INSTITUTO DE COMPUTAÇÃO
UNIVERSIDADE ESTADUAL DE CAMPINAS

A fast quantum algorithm for the closest bichromatic pair problem

Nilton Volpato *Arnaldo Moura*

Technical Report - IC-10-03 - Relatório Técnico
January - 2010 - Janeiro

A fast quantum algorithm for the closest bichromatic pair problem

Nilton Volpato* Arnaldo Moura†

Abstract

We present an algorithm for solving the two-color *bichromatic closest pair* problem using $O(N^{1/2}M^{1/4}\log M \log N)$ queries if $N \leq M \leq N^2$, or $O(M^{1/2}\log^2 N)$ if $M > N^2$. This result contrasts with the classical probabilistic time complexity of $O((NM \log N \log M)^{2/3} + M \log^2 N + N \log^2 M)$. We also show how to solve the *closest pair* problem—that is a special case of the *bichromatic closest pair* problem—using $O(N^{3/4}\log^2 N)$ queries. And, we also show a quantum lower bound of $\Omega(N^{2/3})$ queries for this problem, and discuss some open issues.

1 Introduction

In this article we present a new, and faster than classical, quantum algorithm for the two-color *bichromatic closest pair* and *closest pair* problems. We also provide a lower bound for the *closest pair* problem. Some of these results also appeared in a simplified form in [13].

The *closest pair* problem is defined as follows. Here $[N]$ stands for the set $\{1, \ldots, N\}$.

Definition 1 (Closest pair problem) *Given N unique points $P = \{p_i : i \in [N]\}$ in k-dimensional space and a distance function $d : P \times P \to \mathbb{R}$, find a pair of points which are closest to each other.*

The *closest bichromatic pair*, defined below, is a generalization of the *closest pair* problem. By providing an algorithm for the former we also provide an algorithm for the latter.

Definition 2 (Closest bichromatic pair problem) *Given a set of N unique points $P = \{p_i : i \in [N]\}$ and M unique points $Q = \{q_i : i \in [M]\}$ in k-dimensional space, and a distance function $d : (P \cup Q) \times (P \cup Q) \to \mathbb{R}$, find a pair of points $(p, q) \in P \times Q$ which are closest to each other.*

The points in P and Q may be regarded as colored with different colors, therefore the name *closest bichromatic pair*. The problem may be viewed as finding a point p and a point q that have different colors, such that the distance between p and q is minimum among all such bichromatic pairs.

*Research supported by CNPq grant: 140756/2004-3

†Research supported by CNPq grant: 471460/2004-3

Note that we only consider the problem where each point is colored by one of two colors. In the general *closest bichromatic pair* problem each point may be labeled with a color chosen from a given set, which may contain more than two colors.

The *closest pair problem* is a fundamental one in many applications, and is also a key step in many algorithms, being widely studied in classical computational geometry problems. The *closest bichromatic pair* problem, with two colors, has applications, for instance, in solving the *Euclidean minimum spanning tree* problem, and was described by [2].

To simplify the exposition, we consider that the points are all distinct and have non-negative coordinates. These restrictions are not impeditive and the algorithm can be easily adapted to the general case.

The algorithm we present here relies on a total order of the points. This consideration only makes sense for the unidimensional case, and this is the case we consider. To perform comparisons of distances between pairs of points, we resort to an oracle given as an unitary transformation, which, given two pairs of indices to points (i, j) and (k, l), decides if $d(p_i, p_j) < d(p_k, p_l)$. The complexity measure we use is the number of oracle calls.

The rest of the paper is organized as follows. In Section 2 we expose some important results, we show how one could build a simple algorithm for the two color bichromatic closest pair problem with running time $O(\sqrt{NM})$, and we present the quantum lower bound for the closest pair problem. In Section 3, we present the closest pair algorithm itself, which includes an auxiliar algorithm, together with the corresponding complexity and success probability analyses. Section 4, contain some further discussions and open problems. It is followed by Section 5 with some concluding remarks.

