# Staggered Quantum Algorithm for Element Distinctness

**Alexandre S. Abreu[1], Luis A.B. Kowada[2], Franklin L. Marquezino[1]**

Programa de Engenharia de Sistemas e Computação, Coppe - UFRJ[1]
Instituto de Computação - UFF[2]
(santiago, franklin)@cos.ufrj.br[1], luis@ic.uff.br[2]

## Abstract

*The element distinctness problem consists in finding, given a list of elements, if there is some collision (two or more equal elements). The quantum algorithm for element distinctness was first proposed by Ambainis and uses important concepts of quantum walks. Ambainis' algorithm performs a quantum walk over the edges of a Johnson graph while searches for a marked vertex. Szegedy generalized this process into a quantum walk model on bipartite graphs, leading to the formulation of other important algorithms. Recently, Portugal et al. introduced the staggered quantum walk, a novel quantum walk model which includes Szegedy's model as a particular case. In this work, we reformulate the element distinctness quantum algorithm using a staggered quantum walk. We show that our new formulation requires less memory than the original one while keeping the same query complexity.*

## 1. Introduction

**The problem:** In a given list of indices $[N] = 0, 1, ..., N-1$, where each index represents an element in $[M]$, we have the following considerations:

1. **Element Distinctness.** Given elements $x_0, x_1, ..., x_{N-1} \in [M]$, are there indices $i, j \in [N]$ such that $i \neq j$, and $x_i = x_j$ ?
2. **Element *k*-Distinctness.** Given elements $x_0, x_1, ..., x_{N-1} \in [M]$, are there $k$ indices $i_1, i_2, ..., i_k \in [N]$ such that $i_1 \neq i_2 \neq ... \neq i_k$ and $x_1 = x_2 = ... = x_k$ ?

**Quantum Walks (QW):** Describes the movement of a particle through nodes of a graph [2]. This walk must be reversible and deterministic, as ruled by the postulates of quantum mechanics. Two models are particularly important for this work: Szegedy's model [7] and Staggered model [6].

**Ambainis' algorithm for element distinctness:** Uses concepts of QW and quantum search in graphs to perform a search for a marked vertex on a Johnson graph, requiring $O(r(\log N + \log M))$ qubits of memory [3], and $O(N^{k/k+1})$ queries, where $k$ is the number of collisions in the list.

**Belovs' algorithm for element distinctness:** This quantum algorithm is based in learning graphs and requires $O(N^{1-2^{k-2}/(2^k-1)})$ queries, being more efficient than Ambainis' algorithm [4].

**Staggered quantum algorithm for element distinctness:** Our proposal uses the Staggered quantum walk model [6]. This algorithm has the same complexity of Ambainis' original proposal, however it has the advantage of requiring just $O(r \log N)$ qubits of memory [1].

## 2. Staggered Quantum Walk Model

**Definition 1.** *Let $C$ be a set of cliques of $\Gamma$. We say that $C$ is a* set of disjoint maximal cliques of $\Gamma$ *if the cliques of $C$ are pairwise disjoint and the inclusion of any vertex $v \in V(\Gamma)\backslash C$ to some clique $K \in C$ turns it into a non-complete graph—that is, $K \cup \{v\}$ is not a clique of $\Gamma$—or causes the cliques of $C$ are no longer pairwise disjoint—that is, $(K \cup \{v\}) \cap K' \neq \varnothing$, for some $K' \in C$ such that $K' \cup \{v\} \neq K$. If $C$ is an inclusion-wise maximal set of disjoint maximal cliques of $\Gamma$—that is, the inclusion of any clique $K$ of $\Gamma$ into $C$ turns it a non-disjoint maximal set of cliques of $\Gamma$—, then we say that $C$ is a* tessellation *of $\Gamma$. In the context of the Staggered Quantum Walk model, we call each clique in $C$ a* polygon.

