Presentation notes for Distributed approximate algorithm for bipartite vertex cover

Soumyadeep Paul (BMC202178)

April 17, 2024

We present a logarithmic time approximation algorithm for minimum vertex cover in bipartite graph in the **LOCAL**-model.

Summary

1 LOCAL-model

We work in the **LOCAL**-model for distributed computing. We are given an undirected graph $G = (V, E)$, which we can think of as a network. Each node has a processor and two processors can communicate with each other if they are connected by an edge. All nodes run the same distributed algorithm $\mathcal{A}$. We will additionally assume the graph is connected.

Each node has a unique ID $ID(v) \in \{1, 2, \cdots n^c\}$. The computation proceeds in synchronous communication rounds. In each round, all nodes first perform some local computations and then exchange (unbounded) messages with their neighbours. After some r communication rounds the nodes stop and produce local outputs. Here r is the running time of $\mathcal{A}$ and the output of v is denoted $\mathcal{A}(G, v)$. We ignore the local computation time when measuring complexity of algorithms.

Every computable function can be computed by an algorithm in $O(|V|)$ time by an algorithm where in the i^{th} round each node sends its radius-i neighborhood to its neighbours. When this algorithm terminates all the nodes have complete information about the graph and can run any deterministic sequential algorithm locally.

Example 1. *We demonstrate a simple algorithm in the local model for finding 3 coloring of a path graph.*

This algorithm runs in $O(n)$ rounds but there exists algorithms for this problem which run in $O(log^*(n))$ rounds.

Algorithm 1: Two coloring of path

c = ID_x
for ∞ **do**
  Send c to all neighbours
  Receive messages from all neighbours, let M be the set all messages from neighbours
  if $c \notin 1, 2, 3$ *and* $c > max(M)$ **then**
    $c = min(\{1, 2, 3\} \setminus M)$
  end
end

2 Graph Decompositions

Definition 1. *Let $G = (V, E)$ be a graph. Any subset $W \subset V$ is said to be a block. The strong diameter of a block W, $SD(W)$ is the maximum diameter of any connected component of the graph G_W induced on W. The weak diameter $WD(W)$ is the maximum distance in G between any two vertices in W. The difference between strong diameter and weak diameter is that while calculating weak diameter we allowed to shortcut through vertices not in W. Clearly $WD(W) \leq SD(W)$.*

A partition Π of V into λ disjoint blocks is called a λ-decomposition of G. $SD(\Pi)(WD(\Pi))$ is the maximum strong (weak) diameter of any of its blocks.

We are interested in finding a graph decomposition into a small number of blocks each of a small diameter.

Theorem 1. *Let $p \in (0, 1)$, G be an n vertex undirected graph and $\lambda = \frac{log(n)}{log\left(\frac{1}{1-p}\right)}$. Then there is a λ-decomposition of G with strong diameter at most $\frac{2log(n)}{log\left(\frac{1}{p}\right)}$.*

Proof. For an integer r let $B_r(x)$ be the ball of radius r around x. We call an integer r a safe radius if $p|B_r(x)| < |B_{r-1}(x)|$.

If $1, 2, \cdots, r$ are all unsafe for x, then $|B_j(x)| > \left(\frac{1}{p}\right)^j \forall j \in [r]$, in particular $n \geq \left(\frac{1}{p}\right)^r$. In other words, for every x there exists a safe radius not exceeding $\frac{log(n)}{log\left(\frac{1}{p}\right)}$.

We construct λ-decomposition $V_1, V_2, \cdots$ one block at a time. Pick any vertex x_1 of $G_1 = G$ and let r_1 be the smallest safe radius of x_1. Add all the vertices of $B_{r_1-1}(x)$ to V_1 and define $G_2 = G_1 \setminus B_{r_1}(x)$. Similarly construct x_i, r_i, G_i till we run out of vertices.

Having constructed $V_1, V_2, \cdots, V_{i-1}$, define $G^i = G \setminus (V_1, V_2, \cdots, V_{i-1})$ and apply the process to G_i to obtain V_i.

The construction of the blocks guarantees that its strong diameter is at most twice the largest radius of any of the selected balls. Therefore,

$$SD(W) \leq \frac{2log(n)}{log(1/p)}.$$

Since, the ratio of $|B_{r_j-1}(x)|$ to $|B_{r_j}(x)|$ is at least p for each selected ball,the fraction of vertices of G^i not assigned to V_i is at most $1-p$. Therefore, $|G^i| \leq (1-p)|G^{i-1}|$. This implies

$$\lambda \leq \frac{log(n)}{log(1/(1-p)}$$

□

We now give a randomized distributed algorithm for construction of the blocks which we call *Construct_Block*.