2 Preliminaries

In this section some supporting algorithms are listed. We also present a naïve quantum algorithm for the bichromatic closest pair problem and a lower bound for the quantum closest pair problem.

2.1 Requisites

In our algorithm we make use of the algorithm for finding the minimum [7], a summary of which is given here.

Theorem 1 (Minimum algorithm [7]) *Let $T[0 \dots N-1]$ be an unsorted table of N items, each holding a value from an ordered set. Let*

$$O\,|i\rangle\,|j\rangle \rightarrow \begin{cases} -|i\rangle\,|j\rangle & \text{if } T[i] < T[j] \\ |i\rangle\,|j\rangle & \text{otherwise,} \end{cases}$$

be an oracle that marks all elements that satisfy the less-than condition. Then, there exists a quantum algorithm that finds the index k such that $T[k]$ is minimum with probability at least $1/2$ and with running time $O(\sqrt{N})$.

We also find it useful to outline, in a simplified form, the quantum *amplitude amplification* algorithm [4], which also appears in [6].

Theorem 2 (Amplitude Amplification [4]) *Let $\mathcal{A}$ be any quantum algorithm that uses no measurements, and let $\chi : \mathbb{Z} \to \{0, 1\}$ be any Boolean function. Let α denote the initial success probability of $\mathcal{A}$ finding a solution (i.e. the probability of outputting z such that $\chi(z) = 1$). Then, there exists a quantum algorithm that finds a solution using an expected number of $O(1/\sqrt{\alpha})$ applications of $\mathcal{A}$ and $\mathcal{A}^{-1}$ if $\alpha > 0$, and otherwise runs forever. Also, if α is known then $O(1/\sqrt{\alpha})$ is the worst case complexity of the resulting algorithm.*

It is worth noting that if $\alpha = 0$, then amplitude amplification will not change the success probability in the resulting algorithm, however, we can obtain the same complexity by limiting the running time of the algorithm.

2.2 The naïve algorithm

In the quantum setting, we can build a naïve algorithm for the bichromatic closest pair problem, which is based on Dürr and Høyer's algorithm for finding the minimum [7]. The latter, in turn, is based on Grover's algorithm [8, 9]. First, we can consider that each element being searched is, in fact, a pair of points identified by their indices. So, the pair (p_i, p_j), where $p_i \in P$ and $p_j \in Q$, could be identified as (i, j) and each pair (i, j) can be identified with a number from 0 to $NM - 1$ (and with states $|0\rangle$ to $|NM - 1\rangle$) and can, thus, be represented by $\log N + \log M$ qubits.

Let $T[(0, 0) \dots (N-1, M-1)]$, according to Theorem 1, be a table of NM items, holding all the MN bichromatic pairs. The problem is to find (i, j) such that $d(p_i, p_j)$ is minimum. For this, we can use the following oracle:

$$O\,|i,j\rangle\,|k,l\rangle \to \begin{cases} -\,|i,j\rangle\,|k,l\rangle & \text{if } d(p_i,p_j) < d(p_k,p_l) \\ \;\;|i,j\rangle\,|k,l\rangle & \text{otherwise.} \end{cases}$$

Also, in the first step of the algorithm, we must select a random pair of indices (k, l). The rest of the algorithm follows from the algorithm presented in [7]. For a table of K items, the algorithm running time is $O(\sqrt{K})$. Since we have $K = NM$, the running time of this algorithm is $O(\sqrt{NM})$.

We can use this same algorithm to solve the simple closest pair problem, in which case the complexity reduces to $O(N)$, by making $M = N$ and not allowing a pair of the form (i, i) to be selected.

2.3 Closest-pair lower bound

In the classical deterministic model, it is known that $\Theta(N \log N)$ queries are necessary and sufficient [11] for finding the closest pair. For the classical probabilistic case, there is a number of algorithms that run in $O(N)$ time, for example [12, 10].

Classically, the lower bound stems from the fact that the closest pair problem can be reduced to the element distinctness problem [3, 14]. As the latter has an $\Omega(N \log N)$ lower bound, the former inherits the same bound.

