Dining Philosophers (Short Note)

David Ponarovsky

April 30, 2023

Abstract

In this paper, we improve a wide range of upper bounds for a variety of **NP** problems, by plugging Grover into the work made by [Fom+15]. We emphasize that this work has only a technical value and does not present any new idea. Nevertheless, we think it is worth sharing how easy and straightforwardly integrating quantum might be. We coin the term Groverize, harvesting a quantum improvement by doing almost nothing.

1 Todo.

1. Write the table (sage script). **[COMMENT]** all most done, it's left to handle the parameterized lines.
2. Add definitions. Problem description.
3. Complete the 'proof'.
4. Prove lower bound.
5. Add figures of covering the space by balls and cones.

2 Introduction.

The Dining Philosophers problem is a classic synchronization problem in computer science, where a group of philosophers sit at a round table and alternate between thinking and eating. The problem arises when they share common resources (forks) and a set of rules must be established to prevent deadlocks and starvation.

The impossibility result for the deterministic case states that it is impossible to design a solution to the Dining Philosophers problem if the philosophers behave deterministically and the resource allocation is symmetric. This is because each philosopher would require the same resources as their neighbor at the same time, which leads to deadlocks.Michael Rabin proposed a solution to the randomized case, where the philosophers behave randomly in choosing which fork to pick up first. This randomization breaks the symmetry and prevents deadlocks.

Attempts have also been made to solve the problem using quantumness, with **[COMMENT]** Adi Shamir and Avi Wigderson proposing a quantum analog to the Dining Philosophers problem. In **[COMMENT]** 2003, Dorit Aharonov proposed a quantum solution to the problem, which involves using entanglement to share the forks between philosophers.

An example of a real-world application of the Dining Philosophers problem is in resource allocation in computer networks, where multiple nodes may need access to a shared resource. The problem can also be used as a teaching tool in computer science to illustrate the importance of synchronization and avoiding deadlocks.

An incident related to the problem occurred in **[COMMENT]** 2008, when a bug in the synchronization code of the iPhone's email application caused it to hang, leading to a flurry of frustrated complaints from iPhone users dubbed the "Dining Philosopher's bug".

Definition 1 (Parameterized Computable)**.** *Let L be a problem family, and use the notation $P \in L$ to indicate that P is an instance of L (e.g., if L is a 3-SAT problem), and by $|P|$ to denote the length of the binary string encoding P. L is said to be parameterized computable if there exists a mapping $\phi : L \to \mathbb{N}$ and an algorithm $\mathcal{A}$ such that:*

1. *$\phi(P) \in [|P|]$ for any $P \in L$*
2. *$\mathcal{A}$ returns on P at running time $\mathcal{O}\left(c^{\phi(P)} \cdot Poly(|P|)\right)$ for a fixed c which does not depend on P.*

References

[Fom+15] Fedor V. Fomin et al. *Exact Algorithms via Monotone Local Search*. 2015. arXiv: 1512.01621 [cs.DS].