Distributed Approximations of f-Matchings and b-Matchings in Graphs of Sub-Logarithmic Expansion

Andrzej Czygrinow
School of Mathematical and Statistical Sciences, Arizona State University, Tempe, AZ, USA

Michał Hanćkowiak
Faculty of Mathematics and Computer Science, Adam Mickiewicz University, Poznań, Poland

Marcin Witkowski
Faculty of Mathematics and Computer Science, Adam Mickiewicz University, Poznań, Poland

Abstract

We give a distributed algorithm which given $\epsilon > 0$ finds a $(1-\epsilon)$-factor approximation of a maximum f-matching in graphs $G = (V, E)$ of sub-logarithmic expansion. Using a similar approach we also give a distributed approximation of a maximum b-matching in the same class of graphs provided the function $b : V \to \mathbb{Z}^+$ is L-Lipschitz for some constant L. Both algorithms run in $O(\log^* n)$ rounds in the LOCAL model, which is optimal.

2012 ACM Subject Classification Theory of computation → Distributed computing models

Keywords and phrases Distributed algorithms, f-matching, b-matching

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2021.59

Funding *Andrzej Czygrinow*: Research supported in part by Simons Foundation Grant # 521777.

1 Introduction

A matching in a graph $G = (V, E)$ is a set of edges $M \subseteq E$ such that every vertex $v \in V$ belongs to at most one edge $e \in M$. Although the concept of a matching can be generalized in a few different ways, two natural and useful notions that have been considered in this context are that of an f-matching and a b-matching. Let $G = (V, E)$ be a (simple) graph and let $f : V \to \mathbb{Z}^+$. A set $M \subseteq E$ is called an *f-matching* in G if for every $v \in V$, v is incident to at most $f(v)$ edges from M. Given a function $b : V \to \mathbb{Z}^+$, a *b-matching* in G is a function $x : E \to \mathbb{N}$ such that for every vertex $v \in V$, $\sum_{y \in N(v)} x(vy) \leq b(v)$. Sometimes an f-matching is called a simple b-matching (a capacitated b-matching with bound one for capacities) [15].

In particular, in the case when $f = 1$ $(b = 1)$, then an f-matching (b-matching) is simply a matching. The main difference between f-matchings and b-matchings is that in the case of b-matchings $x : E \to \mathbb{N}$ and the corresponding assignment x for f-matching satisfies $x : E \to \{0, 1\}$. We shall use $\nu_f(G)$ and $\nu_b(G)$ to denote the maximum f-matching and b-matching, that is $\nu_f(G) = \max\{|M| : M \text{ is an } f\text{-matching in } G\}$ and $\nu_b(G) = \max\{\sum_{e \in E} x(e) : x \text{ is a } b\text{-matching in } G\}$. In addition, we use $\nu(G)$ for the size of a maximum matching in G.

We will use the LOCAL model from Peleg's book [14]. In this model a distributed network is modeled as an undirected graph with vertices corresponding to computational units and edges representing bidirectional links between them. The computations are synchronized, and in each round every vertex can send and receive messages from its neighbors, and, in addition, can perform some local computations. Neither the size of messages sent nor the amount of computations is restricted in any way. In addition, vertices have unique identifiers from $\{1, \ldots, n\}$ where n is the order of the graph.

1.1 Related Work

Although distributed algorithms for matchings have been studies extensively, little is known about f-matchings and b-matchings.

A maximal matching gives a 1/2-factor approximation of a maximum and there has been some success in designing distributed algorithms that find a maximal matching. Seminal papers of Hanćkowiak, Karoński and Panconesi [8, 9] give deterministic poly-logarithmic algorithms for the maximal matching problem. Recently, a faster distributed algorithm for the maximal matching problem was given by Fischer [5]. This algorithm runs in $O(\log^2 |V| \log \Delta)$ rounds in graphs $G = (V, E)$ of maximum degree Δ. At the same time Kuhn et al. [12] showed that finding a maximal matching in a graph G of order n and maximum degree Δ requires $\Omega(\log \Delta / \log\log \Delta + \sqrt{\log n / \log\log n})$ rounds.

Distributed approximations for the maximum matching problem which run in a poly-logarithmic number of rounds are known [2, 3]. Quite recently, Ghaffari, Harris and Kuhn [6] gave a $O(\log^2 n \log^5 (\Delta/\epsilon^9))$ algorithm that finds a $(1-\epsilon)$-factor approximation of a maximum matching in a graph on n vertices with maximum degree Δ, and Harris [10] gave fast approximation algorithms for weighted maximum matching and hypergraph matching.

As impressive as these algorithms are, their time complexity is often prohibitively high. Not surprisingly, there has been a lot of interest in designing faster distributed algorithms for special classes of graphs. For planar graphs, there is a distributed deterministic algorithm [4] which given $\epsilon > 0$ finds in a graph G of order n in $O(\log^* n)$ rounds, a matching M such that $|M| \geq (1-\epsilon)\nu(G)$. This algorithm proceeds in two main steps. First, an ad-hoc procedure is used to reduce a given graph $G = (V, E)$ to a graph $G' = (V', E')$ such that (1) $|V'|$ is proportional to the size of a maximum matching and (2) a maximum matching in G' is also maximum in G and (3) G' is still planar. Second, a clustering algorithm is used to partition V' into sets $V_1, \ldots, V_l$ so that it is possible to quickly find optimal solutions in graphs $G'[V_i]$ and combine them to obtain a good approximation in the whole graph G' and so G.

The notions of an f-matching and b-matching are significantly more general. Very little is known about the distributed complexity of f-matchings except what can be concluded from the case $f = 1$. Using the Tutte's construction (see for example [15]) one can reduce b-matchings to matchings but this reduction infuses potentially large complete bipartite graphs.

Hanćkowiak [7] gave a poly-logarithmic running time algorithm for the maximal f-matching problem in general graphs which builds on the approach for matchings [8]. In addition, extending the approach for matchings, Fischer [5] managed to give a deterministic distributed algorithm for $(1/2-\epsilon)$-approximating a maximum f-matching in general graphs that runs in $O(\log^2 \Delta \log 1/\epsilon + \log^* |V|)$ rounds. In the case of a maximum b-matching it is possible to reduce it to the problem of a maximum matching but the reduction leads to graphs which can potentially contain large complete bipartite graphs. Consequently, the graph obtained after reduction can have minors of large cliques. As a result, maximum matching algorithms that exploit sparseness conditions cannot be invoked in the graph obtained by the reduction.

