

Maximum flow is approximable by deterministic constant-time algorithm in sparse networks

Endre Csóka

Eötvös Loránd University, Budapest, Hungary

Abstract

We show a deterministic constant-time parallel algorithm for finding an almost maximum flow in multisource-multitarget networks with bounded degrees and bounded edge capacities. As a consequence, we show that the value of the maximum flow over the number of nodes is a testable parameter on these networks.

1 Introduction

In the last decade it became apparent that a large number of the most interesting structures and phenomena of the world can be described by networks which are so large that the data about them can be collected only by indirect means like random local sampling. This yielded the motivation of property testing and parameter testing, which became an intensively studied field recently (see [3], [12] and all references). There are two special cases that have been treated in most papers. One is the dense graphs, where a positive fraction of all pairs of nodes are connected. The other is the sparse graphs, these are mostly graphs with bounded degrees. In this paper, we deal only with parameter testing of bounded-degree graphs. A parameter tester of bounded-degree graphs means an algorithm which chooses a constant number of random nodes, and to these constant radius neighbourhoods, assigns an estimation (a number) which is at most ε far from the true parameter with at least $1 - \varepsilon$ probability. We call a parameter testable if there exists a tester with arbitrary small errors.

There is a strongly connected concept called constant-time algorithm, introduced by Nguyen and Onak[14]. For example, consider the maximum matching problem. Here, a constant-time algorithm is a "local function" that decides about each edge that whether it chooses to the matching or not, so that the chosen edges form a matching, and its size is at most εn less than the size of the maximum matching, with at least $1 - \varepsilon$ probability. Locality means that it depends only on the constant-size neighbourhood of the edge, including some random numbers, as follows. Randomization is strictly required to break symmetry, for example in a regular large-girth graph. The idea of Nguyen and Onak was to assign independent random numbers to the nodes uniformly from $[0, 1]$,

and the neighbourhood consists not only of the induced subgraph of nodes at most constant far from the chosen node, but also of the random numbers of these nodes. In that paper, they showed a constant-time algorithm for this and some other problems.

About the connection of the two concepts, notice that if we have a constant-time algorithm producing the maximum matching then the ratio of the size of the maximum matching and the number of nodes is a testable parameter. Because we can make a tester which simply calculates the probability that the chosen random node would be covered by the matching produced by the constant-time algorithm, and the average of these probabilities is a good approximation.

In this paper, we show a constant-time *deterministic* algorithm for a version of the maximum flow problem. This determinism means that we will get the analogous result using no random numbers.

If we delete all edges from all sources or targets then the value of the maximum flow decreases to 0 while the distribution of the local neighbourhoods remains asymptotically the same. That is why the value of the maximum flow in a graph with 1, or even with $o(n)$ sources or targets cannot be tested in any reasonable way. Similarly, one new edge with high capacity between a source and a target would increase the value of the maximum flow by this arbitrary large value. These are some reasons why we will deal only with multiple sources and targets and bounded capacities.

2 Model and results

There is an input graph G with degrees bounded by d . Its vertices are separated into the disjoint union of the colour sets R (regular), S (source) and T (target). Each edge of the graph is considered as two directed edge in the two directions, and we have a capacity function $c : \vec{E}(G) \rightarrow [0, M]$ of the directed edges. The word "graph" will include these two structures and we handle d and M as global constants throughout the paper. Let $|V(G)| = n$, denote by $out(v)$ the set of edges starting from a node v , and for an $e \in \vec{E}(G)$, let $-e$ denote the edge e in the opposite direction. We define the flow as a function $f : \vec{E}(G) \rightarrow \mathbb{R}$ for which $\forall e \in \vec{E}(G)$ satisfies $f(-e) = -f(e)$ and $f(e) \leq c(e)$ and $\forall v \in S : \sum_{e \in out(v)} f(e) \geq 0$ and $\forall v \in T : \sum_{e \in out(v)} f(e) \leq 0$ and $\forall v \in R : \sum_{e \in out(v)} f(e) = 0$. The value of a flow f is $|f| = \sum_{s \in S; e \in out(s)} f(e)$. Denote a maximum flow by $f^* = f^*(G)$.

