コンピュータと数学：計算の複雑さから量子コンピュータへ

名古屋大学大学院多元数理科学研究科
ルガル フランソワ

本稿は 2025 年度年会の市民講演会（2025 年 9 月 20 日）での講演内容を，配布・記録用の文章としてまとめたものである．計算を数学的対象として捉える「計算の複雑さ理論」を軸に，P ≠ NP 予想の歴史的背景，数独や社会的ネットワークに現れる計算問題，NP 完全性理論，近似計算の困難性，そして量子計算と量子アルゴリズム（特にショアのアルゴリズム）の意義を概説する．

1　P ≠ NP 予想の歴史的背景

P ≠ NP 予想を理解する第一歩は，「問題を解くこと」と「解が正しいと確認すること」の違いである．例えばパズルや難問では，正しい解が与えられれば検算は容易だが，そもそも解を見つけるためには膨大な試行錯誤が必要になることがある．

計算の複雑さ理論では，入力サイズ（問題の大きさ）を n としたとき，計算に必要な手順数が n の多項式で抑えられるものを「効率よく解ける」と見なす．このとき，P は多項式時間で解ける問題の集合（直観的には「速く解ける」問題）と定義される．NP は，解が与えられたとき，その正しさを多項式時間で検証できる問題の集合と定義される．

P ≠ NP 予想は「P と NP は一致しない」，すなわち

解を速く確認できても，必ずしも解を速く見つけられるわけではない

という主張である．P ≠ NP を証明するには，「どんなアルゴリズムを使っても速く解けない」問題が NP の中に存在することを示さねばならない．しかし「存在しない」「不可能である」ことの証明は一般に難しい．ここに P ≠ NP の問いの深い困難がある．

また，P ≠ NP 予想は単に計算機科学の内部問題ではなく，数学の証明の仕方，暗号の安全性，最適化の限界など多方面に影響する．例えば現代暗号は「ある計算問題が現実的時間では解けない」ことに依存して設計されることが多く，P=NP が成立すると暗号の枠組みが根底から揺らぐと考えられている．

P ≠ NP の問いは 1950 年代から本質的な形で意識されていた．ゲーデルはフォン・ノイマンへの手紙（図 1）の中で，「証明の探索を機械化した場合に，どれほど速く見つかるか」という趣旨の問いを述べている．また，ナッシュも 1955 年に NSA 宛ての手紙（図 2）で，同様の計算困難性の重要性に触れているとされる．こうした背景は，P ≠ NP が単なる技術的問題ではなく，数学と計算の関係そのものに関わる問いであることを示唆している．