1.2 Results

We will give approximation algorithms for f-matchings and b-matchings. Although our primary interest comes from graphs that are K_m-minor-free for a fixed integer m, it is possible to phrase the results in terms of graphs of sub-logarithmic expansion which generalize the former class. (See Section 2 for definitions.)

Our first contribution is a distributed algorithm which given a graph $G = (V, E)$ of sub-logarithmic expansion and $\epsilon > 0$ finds a $(1-\epsilon)$-factor approximation of a maximum f-matching in G. The algorithm runs in $O(\log^* |V|)$ rounds (Theorem 7). The algorithm proceeds in two main steps, the first, described in detail in what follows, reduces the input graph to a graph in which a maximum f-matching is proportional to the number of vertices and the second invokes the clustering procedure of Amiri et al. [1].

The case of b-matchings is surprisingly more subtle in the realm of sparse graphs and our approach requires an additional assumption about function b. Specifically, we shall require that b is L-Lipschitz for some constant L, which is known to the algorithm. Under the same regime as f-matchings, we again give a $O(\log^* |V|)$-time distributed $(1-\epsilon)$-factor approximation (Theorem 17).

It is known [4] that finding a constant approximation of a maximum matching or maximum independent set in a cycle on n vertices requires $\Omega(\log^* n)$ rounds and so the running time in Theorem 7 and Theorem 17 cannot be improved.

Algorithms for both theorems use in their second main step, i.e. the clustering procedure, the procedure of Amiri et al. [1] which is based on the algorithm of Czygrinow et al. [4]. However the first main step of both algorithms, i.e. the reduction phase is new and relies on the Gallai-Edmonds theorem [11]. The rest of the paper is structured as follows. In Section 2 we introduce necessary terminology. In Section 3 we discuss f-matchings and Section 4 is devoted to b-matchings.

2 Preliminaries

For a positive integer r and a graph H, we say that H is a *minor of depth* r of graph $G = (V, E)$ if for some subgraph $G' = (V', E')$ of G it is possible to partition V' into sets $V_1, \dots, V_l$ so that for every i, $G'[V_i]$ has radius at most r and the graph obtained by contracting each V_i to a vertex is isomorphic to H. We will set $\nabla_r(G) = \max_H \frac{|E(H)|}{|V(H)|}$ where the maximum is taken over all minors H of depth r of G. A graph G is said to have a *bounded expansion* if there exists $g : \mathbb{N} \to \mathbb{N}$ such that for every $r \in \mathbb{N}$, $\nabla_r(G) \le g(r)$. Note that surprisingly many classes of graphs have bounded expansion and we refer to [13] for an extensive discussion. In this paper, we will consider graphs G of sub-logarithmic expansion, that is graphs G such that $\nabla_r(G) \le g(r)$ for some $g(r) \in o(\log r)$. The class was introduced by Amiri et al. [1] where it is shown that the clustering algorithm from [4] works in this more general setting. In the case graph G is K_m-minor-free for some fixed m, $\nabla_r(G)$ can be bounded from above by a constant which depends on m only and so graphs of sub-logarithmic expansion include graphs that are K_m-minor-free for a fixed m. In addition, we obviously have $\nabla_r(G) \le \nabla_{r+1}(G)$.

Our analysis relies on the Gallai-Edmonds theorem. To state it we need some additional terminology. A *k-factor* of a graph G is a spanning k-regular subgraph of G. In particular, a 1-factor is a perfect matching. For a graph H, we use $\mathcal{C}_H$ to denote the set of (connected) components of H. A graph H is called *factor-critical* if, for every $v \in V(H)$, graph $H - v$ has a 1-factor. Note that a component with only one vertex is factor-critical. Gallai-Edmonds theorem (see for example Kotlov [11]) will play a major role in our analysis. For a graph $G = (V, E)$ let A be the set of vertices $v \in V$ such that there is a maximum matching in G that does not cover v. Let $B := N(A) = \{v \in V \setminus A \mid \exists_{w \in A} vw \in E\}$, and let $C := V \setminus (A \cup B)$. Clearly, $\{A, B, C\}$ is a partition V (although some sets can be empty) which we will call a *Gallai-Edmonds decomposition* of graph G. We have the following theorem.

▶ **Theorem 1** (Gallai-Edmonds Theorem). *Let $G = (V, E)$ be a graph and let $\{A, B, C\}$ be a Gallai-Edmonds decomposition of G. Then the following conditions hold.*

(a) *Every odd component H of $G - B$ is factor-critical and $V(H) \subseteq A$.*

(b) *Every even component H of $G - B$ has a perfect matching and $V(H) \subseteq C$.*

(c) *For every non-empty subset $X \subseteq B$, $N(X)$ has vertices from more than $|X|$ odd components of $G - B$.*

In particular, in view of Hall's theorem, B is matchable to the set of odd components in $G - B$.

Fix $g : \mathbb{N} \to \mathbb{N}$ such that $g(r) = o(\log r)$ and $\nabla_r(G) \leq g(r)$. It will be convenient to define $D := 2 \cdot g(1)$. Although the algorithms do not need to have perfect information about g, they certainly need to know D. By definition, for every subgraph H of G we have

$$\sum_{v \in V(H)} deg_H(v) \leq D \cdot |V(H)|. \tag{1}$$

We will call a clique S an i-clique if $|S| = i$.

▶ **Lemma 2.** *Let G be a graph of order n, D be such that $\nabla_0(G) \leq D/2$ and let $i \geq 2$. Then there is a vertex $v \in V(G)$ that belongs to at most $\binom{D}{i-1}$ i-cliques.*

Proof. Note that the statement is true when $i = 2$. For $i \geq 3$, let v be a vertex of minimum degree which by (1) is at most D. Then every i-clique containing v is a subset of $N[v]$. ◀

3 f-matchings

Assume that G satisfies $\nabla_r(G) \leq g(r)$ and $D = 2 \cdot g(1)$. We call $u, v \in V(G)$ i-clones if $N(u) = N(v)$ and $|N(u)| = i$. Note that we consider open neighborhoods and so if uv is an edge then u and v are not clones. The relation of being i-clones, $\sim_i$, is symmetric, transitive, and reflexive on the set $V_i := \{v \mid deg(v) = i\}$. Let $v_1, \ldots, v_l$ be i-clones such that $N(v_1) = \{u_1, \ldots, u_i\}$ and $l \geq i + 1$. Since for every $j \in \{1, \ldots, i\}$, $deg(u_j) \geq l \geq i + 1$, u_j cannot be an i-clone of any other vertex.

