A Quantum Algorithm for Finding k-Minima

Kohei Miyamoto[1] Masakazu Iwamura[1] Koichi Kise[1]

[1]Department of Computer Science and Intelligent Systems, Graduate School of Engineering, Osaka Prefecture University

Introduction

Finding k-minima, which finds the k smallest from N data indices, is an important problem, as it can be applied to k-nearest neighbor search and other quantum machine learning methods such as clustering and classification. The primary difficulty of the problem is that it requires to return k answers. These machine learning methods are very important for analyzing big data. If the power of the quantum computer adapts to big data, big data that is hard to calculate on the classical computer can be analyzed.

Definitions of Symbols

D	Set of indices	Let D be the set of all indices of data. $\lvert D\rvert = N$.
f	Oracle	Let $f(x), x \in D$ be a function that returns 1 or 0. $f(x) = 0, 1$
M	Set of indices	A set of indices that satisfy $f(x) = 1$ (Marked).
g	Mapping function	Let $g(x)$ be a function that maps index x to the corresponding value.

Problem Definition of Finding k-Minima

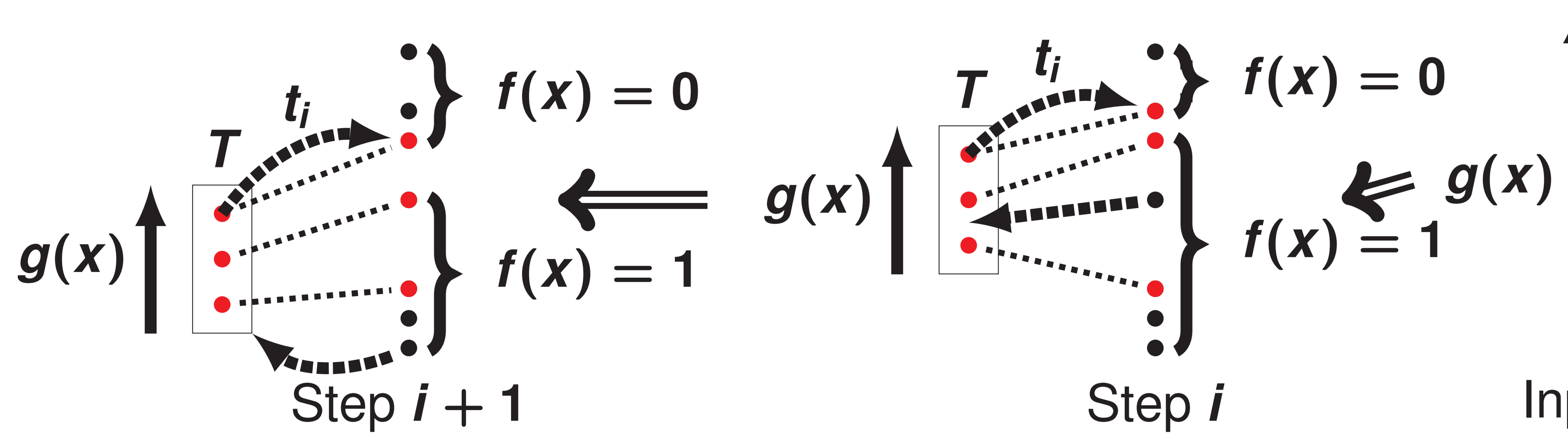
Input	:	Set of indices D.
Output	:	k indices from D that have minimum **values**.
Query Complexity	:	How many times oracle f is called.

Dürr's Algorithm (Existing Method)

Let T be a set of k threshold indices.

T is updated in a step-by-step manner and each updating step replaces the index that has the maximum value with the found index by AA. This is a kind of a greedy algorithm.

$$f_T(x) = \begin{cases} 0, & \text{if } g(x) \geq g(t) \text{ for some } t \in T, \\ 1, & \text{if } g(x) < g(t) \text{ for some } t \in T. \end{cases} \tag{1}$$



Query complexity is $\mathcal{O}(\sqrt{kN})$.