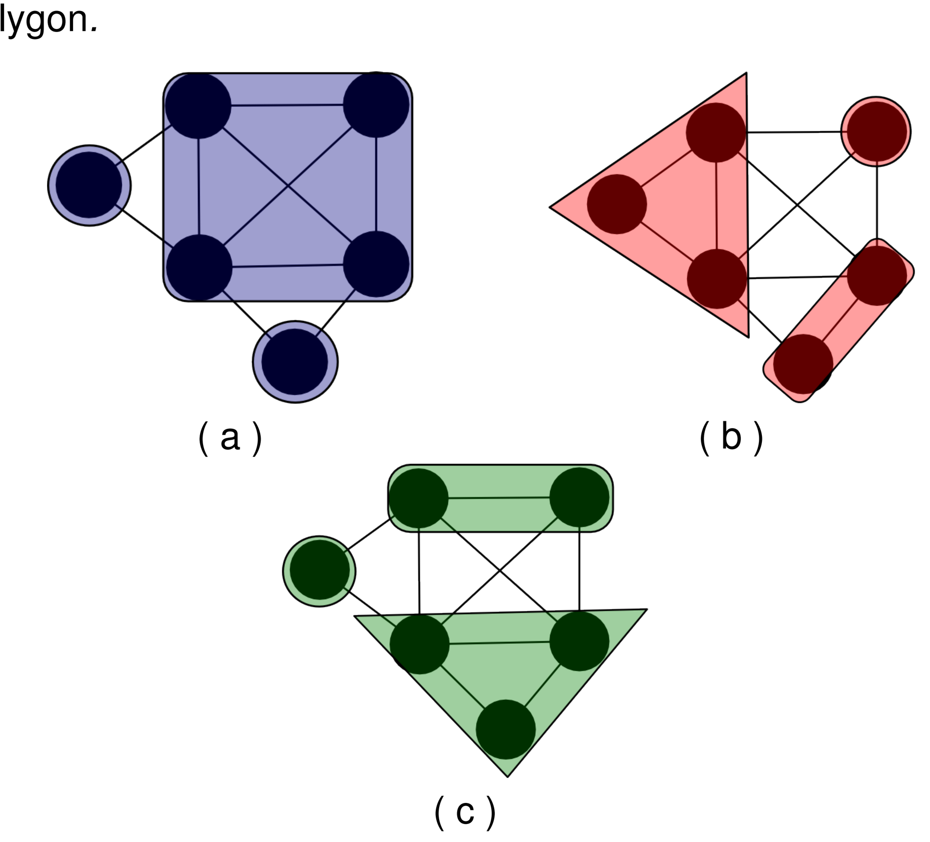

**Figure 1:** *Different tessellations of the same graph. Each tessellation is a set of disjoint-maximal cliques in the graph. Notice that in each tessellation, every vertex is covered by a polygon.*

**Definition 2.** *Let $C_1, C_2, ..., C_T$ be tessellations of $\Gamma$. We say that $\Gamma$ is $T$-tessellable if and only if $C_1, C_2, ..., C_T$ is the smaller family of tessellations in $\Gamma$ such that $C_1 \cup C_2 \cup ... \cup C_t$ covers all vertices and all edges of $\Gamma$.*

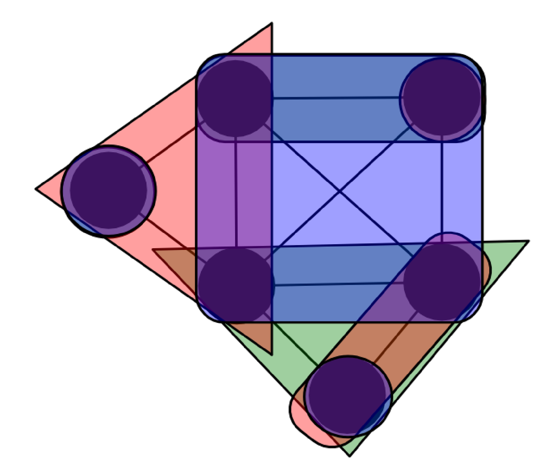

**Figure 2:** *A graph $\Gamma$ covered by tessellations present in Fig 1. This graph is 3-tessellable because if we remove any tessellation, the graph will not be fully covered.*

Polygons of different tessellations necessarily have at least one vertex in common, and each polygon defines a unit vector in a Hilbert Space $\mathcal{H}^N$. Each tessellation defines a unitary orthogonal reflexive operator as

$$U_i = 2\left(\sum_{k=0}^{m_i-1} |\varphi_i\rangle\langle\varphi_i|\right) - I, \tag{1}$$

where

$$|\varphi_i\rangle = \sum_{j \in \psi_i} \psi_{i,j}|j\rangle, \tag{2}$$

and $m_i$ is the number of polygons in tessellation $C_i$, and $\psi_{i,j}$ are nonzero complex amplitudes of the unit vector $|\varphi_i\rangle$, that represents a polygon in $\Gamma$ with vertices where $j$ is a vertex in the polygon represented by the unit vector $|\varphi_i\rangle$.