In the first phase of the algorithm we trim graph G, discard some edges, and delete isolated vertices so that for the trimmed graph G' we have $\nu_f(G') = \nu_f(G)$ and $\nu_f(G') = \Omega(|G'|)$. Phase 1 is split into two procedures Discard Edges and Trimming.

Procedure Discard Edges (G, i).

- Let G' be the graph obtained from G by deleting the following edges. For every equivalence class $\{v_1, \ldots, v_l\}$ of $\sim_i$ in V_i which satisfies $f(v_1) \geq f(v_2) \geq \cdots \geq f(v_l)$:
  - If $l \geq D$ and $N(v_1) = \{u_1, \ldots, u_i\}$ then, assuming $f(u_1) \leq f(u_j)$ for every $j \in \{1, \ldots, i\}$, delete edges $u_1 v_j$ for $j > \max\{f(u_1), D\}$.
- Return G'.

We will first show that graph G' obtained by Discard Edges has the same size of a maximum f-matching as G does.

▶ **Lemma 3.** *Let $i \in \mathbb{Z}^+$ and let G' denote the graph returned by* Discard Edges(G, i). *Then*

$$\nu_f(G') = \nu_f(G).$$

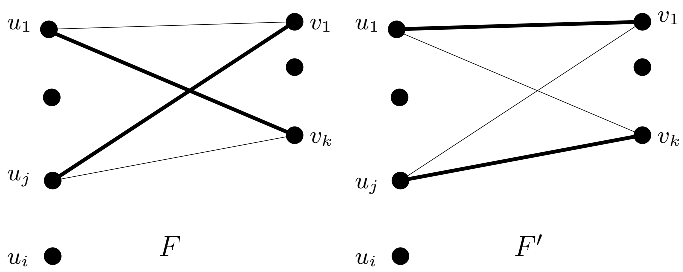

Figure 1 Case 2 of the proof of Lemma 3.

Proof. Clearly $\nu_f(G') \leq \nu_f(G)$. To prove the opposite inequality consider f-matching F in G of size $\nu_f(G)$ that contains as few edges from $E(G) \setminus E(G')$ as possible. Note that only edges incident to u_1 are removed from G. Suppose F contains u_1v_k for some $k > \max\{f(u_1), D\}$. Thus from all edges $u_1v_1, \ldots, u_1v_{f(u_1)}$ that are in G', at least one of them, say u_1v_1, is not in F.

Case 1: If $u_jv_1 \in F$ for at most $f(v_1) - 1$ edges u_jv_1, then consider $F' = F - u_1v_k + u_1v_1$ and note that F' is an f-matching of size $|F|$ with fewer edges from $E(G) \setminus E(G')$ than F.

Case 2: For some $j \in \{1, \ldots, i\}$, $u_jv_1 \in F$ but $u_jv_k \notin F$. Then consider $F' = F - u_1v_k - u_jv_1 + u_1v_1 + u_jv_k$.

If none of the cases 1 and 2 hold, then v_1 is incident to $f(v_1)$ edges $v_1u_j \in F$ such that $j \geq 2$ and v_k is incident to $f(v_1) + 1$ edges from F because we have that $u_jv_k \in F$ whenever $u_jv_1 \in F$ and additionally $u_jv_k \in F$. Thus $f(v_k) > f(v_1)$, but by definition of the algorithm, $f(v_k)$ is at most $f(v_j)$ for every $j \leq \max\{f(u_1), D\}$. ◀

We now consider the main trimming procedure.

Procedure Trimming(G).

- For $i = D - 1$ downto 1, let $G :=$ Discard Edges(G, i).
- Let H be obtained from G by deleting all isolated vertices.
- Return H.

Let H be the graph returned by Trimming(G) for the original graph G. In addition, we use G_i to denote graph G for which we call Discard Edges(G, i) in the ith iteration of the loop in step one. Before proving our main lemma, we show the following observation.

▶ **Lemma 4.** *Let $1 \leq k \leq D - 1$, and let $Y = \{y_1, \ldots, y_j\}$, $|Y| = j$, be k-clones in H such that $N_H(y_1) = \{u_1, \ldots, u_k\}$. Then*

$$j \leq D \cdot f^*,$$

where $f^* = \min_{1 \leq i \leq k}\{f(u_i)\}$.

Proof. We claim that $y_1, \ldots, y_j$ are k-clones in G_k. First note that in view of the main step of Discard Edges, if x is a vertex in G such that $deg_{G_l}(x) \geq D$ for some $l < D$, then $deg_H(x) \geq D$. Indeed, v_j has degree less than D as $l < D$, moreover for u_1 edges u_1v_j can be deleted, but Discard Edges always keeps at least D edges incident to u_1.

We clearly have $N_H(y_l) \subseteq N_{G_k}(y_l)$ for every $l = 1, \ldots, j$. Suppose $k = deg_H(y_l) < deg_{G_k}(y_l)$ for some l. Then y_l had some edges deleted from the set of edges incident to y_l in G_k in iteration i for some $i \leq k$. However in iteration i, we delete edges incident to vertices of degree i or vertices of degree larger than D, keeping at least D of them. Thus either $deg_H(y_l) < k$,which is not possible, or $deg_H(y_l) \geq D$, contradicting $k \leq D - 1$.

Since $y_1, \ldots, y_j$ are k-clones in G_k, we have $j \leq \max\{f^*, D\}$ after step two of DISCARD EDGES(G_k, k). Consequently, $j \leq D \cdot f^*$ because f^* and D are at least one. ◀

By Lemma 3, we have

$$\nu_f(H) = \nu_f(G), \tag{2}$$

and H has no isolated vertices.

We will now establish our second fact which shows that $\nu_f(H)$ is proportional to the order of H.

