*Combinatorics, Probability and Computing* (2001) 10, 397–416. © 2001 Cambridge University Press
DOI 10.1017/S0963548301004849 Printed in the United Kingdom

# Spanning Trees in Dense Graphs

JÁNOS KOMLÓS[1], GÁBOR N. SÁRKÖZY[2]
and ENDRE SZEMERÉDI[1,3]

[1] Department of Mathematics, Rutgers University, Piscataway, NJ 08854, USA

[2] Computer Science Department, Worcester Polytechnic Institute, Worcester, MA 01609, USA

[3] Hungarian Academy of Sciences, Budapest, Hungary

*Received 27 December 1997; revised 7 April 2001*

In this paper we prove the following almost optimal theorem. For any $\delta > 0$, there exist constants $c$ and $n_0$ such that, if $n \geqslant n_0$, $T$ is a tree of order $n$ and maximum degree at most $cn/\log n$, and $G$ is a graph of order $n$ and minimum degree at least $(1/2 + \delta)n$, then $T$ is a subgraph of $G$.

## 1. Introduction

### 1.1. Notation and definitions

We will sometimes use $+$ and $\Sigma$ to denote disjoint unions of sets. $V(G)$ and $E(G)$ denote the vertex set and the edge set of the graph $G$, and we write $v(G) = |V(G)|$ (order of $G$) and $e(G) = |E(G)|$ (size of $G$). $(A, B)$ or $(A, B, E)$ denote a bipartite graph $G = (V, E)$, where $V = A + B$, and $E \subset A \times B$. In general, given any graph $G$ and two disjoint subsets $A, B$ of $V(G)$, the pair $(A, B)$ is the graph restricted to $A \times B$. $N(v)$ is the set of neighbours of $v \in V$. Hence the size of $N(v)$ is $|N(v)| = \deg(v) = \deg_G(v)$, the degree of $v$. $\delta(G)$ stands for the minimum, and $\Delta(G)$ for the maximum degree in $G$. More generally, for $A \subset V(G)$ we write $N(A) = \cup_{v \in A} N(v)$. $N(u, v) = N(u) \cap N(v)$ is the set of common neighbours. For a vertex $v \in V$ and set $U \subset V - \{v\}$, we write $\deg(v, U)$ for the number of edges from $v$ to $U$. The length of a path is the number of its edges, and the distance between two vertices $x$ and $y$, denoted by $\mathrm{dist}(x, y)$, is the minimum length of an $x - y$ path. We let $e(A, B)$ denote the number of edges of $G$ with one end-point in $A$ and the other in $B$. For non-empty $A$ and $B$,

$$d(A, B) = \frac{e(A, B)}{|A||B|}$$

is the *density* of the graph between $A$ and $B$.

https://doi.org/10.1017/S0963548301004849 Published online by Cambridge University Press

**Definition 1.** The pair $(A, B)$ is *$\varepsilon$-regular* if

$$X \subset A,\ Y \subset B,\ |X| > \varepsilon|A|,\ |Y| > \varepsilon|B|$$

imply

$$|d(X, Y) - d(A, B)| < \varepsilon;$$

otherwise it is $\varepsilon$-irregular.

**Definition 2.** $G = (A, B, E)$ is said to be *$(\varepsilon, \delta)$-super-regular* if it is $\varepsilon$-regular, and

$$\deg(a) > \delta|B| \quad \text{for all } a \in A, \quad \text{and} \quad \deg(b) > \delta|A| \quad \text{for all } b \in B.$$

Given a rooted tree and a vertex $v$, we write $A(v)$ for the set of ancestors of $v$, $C(v)$ for the set of children of $v$, $G(v)$ for the set of grandchildren of $v$, and $T(v)$ for the set of descendants of $v$ (including $v$ itself). In a tree $T$ we denote the set of leaves by $L(T)$. In a star $S$ we denote the middle vertex (or the root) by $M(S)$. A rooted forest is a forest of rooted trees. We will use $a \ll b$ to denote that $a$ is sufficiently small compared to $b$. For simplicity, we do not always compute these dependences, although it could be done.

### 1.2. Packings and subtrees in dense graphs

**Definition 3.** An *embedding* of a graph $G = (V, E)$ into a graph $G' = (V', E')$ is an edge-preserving one-to-one map from $V$ to $V'$, that is, an injection $\varphi : V \to V'$ such that $\{u, v\} \in E$ implies $\{\varphi(u), \varphi(v)\} \in E'$.