Given a p and B, we call the following a truncated geometric distribution:

$$P(X=j) = p^j(1-p)\forall j \in \{0,1,2,\cdots,B-1\}$$
$$P(X=B) = p^B.$$

First each vertex x selects an integer radius r_x according to the truncated geometric distribution (we will choose the p and B later). It then broadcasts (ID_x, r_x) to every node within distance r_x of it. Now each node z selects its center node, $C(z)$ to be the highest ID whose broadcast it received. If $C(z) > d(z, C(z))$, z joins the block else it waits for the next iteration of *Construct_Block*.

Lemma 1. *If Construct_Block is applied to G with n vertices and S be the set of vertices comprising the block selected:*

1. $WD(S) \leq 2B$
2. $\forall x \in V_G$, *probability that it belongs to S is at least* $p(1-p^B)^n$.

Proof. 1. To prove the first part we just need to prove that for any connected subset T of S, $C(y)$ is the same vertex for all $y \in T$.

We give a proof of this fact by contradiction, say there exists adjacent vertices, y and z with $C(y) \neq C(z)$. We assume, WLOG, $ID_{C_y} > ID_{C_z}$. By the definition of S, $r_{C_y} > d(C(y), y)$. Since y and z are neighbours, $r_{C_y} \geq d(C(y), z)$. Therefore z received the broadcast sent by $C(y)$. This contradicts the fact that $C(z) < C(y)$.

2. We have,

$$P(y \in S) \geq \sum_{d(z,y)<B} P(y \in S | C(y) = z) P(C(y) = z).$$

We define the following events

(a) $D_z : r_z \geq d(z, y)$

(b) $E_z : r_z > d(z, y)$

(c) F_z: For every vertex w with ID higher z, $r_{C_w} < d(w, y)$.

We then have,

$$P(y \in S | C(y) = z) = P(E_z \wedge F_z | D_z \wedge F_z) = P(E_z \wedge F_z)/P(D_z \wedge F_z) = P(E_z)/P(D_z) = p.$$

Since, $P(D_z) = p^{d(z,y)}, P(E_z) = p^{d(z,y)+1}$. Thus,

$$P(y \in S) \geq p \sum_{d(z,y)<B} P(C(y) = z) \geq pP(d(C(y), y) < B) \geq pP(r_z \neq B, \forall z) \geq p(1-p^B)^n.$$

□

3 Distributed algorithm for minimum vertex cover

Theorem 2. *Let $\epsilon > 0$. An expected $(1 + O(\epsilon))$ approximation of minimum vertex cover can be found in time $O\left(\frac{log(n)}{\epsilon}\right)$ on graphs of maximum degree $\Delta = O(1)$.*

Proof. We run *Construct_Block* algorithm with $p = 2^{-\epsilon}, B = \frac{2log(n)}{log(1/p)}$.

By Lemma 1, each component of G_S, has a weak diameter at most $B = \frac{4log(n)}{log(1/p)}$.

We also have,

$$\lim_{n\to\infty}(1 - p^B)^n = \lim_{n\to\infty}(1 - n^{-2})^{n^2\times n^{-1}} = \lim_{n\to\infty} e^{-n^{-1}} = 1.$$

Therefore $E[|S|] \geq np(1 + o(1)) \geq (1 + o(1))n(1 - \epsilon)$.

Now, let C be a component of G_S, every node can discover the structure of C in time $O\left(\frac{log(n)}{\epsilon}\right)$ exploiting weak diameter. Therefore, every node of C can internally compute the same optimal solution for vertex cover.

We then output as a vertex cover for G the union of vertex cover of each component of G_S and $V \setminus S$. This results in a solution of size at most

$$OPT_{G_S} + \epsilon n \leq OPT_G + \epsilon n.$$

But we have $OPT_G \geq \frac{|E|}{\Delta} = \Omega(n)$ for a connected graph. Therefore this is an expected (1 + O(ϵ))-approximation of minimum vertex cover. □

4 References

1 Linial, N., Saks, M. Low diameter graph decompositions. Combinatorica 13, 441–454 (1993). https://doi.org/10.1007/BF01303516

2 Göös, M., Suomela, J. No sublogarithmic-time approximation scheme for bipartite vertex cover. Distrib. Comput. 27, 435–443 (2014). https://doi.org/10.1007/s00446-013-0194-z