▶ **Lemma 5.** *Let G be a graph such that $\nabla_1(G) \leq D/2$ and let H be obtained by calling* TRIMMING(G)*. Then $\nu_f(H) \geq c_D|V(H)|$ for some c_D which depends on D only.*

Proof. We will use Theorem 1. Let $\{A, B, C\}$ be a Gallai-Edmonds decomposition of H. Then the size of a maximum matching in H is $\Omega(|B| + \sum_W |W|)$ where the sum is taken over components W of $H - B$ of order at least two. Since a maximum f-matching has size larger than or equal to the size of a maximum matching,

$$\nu_f(H) = \Omega(|B| + \sum_W |W|).$$

Let X be the set of components of $H - B$ of size one. We will identify each component from X with the vertex it contains. Since H has no isolated vertices, every component from X has at least one neighbor in B. Let Y be the set of components in X which have at least D neighbors in B. We have $D \cdot |Y| \leq |E_H(Y, B)| \leq D(|Y| + |B|)/2$ from (1), and so $|Y| \leq |B|$. Thus $\nu_f(H) = \Omega(|Y|)$. For $i = 1, \ldots, D - 1$, let $X_i \subseteq X$ denote the set of components that have exactly i neighbors in B. In the rest of the argument we will show that for $i \in \{1, \ldots, D - 1\}$, $\nu_f(H) = \Omega_D(|X_i|)$. Given that this is the case, we can conclude

$$\nu_f(H) = \Omega(|B| + \sum_W |W| + |Y| + \sum_{i<D} |X_i|) = \Omega_D(|V(H)|).$$

▷ Claim 6. For $i \in \{1, \ldots, D - 1\}$, $\nu_f(H) = \Omega_D(|X_i|)$.

Proof. Let $H^* := (B, \emptyset)$ be the edge-less graph on the set B. Order the vertices in X_i and proceed one by one for as long as possible using the following procedure. Take $y \in X_i$. If $N_H(y)$ is not an i-clique in H^*, then take two non-adjacent vertices v_1, v_2 from $N_H(y) \subseteq B$, add an edge between them to H^*, and delete y from X_i. Formally, we delete all edges incident to y except yv_1, yv_2 and contract the edge yv_1. Let X_i^* denote the set of deleted vertices. Then $|X_i^*| = |E(H^*)| \leq \nabla_1(G)|B| \leq D|B|/2$. We will now bound the number of remaining vertices. Unlike the previous part of the argument, we will show that there is an f-matching F in H such that $|F| \geq \Omega_D(|X_i \setminus X_i^*|)$.

For every vertex $y \in X_i \setminus X_i^*$, $N_H(y)$ is an i-clique in H^*, and if $y, y' \in X_i \setminus X_i^*$ are such that $N_H(y) = N_H(y')$, then y and y' are i-clones. We will put weights on these cliques. Specifically, the weight of an i-clique T is the number of vertices y in $X_i \setminus X_i^*$ such that $N_H(y) = T$. By Lemma 4, for every i-clique T in H^*, the weight of T satisfies $\omega(T) \leq D \cdot f_T$, where $f_T = \min_{u \in T} f(u)$. In addition, we have

$$\sum_T \omega(T) = |X_i \setminus X_i^*|. \tag{3}$$

We will now assign i-cliques to vertices from B so that the following conditions hold:

- If T is assigned to v, then $v \in T$;
- There are at most $\binom{D}{i-1}$ cliques assigned to every vertex v;
- Every i-clique in H^* is assigned to exactly one vertex from B.

This is possible, because by Lemma 2, there is a vertex v in B which is in at most $\binom{D}{i-1}$ i-cliques T in H^*. We assign these cliques to v and continue the process (note that $\nabla_0(H^* - v) \leq \nabla_0(H^*)$). Let E_v denote the set of cliques assigned to v. For every $T \in E_v$, the weight satisfies $\omega(T) \leq D \cdot f_T \leq D \cdot f(v)$ because $v \in T$. Let $T_v \in E_v$ be an i-clique in E_v of the largest weight. Then, by (3),

$$\sum_v \omega(T_v) \geq \sum_v \sum_{T \in E_v} \omega(T) / \binom{D}{i-1} = \frac{\sum_T \omega(T)}{\binom{D}{i-1}} = |X_i \setminus X_i^*| / \binom{D}{i-1}.$$

Since $f(x) \geq 1$ for every x, we can select $\lceil \omega(T_v)/D \rceil \leq f(v)$ vertices $y \in X_i \setminus X_i^*$ to get star F_v with center at v and selected vertices y as its leaves. Since $F_u \cap F_v = \emptyset$ when $u \neq v$, $\bigcup F_v$ is an f-matching in H of size at least $|X_i \setminus X_i^*| / (D \cdot \binom{D}{i-1})$. Consequently,

$$|X_i| = |X_i^*| + |X_i \setminus X_i^*| = O_D(|B|) + O_D(\nu_f(H)) = O_D(\nu_f(H)). \qquad \triangleleft$$

That concludes the proof of Lemma 5. ◀

We will now proceed to Phase 2 of the algorithm. For this phase we need to know that the graph G has sub-logarithmic expansion g. Note that to use the algorithm in Phase 2, constant c_D (or a bound for it) must be provided and, in addition, the algorithm from [1] which is used in step 2 requires some knowledge of the function g.

Procedure Approximation(G, ϵ, g).

- Use Trimming(G) to obtain graph H.
- Use the algorithm from [1] to find a partition of $V(H)$ into sets $V_1, \ldots, V_l$ such that $diam(V_i) = O_{\epsilon,D}(1)$ and the number of edges with endpoints in different sets V_i is at most $\epsilon \cdot c_D |H|$.
- Find an optimal solution F_i in $H[V_i]$ and return $\bigcup F_i$.

▶ **Theorem 7.** *There is a distributed algorithm which given a graph G on n vertices of expansion g such that $g(r) = o(\log r)$ and $\epsilon > 0$ finds an f-matching F in G such that $|F| \geq (1 - \epsilon)\nu_f(G)$. The algorithm runs in $O_{\epsilon,D}(\log^* n)$ rounds.*

Proof. Use Approximation(G, ϵ, g). By (2), $\nu_f(H) = \nu_f(G)$ and by Lemma 5, $\nu_f(H) \geq c_D|H|$. We have

$$\nu_f(G) \leq \sum_i |F_i| + \epsilon \cdot c_D |V(H)| \leq |F| + \epsilon \nu_f(G).$$

Trimming runs in the number of rounds which depends on D only and the running time of the algorithm from [1] is $O(\log^* n)$. ◀

4 b-matchings

In this section, we will discuss maximum b-matchings. All graphs are finite and simple. As before, we will consider graphs with sub-logarithmic expansion g and set $D := 2 \cdot g(1)$.