Such a map $\varphi$ induces an injection from $E$ into $E'$; we will use the notation $\varphi$ for that map, too. In particular, we will write $\varphi(E)$ for the image set of the edges of $G$.

**Definition 4.** Given a set of graphs $G_1, G_2, \ldots, G_l$, we say that $G_1, G_2, \ldots, G_l$ can be *packed* into $G$ if we can find embeddings $\varphi_i$ of $G_i$ into $G$ such that the edge sets $\varphi_i(E(G_i))$ are pairwise disjoint. If $G = K^n$, the complete graph on $n$ vertices, then we say simply that there is a packing of $G_1, G_2, \ldots, G_l$.

The notion of packing plays an important role in the investigation of computational complexity of graph properties. Thus it is not surprising that in recent research literature there is considerable interest in packing-type results and problems (see, *e.g.*, [2, 3, 4, 13]).

Along these lines, solving an old conjecture of Bollobás [2] we proved the following [6].

**Theorem 1.1.** *Let $\Delta$ and $c < 1/2$ be given. Then there exists a constant $n_0$ with the following properties. If $n \geqslant n_0$, $T$ is a tree of order $n$ with $\Delta(T) \leqslant \Delta$, and $G$ is a graph of order $n$ with $\Delta(G) \leqslant cn$, then there is a packing of $T$ and $G$.*

We proved this theorem in the following equivalent embedding form.

**Theorem 1.1′.** *Let $\Delta$ and $\delta > 0$ be given. Then there exists a constant $n_0$ with the following properties. If $n \geqslant n_0$, $T$ is a tree of order $n$ with $\Delta(T) \leqslant \Delta$, and $G$ is a graph of order $n$ with $\delta(G) \geqslant ((1/2) + \delta)n$, then $T$ is a subgraph of $G$.*

https://doi.org/10.1017/S0963548301004849 Published online by Cambridge University Press

In this form, the theorem can be considered as a generalization of Dirac's theorem on Hamiltonian paths. The proof in [6] laid the foundations for a series of papers in which we developed a new method based on the Regularity Lemma and the Blow-up Lemma (see [6, 8, 9, 10, 7, 11]). The method is usually applied to finding certain constant maximum degree spanning subgraphs in dense graphs. Typical examples are spanning trees (above), Hamiltonian cycles or powers of Hamiltonian cycles [10, 7] or $H$-factors for a fixed graph $H$ [11].

Returning to Theorem 1.1′ and the original paper [6], it is a natural question (first asked by P. Erdős) whether in this theorem the constant degree requirement on $T$ can be relaxed. The purpose of this paper is to prove the following almost optimal result in this direction.

**Theorem 1.2.** *Let $\delta > 0$ be given. Then there exist constants $c$ and $n_0$ with the following properties. If $n \geqslant n_0$, $T$ is a tree of order $n$ with $\Delta(T) \leqslant cn/\log n$, and $G$ is a graph of order $n$ with $\delta(G) \geqslant \big((1/2) + \delta\big)\, n$, then $T$ is a subgraph of $G$.*

This result is optimal apart from a constant factor. In fact, let $c_1$ be a sufficiently large constant, and let $T$ and $G$ be as follows: $T$ is a rooted tree with root $r$ and depth 2, $\deg(r) = (\log n)/c_1$ and the degrees of the children of $r$ are as equal as possible; $G$ is a random $n$-graph with edge-probability 0.9. An easy calculation shows that, with high probability for large $n$, $G$ satisfies $\delta(G) > 0.8n$ but $T$ is not a subgraph of $G$.

In addition to being an almost optimal result, Theorem 1.2 is the first case when we were able to apply the Regularity Lemma/Blow-up Lemma method for finding spanning subgraphs with higher than constant maximum degree.

## 2. Main tools

### 2.1. Regularity Lemma

In the proof the following lemma of the third author plays a central role.

**Lemma 2.1 (Regularity Lemma).** *For every $\varepsilon > 0$ and positive integer $m$ there are positive integers $M = M(\varepsilon, m)$ and $N = N_0(\varepsilon, m)$ with the following property: for every graph $G$ with $n \geqslant N$ vertices there is a partition of the vertex set into $k + 1$ classes (clusters)*

$$V = C_0 + C_1 + C_2 + \cdots + C_k$$

*such that*

- $m \leqslant k \leqslant M$,
- $|C_0| < \varepsilon n$,
- $|C_1| = |C_2| = \cdots = |C_k|$,
- *at most $\varepsilon k^2$ of the pairs $(C_i, C_j)$ are $\varepsilon$-irregular.*

The proof can be found in [15], although an earlier version appeared in [16].