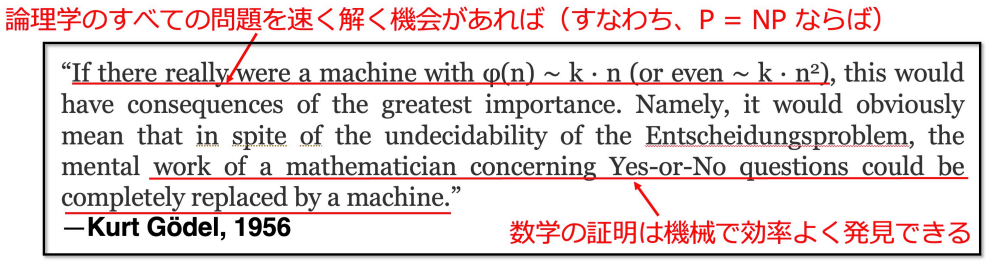


図 1: ゲーデルの手紙の内容

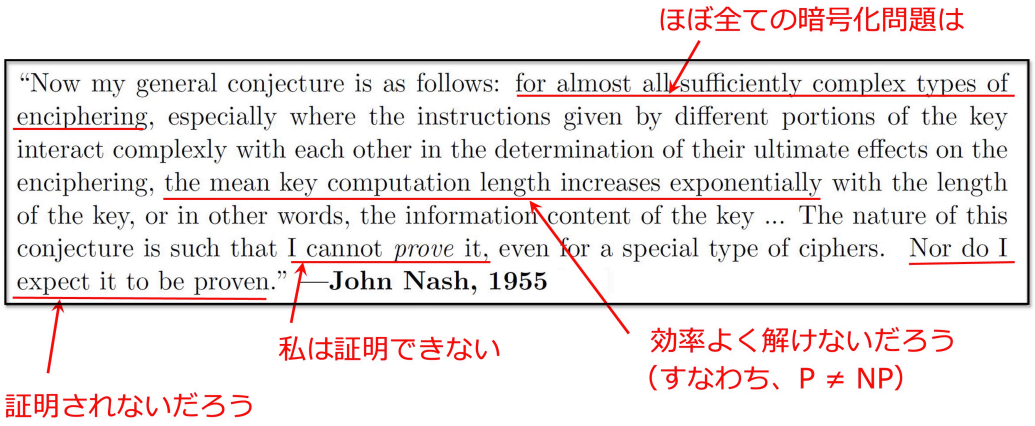


図 2: ナッシュの手紙の内容

2　コンピュータの数学：計算モデルと計算時間

計算を数学的に扱うためには，計算そのものを形式化する必要がある．1936 年，テューリングはテューリングマシンという抽象的計算モデルを導入し，「計算可能性」の概念を明確に定式化した [11]．テューリングマシンは非常に単純な規則で動くが，現代の計算機が行える計算を本質的にすべて表現できると考えられている．

この形式化の価値は，計算ができる／できないという二分だけでなく，「どれくらい時間がかかるか」「どれくらいの資源（メモリなど）を使うか」という定量的評価を可能にした点にある．計算の複雑さ理論は，この「時間」や「資源」を数学的対象として扱う分野である．

簡単な例として，足し算と掛け算を考えよう．入力サイズ n を「桁数」として，必要なステップ数を見積もると，

- n 桁の足し算はおよそ $O(n)$ ステップ
- n 桁の掛け算はおよそ $O(n^2)$ ステップ

となる．もちろん実際には高速な掛け算アルゴリズムが存在し，より良い計算量が得られるが，ここで重要なのは「入力が大きくなると計算時間がどのように増えるか」という見方である．

このような見積もりにより，多項式時間で解ける問題は入力が増えても計算時間が比較的穏やかに増えると期待できる．一方，指数時間（例えば 2^n）が必要な問題では，入力が少し増えただけで計算時間が爆発的に増加する．

次に，素因数分解の計算について考えよう．素因数分解は「与えられた整数を素数の積に分解する」という基本的な問題であり，古典計算においては，一般に非常に難しいと考えられている．（正確には，最良の既知アルゴリズムは準指数時間であり，多項式時間アルゴリズムは知られていない．）暗号（RSA など）はこの困難性に依存しているため，素因数分解は計算の複雑さ理論と社会的応用の接点としても重要である．

3　さまざまな問題：数独・クリーク・部分和

数独は多くの人に親しまれたパズルだが，数学的には興味深い性質をもつ．通常の数独（9×9）は人間向けに設計されているため解けるが，サイズを大きくして 16×16，25×25，一般に $N \times N$（ただし N は平方数）へ拡張すると状況が一変する（図 3）．