In the quantum case we can use a similar argument, reducing the quantum closest pair problem to the quantum element distinctness problem. The element distinctness problem has a lower bound of $\Omega(N^{2/3})$ queries [1, 5]. Since solving the quantum closest pair problem allows us to solve the quantum element distinctness problem, we obtain a lower bound of $\Omega(N^{2/3})$ for the closest pair problem on a quantum computer.

The reduction is as follows. We are given an input to the element distinctness problem consisting of a set of N elements $\{x_i : i \in [N]\}$ and an oracle which can compare the elements for equality. We consider it as being an input to the closest pair problem, where x_i is the position of point p_i, and define the distance function as $d(p_i, p_j) = 0$ if $x_i = x_j$ and $i \neq j$, or $d(p_i, p_j) = 1$ otherwise. It is easy to see that the closest pair has distance zero if and only if the element distinctness problem has a non unique element.

It is worth noting that this lower bound for the element distinctness problem is based on evaluation queries, that is, given i one can obtain the element value x_i. Our algorithm is based on comparison queries, that is, given i and j one can only decide if $x_i < x_j$. The first model is stronger than the second, because, by evaluating, one can still compare the elements, but not the other way around, so this bound is valid for both cases.

3 The algorithm

Our purpose is to design an algorithm for solving the two-color closest bichromatic pair problem. We start by discussing about an algorithm to solve a similar problem. This algorithm is based on the one presented in [6], which also contains other interesting remarks.

Instead of finding the bichromatic pair which have the minimum distance, we are looking for any bichromatic pair whose distance is less than a threshold distance, given as the distance between two points whose indices (k, l) are passed as a parameter to the algorithm. The return of this algorithm is an index i, such that $d(p_i, p_j) < d$, where $d = d(p_k, p_l)$ is the distance threshold, $p_i \in P$, and $p_j \in Q$, which must be computed, is the closest point to p_i.

To obtain p_j one can just use a simple application of the algorithm for finding the minimum, which is able to find the closest point to a given one, that is, to find the index j given index i, in $O(\sqrt{|Q|}) = O(\sqrt{M})$ time.

Algorithm A(k,l)

1. Select a random subset $A \subset [N]$ of size L.
2. Select a random subset $B \subset [M]$ of size L^2.
3. Sort the elements of A according to their distances to the origin.
4. Use Grover's Algorithm on the elements of the B set to search for pairs in $A \times B$ such that their distance is less than the distance threshold $d(p_k, p_l)$. For this, use the following oracle:
   - Mark an item $i \in B$ if the distance from i to its closest point in A is less than $d(p_k, p_l)$.
   - Use binary search to determine the point in A closest to a given point, and let j be the index to this point.

We analyze the query complexity of Algorithm A, by choosing $L = \min\{N, \sqrt{M}\}$. Step 3 takes $L \log L + O(L)$ comparisons, using classical sorting. Step 4 takes $O(\sqrt{|B|})$ applications of the oracle, which, in turn, does $O(\log |A|)$ comparisons, yielding a complexity of $O(L \log L)$. This results in an overall $O(L \log L)$ query comparisons for Algorithm A.

Now, for the success probability of algorithm A. If there are no pair of points such that their distance is less than d, then the algorithm will not succeed. Suppose there is at least one pair of points such that their distance is less than the threshold, say (p_x, p_y). Then, the probability of (x, y) belonging to (A, B) is at least $(L/N)(L^2/M) = L^3/(MN)$, and if indeed $(x, y) \in A \times B$, then step 4 will find this (or some other) pair of points with probability at least $1/2$ in at most $O(L \log L)$ queries. Hence, the overall success probability of Algorithm A is at least $\alpha = L^3/(2MN)$. This probability is small, but can be amplified by taking advantage of the amplitude amplification algorithm.

