Algorithmic Manifold and Application to P versus NP Problem

Takuya Yabu(takuya.yabu@live.jp)

1. Abstract and Introduction

About P versus NP problem [1], it has been studied for long time. Recent literature [2] has shown that the existing proof method using the diagonal argument or the circuit complexity is not effective. On the other hand, as another approach, calculation of time complexity based on the geometric method is also performed [3], but it is limited to the quantum algorithm, and it is an application example to the existing method of lower band derivation of quantum circuit complexity, it is essentially unchanged.

In this paper, I introduce algorithmic manifolds that explain algorithms by geometric method and show that they are topologically homogeneous with respect to P versus NP problem. And I will also discuss polynomial-time reduction method of NP problem for class P.

2. Algorithmic manifold

Think about the Riemannian manifold defined by the following metric.

$$ds^2 = dn^2 + O\left(n^{k(l)}\right)dl^2 \quad (1)$$

Here, n is the data capacity to be input to the Turing machine to be processed, and l is the length of the algorithm of the Turing machine to be processed, and $O\left(n^{k(l)}\right)$ is the time complexity of the algorithm. $O\left(n^{k(l)}\right)dl$ means the computation time of the program. The Riemannian manifold defined by the above metric is called the algorithmic manifold below.

From equation (1), the Gauss curvature K is given by the following equation.

$$K = \frac{O\left(n^{(k(l)-1)^2}\right)}{O\left(n^{k(l)^2}\right)} > 0 \quad (2)$$

Here, consider the value of n. When n is a sufficiently large value n_s, $K \to 0$, and when $n = 0$, computational complexity is on the order of constant time, so if you set it to 1, $K = 1$. That is, in the R^2, since the value of the closed set of $[0, n_s]$ is bounded with $[0,1]$, the algorithmic manifold is compact and complete.

The Gauss curvature K of the algorithmic manifold is positive everywhere. Therefore, the integral value of the entire algorithmic manifold of the Gauss curvature K is also positive. Also, since the algorithmic manifold is compact and bounded, the algorithmic manifold is homeomorphic to 2-sphere from Gauss-Bonnet theorem.

3. The topological aspect of P versus NP problem

The algorithmic manifold representing the deterministic Turing machine for solving class P problems is homeomorphic to 2-sphere. Also, considering replicating a deterministic Turing machine for each branch, a non-deterministic Turing machine to solve the problems of class NP is nothing but replicating algorithmic manifolds for each branch. The connected sum of the plurality of algorithmic manifolds represents a non-deterministic Turing machine. Therefore, when looking at a non-deterministic Turing machine topologically, it is a connected sum for each branch of a 2-sphere, so that it is homeomorphic to 2-sphere as a result.

Therefore, class P and class NP are homeomorphic, that is,

$$\mathrm{P} = \mathrm{NP} \tag{3}$$

4. Parallel transport and geodesics on the algorithmic manifold

The algorithmic manifold is the Riemannian manifold. As a result, the metric on the algorithmic manifold is preserved and the torsion tensor is zero. Consequently, the inner product on the algorithmic manifold is preserved as a result of the parallel transport. That is, if the end points of the curve are set as the start point and the end point of the algorithm on the algorithm manifold, the parallel transport of the curve part can be shifted at the start point and the end point on the curve of another algorithm without changing the property of the problem. Therefore, the parallel transport is the same as polynomial-time reduction to another problem. In the case of class P, polynomial-time reduction is possible in all problems, so that $k(l)$ in equation (1) can be expressed by one function $k_{\mathrm{P}}(l)$. Also in the case of class NP, I can also convert polynomial time to NP-complete problem in all problems, so I can also represent it with one function $k_{\mathrm{NPc}}(l)$.

Also, the algorithmic manifold is also a compact and complete Riemannian manifold. Therefore, according to Hopf - Rinow theorem, there exists the length minimizing geodesic connecting any two points on the algorithmic manifold. Given that two points are the start and end points of the algorithm, the length minimizing geodesic represents the optimal algorithm. As for the class NP as well as the class P, if the start point and the end point of the algorithm are expressed on the algorithm manifold, the length minimizing geodesic can be determined, so it can be converted to the problem of the deterministic Turing machine. Also from this point, equation (3) holds.

5. Conclusion

I defined the Riemannian manifold with the metric of equation (1) as the algorithmic manifold, described the properties derived therefrom, and clarified that P = NP topologically. Also, parallel transport on the algorithmic manifold corresponds to polynomial-time reduction, indicating that the length minimizing geodesic exists between arbitrary starting and ending points on the algorithmic manifold, and for the case of class P, $k(l)$ in equation (1) can be expressed by one function $k_{\mathrm{P}}(l)$. Also in the case of class NP, I can convert it to NP-complete problem and showed that it can be represented by one function $k_{\mathrm{NPc}}(l)$. Finally, it was clarified that the length minimizing geodesic, that is, the optimal algorithm can be described by deterministic Turing machine with respect to the problem of class NP. The future task is to clarify the relation between $k_{\mathrm{P}}(l)$ and l, that is, the length of the algorithm and the algorithm, to determine the algorithm from $k_{\mathrm{P}}(l)$ and to determine $k_{\mathrm{P}}(l)$ from the algorithm. Thus, it can be shown that NP is actually converted to P.

References

[1] S. Cook, "The P versus NP Problem," http://www.claymath.org/sites/default/files/pvsnp.pdf, 2000.

[2] A. A. Razborov , S. Rudich, "Natural Proofs," In Proceedings of the 26th ACM Symposium on Theory of Computing, 1994.

[3] M. A. Nielsen, M. R. Dowling, M. Gu , A. C. Doherty, "Quantum Computation as Geometry," Science, 2006.