**Proposition 1.** *[5] A graph is two-tessellable if and only if its clique graph is two-colorable.*

## 3. Staggereg quantum algorithm for Element Distinctness

Let define $r = \lfloor N^{k/k+1} \rfloor$, where $k$ is the number of collisions that we have in the starting list. In our approach we have two sets represented by $S$ and $y$, where $S \subseteq [N]$ with size $r$ and $y \subseteq [N] \backslash S$ with unit size. $S$ represents the indices of the elements in our list. For each $S$ we have $N - r$ unitary sets $y$ associated. We have a graph $\Gamma$ where each vertex $v = (S, y)$, *i.e.*, we have $N - r$ vertices with the same set $S$, but differing by sets $y$.

**Definition 3.** *We define a graph $\Gamma$ with $\binom{N}{r}(N-r)$ vertices. A vertex $v$ corresponds to $(S, y)$. There are an edge between two vertices $v$ and $v'$, for $v = (S, y)$ and $v' = (S', y')$, if and only if (i) $S' = S$ and $y \neq y'$, or (ii) $S' = S \cup \{y\} \backslash \{y'\}$.*

**Definition 4.** *We can define two tessellations in graph $\Gamma$. The first tessellation, called $\alpha$-tessellation, is defined by polygons that cover cliques, where for every pair of vertices $(v, v')$ such that $v = (S, y)$ and $v' = (S', y')$, we have $S = S'$ and $y \neq y'$. The second tessellation, called $\beta$-tessellation, is defined by polygons that cover cliques, where for every pair of vertices $(v, v')$, such that $v = (S, y)$ and $v' = (S', y')$, we have $S' = S \cup \{y\} \backslash \{y'\}$.*

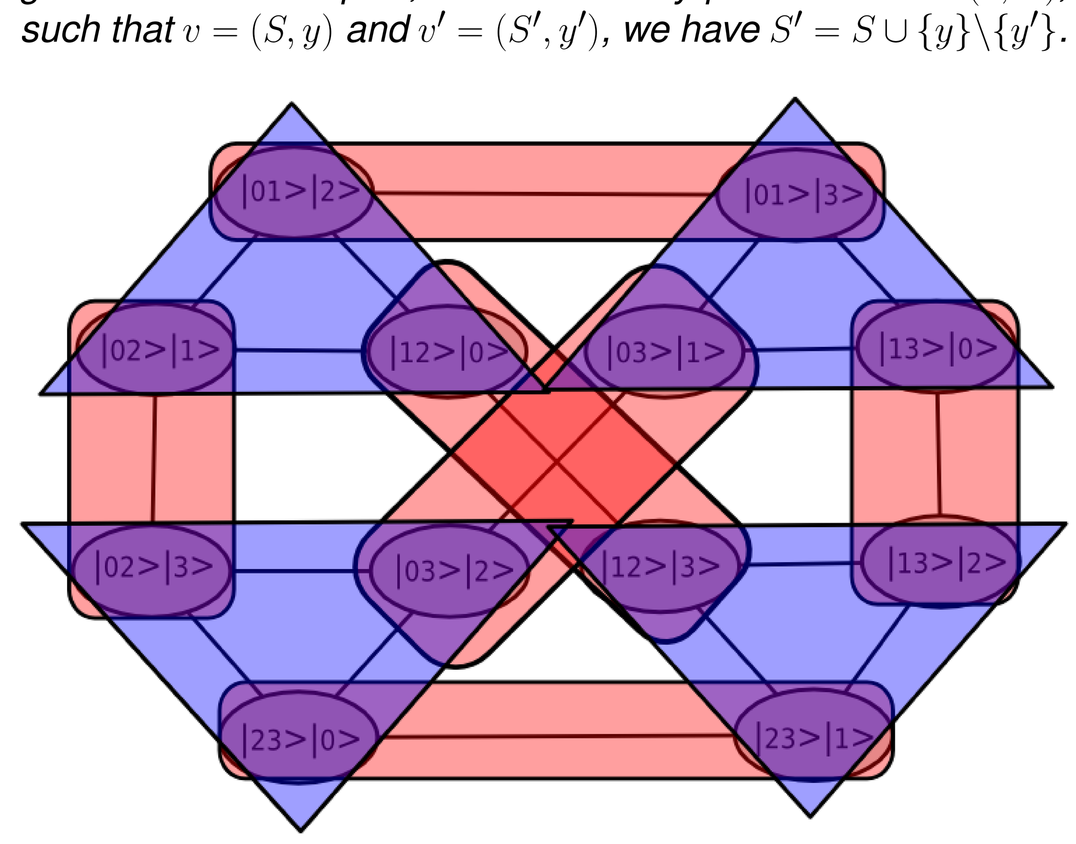


**Figure 3:** *Graph $\Gamma$ generated from $[N] = \{0, 1, 2, 3\}$, where $k = 2$, $N = 4$ and $r = 2$. We have $\alpha$-tessellation in red and $\beta$-tessellation in blue.*