Proposed Algorithm

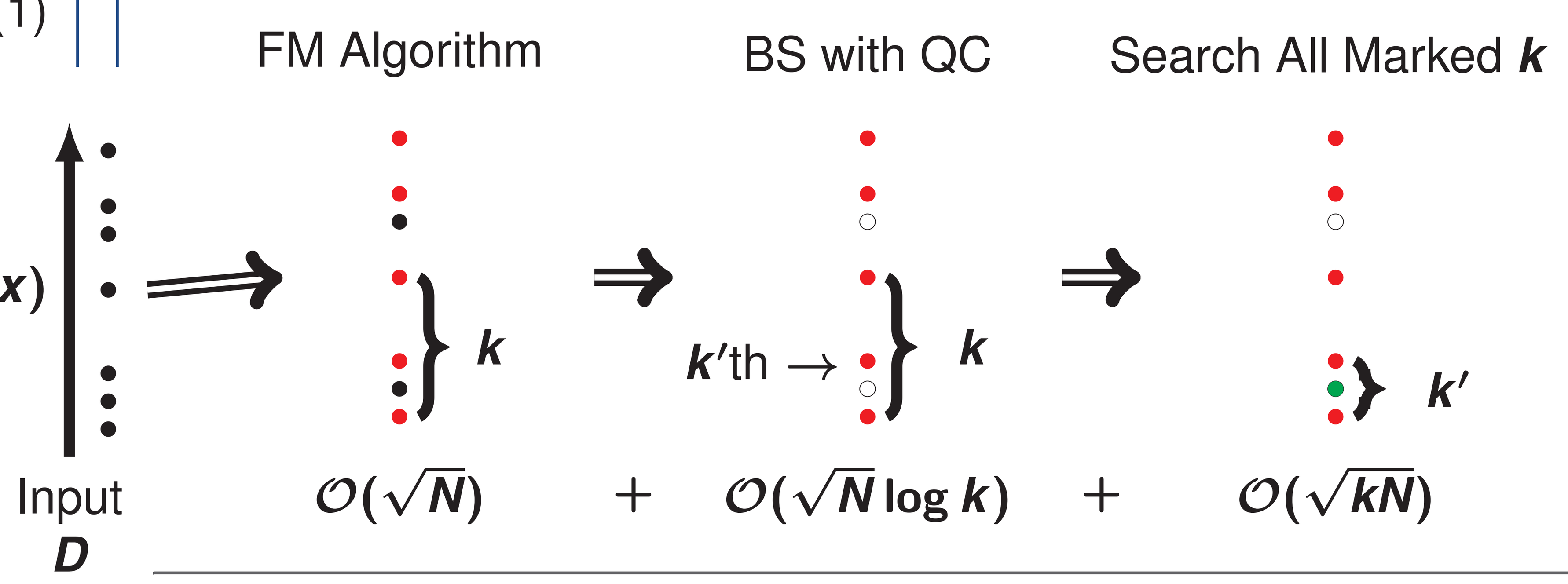
Let T be a already found indices.

Step 1 Set T to last k indices of FM algorithm's threshold indices.

Step 2 Search a k'th index by Binary Search (BS) that has the number of k' indices less than that value. The number is determined by Quantum Counting (QC).

Step 3 Search all k' indices less than the value of k'th index except for indices in T.

$$f'_{t_k}(x) = \begin{cases} 0, & \text{if } g(x) \geq g(t_k) \text{ or } x \in T, \\ 1, & \text{if } g(x) < g(t_k) \text{ and } x \notin T. \end{cases} \tag{2}$$