The rooted neighbourhood of a vertex v or edge e of radius r , denoted by $h_r(v)$ and $h_r(e)$, means the induced subgraph (with colours and edge weights) of the vertices at a distance at most r from v or e , with a mark at v or e . The set of all possible neighbourhoods are denoted by $H_r^{(1)}$ and $H_r^{(2)}$. A local (flow) assignment means a function $L : H_r^{(2)} \rightarrow \mathbb{R}$ for which the function $e \rightarrow L(h_r(e))$

is a flow for each graph.

Theorem 1. *For each $\varepsilon > 0$ there exists a constant-time algorithm producing a flow with value $\geq |f^*| - \varepsilon n$.*

We emphasize that Theorem 1 is about not a stochastic but a deterministic assignment. So from this aspect, Theorem 1 is stronger than necessary to prove Theorem 2. We get this determinism only by averaging on all random labellings, since the space of flows is a convex set and the values of flows is a linear function.

Theorem 2. *$|f^*(G)|$ is testable, namely, for all $\varepsilon > 0$ there exist $k, r \in \mathbb{N}$ and a function $g : (H_r^{(1)})^k \rightarrow \mathbb{R}$ satisfying that if the vertices v_1, v_2, \dots, v_k are chosen independently with uniform distribution then $E(|f^*(G)|/n - g(h_r(v_1), h_r(v_2), \dots, h_r(v_k))) < \varepsilon$.*

We note that for all $\varepsilon > 0$, having an approximation with expectedly $< \varepsilon n$ error is stronger but often equivalent to having this with $< \varepsilon n$ error with at least $1 - \varepsilon$ probability.

3 Proofs

First we prove Theorem 1 using the following lemmas, and Theorem 2 will be an easy consequence of it.

An augmenting path of a flow f is a directed path $u = (e_1, e_2, \dots, e_k)$ from S to T with $f(e_i) < c(e_i)$ for each set e_i . The capacity of u means $cap(u) = cap(u, f) = \min_i (c(e_i) - f(e_i))$, and we identify an augmenting path u with the flow $u : \vec{E}(G) \rightarrow \mathbb{R}$, $u(e) = \{1 \text{ if } \exists i : e = e_i; -1 \text{ if } \exists i : e = -e_i; 0 \text{ otherwise}\}$, which we also call path-flow. Augmenting on such a path u means the incrementation of f by $cap(u) \cdot u$.

Lemma 3. *If for a flow f , there is no augmenting path with length at most l , then $|f| \geq |f^*| - \frac{2M}{l}n$.*

Proof. With the identically $2M$ capacity function, $f^* - f$ is a flow, so we can split it into the sum of path-flows u_1, u_2, \dots, u_q and a circulation u_0 that follow the directions of the flow $f^* - f$, namely, $\forall i \in \{0, \dots, q\}, e \in \vec{E}(G) : sgn(u_i(e)) \in \{0, sgn((f^* - f)(e))\}$. Thus,

$$\begin{aligned} |f^*| - |f| &= |f^* - f| = \sum_{i=1}^q |u_i| = \frac{1}{2l} \sum_{i=1}^q 2l|u_i| \\ &\leq \frac{1}{2l} \sum_{i=1}^q \sum_{e \in \vec{E}(G)} |u_i(e)| = \frac{1}{2l} \sum_{e \in \vec{E}(G)} \left| \sum_{i=1}^q u_i(e) \right| \leq \frac{1}{2l} \sum_{e \in \vec{E}(G)} 2M \leq \frac{dnM}{l}. \quad \square \end{aligned}$$

Lemma 4. *If for a flow f , there is no augmenting path shorter than k , then augmenting on a path with length k does not create a new augmenting path with length at most k .*