Algorithm Bichromatic Closest Pair

1. Choose uniformly a random index $k \in [N]$. Compute the index l such that $p_l \in Q$ is the closest point to $p_k \in P$.
2. Repeat the following steps for $2m$ times (m is defined below):
   (a) Apply amplitude amplification on Algorithm A with distance threshold $d(p_k, p_l)$.
   (b) Observe the outcome, obtaining i and computing j, such that $p_j \in Q$ is the closest point to $p_i \in P$.
   (c) If $d(p_i, p_j) < d(p_k, p_l)$, then set (k, l) to (i, j).
3. Return (k, l).

We first analyze the query complexity of step 2a. As we remarked above, the success probability of Algorithm A is at least $\alpha = L^3/(2MN)$, hence the amplitude amplification step requires a worst case number of $O(1/\sqrt{\alpha}) = O(\sqrt{MN/L^3})$ applications of Algorithm A. Therefore, taking into consideration that $L = \min\{N, \sqrt{M}\}$, the total number of queries for step 2a is $O(\sqrt{N} M^{1/4} \log M)$ if $N \leq M \leq N^2$, or $O(\sqrt{M} \log N)$ if $M > N^2$. The query complexity of step 1 and step 2b is $\sqrt{M}$ due to the closest point search. Step 2c, uses one oracle query.

Now, for the number of times we need to repeat step 2. We want to derive the expected time to find the minimum.

Lemma 1 *The expected number of times that step 2 should be repeated so that (k, l) holds the closest pair is at most $m = 2 \log N + 2$.*

Proof. We can rank, from 1 to N, each element selected in steps 1 and 2b because we just select one of the points from set $[N]$, the other is deterministically obtained.

Define $S(r)$ as the number of times step 2 should be repeated before (k, l) holds the closest pair if we choose the element with rank r as a threshold. By calculating $S(N)$ we can find an upper bound to the expected number of times step 2 should be repeated, because selecting the element with rank N is the worst case for the first step of the algorithm.

In any step, any element with a distance less than the threshold can be chosen with equal probability. Also, in step 2a, the algorithm may fail with probability at most 1/2, in which case we re-execute the step with the same element as a threshold. So we can define $S(N)$ recursively as:

$$S(N) = 1 + \frac{1}{2}\left(\frac{1}{N-1}\sum_{i=1}^{N-1} S(i)\right) + \frac{1}{2}S(N),$$

which, by considering $S(1) = 0$, has as solution $S(N) = 2H_{N-1}$, where H_k is the k-th harmonic number. So $S(N) \leq 2\log(N-1) + 2 \leq 2\log N + 2$. ■

Lemma 2 *To achieve at least* $1/2$ *probability of success, we should run step 2 of the algorithm for at least* 2 *times the expected number, i.e.,* $2m = 2(2\log N + 2)$ *times.*

Proof. Simply apply Markov's Inequality. ■

Theorem 3 *The number of queries performed by the algorithm for finding the two-color closest bichromatic pair is* $O(N^{1/2}M^{1/4}\log M \log N)$ *if* $N \leq M \leq N^2$*, or* $O(M^{1/2}\log^2 N)$ *if* $M > N^2$*.*

Proof. From Lemmas 1 and 2, and the above complexity analysis, step 2a is executed $4\log N + 4$ times. By considering the number of queries in each execution, the result easily follows. ■

Corollary 1 *The number of queries performed by the algorithm for finding the closest pair is* $O(N^{3/4}\log^2 N)$*.*

Proof. Let $M = N$, and modify the algorithm to avoid selecting points of the form (i, i). ■

4 Remarks and open issues

Let $Q_2(P)$ be the worst-case number of queries required for solving problem P by a quantum bounded error algorithm. Then, the query comparison complexity of the quantum closest pair problem currently is:

$$\Omega(N^{2/3}) \leq Q_2(\textbf{Closest-Pair}) \leq O(N^{3/4}\log^2 N).$$

For the bichromatic closest pair problem, there is still no quantum lower bound, even for the two-color case. Such lower bound would preferably involve variables N and M, regarding the number of points labeled with each color.

The specific algorithm used relies on sorting, so the derived upper bound applies only to the 1-dimensional case. We also use an oracle for comparing distances, so our queries are comparison-based.