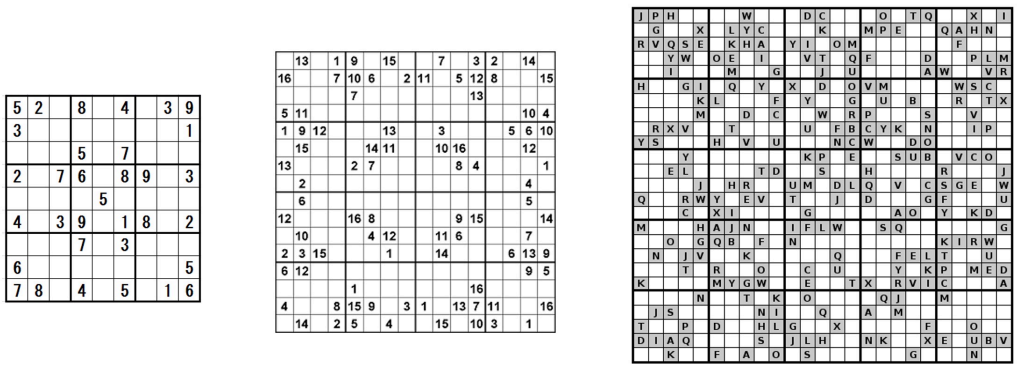
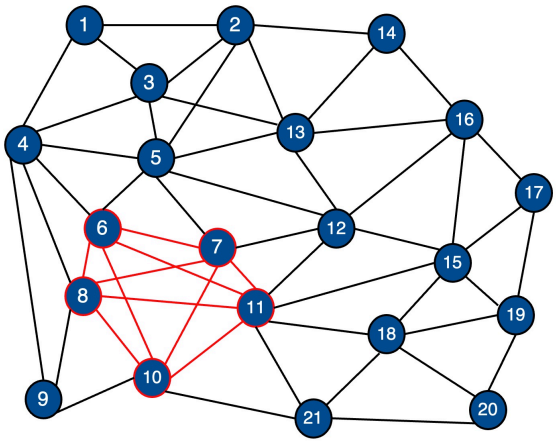


図 3: 通常の数独（左），16×16 数独（中），25×25 数独（右）

重要なのは，数独の解が与えられたとき，それが正しいかどうかの確認は簡単である点である．行・列・ブロックごとに条件をチェックすればよいので，検証は多項式時間で行える．しかし，解を見つける作業は一般には爆発的に難しくなると考えられている．この「検証は簡単だが探索が難しい」構造が，NP の典型的な特徴である．

次に，LINE などの友だち関係を抽象化して考える．人を頂点，友だち関係を辺とすることで，社会的ネットワークはグラフとして表現できる．このとき，あるグループが「互いに

全員が友だち」であることを数学的に表すと，それはグラフのクリーク（clique）に対応する．例えば，次のグラフにおいて，



頂点 $\{6, 7, 8, 10, 11\}$ はサイズ 5 のクリークをなす．クリーク問題の一例として，

「グラフの中に，サイズ k のクリークが存在するか？」

という判定問題を考える．特定の頂点集合が与えられれば，それがクリークであるかは各辺の有無を調べればよく，検証は容易である．しかし，一般のグラフに対して「どこかにクリークがあるか」を探すのは，組合せが膨大であるため難しい．この問題も NP の代表例の一つである．

部分和問題（Subset Sum）も典型例である．整数 $a_1, \ldots, a_m$ が与えられたとき，

「いくつかを選んで和を T にできるか？」

を問う．例えば，$T = 1,000,000$ の場合，次の 38 個の数が与えられたとき，下線を引いた 19 個の数の和がちょうど T になる．

14,175	19,300	26,343	41,867	58,306	69,189	82,027	97,042
15,055	19,731	28,725	43,155	61,848	72,936	82,623	97,507
16,616	22,161	29,127	46,298	65,825	74,287	82,802	99,564
17,495	23,320	32,257	56,734	66,042	74,537	82,988	
18,072	23,717	40,020	57,176	68,634	81,942	90,467	

どの要素を選ぶかは 2^m 通りあるため，素朴に全探索すると指数時間がかかる．一方，解（選び方）が与えられれば足し算で確かめられるので検証は容易である．このように，NP に現れる難しさはしばしば「組合せ爆発」として理解できる．