▶ **Definition 8.** *Given a (simple) graph $G = (V, E)$ and a function $b : V \to \mathbb{Z}^+$, let $\nu_b(G) = \max\{\sum_{e \in E} x(e)\}$ where the maximum is taken over all functions $x : E \to \mathbb{N}$ such that for every vertex $v \in V$, $\sum_{y \in N(v)} x(vy) \leq b(v)$.*

In our reduction algorithm, we will assume that b is L-Lipschitz for some given constant L, that is $b : (V, d_1), \to (\mathbb{Z}^+, d_2)$ satisfies $d_2(b(u), b(v)) \leq Ld_1(u, v)$ for any $u, v \in V$, where d_1 is the metric determined by the distance in graph $G = (V, E)$ and $d_2(a, b) = |a - b|$. The condition will be of no relevance if u and v are in different connected components of G.

To start the analysis we have the following observation.

▶ **Fact 9.** *Let $G = (V, E)$ be a graph with $\nabla_0(G) \leq D/2$ and let $b : V \to \mathbb{Z}^+$. Define $\omega : E \to \mathbb{Z}^+$ as*

$$\omega(uv) := \min\{b(u), b(v)\}. \tag{4}$$

Then (a) $\omega(E) \leq D \cdot b(V)$ and (b) $\nu_b(G) \leq \omega(E)$.

Proof. We shall prove (a) by induction on $|V|$. The base case is obvious. For the inductive step, by (1), the average degree of G is at most D and so there is a vertex $v \in V$ of degree at most D. By induction, $\omega(E(G - v)) \leq D \cdot b(V \setminus \{v\})$ and the weight on edges incident to v is at most $D \cdot b(v)$ by (4). For (b), if $x : E \to \mathbb{N}$ is such that $x(E) = \nu_b(G)$, then for every $e \in E$, $x(e) \leq \omega(e)$. ◀

The general idea behind the approach is the same as in the case of f-matchings. We first reduce graph G using the notion of i-clones and then apply clustering. However, the reduction and the fact that it accomplishes the desired result (Lemma 16) require additional care.

Procedure Modify(G, b, i).

- Let G' be the graph obtained from G by deleting the following edges. For every equivalence class $\{v_1, \ldots, v_l\}$ of $\sim_i$ in V_i in parallel:
  - If $l > D$ and $N(v_1) = \{u_1, \ldots, u_i\}$, then delete vertices $v_{D+1}, \ldots, v_l$. Set $b'(v_D) := b(v_D) + \cdots + b(v_l)$ and $b'(w) := b(w)$ for any other vertex w.
- Return (G', b').

For a graph $G = (V, E)$ and $v \in V$, we used $E_G(v)$ to denote the set of edges $e \in E$ such that $v \in e$.

▶ **Fact 10.** *Let (G', b') denote the pair returned by* Modify(G, b, i).
(a) *Then $\nu_{b'}(G') = \nu_b(G)$.*
(b) *For every $u \in V(G')$ either $deg_{G'}(u) \geq D$ or $N_{G'}(u) = N_G(u)$.*

Proof. We will first show part (a). Let $x : E(G) \to \mathbb{N}$ be a maximum b-matching in G and let $a_p = \sum_{k=1}^{i} x(v_p u_k)$. Let x' be obtained by setting $x'(v_D u_k) := \sum_{p=D}^{l} x(v_p u_k)$ and $x'(v_j u_k) := x(v_j u_k)$ for $j < D$. Then $x' : E(G') \to \mathbb{N}$ and for every $k \in \{1, \ldots, i\}$,

$$\sum_{e \in E_G(u_k)} x(e) = \sum_{e \in E_{G'}(u_k)} x'(e)$$

as the total value of x on edges between u_k and $v_1, \dots, v_l$ stays the same. Obviously, for $j < D$, we have $\sum_{k=1}^{i} x'(v_j u_k) = \sum_{k=1}^{i} x(v_j u_k)$ and, in addition,

$$\sum_{k=1}^{i} x'(v_D u_k) = \sum_{p=D}^{l} a_p \leq \sum_{p=D}^{l} b(v_p) = b'(v_D)$$

as $a_p \leq b(v_p)$. Thus x' is a b'-matching in G' of the same value as x.

Similarly, let $x' : E(G') \to \mathbb{N}$ be a maximum b'-matching in G'. Then $\sum_{e \in E_{G'}(u_k)} x'(e) \leq b'(u_k) = b(u_k)$ for every $k \in \{1, \dots, i\}$, and $\sum_{k=1}^{i} x'(v_D u_k) \leq b'(v_D)$. For $p > D$, taking a_p to be the maximum value such that $0 \leq a_p \leq b(v_p)$ and $\sum_{k=1}^{i} x'(v_D u_k) - a_p \geq 0$, we can write $a_p = \sum_{k=1}^{i} a_{pi}$ where $0 \leq a_{pi} \leq x'(v_D u_k)$ and assign a_{pi} to $v_p u_k$ so that the total of a_p is reassigned from $E(\{v_D\}, \{u_1, \dots, u_i\})$ to $E(\{v_p\}, \{u_1, \dots, u_i\})$. Proceeding one by one with $p = D+1, \dots, l$ gives $x : E(G) \to \mathbb{N}$ such that $x(E(G)) = x'(E(G'))$ and x is a b-matching in G.

For part (b), removing vertices v_j for $j > D$ only affects the neighborhoods of vertices $u_1, \dots, u_i$. However, for every $k \in \{1, \dots, i\}$, $deg_{G'}(u_k) \geq D$ as we keep D of i-clones from $v_1, \dots, v_l$. ◀

After Modify(G, b, i) we obtain a graph (G', b'), where some vertices were removed from G and for some vertices v, we have $b(v) < b'(v)$. In this cases v will be called *special*. Note that it is easy to reverse Modify and obtain x on G from x' on G' by making special vertices distribute x' to deleted vertices as described in the proof.

We will now obtain a reduction of G which will be used in further computations.

Procedure Reduction(G, b).