Proof. Let the residual graph mean the graph without edge capacities $G_f = (S(G), R(G), T(G), \{e \in \vec{E}(G) | f(e) < c(e)\})$. Then the augmenting paths of G can be identified with the paths in G_f from S to T . So if the length of the shortest augmenting path of a flow is k then it means that the length of the shortest path in G_f from S to T is k . Let the movement of an edge in G mean the difference of the distances of its endpoint and starting point from S in G_f . Augmenting on a shortest path adds only such edges to the residual graph on which f decreases, which are the reverse edges of the path. All these edges have movement -1 (calculated before augmenting). So if a path becomes an augmenting path at this augmenting step then all its edges have movements at most 1 and contain an edge with movement -1 , so its length is at least $k+2$. \square

Let us label all paths u with length at most l with independent random variables $x(u)$ chosen uniformly from $[0, 1)$, forbidding any two labels to be equal. We define chain as a sequence u_1, u_2, \dots, u_s of such paths for which $|u_1| + x(u_1) > |u_2| + x(u_2) > \dots > |u_s| + x(u_s)$ and $\forall i \in \{1, 2, \dots, s-1\}$ there exists a common undirected edge of u_i and u_{i+1} (henceforth: these intersect each other).

Lemma 5. *For each $l \in \mathbb{N}$ and $\varepsilon > 0$ there exists a $q = q(l, \varepsilon) \in \mathbb{N}$ for which for every graph G (with degrees bounded by d) and its undirected edge e , with random labelling, the probability that there exists a chain u_1, u_2, \dots, u_q for which u_1 contains e is at most ε .*

Proof. There exists an upper bound $z = z(l)$ for the number of paths with length at most l intersecting a given path of length at most l . Hence, there are at most z^q sequences of paths u_1, u_2, \dots, u_q for which e is in u_1 and $\forall i \in \{1, 2, \dots, q-1\}$, u_i intersects u_{i+1} . All such sequences contain $\lceil q/l \rceil$ paths of the same length. The ordering of their labels the $\lceil q/l \rceil!$ permutations with the same probability, so the probability that the labels are decreasing is $1/\lceil q/l \rceil!$. This event is necessary for the sequence to be a chain. Denote the number of chains in the lemma by the random variable X (with respect to the random labelling). We have

$$P(X \geq 1) \leq E(X) \leq \frac{z^q}{\lceil \frac{q}{l} \rceil!} \rightarrow 0 \text{ where } q \rightarrow \infty,$$

which proves the lemma for some large enough number q . \square

Proof of Theorem 1. Consider the variant of the Edmonds–Karp algorithm where we augment on the one of the shortest augmenting paths with the lowest label, and we stop the algorithm when no augmenting path with length at most l remains. In other words, we start from the empty flow, we take the paths u with length at most l in the increasing order of $|u| + x(u)$, and with each path, we increase the actual flow f by $\text{cap}(f, u) \cdot u$. We denote this algorithm by A_1 and the resulting flow by $f_1 = f_1(G, x)$.

Consider now the variant of the previous algorithm where we skip augmenting on each path which can be obtained as the first element of any chain

with length s . We denote this algorithm by A_2 and the resulting flow by $f_2 = f_2(G, x)$. The next lemma shows that $f_2(e)$ is a constant-time algorithm.

Lemma 6. *For each edge e and labelling x , $f_2(G, x)(e) = f_2(h_{sl}(e), (x|V(h_{sl}(e))))(e)$.*

Proof. Let us consider the run of the two algorithms in parallel so that when the first one takes a path u in G then if u is in $h_{sl}(e)$ then the second algorithm takes u as well, otherwise it does nothing. If at a point, the two flows differ at an edge $e' \in \vec{E}(h_{sl}(e))$ then there must have been a path u through e' on which the two algorithms augmented by different values. There are three possible reasons of it:

1. u is not in $h_{sl}(e)$;
2. u can be obtained as the first term of some chain in G with length s , but not in $h_{sl}(e)$;
3. u has an edge e'' in $h_{sl}(e)$ at which the values of the two flows were different before taking u .

Assume that at the end, the two flows are different on e . Using the previous observation initially with $e' = e$, let us take a path u through e' on which the two augmentations were different, and consider which of the three reasons occurred. As long as the third one, repeat the step with choosing e' as the e'' of the previous step. Since by each step we jump to an earlier point of the runs, we must get another reason sooner or later. Denote the paths considered during the process by u_1, u_2, \dots, u_t . (Note that these are in reverse order on the augmenting timeline.)