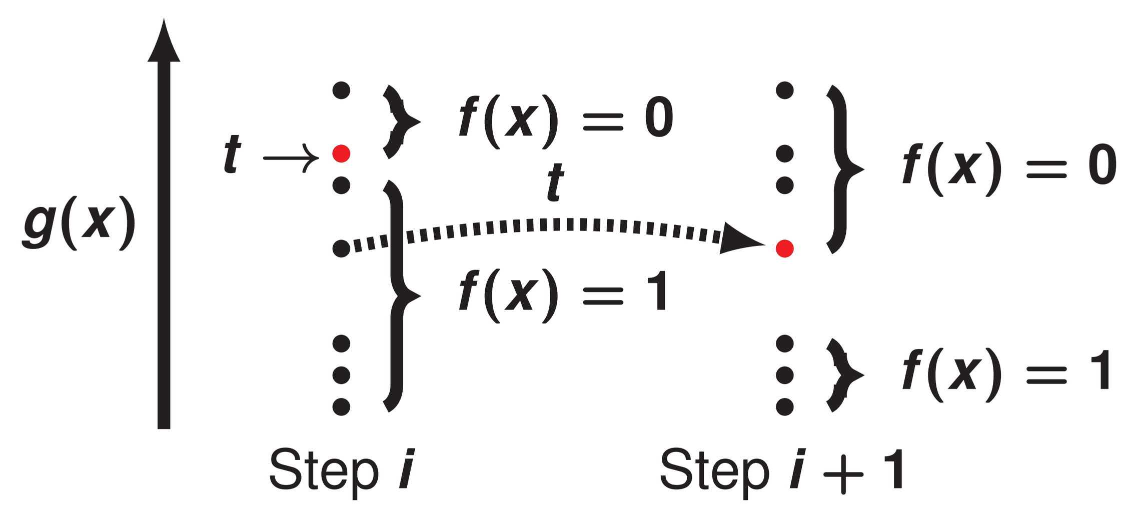
Query complexity is $\mathcal{O}(\sqrt{kN})$.

Finding Minimum Algorithm

Finding Minimum (FM) Algorithm finds the minimum data from N data with the query complexity $\mathcal{O}(\sqrt{N})$ by using AA. This complexity is smaller than that of the linear search in the classical computer ($\mathcal{O}(N)$).

$$f_t(x) = \begin{cases} 0, & \text{if } g(x) \geq g(t), \\ 1, & \text{if } g(x) < g(t). \end{cases} \tag{3}$$

Query complexity is $\mathcal{O}(\sqrt{N})$.



Search All Marked k-Indices Algorithm

Search All Marked k Indices Algorithm is an extension of Amplitude Amplification (AA). AA is a quantum database search algorithm and is a generalization of Grover's algorithm [1]. From N indices, AA searches one of k indices that satisfy some certain condition with the query complexity of $\mathcal{O}(\sqrt{N/k})$.

Query complexity of AA is $\mathcal{O}(\sqrt{N/k})$ where $\lvert M\rvert = k$.

By contrast, Search All Marked k Indices Algorithm searches all $i \in M$ from D.

$$\mathcal{O}_q\left(\sqrt{\frac{N}{k}} + \sqrt{\frac{N}{k-1}} + \cdots + \sqrt{N}\right) = \mathcal{O}_q(\sqrt{kN}), \tag{4}$$

$$\text{because } \sum_{t=1}^{k} \sqrt{\frac{1}{t}} < 1 + \int_1^k \sqrt{\frac{1}{t}}dt = 2\sqrt{k} - 1. \tag{5}$$

Query Complexity of Search All Marked k-indices is $\mathcal{O}(\sqrt{kN})$.

Contribution

We proposed a new *finding k-minima* algorithm and derived its query complexity. Our algorithm is easier to understand than Dürr's algorithm [2] and written in a clear form. Moreover, our algorithm consists of several steps and can stop the algorithm in each step. Finding k-minima algorithm can be applied to many kinds of algorithms or applications. For example, k-nearest neighbor search and k-nearest neighbor clustering and classification. If these methods solve problems faster than the classical computer, the data that can be analyzed is greatly increased.

References

L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, 1996, pp. 212–219.

C. Dürr, M. Heiligman, P. Høyer, and M. Mhalla, "Quantum query complexity of some graph problems," *SIAM Journal on Computing*, vol. 35, no. 6, pp. 1310–1328, 2006.