We will also use the following easy consequence of the Regularity Lemma (see [6]).

https://doi.org/10.1017/S0963548301004849 Published online by Cambridge University Press

**Lemma 2.2 (Degree form of the Regularity Lemma).** *For every $\varepsilon > 0$ and positive integer $m$ there are positive integers $M = M(\varepsilon, m)$ and $N = N_0(\varepsilon, m)$ with the following property: if $G = (V, E)$ is any graph with $n \geqslant N$ vertices and $d \in [0, 1]$ is any real number, then there is a partition of the vertex set into $k+1$ clusters $C_0, C_1, \ldots, C_k$, and there is a subgraph $H \subset G$ with the following properties:*

- $V(H) = V(G)$,
- $m \leqslant k \leqslant M$,
- $|C_0| \leqslant \varepsilon n$,
- $|C_1| = |C_2| = \cdots = |C_k|$,
- $\deg_H(v) > \deg_G(v) - (d + \varepsilon)n$ *for all* $v \in V$,
- *all* $H|_{C_i}$ *are empty (the* $C_i$*s are independent in* $H$*),*
- *all pairs* $H|_{C_i \times C_j}$, $1 \leqslant i < j \leqslant k$, *are* $\varepsilon$*-regular, each with a density either* 0 *or exceeding* $d$.

*Furthermore, if $G$ satisfies $\delta(G) \geqslant (\frac{1}{2} + 2d + 2\varepsilon)v(G)$, then the following additional conditions can also be met:*

- $k$ *is even,*
- *the clusters can be matched* $\sum C_i = (\sum A_i) + (\sum B_i)$ *in such a way that* $H$ *restricted to any pair* $(A_i, B_i)$ *is* $(\varepsilon, d)$*-super-regular.*

We will refer to the pairs $(A_i, B_i)$ as the 'edges of the 1-factor'.

In [6] the proof of Theorem 1.1′ reduced the general embedding problem to two special cases, namely, embedding forests of stars and forests of paths into bipartite graphs. Here we again use these two special cases in Section 4.1.

### 2.2. Embedding a forest of stars

Given a bipartite graph $G = (A, B, E)$ and a vector $\underline{d} = (d_a : a \in A)$, $\sum d_a \leqslant |B|$ of positive integers, we say that $G$ has a $\underline{d}$-matching from $A$ to $B$ if there is a partition $B = B_0 + \sum_{a \in A} B_a$ such that, for all $a \in A$,

$$|B_a| = d_a \quad \text{and} \quad \{a\} \times B_a \subset E.$$

**Lemma 2.3 (Embedding a forest of stars).** *Let $G = (A, B, E)$ be a bipartite graph, and write*

$$\delta_A = \min_{a \in A} \deg(a), \quad \delta_B = \min_{b \in B} \deg(b).$$

*We are also given a vector $\underline{d} = (d_a : a \in A)$, $\sum d_a \leqslant |B|$ of positive integers, and we write $\Delta = \max d_a$.*

*If $G$ has the weak expanding property*

$$(X \subset A,\ Y \subset B,\ |X| > \delta_A/\Delta,\ |Y| > \delta_B) \quad \textit{imply } e(X, Y) > 0,$$

*then there is a $\underline{d}$-matching from $A$ to $B$.*

This lemma is a relatively straightforward consequence of Hall's theorem (for details see [6]).

https://doi.org/10.1017/S0963548301004849 Published online by Cambridge University Press

### 2.3. Embedding a forest of paths

**Definition 5.** A *four-layer graph* is a graph $G = (V, E)$ where $V = V_1 + V_2 + V_3 + V_4$, $|V_i| = m$, $1 \leqslant i \leqslant 4$, and $E \subset \sum_{i=1}^{3} A_i \times A_{i+1}$.

A *four-layer super-regular graph* is a four-layer graph in which the graphs $G_{|V_i + V_{i+1}}$ are all super-regular.

**Lemma 2.4.** *For every $\delta > 0$, there are $\varepsilon, m_0 > 0$ such that the following holds for all $m \geqslant m_0$. If $G$ is a four-layer $(\varepsilon, \delta)$-super-regular graph on $4m$ vertices, then for any one-to-one map between $V_1$ and $V_4$ there exists a set of $m$ vertex-disjoint paths of order 4, each one connecting mapped pairs.*

We note that three layers would not be sufficient, since a vertex on layer 1, while certainly connected (by paths of order 3) to at least $(1 - \varepsilon)n$ vertices of layer 3, is not necessarily connected to *all* vertices of layer 3. Furthermore, it would be much easier to prove a 5-layer version of the lemma: namely, one can choose a random one-to-one map between $V_1$ and $V_3$, and this has to be modified at only a few vertices. We just used the 4-layer version because it is optimal.