Consider the case when the reason for u_t was the first reason. The set of all edges of all of these t paths is connected, it contains at most tl edges, it contains e and an edge at least sl away from e , so $tl > sl$, whence $t > s$. Thus u_1, u_2, \dots, u_s is a chain with a connected edge set with size at most sl , so this chain is in $h_{sl}(e)$, that is why neither runs should have been augmented on u_1 , contradicting with the definition of u_1 .

On the other hand, if the reason for u_t was the second reason then by appending u_1, u_2, \dots, u_t with the chain from u_t with length s , as its subchain, we get a chain starting with u_1 with length s , and it provides the same contradiction. \square

We prove that if f_2 is the output of A_2 with $l = 2dM/\varepsilon$ and using the function q of Lemma 5, with $s = q(l, \varepsilon/(4dM))$ satisfies the following inequality.

$$E(|f_2|) \geq E(|f_1|) - \frac{\varepsilon}{2}n \geq |f^*| - \varepsilon n \quad (1)$$

f_1 contains no augmenting path with length at most l , so using Lemma 3,

$$|f_1| \geq |f^*| - \frac{dM}{l}n = |f^*| - \frac{dM}{\frac{2dM}{\varepsilon}}n = |f^*| - \frac{\varepsilon}{2}n.$$

Consider now the first inequality.

Lemma 7. *If $f_1(x)(e) \neq f_2(x)(e)$ then there exists a path through e which is the first term of a chain with length s .*

Proof. Let us consider the run of A_1 and A_2 in parallel so that at the same time these take the same edge. If at a point of the runs, the two flows differ in an edge e' then there must have been a path u through e' on which the two algorithms augmented by different values. There are two possible reasons of it:

1. u is the first term of a chain with length s ;
2. u has an edge e'' on which the values of the two flows were different before taking u .

Assume that at the end, the two flows are different at e . Using the previous observation, let us take a path u through e' on which the two augmentation were different, and consider which of the two reasons occurred. As long as the latter one, repeat the step with choosing e' as the e'' of the previous step. Since by each step we jump to an earlier point of the runs, we must get the first reason in finite many steps. Denote the paths considered during the process by u_1, u_2, \dots, u_t . Then appending u_1, u_2, \dots, u_t with the chain from u_t with length s , as its subchain, we get a chain starting with u_1 with length s . \square

If $f_1(x)(e) \neq f_2(x)(e)$ then by Lemma 7, there exists a chain with length $q(l, \frac{\varepsilon}{4dM})$, and Lemma 5 says that this has probability at most $\frac{\varepsilon}{4dM}$. But even if this occurs, $f_1(x) - f_2(x) \leq M - (-M) = 2M$. That is why,

$$\begin{aligned} E(|f_1|) - E(|f_2|) &= E(|f_1(x)| - |f_2(x)|) = E(|(f_1(x) - f_2(x))|) \\ &= E\left(\sum_{s \in S; e \in \text{out}(s)} f_1(x)(e) - f_2(x)(e)\right) \leq \sum_{s \in S; e \in \text{out}(s)} \frac{\varepsilon}{4dM} \cdot 2M \leq \\ &\leq dn \cdot \frac{\varepsilon}{4dM} \cdot 2M \leq \frac{\varepsilon}{2}n. \end{aligned}$$

We are finished proving (1).

