (D_2, \mu_2)$ と記述する．

（1） f は多項式時間計算可能な Σ^* から Σ^* への関数，

（2） f により， $D_1 \leq_m^p D_2$ ，

（3） $f(\mu_1) \sqsubseteq \mu_2$.

（注：この定義の条件は Levin [Le86] のそれより強い．私は [Le86] の定義には誤りがあると思う．）

これらの定義の意味についてここでは詳しく述べないが，つぎのことが成り立つように定義したと考えてよいだろう．

定理： $(D_1, \mu_1) \propto (D_2, \mu_2) \wedge (D_2, \mu_2) \in \text{ave-P} \rightarrow (D_1, \mu_1) \in \text{ave-P}$.

dist-NP完全性の定義は従来通りつぎのようになる：

定義： (D, μ) は，つぎの条件を満たすときdist-NP完全である．

（1） $(D, \mu) \in$ dist-NP,

（２）　すべての(D', μ')が，$(D', \mu') \alpha (D, \mu)$.

Levin の仕事の大事な点は，このような dist-ＮＰ完全な問題を実際に示した点にあると思う．ここでは，Gurevich [Gu87] が示した dist-ＮＰ完全問題を考えよう．

$$K = \{ \langle M, x, 0^l, 0^t \rangle : \exists w \in \Sigma^l,\ M \text{ accepts } \langle x, w \rangle \text{ in } t \text{ steps} \},$$

$$\hat{\mu}_K(\langle M, x, 0^l, 0^t \rangle) = \frac{6}{\pi^2 (m+n+l+t)^2 \cdot 2^m \cdot 2^n}$$

ただし，M は決定性チューリング機械，m はMの記述の長さ，
そして，n はxの長さ．

定理：(K, μ_K) は dist-ＮＰ完全．

つまり，ave-P $\supseteq$ dist-ＮＰ？という問題は，$(K, \mu_K) \in$ ave-P？という具体的な問題の難しさに帰着されたのだ．これは“妥当な入力分布”という面からみても重要である．問題Kでは，その平均計算時間が入力分布μ_Kのもとで多項式以下ならば，どんな入力分布のもとでも多項式時間以下に解くことができるのである．この意味で，(K, μ_K)はKの平均時間計算量を本質的に表しているといえよう．ハミルトン閉路問題にたいしても，このような入力分布を定義できる．（コメント：Levin らは，“自然な”入力分布を好んでいるようだ．ところが多くの場合，自然な入力分布で完全問題をつくるのは難しい [Gu87]．これはコーディングの問題からくるのだと思う．しかし，なぜ自然な入力分布にこだわるのだろう？）

従来の計算量のクラスとの関連について述べておこう．つぎの定理は簡単に証明できる．

定理：

（１）　$P = NP \rightarrow$ dist-NP $\subseteq$ ave-P，

（２）　DEXT $\neq$ NEXT $\rightarrow$ dist-NP − ave-P $\neq \emptyset$.

３．暗号の安全性を調べるのに適した枠組み

最近では，ある問題の“計算の複雑さ”用いて暗号系を作ることが行われて

いる．そうした暗号系では，生成した暗号鍵の平均的安全性が問題となる．具体例を示そう．つぎのようなアルゴリズムＸを考える：

アルゴリズムＸ：
入力nにたいし，乱数を利用して約nビットの素数の組(p,q)と
その積$m=p\cdot q$を出力する．

このＸの出力のうちmだけを見て，(p,q)を求める問題を考える．この種の問題が暗号鍵の安全性の問題と思ってよいだろう．さて，この問題（すなわち，ある種の整数にたいする素因数分解）の平均的難しさを解析したい．それには，入力分布をＸが　を出力する確率に基づいて決めるのが妥当だろう．ところが，そのような入力分布の中には，Levin の条件(iii) を満たさないものがあり，Levin 流の枠組みではこの問題を議論できないのである．

Ben-David ら [BDGL89] は，条件 (iii) の代わりにつぎのような条件 (iv) を提案している：

(iv)　乱数を利用して$\hat{\mu}$の確率分布で入力例を生成する多項式時間アルゴリズムが存在する．

そして，ＮＰ問題Dと，(i), (ii), (iv) を満たす分布関数μの組(D,μ)のクラスを sample-ＮＰ と定義した．このクラスは，我々が今考えた問題にピッタリのクラスである．このクラスについても完全問題の存在が示されている．

4．最後に

この分野はいまのところあまり注目されていない．しかし，ことによったら宝の山かもしれない，と私は思うのですが，‥‥

18

参考文献

[BCGL89] S. Ben-David, B. Chor, O. Goldreich, and M. Ludy, On the theory of average case complexity, in "Proc. 21st STOC", ACM (1989), 204-216.

[Gu87] Y. Gurevich, Complete and incomplete randomized NP problems, in "Proc. 28th FOCS", IEEE (1987), 111-117.

[HU79] J. Hopcroft and J. Ullman, "Introduction to Automata Theory, Languages, and Computation", Addison-Wesley (1979).
(邦訳) オートマトン 言語理論 計算論 II, サイエンス社.

[Jo84] D. Johnson, The NP-completeness column: an ongoing guide, J. Algorithms 5 (1984), 284-299.

[Le86] L. Levin, Average case complete problems, SIAM J. Comput. 15 (1986), 285-286.

[Mi89] S. Miyano, propsosed in this meeting.