4　計算の複雑さの理論：NP 完全性

1970 年代に，クック，レヴィン，カープらによって NP 完全性理論が確立された [4, 8]．NP 完全問題とは，ざっくり言えば

NP の中で「最も難しい」問題

である．より正確には，NP 完全問題 X は

- $X \in$ NP（検証が多項式時間でできる）
- 任意の $Y \in$ NP が多項式時間で X に帰着できる

という性質をもつ．帰着とは「Y の入力を多項式時間で変換して X の入力にし，X が解ければ Y が解けるようにする」操作である．

この理論のインパクトは大きい．もし何か一つでも NP 完全問題が多項式時間で解ければ，NP のすべての問題が多項式時間で解ける．つまり P=NP となる．逆に，P $\neq$ NP を示すには NP 完全問題のどれか一つが多項式時間では解けないことを示せばよい（ただし，これが難しい）．

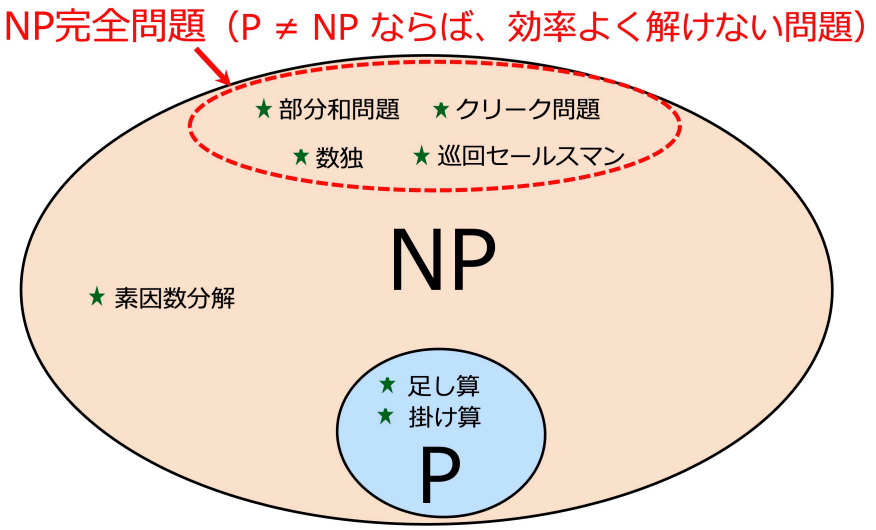


図 4: P $\neq$ NP のイメージ図

クリーク問題と部分和問題の NP 完全性は 1972 年にカープにより証明された後，様々な問題の NP 完全性が示されてきた．数独の NP 完全性も 2003 年に証明された [12] ため，先ほど述べた数独が難しいという主張は，NP 完全性理論により数学的に裏付けられる．もち

ろん，これは「実際に絶対に解けない」という意味ではない．入力が小さいなら解けるし，工夫により多くの実例は解ける．しかし「入力が大きくなるにつれて一般の場合には効率的解法がない」と理解すべきである．

P と NP の関係について，直観的な図としては「P が NP の真部分集合である」という像がしばしば描かれる (図 4)．すなわち，NP には「検証は簡単でも探索は難しい」問題が多数含まれ，その典型が NP 完全問題群である，という理解である．

5　近似の難しさ：巡回セールスマン問題

巡回セールスマン問題（Traveling Salesman Problem, TSP）は，1930 年代から研究されている代表的最適化問題である．都市と都市の距離が与えられたとき，