The fact that distances between points in space may be non-rational numbers, even if all the coordinates are integers, may pose some problems if we were to use evaluation queries for the closest pair problem. The only exception to this is the 1-dimensional case. Hence, it would be possible to solve the closest pair problem using evaluation queries and the traditional distance function only in 1 dimension. For higher dimensions, it would be necessary to use comparison queries or simpler distance functions returning only rational numbers.

Also, it seems that the lower bound for the closest pair problem using comparison-based oracles could be raised, possibly by a factor of $\log N$. Note that evaluation queries are stronger than comparison queries, and the former can simulate the latter by using binary search, which usually involves doing $\log N$ additional operations.

5 Conclusion

We presented a faster than classical algorithm for solving the two-color *bichromatic closest pair* problem and, as a special case, the *closest pair* problem. Both algorithms are assymptotically superior than their classical deterministic and probabilistic versions. They also provide an speedup against a naïve approach.

There is still room for improvement, by raising the lower bound or by reducing the upper bounds. This might be true specially for the *closest pair* problem, for which we obtained the aforementioned lower bound.

There is still no lower bound for the closest pair problem for an oracle exclusively based on comparisons.

References

[1] Scott Aaronson and Yaoyun Shi. Quantum lower bounds for the collision and the element distinctness problems. *Journal of the ACM*, 51:595–605, 2004.

[2] Pankaj K. Agarwal, Herbert Edelsbrunner, Otfried Schwarzkopf, and Emo Welzl. Euclidean minimum spanning trees and bichromatic closest pairs. In *SCG '90: Proceedings of the sixth annual symposium on Computational geometry*, pages 203–210, New York, NY, USA, 1990. ACM Press.

[3] Michael Ben-Or. Lower bounds for algebraic computation trees. In *STOC '83: Proceedings of the fifteenth annual ACM symposium on Theory of computing*, pages 80–86, New York, NY, USA, 1983. ACM Press.

[4] Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. 2000. `http://arxiv.org/abs/quant-ph/0005055`.

[5] Gilles Brassard, Peter Høyer, and Alain Tapp. Quantum cryptanalysis of hash and claw-free functions. *Lecture Notes in Computer Science*, 1380:163–169, 1998.

[6] Harry Buhrman, Christoph Durr, Mark Heiligman, Peter Hoyer, Frederic Magniez, Miklos Santha, and Ronald de Wolf. Quantum algorithms for element distinctness, 2000. `http://arxiv.org/abs/quant-ph/0007016`.

[7] Christoph Durr and Peter Hoyer. A quantum algorithm for finding the minimum, 1996. `http://arxiv.org/abs/quant-ph/9607014`.

[8] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proc. 28th Annual ACM Symposium on the Theory of Computation*, pages 212–219, New York, NY, 1996. ACM Press, New York. `http://arxiv.org/abs/quant-ph/9605043`.

[9] Lov K. Grover. Quantum mechanics helps in searching for a needle in a haystack. *Phys. Rev. Lett.*, 79:325, 1997. `http://arxiv.org/abs/quant-ph/9706033`.

[10] Samir Khuller and Yossi Matias. A simple randomized sieve algorithm for the closest-pair problem. *Inf. Comput.*, 118(1):34–37, 1995.

[11] Franco P. Preparata and Michael Ian Shamos. *Computational Geometry an introduction.* Springer-Verlag New York Inc., 1985.

[12] M. Rabin. Probabilistic algorithms. *Algorithms and Complexity, Recent Results and New Directions*, pages 21–39, 1976.

[13] Nilton Volpato and Arnaldo Moura. An $O(N^{3/4} \log^2 N)$ quantum algorithm for the 1-dimensional closest-pair problem. In *WECIQ 2006: Workshop-Escola de Computação e Informação Quântica*, October 2006.

[14] Andrew Chi-Chih Yao. Lower bounds for algebraic computation trees with integer inputs. *SIAM J. Comput.*, 20(4):655–668, 1991.