The two operators generated by tessellations $\alpha$ and $\beta$, respectively, are

$$U_0 = 2\Bigg[\sum_{i=0}^{\binom{N}{r}-1}\Bigg(\frac{1}{N-r}|S_i, y\rangle\langle S_i, y| + \sum_{\substack{y' \neq y \\ y' \notin S}} \frac{1}{N-r}|S_i, y'\rangle\langle S_i, y'|\Bigg)\Bigg] - I \tag{3}$$

and

$$U_1 = 2\Bigg[\sum_{i=0}^{\binom{N}{r}-1}\Bigg(\frac{1}{r+1}|S_i, y\rangle\langle S_i, y| + \sum_{\substack{y' \neq y \\ y' \in S}} \frac{1}{r+1}|S_i', y'\rangle\langle S_i', y'|\Bigg)\Bigg] - I, \tag{4}$$

where $|S_i'\rangle = |S_i \cup \{y\} \backslash \{y'\}\rangle$.

Our algorithm uses just $O(r \log N)$ qubits of memory, corresponding to a Hilbert Space of dimension $\binom{N}{r}(N-r)$.

**Proposition 2.** *The graph $\Gamma$ constructed by Def. 3 can be cast into a bipartite Johnson graph in Szegedy's model.*

**Proposition 3.** *A graph $\Gamma$ constructed by Def. 3 is two-tessellable.*

**Proposition 4.** *A graph $\Gamma$ constructed by Def. 3 and Def. 4 always has at most one vertex in intercession of two polygons $\alpha_i$ and $\beta_j$ of different tessellations $\alpha$ and $\beta$, respectively.*

**Algorithm 1:** *Element k-distinctness Algorithm (Single-solution)*

1. Generate the uniform superposition
$$\frac{1}{\sqrt{\binom{N}{r}(N-r)}} \sum_{|S|=r, y \notin S} |S, y\rangle.$$
2. $t_1 = O((N/r)^{k/2})$ times repeat:
   (i) Apply the conditional phase flip ($|S, y\rangle \rightarrow -|S, y\rangle$), such that $x_{i_1} = ... = x_{i_k}$ for $i_1 \neq ... \neq i_k$, and $i_1, ..., i_k \in S$.
   (ii) Apply $U_1 U_0$ $t_2 = O(\sqrt{r})$ times.
3. Measure the final state. Check if S contains a $k$-collision and answer "There is a $k$-collision" or "There is no $k$-collision" according to the result.

## Acknowledgment

A.S.A. thank CNPq for financial support. F.L.M. acknowledge financial support from CNPq/Universal, CNPq/PQ and CAPES/PCT.

## References

[1] A. S. Abreu, L. A. B. Kowada, and F. L. Marquezino. Staggered quantum algorithm for element distinctness. Work in progress. Available in http://www.cos.ufrj.br/ franklin/workinprogress.html, 2016.

[2] Y. Aharonov, L. Davidovich, and N. Zagury. Quantum random walks. *Physical Review A*, 48(2):1687, 1993.

[3] A. Ambainis. Quantum algorithm for element distinctness. *SIAM Journal on Computing*, 37(1):210—-239, 2007.

[4] A. Belovs. Learning-graph-based quantum algorithm for k-distinctness. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 207–216, Oct 2012.

[5] R. Portugal. Staggered quantum walks on graphs. *arXiv preprint arXiv:1603.02210*, 2016.

[6] R. Portugal, R. Santos, T. Fernandes, and D. Gonçalves. The staggered quantum walk model. *Quantum Information Processing*, 15(1):85–101, 2016.

[7] M. Szegedy. Quantum speed-up of markov chain based algorithms. In *Foundations of Computer Science, 2004. Proceedings. 45th Annual IEEE Symposium on*, pages 32–41. IEEE, 2004.