> 「すべての都市を一度ずつ訪れて出発点に戻る巡回路のうち，総距離が最小のものを求めよ」

という問題である．

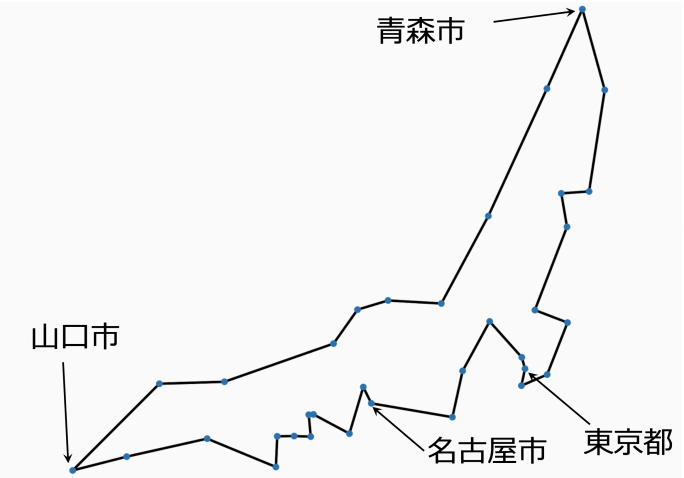


図 5: 日本（本州）の全県庁所在地の最短巡回路（ソフトウェア ConcordeTSP による計算）

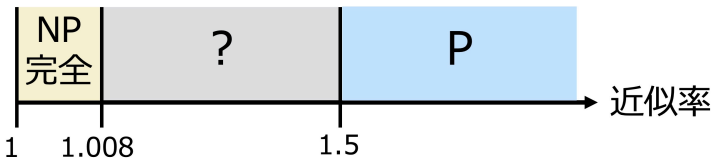
例えば日本（本州）の各都道府県の県庁所在地をすべて巡る最短巡回路（図 5），あるいはドイツの 15,112 都市規模で最短巡回路 [1] を求める，さらには製造工程（マイクロチップ製造等）での経路最適化など，TSP は実問題としても頻出する．しかし一般には TSP は NP 完全（より正確には NP 困難）であり，最適解を求めるのは難しい．

ここで重要になるのが「近似解」である．最適解が難しいなら，最適値の $1+\varepsilon$ 倍以内の解を効率よく求められないか，という発想である．例えば対称距離（距離が三角不等式を満たす）を仮定すると，TSP には有名な 1.5 近似アルゴリズムが存在する（クリストフィデスのアルゴリズム [2]）．

一方で，1990 年代以降の研究で，

近似であっても一定以上良い近似は難しい

という現象が，数学的に明確になってきた．すなわち，NP 完全性理論は「最適解が難しい」ことを述べるが，近似困難性理論は「ほどほどの精度であっても難しい」ことを述べる．特に，TSP の近似において，最適解の 1.008 倍以内の解を求めることは NP 完全（正確には NP 困難）であることが知られており [9]，次のような状況になっている．



近似率が 1.008 と 1.5 の間において，多項式時間の近似アルゴリズムが存在するかどうかは，長年の未解決問題である．

補足であるが，最近の論文 [5] では，TSP の 1.4999999999999999999999999999999999999（$= 1.5 - 10^{-36}$） 近似を求める多項式時間アルゴリズムが示された．1.5 は長年「超えられない壁」と考えられてきたので，それを（ごくわずかでも）理論的に下回ったこと自体が非常に重要で，大きなブレークスルーである．

一方で，TSP は長い研究史の中で着実に進歩してきた．非常に大規模な実例に対しても，巧妙な工夫で最適解に近い解が得られる場合がある．ここには理論（最悪ケース解析）と実務（典型例での性能）のギャップも存在し，このギャップ自体が興味深い研究対象となっている．

6　量子計算と量子アルゴリズム

1980 年代に，マニンやファインマン，ドイッチュらが量子計算の基本概念を提案した．ファインマンが強調したのは，量子力学的な現象を古典コンピュータでそのままシミュレートしようとすると，状態数が指数的に増え，計算資源が爆発するという点である [3]．ならば，量子力学そのものを計算機の原理として利用すればよい，というのが量子計算の出発点である．

量子計算が世に広く注目されるきっかけとなったのは，1994 年のショアのアルゴリズムである [10]．ショアは，素因数分解という重要問題が**量子コンピュータ上で多項式時間で解ける**ことを示した．この結果は，古典計算の世界では「おそらく難しい」と信じられていた問題が，計算モデルを変えることで一気に「効率的に解ける」側に移る可能性を示した点で画期的である．素因数分解以外にも，線形代数における様々な問題 [7] や化学におけるエネルギーの計算 [6] に関して，量子計算が古典計算を凌駕することが知られている．