Now, let $\bar{f}_2(e) = E(f_2(x)(e))$. It is a flow because it is easy to check that it satisfies all requirements, and $|\bar{f}_2| = E(|f_2|) \geq |f^*| - \varepsilon n$. Furthermore, $\bar{f}_2(e)$ depends only on $h_{sl}(e)$, so it can be calculated by a constant-time algorithm. Consequently, this assignment satisfies the requirements of the theorem. \square

Proof of Theorem 2. Let \bar{f}_2 be the flow constructed by the constant-time algorithm of the previous proof with error bound $\varepsilon/2$, which therefore satisfies $|\bar{f}_2| \in [||f^*| - \frac{\varepsilon}{2}|, |f^*|]$, and let r be the radius used there plus 1. Using the notion $I(b) = \{1 \text{ if } b \text{ is true, } 0 \text{ if false}\}$ for an event b , let

$$g(h_r(v_1), h_r(v_2), \dots, h_r(v_k)) = \frac{1}{k} \sum_{i=1}^k (I(v_i \in S) \sum_{e \in \text{out}(v_i)} \bar{f}_2(e)). \quad (2)$$

As $I(v_i) \sum_{e \in \text{out}(v)} \bar{f}_2(e) \in [0, dM]$, the Law of Large Numbers says that (2) stochastically uniformly (with respect to G) converges to the following.

$$\frac{1}{n} \sum_{v \in V(G)} (I(v \in S) \sum_{e \in \text{out}(v)} \bar{f}_2(e)) = \frac{1}{n} \sum_{s \in S} \sum_{e \in \text{out}(v)} \bar{f}_2(e) = |\bar{f}_2| \in [|f^*| - \frac{\varepsilon}{2}, |f^*|].$$

This implies that, for a large enough k , these k , r and g satisfy the requirements. \square

4 Acknowledgement

Thank you for László Lovász for the question and his help in making this paper.

References

- [1] *Alon, Shapira: A characterization of the (natural) graph properties testable with one-sided error*; SIAM Journal on Computing, vol. 37, no. 6, pp. 1703-1727, 2008
- [2] *R Andersen, Y. Peres: Finding sparse cuts locally using evolving sets* CoRR, vol. abs/0811.3779, 2008
- [3] *I Benjamini, O Schramm: Recurrence of distributional limits of finite planar graphs* Electron. J. Probab. 6 no. 23, (2001) 13 pp. (electronic).
- [4] *Benjamini, Schramm, Shapira: Every minorclosed property of sparse graphs is testable* Symposium on Theory of Computing (STOC), 2008, pp. 393-402
- [5] *Bogdanov, Obata, Trevisan: A lower bound for testing 3-colorability in bounded-degree graphs*; Foundations of Computer Science (FOCS) 2002, pp. 93-102
- [6] *Czumaj, Shapira, Sohler: Testing hereditary properties of nonexpanding bounded-degree graphs*; SIAM Journal on Computing, vol. 38, no. 6, pp. 2499-2510, 2009
- [7] *G Elek: Parameter testing with bounded degree graphs of subexponential growth* Random Structures and Algorithms (to appear)
- [8] *G Elek, G Lippner: Borel Oracles. An analytical approach to constant-time algorithms* Proceedings of the American Mathematical Society (to appear)
- [9] *E Fischer: On the strength of comparisons in property testing* Electronic Colloquium on Computational Complexity (ECCC) 8 (2001)

- [10] *O Goldreich, D Ron: Property testing in bounded degree graphs*
Algorithmica, vol. 32, no. 2, pp. 302-343, 2002
- [11] *Lenzen, Oswald, Wattenhofer: What can be approximated locally? Case study: Dominating sets in planar graphs*; SPAA, 2008, pp. 46-54.
- [12] *L Lovász: Very large graphs*
Current Developments in Mathematics, 2008 (2009), pp. 67-128
- [13] *Marko, Ron: Approximating the distance to properties in bounded-degree and general sparse graphs*; ACM Transactions on Algorithms, vol. 5, no. 2, 2009
- [14] *Nguyen, Onak: Constant-Time Approximation Algorithms via Local Improvements*
49th Annual IEEE Symposium on Foundations of Computer Science, 2008, 327-336
- [15] *Parnas, Ron, Rubinfeld: Tolerant property testing and distance approximation*
Journal of Computer and System Sciences, vol. 72, no. 6, pp. 1012-1042, 2006
- [16] *Yoshida, Yamamoto, Ito: An improved constant-time approximation algorithm for maximum matchings*; Symposium on Theory of Computing (STOC), 2009