### 2.4. Random $\varepsilon$-regular subgraphs

Our final tool is the following lemma.

**Lemma 2.5.** *Let $0 < \varepsilon \ll d$. There exist constants $C, n_0 > 0$ and $\varepsilon \ll \varepsilon' \ll d$ with the following properties. Let $n \geqslant n_0$, $m \geqslant m' \geqslant n_0$, and let $(A, B)$ be an $\varepsilon$-regular pair of density $d$ with $|A| = m$ and $|B| = n$. Let us select at random a subset $A' \subset A$ with $|A'| = m'$. Then with probability close to 1 we have the following:*

(1) *the pair $(A', B)$ is $\varepsilon'$-regular,*
(2) *if $m' \geqslant C \log n$, then (with probability $1 - O(ne^{-m'})$) every vertex $v \in B$ has degree $\deg(v, A') \geqslant \deg(v, A)\frac{m'}{m} - \varepsilon m'$.*

Indeed, (2) is just the consequence of the law of large numbers.

For (1), observe that, with high probability (about $1 - O(m'e^{-\varepsilon m'})$), apart from at most $4\varepsilon|A'|^2$ exceptional pairs in $A'$, for a pair $\{x, x'\} \subset A'$ we have

$$|N(x, x')| \geqslant (d - \varepsilon)^2|B|.$$

This so-called quasi-random property is known to imply (1) (see [14, 1, 5]).

## 3. Sketch of the proof of Theorem 1.1′

In [6] the proof of Theorem 1.1′ followed the following rough outline. We started by splitting the graph $G$ into a 1-factor of super-regular pairs as guaranteed by Lemma 2.2, and splitting the tree $T$ into a constant number of small trees. These small trees were assigned to the edges of the 1-factor first in an approximate manner, and then, moving some vertices around, we made this assignment exact. Then we embedded the bulk of $T$

https://doi.org/10.1017/S0963548301004849 Published online by Cambridge University Press

with the application of a simple greedy technique, and finally an application of Lemma 2.3 or Lemma 2.4 concluded the proof.

## 4. Proof of Theorem 1.2

### 4.1. Special cases when the proof of Theorem 1.1′ works

Here we will use the following main parameters:

$$\beta \ll \varepsilon' \ll \alpha \ll \delta. \tag{4.1}$$

Let us make $T$ rooted by picking an arbitrary vertex $r$ as the root. The proof method in [6] works with minor modifications for $\Delta(T) \leqslant cn/\log n$ as well in the following special cases:

- $T$ contains at least $\alpha n$ vertex-disjoint paths of length 3 (induced, with counting edges),
- $T$ contains at least $\alpha n$ non-leaf vertices with at least one leaf child.

We sketch briefly that this is in fact the case. For further details see [6].

We decompose $T$ into small trees. Find a vertex $v$ such that

- $|T(v)| \geqslant \beta n$, and
- $|T(u)| < \beta n$ for every $u \in C(v)$.

Each subtree $T(u)$ is a piece of the decomposition and the $(u,v)$ edges are called bridges. We cut these pieces out and continue with the remaining part of $T$. The obtained pieces in this decomposition are all of size less than $\beta n$, and their number is at most

$$\left(\frac{1}{\beta}+1\right)\frac{cn}{\log n}.$$

We decompose $G$ into a matching of clusters using Lemma 2.2, with $\varepsilon'$ playing the role of $\varepsilon$. We define two types of buffer vertices $B_1, B_2$ in $T$. In both cases these vertices are disjoint from the bridges. In the first case they are $\lfloor \alpha n/2 \rfloor$ vertex-disjoint paths of length 3, where the first type of buffer vertices ($B_1$) are the end-points of the paths, and the other type ($B_2$) is the set of the middle vertices on the paths. In the second case, $B_1$ is a set of $\lfloor \alpha n/2 \rfloor$ non-leaf vertices with at least one leaf child, and $B_2$ is the set of their leaf children.