もちろん，これは P ≠ NP 予想を直接解決するものではない．量子計算により効率よく解ける問題のクラス（BQP と呼ばれる）は，NP 完全問題を含まないと一般に信じられている（図 6）．すなわち，量子コンピュータは NP 完全問題を効率よく解けるとは考えられていない．しかし「計算の難しさは計算モデルに依存する」こと，そして新しい計算モデルが暗号や最適化の景色を変え得ることを強く印象づけた．

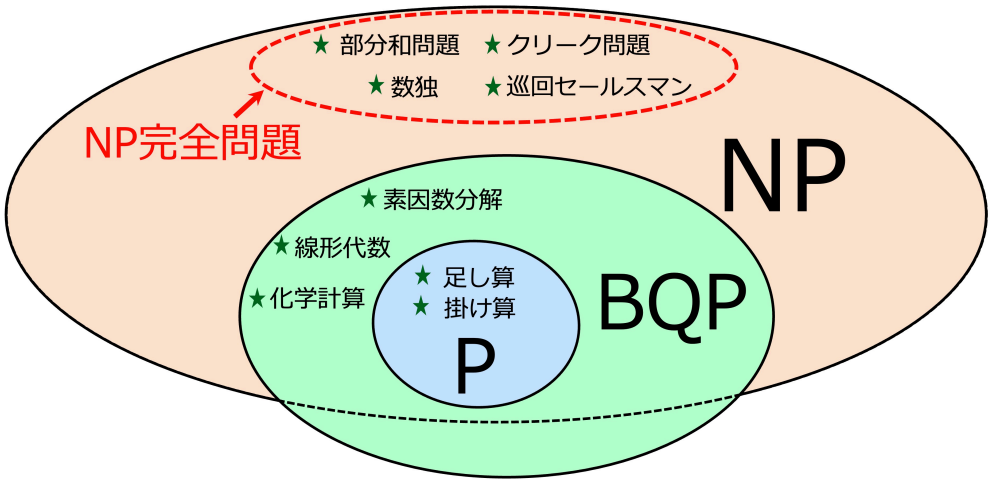


図 6: P，NP，BQP のイメージ図

1990 年代後半から 2000 年代にかけて「第一次量子計算ブーム」と呼ばれる盛り上がりがあった．その後も実験技術が進展し，「第二次量子計算ブーム」に入り，現在では複数の企業・研究機関が量子プロセッサを開発している（図 7）．ただし，現状の量子コンピュータはノイズが大きく，量子ビット数も限定されるため，実用的な規模でショアのアルゴリズムを実行するには多くの課題が残っている．

量子計算の能力を評価するには，何が「量子でしかできない」あるいは「量子が圧倒的に有利」なのかを理論的に理解する必要がある．計算の複雑さ理論は「どの問題でどの程度の優位性が期待できるか」「必要な資源は何か」を明確にする役割を担う．

