Quantum Query Complexity of Some Graph Problems

Dürr, Heiligman, Høyer, Mhalla, SIAM J. Computing 2006.

Quantum algorithms, basic graph problems (query complexity), two representations:

Matrix model: algorithm is given n there is a black box that given i and j, returns whether vertices i and j are adjacent.

Array model: algorithm is given n and $d_1, \ldots, d_n$. there is a black box that given i and j, returns the j-th neighbour of vertex i.

representation	matrix	array
non-quantum	$\Theta(n^2)$	$\Theta(n^2)$
quantum	$\Theta(n\sqrt{n})$	$\Theta(n)$

Randomized polynomial algorithms with bounded error probability

Other problems: strong connectivity, minimum spanning trees, single source shortest paths

Main tool. Amplitude amplification, a generalization of Grover's search algorithm: Consider a function $F : \{1, \dots, n\} \to \{0, 1\}$. Let $A = \{x : F(x) = 1\}$, and $a = |A|$. There is a quantum algorithm that given black-box access to F,

1. If $a > 0$, outputs a random element in A asking an expected number of $0.9\sqrt{n/a}$ queries.

2. If $a = 0$, may run forever.

Algorithm for matrix model. Start with n trees: in each iteration search for an edge that connects two trees.

Running time analysis if graph is connected: when there are $k > 1$ trees, it takes on average $0.9\sqrt{\frac{n(n-1)}{k-1}} = O(\sqrt{n^2/k})$ queries to reduce the number of trees by 1. Total expected running time is thus of order

$$\sum_{k=2}^{n} \sqrt{n^2/k} = n \sum_{k=2}^{n} \frac{1}{\sqrt{k}} \leq n \int_{x=1}^{n} \frac{1}{\sqrt{x}} = n2\sqrt{x}|_1^n = n\sqrt{n}$$

Algorithm for array model.

Phase 1: Partition the vertex set into pieces (sets inducing connected subgraphs).

Phase 2: Merge the pieces iteratively: in every iteration, choose a piece with minimum total degree, and find an outgoing edge; then merge two pieces using that edge.

Lemma 1. *Using $O(n)$ queries we can partition the vertices into several pieces such that for each piece C, its total degree $t(C) < |C|^2$.*

Lemma 2. *Suppose that after Phase 1 the pieces have total degrees $t_1, \ldots, t_k$. If graph is connected, then the expected query complexity of (2) is*

$$O\left(\sqrt{t_1} + \sqrt{t_2} + \cdots + \sqrt{t_k}\right) = O(n)$$

Thus we find an algorithm with expected query complexity $O(n)$ for connected graphs. To get a polynomial time algorithm, we run it and if it did not succeed after three times the expected time, stop it and output *DISCONNECTED*.

Lemma 1. Using $O(n)$ queries we can partition the vertices into several pieces such that for each piece C, its total degree $t(C) < |C|^2$.

Proof. Initially there is no piece, and all vertices are uncovered.
(*) while there exist uncovered vertices,
$u =$ an uncovered vertex with maximum degree d.
Go through neighbours of u in order, inserting them into a buffer B
If a neighbour v is encountered that is covered, add $u \cup B$ to the piece containing v, and go to (*)
Otherwise, i.e. if all neighbours of u are uncovered, u and its neighbours form a new piece; go to (*)

Case 2: a newly built piece has $d+1$ vertices and total degree $\leq d(d+1) < (d+1)^2$.

Case 1: $u \cup B$ is added to piece P: let $b = |B|$. Note $d \leq |P|$.

$$\begin{aligned} t(P \cup u \cup B) &\leq t(P) + d + |B|d \\ &< |P|^2 + |P|(b+1) \\ &< (|P| + b + 1)^2 \\ &= (|P \cup u \cup B|)^2 \end{aligned}$$

□

Lemma 2. Suppose that after Phase 1 the pieces have total degrees $t_1, \ldots, t_k$. If graph is connected, then the expected query complexity of (2) is

$$O\left(\sqrt{t_1} + \sqrt{t_2} + \cdots + \sqrt{t_k}\right) = O(n)$$

Proof. (Define the notion of a half-edge.) In a given iteration, if P is the piece with minimum $t(P)$, that iteration takes $0.9\sqrt{t(P)/\delta(P)} \leq \alpha\sqrt{t(P)}$ queries on average. We distribute this cost among the half-edges of P, so each half-edge pays $\leq \alpha/\sqrt{t(P)}$.

Fix a half-edge e, and assume during Phase 2 it is contained in pieces with total degrees $t(P_0) = m_0 < m_1 < \ldots$. Then $m_{i+1} \geq 2m_i$ for all i, hence e pays

$$\leq \sum_{i=0}^{\infty} \frac{\alpha}{\sqrt{m_i}} \leq \sum_{i=0}^{\infty} \frac{\alpha}{\sqrt{2^i m_0}} = \frac{\alpha}{\sqrt{m_0}} \sum_{i=0}^{\infty} 2^{-i/2} \leq \frac{4\alpha}{\sqrt{m_0}}$$

Hence, for an initial piece P_0, the half-edges of P_0 pay $\leq t(P_0)\frac{4\alpha}{\sqrt{t(P_0)}} = 4\alpha\sqrt{t(P_0)}$ in total $\square$