---

1. For $i = D-1$ downto 1, let $(G, b) :=$Modify(G, b, i).
2. Return (G, b).

▶ **Lemma 11.** *Let (G', b') denote the pair returned by* Reduction(G, b). *Then we have the following:*

(a) $\nu_{b'}(G') = \nu_b(G)$;

(b) *For every $i < D$, if $S \subseteq V$ has size i, then there are at most D i-clones v in G' such that $N_{G'}(v) = S$.*

Proof. The first part follows from Fact 10 (a). For the second part, if $w_1, \dots, w_l$ are i-clones in G' and $N_{G'}(w_1) = S$, then, by Fact 10 (b), we have $N_G(w_k) = S$ for every $k \in \{1, \dots, l\}$. Consequently, $w_1, \dots w_l$ are i-clones in the original graph G and all graphs obtained in step one of Reduction(G, b). Therefore, in the ith iteration, all but at most D of $w_1, \dots, w_l$ are removed, and so $l \leq D$. ◀

To prove our main fact (Lemma 16) we need some additional preparation. In the proof of the main lemma, we will use the Tutte's construction that reduce the problem of b-matchings to matchings.

▶ **Definition 12.** *Let $\tilde{G} = (\tilde{V}, \tilde{E})$ be obtained from $G = (V, E)$ as follows. Replace vertex v from V with an independent set U_v of size $|U_v| = b(v)$ so that for $i \neq j$, $U_v \cap U_w = \emptyset$. If $vw \in E$ then add all edges xy to $\tilde{E}$ for every $x \in U_v$ and $y \in U_w$.*

The following fact is easy to see [16]:

▶ **Fact 13.** $\nu_b(G) = \nu(\tilde{G})$.

Let $\{\tilde{A}, \tilde{B}, \tilde{C}\}$ be a Gallai-Edmonds decomposition of $\tilde{G}$. Recall that properties of the decomposition are given before Theorem 1.

▶ **Lemma 14.** *Let $G = (V, E)$ and $\tilde{G} = (\tilde{V}, \tilde{E})$. Then partition $\{U_v \mid v \in V\}$ of $\tilde{V}$ is a refinement of $\{\tilde{A}, \tilde{B}, \tilde{C}\}$.*

Proof. Let $v \in V$ and let $U_v = \{v_1, \ldots, v_k\}$.

- Suppose for some i, $v_i \in \tilde{A}$ and let $\tilde{M}$ be a maximum matching in $\tilde{G}$ that does not cover v_i. If v_j is covered by $\tilde{M}$, then $v_j w \in \tilde{M}$ for some $w \in \tilde{V}$. By the construction, $v_i w \in \tilde{E}$ and so $\tilde{M} - v_j w + v_i w$ is a maximum matching that does not cover v_j. Consequently, $v_j \in \tilde{A}$.
- If $v_i \in \tilde{B}$ then $v_i \notin \tilde{A}$ and $v_i w \in \tilde{E}$ for some $w \in \tilde{A}$. For any $j \in \{1, \ldots, k\}$, by the previous observation $v_j \notin \tilde{A}$ and by the construction $v_j w \in \tilde{E}$. Thus $v_j \in \tilde{B}$.
- If $v_i \in \tilde{C}$ then for any j, $v_j \in \tilde{C}$, because otherwise $v_j \in \tilde{A} \cup \tilde{B}$ which in view of previous observations gives $v_i \in \tilde{A} \cup \tilde{B}$. ◀

Lemma 14 easily gives the following fact.

▶ **Fact 15.** *Let $\tilde{H}$ be a component of $\tilde{G} - \tilde{B}$ that satisfies $|V(\tilde{H})| \geq 2$ and let $v \in V(G)$. If $U_v \cap V(\tilde{H}) \neq \emptyset$, then $U_v \subseteq V(\tilde{H})$.*

Proof. Suppose $u \in U_v \cap V(\tilde{H})$. Since $|V(\tilde{H})| \geq 2$, there is $w \in V(\tilde{H})$ such that $wu \in \tilde{E}$. By construction, for every $u' \in U_v$, $u'w \in \tilde{E}$. By Theorem 1, $u \notin \tilde{B}$ and so by Lemma 14, $u' \notin \tilde{B}$. Consequently, since $\tilde{H}$ is a component in $\tilde{G} - \tilde{B}$, $u' \in V(\tilde{H})$. ◀

We can now prove the main lemma. We will define ω on the edges set of G' where (G', b') is returned by Reduction(G, b) for the original graph G and function b, that is $\omega : E(G') \to \mathbb{Z}^+$ as $\omega(uv) = \min\{b'(u), b'(v)\}$.

▶ **Lemma 16.** *For every $D, L \in \mathbb{Z}^+$ there is $c_{D,L} > 0$ such that the following holds. Let G be a graph with $\nabla_1(G) \leq D/2$, let $b : V \to \mathbb{Z}^+$ be L-Lipschitz, and let (G', b') be the pair returned by* Reduction(G, b). *Then*

$$\nu_{b'}(G') \geq c_{D,L} \cdot \omega(E(G')). \tag{5}$$

Proof. Consider $\tilde{G}$ obtained from G' and let $\{\tilde{A}, \tilde{B}, \tilde{C}\}$ be the Gallai-Edmonds decomposition of $\tilde{G}$. By Lemma 14, suppressing each set U_v to v gives partition $\{A, B, C\}$ of $V(G)$ defined as $A = \bigcup_{U_v \subseteq \tilde{A}} \{v\}$, $B = \bigcup_{U_v \subseteq \tilde{B}} \{v\}$, $C = \bigcup_{U_v \subseteq \tilde{C}} \{v\}$.

Let $\tilde{H}$ be a component of $\tilde{G} - \tilde{B}$ such that $|V(\tilde{H})| \geq 2$ and let H be obtained by suppressing each $U_v \subseteq V(\tilde{H})$ to v. If $|V(\tilde{H})|$ is even then $\tilde{H}$ has a perfect matching $\tilde{M}$ by Theorem 1, and by Lemma 14, H contains a b'-matching $x_H : E(H) \to \mathbb{N}$ such that for each vertex $v \in V(H)$, $\sum_{e \in E_H(v)} x_H(e) = b'(v)$. Thus

$$x_H(E(H)) = \sum_{v \in V(H)} b'(v). \tag{6}$$

If $|V(\tilde{H})|$ is odd, then $|V(\tilde{H})| \geq 3$ and $\tilde{H}$ is factor-critical. Consequently, there is a b'-matching x_H in H such that $\sum_{e \in E_H(v)} x_H(e) = b'(v) - 1$ for any specified vertex $v \in V(H)$ and $\sum_{e \in E_H(w)} x_H(e) = b'(w)$ for every vertex $w \in V(H) \setminus \{v\}$. Therefore,

$$x_H(E(H)) = \left(\sum_{v \in V(H)} b'(v) \right) - 1 > \frac{1}{2} \sum_{v \in V(H)} b'(v). \tag{7}$$