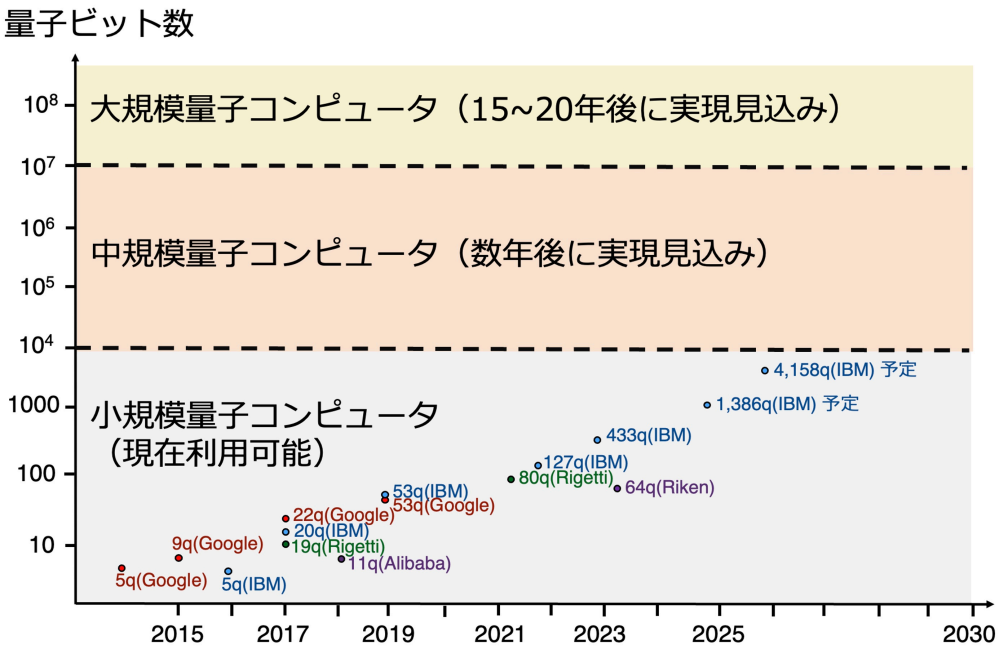


図 7: 量子コンピュータの開発の現状と展望

7　おわりに

本稿では，計算を数学的に捉える視点から，P ≠ NP 予想，NP 完全性，近似困難性，量子計算という流れを概観した．強調したいのは，計算機科学が単なる工学ではなく，**計算という現象を数学として研究する分野**であるという点である．そして量子計算は，計算モデルを拡張することで「何が容易で何が困難か」という境界を塗り替える可能性を示した．

今後，量子コンピュータがどの程度の規模で実現し，どの領域で本質的な利点をもたらすかは，理論・実験の両面からの研究に支えられて明らかになっていく．計算の複雑さ理論は，その過程で重要な指針を与え続けると期待される．

参考文献

[1] D. Applegate, R. Bixby, V. Chvátal, W. Cook. Implementing the Dantzig–Fulkerson–Johnson algorithm for large traveling salesman problems. *Mathematical Programming*, 97, 91–153, 2003.

[2] N. Christofides. “Worst-case analysis of a new heuristic for the travelling salesman problem,” Report 388, Graduate School of Industrial Administration, CMU, 1976.

[3] R. P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21, 467–488, 1982.

[4] S. A. Cook. The complexity of theorem-proving procedures. *Proceedings of the 3rd annual ACM symposium on Theory of computing (STOC 1971)*, 151–158, 1971.

[5] A. R. Karlin, N. Klein, S. O. Gharan. A (slightly) improved approximation algorithm for metric TSP. *Proceedings of the 53rd annual ACM symposium on Theory of computing (STOC 2021)*, 32–45, 2021.

[6] S. Gharibian, F. Le Gall. Dequantizing the quantum singular value transformation: Hardness and applications to quantum chemistry and the quantum PCP conjecture. *Proceedings of the 54th ACM Symposium on Theory of Computing (STOC 2022)*, pp. 19–32, 2022.

[7] A. W. Harrow, A. Hassidim, S. Lloyd. Quantum algorithm for linear systems of equations. *Physical Review Letters. 103 (15) 150502*, 2009.

[8] R. M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, 85–103, 1972.

[9] C. H. Papadimitriou, S. Vempala. On the approximability of the Traveling Salesman Problem. *Combinatorica* 26, 101–120, 2006.

[10] P. W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. *SIAM Journal on Computing*, 26 (5): 1484–1509, 1997.

[11] A. M. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, s2-42: 230–265, 1937.

[12] T. Yato, T. Seta. Complexity and completeness of finding another solution and its application to puzzles. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E86-A (5): 1052–1060, 2003.