The crucial observation is that the number of bridges is much less than the number of both types of buffer vertices. We first assign the pieces to clusters as follows. We assign the next piece to the next pair where we have so far assigned fewer vertices to both clusters than the actual number of vertices in the two clusters. We fill up the two clusters in a balanced way and we ensure that we assign sufficiently many $B_1$ (and thus also $B_2$) buffer vertices to each cluster. This can easily be achieved by a simple greedy strategy. Then in each pair we consider the assigned buffer vertices and we set aside random buffer zones for them (separately for $B_1$ and $B_2$). We start the actual embedding.

- We embed the pieces in a top-down breadth-first manner, first finishing the embedding of a whole piece (apart from the $B_2$ buffer vertices) before moving over to the next piece. Thus, we first embed the piece $P_1$ containing $r$ and we put the pieces that are connected to $P_1$ through bridges in a left-to-right ordering into a first-in first-out queue. We remove the first piece $P$ from the queue, we embed it, and again we put the pieces

https://doi.org/10.1017/S0963548301004849 Published online by Cambridge University Press

that are connected to $P$ through bridges in a left-to-right ordering into the end of the queue. We continue in this fashion until the queue is empty, by always embedding the first piece from the queue, and putting the pieces that are connected to this piece through bridges in a left-to-right ordering into the end of the queue.

- We embed the non-bridge vertices of a piece into the pair where the piece is assigned with a greedy strategy. That is, we always embed into a vertex that has many neighbours in the remaining non-buffer zone vertices, so we can continue the embedding.
- We handle the bridges as follows. Suppose $(u, v)$ is a bridge between pieces $P$ and $P'$ where $u \in P', v \in P$ and $u \in C(v)$. Assume that $v$ is embedded into cluster $C$, and $P'$ is assigned to $(G, G')$ such that the children of $u$ are assigned to $G$. Find a $D$ such that $v$ has many neighbours in $D$, and $(D, G)$ is an $\varepsilon$-regular pair with high density. Embed $u$ into a vertex (buffer vertex if the rest of $N_D(v)$ is occupied already) in $D$ with large degree in $G$. We embed the children of $u$ into non-buffer vertices among these neighbours and continue the greedy strategy for the embedding of $P'$.
- We embed the non-buffer vertices into non-buffer zone vertices, and we embed the $B_1$ buffer vertices into their buffer zones. The $B_2$ buffer vertices (so the middle vertices of the paths in the first case, and the leaves in the second) remain unembedded at this point.
- When we run out of empty non-buffer vertices or empty $B_1$ buffer zone vertices in a cluster, we combine these two sets and continue the embedding. Finally when this set becomes small as well, we combine it with the $B_2$ buffer zone and finish the embedding apart from the $B_2$ buffer vertices.
- After an adjusting step, since the number of bridges is small, for the embedding of the $B_2$ buffer vertices in each pair we can use Lemma 2.3 or Lemma 2.4, respectively, in the two cases.

The missing details can be found in [6]. Therefore we can assume that neither of these two cases holds. This implies that $T$ contains mostly leaves; more precisely it contains at least $(1 - \sqrt{\alpha})n$ leaves.

We distinguish two cases, according to the distribution of the leaves among the levels

$$V_i = \{v : v \in T, \mathrm{dist}(r, v) = i\}.$$

In the remainder of the proof we will use the following main parameters:

$$\alpha \ll \gamma \ll \varepsilon \ll d \ll \delta. \tag{4.2}$$

### 4.2. The well-distributed case

In this case we assume

$$|V_i| < \gamma n \quad \text{for all } i, \tag{4.3}$$

that is, the leaves are distributed relatively evenly among the levels. The other case, when at least one level contains a large portion of the leaves, is discussed in the next section.

We break up the proof into a few steps.

https://doi.org/10.1017/S0963548301004849 Published online by Cambridge University Press

**Step 1:** Decomposition of $T$.
We define

$$U_j = \bigcup_{i \equiv j \pmod{1/d}} V_i \quad \text{for } 0 \leqslant j < 1/d$$

(for simplicity we assume that $1/d$ is an integer). Let us consider $U_j$ with the maximum number of leaves in it ($\geqslant d(1-\sqrt{\alpha})n$). The levels in $U_j$ divide $T$ into regions $R_0, R_1, R_2, \ldots$, where

$$\begin{aligned} R_0 &= \{V_t : 0 \leqslant t \leqslant j\}, \quad \text{and} \\ R_i &= \left\{V_t : \frac{i-1}{d} + j + 1 \leqslant t \leqslant \frac{i}{d} + j\right\} \quad \text{for } i = 1, 2, \ldots. \end{aligned}$$