In addition, by Lemma 1, we know that $\tilde{B}$ is matchable to the set of odd components in $\tilde{G} - \tilde{B}$, and so, by Lemma 14, there is a b'-matching x_B in G' such that

$$x_B(E(B, V \setminus B)) = |\tilde{B}| = \sum_{v \in B} b'(v). \tag{8}$$

Let $G_1 := G'[B \cup \bigcup V(H)]$ where the union is over all components H of $G' - B$ that satisfy $|V(\tilde{H})| \geq 2$. By (6), (7), there is b'-matching x in G_1 such that

$$x(E(G')) \geq \frac{1}{2} \sum_H \sum_{v \in V(H)} b'(v) \tag{9}$$

where the sum is taken over all components H in $G' - B$ of order at least two. Since $\frac{1}{2}(c+d) \leq \max\{c, d\}$, by (8) and (9), there is a b'-matching x in G' (not necessarily G_1), such that

$$x(E(G')) > \frac{1}{2} \left(x_B(E(B, V \setminus B)) + \frac{1}{2} \sum_H \sum_{v \in V(H)} b'(v) \right) \geq b'(V(G_1))/4 > \omega(E(G_1))/(4 \cdot D)$$

where the last inequality follow from Fact 9 (a).

We will now bound the weight of the edges of G' that are not in G_1. Let $\tilde{U}$ be the set of components of size one in $\tilde{G} - \tilde{B}$, and let U be obtained by suppressing each $U_v \subseteq \tilde{U}$ to v. Note that each $u \in U$ is a component of size one in $G' - B$.

Let $U' \subseteq U$ denote the set of vertices $u \in U$ such that $deg_{G'}(u) = |N_{G'}(u) \cap B| \geq D$. From (1), for every set $S \subseteq U'$,

$$D|S| \leq |E(G'[S, N_{G'}(S)])| \leq D(|N_{G'}(S)| + |S|)/2$$

which gives $|S| \leq |N_{G'}(S)|$. Thus, by Hall's theorem, there is a matching of U' in $G'[U', B]$. Let $\{uv_u \mid u \in U', v_u \in B\}$ be such a matching, and let $H := G'[U', B]$. We have $|U'| \leq |B|$ and there is orientation $\overrightarrow{H}$ of the edges of H such that $\Delta^+(\overrightarrow{H}) \leq D$. By definition of ω, for $v \in B$, $\omega(E(v, N^+(v)) \leq D \cdot b'(v)$. Similarly for $u \in U'$,

$$\omega(E(u, N^+(u)) \leq D \cdot b'(u) = D \cdot b(u) \leq D \cdot (b(v_u) + L) \leq D \cdot (b'(v_u) + L) \leq D(L+1) b'(v_u)$$

because $b \leq b'$, b is L-Lipschitz, and $\textsc{Reduction}(G, b)$ does not change the values of b for vertices of degree at least D. Consequently,

$$\omega(E(H)) \leq D(L+2) \sum_{v \in B} b'(v) = D(L+2) \cdot x_B(E(B, V \setminus B)).$$

Let $U'' := U \setminus U'$ and for $1 \leq i < D$, let $U_i \subseteq U''$ denote the set of vertices $u \in U''$ such that $deg_{G'}(u) = |N_{G'}(u) \cap B| = i$. Let $S_v = N_{G'}(v) \cap B$ for $v \in U_i$, and note that by Lemma 11 part (b) for any $v \in U_i$ there are at most D vertices $w \in U_i$ such that $S_w = S_v$. Out of these vertices w all but at most one satisfy $b(w) = b'(w)$ and potentially there is one vertex, called special vertex, w such that $b(w) < b'(w)$.

Before continuing with the main line of the argument we observe that if $w \in S_v$ for some $v \in U_i$, then $b(w) = b'(w)$. Indeed, for $b'(w)$ to be larger than $b(w)$, w would have to have at least D clones in G and v would be the neighbor of all of them. However, $deg_{G'}(v) = i < D$ and by Fact 10 (b), $N_G(v) = N_{G'}(v)$ and so v cannot have D neighbors in B.

We will now go back to the main line of the argument which is similar to the proof of Claim 6. Starting with $H^* := (B, \emptyset)$, we add edges to H^* as in the proof of Claim 6, i.e. if for $u \in U_i$, there are $v, w \in N_{G'}(u) \subseteq B$ such that vw is not already in $E(H^*)$, then add vw to $E(H^*)$ and remove u from U_i. We continue the process for as long as possible. After it ends, H^* satisfies $|E(H^*)| \leq D|B|/2$ because $\nabla_1(G) \leq D/2$.

Let U'_i denote vertices from U_i that correspond to edges of H^* and let $U''_i = U_i \setminus U'_i$. As before, there is an orientation of the edges of H^* such that the maximum out-degree is at most D. If the arc (v_1, v_2) is obtained by suppressing vertex u from U_i, then we say that u belongs to v_1. Let W_v denote the set of vertices that belong to v. Then $|W_v| \leq D$ and for every $u \in W_v$, $\omega(uv) \leq b'(v) = b(v)$ by the previous observation. Since b is L-Lipschitz, the total weight of edges incident to vertices from W_v is at most $Di(2L+1)$ because if $u \in W_v$ and $uv' \in E(G')$ for some $v' \in B$ then $\omega(uv') \leq b'(v') = b(v') \leq b(v) + 2L \leq (2L+1)b(v) = (2L+1)b'(v)$. Consequently, the total weight of edges incident to vertices from U'_i is a most $Di(2L+1)\sum_{v \in B} b'(v) = Di(2L+1)x_B(E(B, V \setminus B))$.

Now consider U''_i. By construction, for every $u \in U''_i$, $N_{H^*}(u)$ is a clique on i vertices. Since $\nabla_0(H^*) < D/2$, by Lemma 2, we can assign i-cliques in H^* to vertices from B so that each vertex has at most $\binom{D}{i-1}$ cliques assigned to it. If $K_1, \ldots, K_l$ are assigned to v, then the number of vertices u in U''_i such that $N_{H^*}(u) = K_j$ for some $j \in \{1, \ldots, l\}$ is a most $Dl \leq D\binom{D}{i-1}$, and the weight on edges incident to them is at most $Di(2L+1)\binom{D}{i-1}b'(v)$. Consequently, the weight of edges incident to vertices from U''_i is at most $Di(2L+1)\binom{D}{i-1}\sum_{v \in B} b'(v) = Di(2L+1)\binom{D}{i-1}x_B(E(B, V \setminus B))$. As a result, the weight of edges incident to vertices from U_i is at most $Di(2L+1)(1+\binom{D}{i-1})\nu_{b'}(G')$. Summing over $i = 1, \ldots, D-1$ shows that the weight of edges of G' that are not in G_1 is $O(\nu_{b'}(G'))$, completing the proof of (5). ◀

As in the case of f-matchings, next algorithm uses constant $c_{D,L}$ and so it needs to know L and $D = 2 \cdot g(1)$, but the clustering procedure from [1] also needs some information about g.

Procedure b-MATCHING APPROXIMATION(G, b, g, L, ϵ).

- Use REDUCTION(G, b) to obtain graph G' and function b'.
- Define the weights ω on $E(G')$ as in (4). Use the algorithm from [1] to find a partition of $V(G')$ into sets $V'_1, \ldots, V'_l$ such that $diam_{G'}(V'_i) = O_{\epsilon,L,D}(1)$ and the total weight of edges with endpoints in different sets V'_i is at most $\epsilon \cdot c_{D,L}\omega(E(G'))$.
- Find an optimal solution x'_i in $G'[V'_i]$ and let $x' : E(G') \to \mathbb{N}$ be given by $x'(e) := x'_i(e)$ if $e \in E(G'[V'_i])$ and $x'(e) := 0$ otherwise. This gives (using the formula in MODIFY) solution x in G. Return x.

▶ **Theorem 17.** *There is a distributed algorithm which given graph $G = (V, E)$ on n vertices, $\epsilon > 0$ and two functions b and g such that $\nabla_r(G) \leq g(r) = o(\log r)$, and $b : V' \to \mathbb{Z}^+$ is L-Lipschitz, finds a b-matching x in (G, b) such that $\sum_{e \in E} x(e) \geq (1-\epsilon)\nu_b(G)$. The algorithm runs in $O_{\epsilon,L,g}(\log^* n)$ rounds.*

Proof. Use b-MATCHING APPROXIMATION(G, b, g, L, ϵ) and note that x' is a b'-matching in G', and so x is a b-matching in G. By Lemma 11, $\nu_{b'}(G') = \nu_b(G)$ and by Lemma 5, $\nu_{b'}(G') \geq c_{D,L}\omega(E(G'))$. Consequently,

$$\begin{aligned}\nu_b(G) = \nu_{b'}(G') &\leq \sum_i x'_i(E(G'[V'_i])) + \epsilon \cdot c_{D,L}\omega(E(G')) \\ &\leq \sum_{e \in E(G)} x(e) + \epsilon\nu_{b'}(G') = \sum_{e \in E(G)} x(e) + \epsilon\nu_b(G).\end{aligned}$$

REDUCTION runs in the number of rounds which depends on D only and the running time of the algorithm from [1] is $O_{\epsilon,L,g}(\log^* n)$. Obtaining solution x from x' requires only a constant number of steps. ◀

References

1 S. A. Amiri, S. Schmid, and S. Siebertz. Distributed dominating set approximations beyond planar graphs. *ACM Transactions on Algorithms, Vol 15, Issue 3, 39:1–39:18.*, 2019.

2 A. Czygrinow and M. Hanćkowiak. Distributed algorithm for better approximation of the maximum matching. *Proc. Computing and Combinatorics: 9th Annual International Conference, COCOON 2003, 242–251.*, 2003.

3 A. Czygrinow, M. Hańćkowiak, and E. Szymańska. A fast distributed algorithm for approximating the maximum matching. *Proceedings of the Annual European Symposium on Algorithms (ESA), 2004, LNCS 3221, 252–263.*, 2004.

4 A. Czygrinow, M. Hanćkowiak, and W. Wawrzyniak. Fast distributed approximations in planar graphs. *International Symposium on Distributed Computing, DISC, Arcachon, France, September 2008, LNCS 5218, 78–92.*, 2008.

5 M. Fischer. Improved deterministic distributed matching via rounding. *31st International Symposium on Distributed Computing (DISC 2017), LIPIcs 91, 17:1–17:15.*, 2017.

6 M. Ghaffari, D. Harris, and F. Kuhn. On derandomizing local distributed algorithms. *Proc. 59th IEEE Symposium on Foundations of Computer Science, 662–673.*, 2018.

7 M. Hanćkowiak. Private communication.

8 M. Hanćkowiak, M. Karoński, and A. Panconesi. On the distributed complexity of computing maximal matchings. *Proc. of the ACM-SIAM Symposium on Discrete Algorithms (SODA), 219–225.*, 1998.

9 M. Hanćkowiak, M. Karoński, and A. Panconesi. A faster distributed algorithm for computing maximal matchings deterministically. *Proc. of the ACM Symposium on Principles of Distributed Computing (PODC), 219–228.*, 1999.

10 D. Harris. Distributed local approximation algorithms for maximum matching in graphs and hypergraphs. *arXiv*, 2018. `arXiv:1807.07645`.

11 A. Kotlov. Short proof of the gallai-edmonds structure theorem. *arXiv*, 2000. `arXiv:math/0011204`.

12 F. Kuhn, T. Moscibroda, and R. Wattenhofer. What cannot be computed locally? *Proc. 23rd Annual ACM Symposium on Principles of Distributed Computing (PODC 2004). New York, USA: ACM Press, 300–309.*, 2004.

13 J. Nesetril and P. Ossona de Mendez. Sparsity: Graphs, structures, and algorithms. *Algorithms and combinatorics 28, Springer, pp. I-XXIII, 1-457.*, 2012.

14 D. Peleg. Distributed computing: A locality-sensitive approach. *Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.*, 2000.

15 A. Shrijver. Combinatorial optimization. *Springer*, 2002.

16 W.T. Tutte. A short proof of the factor theorem for finite graphs. *Canadian Journal of Mathematics 6, pp. 347–352.*, 1954.