Since $R_0$ may be a short region, for simplicity we remove $V_j$ from $U_j$. $U_j$ still contains at least $dn/2$ leaves. We consider the grandparents in $T$ of the leaves in $U_j$, and we denote these vertices by $r_1, r_2, \ldots, r_l$ (roots) in a top-down, breadth-first order. Let $T_i$ denote the depth-2 subtree of $T$ with root $r_i$ and leaves in $U_j$. This subforest (denoted by $F$) will play a major role in the rest of the proof in the well-distributed case. The leaves in most of this subforest will be embedded only at the very end to the buffer zones by using a König–Hall argument. Note that $r_i$ may have children in $T$ not in $T_i$, and also a child of $r_i$ in $T_i$ may have children not in $T_i$.

**Step 2:** Decomposition of $G$.
We partition $G$ into clusters using Lemma 2.2 with $\varepsilon$ and $d$ as in (4.2). Note that we will not use the matching part of Lemma 2.2 but use a covering by stars instead.

**Step 3:** Construction of the star covering.
We define the following so-called *reduced graph* $G_r$: The vertices of $G_r$ are the clusters $C_i$, $1 \leqslant i \leqslant k$, in the partition and there is an edge between two clusters if they form an $\varepsilon$-regular pair in $H$ with density exceeding $d$. Since, in $H$,

$$\delta(H) > \delta(G) - (d+\varepsilon)n \geqslant \left(\frac{1}{2} + \delta\right) - (d+\varepsilon)n,$$

an easy calculation shows that in $G_r$ we have

$$\delta(G_r) \geqslant \left(\frac{1}{2} + \frac{\delta}{2}\right) k. \tag{4.4}$$

This implies that, for all $C_1, C_2 \in G_r$, we have

$$|N_{G_r}(C_1) \cap N_{G_r}(C_2)| \geqslant \delta k. \tag{4.5}$$

This time, for the finishing König–Hall-type argument we need a covering (of most of the clusters) by a constant number of stars in $G_r$ instead of a covering by independent edges (matching). We are going to construct a constant number of stars $S_1, S_2, \ldots, S_s$ in $G_r$ with the following properties.

(1) $L(S_i) \cap L(S_j) = \emptyset$ for every $1 \leqslant i < j \leqslant s$ (but note that $M(S_i) \in S_j$ or $M(S_j) \in S_i$, or even $M(S_i) = M(S_j)$, is allowed).

https://doi.org/10.1017/S0963548301004849 Published online by Cambridge University Press

(2) For every $S_i, 1 \leqslant i \leqslant s$ we have the following:

$$|N_{G_r}(M(S_j)) \cap L(S_i)| \geqslant d^2|L(S_i)| \quad \text{for every } 1 \leqslant j \leqslant s, j \neq i.$$

These clusters in $L(S_i)$ are called bridge clusters from $S_i$ to $S_j$.

(3) $n - \sum_{i=1}^{s} |L_G(S_i)| \leqslant \varepsilon n$, where $L_G(S_i)$ denotes the set of vertices in $G$ in the leaf clusters of $S_i$.

We construct these stars in the following way. For the first star $S_1$, take an arbitrary cluster as $M(S_1)$, and $L(S_1)$ is the neighbour of $M(S_1)$ in $G_r$, so from (4.4) we have

$$|L(S_1)| \geqslant \left(\frac{1}{2} + \frac{\delta}{2}\right) k.$$

In addition, select randomly a subset $L'(S_1) \subset L(S_1)$ of size $d|L(S_1)|$ (assuming that this is an integer). By the law of large numbers and (4.5), with high probability, we have

$$|N_{G_r}(C) \cap L'(S_1)| \geqslant \frac{\delta}{2}|L'(S_1)| = \frac{\delta}{2} d|L(S_1)| \gg d^2|L(S_1)| \tag{4.6}$$

for every cluster $C$, since, from (4.5),

$$|N_{G_r}(C) \cap L(S_1)| \geqslant \delta k.$$

Fix a subset $L'(S_1)$ for which (4.6) is true for every cluster $C$ in $G_r$.

For $S_2$, if there is a cluster in $G_r \setminus S_1$ which covers at least half of the clusters in $G_r \setminus S_1$, then this cluster is $M(S_2)$ and its neighbours in $G_r \setminus S_1$ are $L(S_2)$. Otherwise, from (4.4), we have $d_{G_r}(S_1, G_r \setminus S_1) > 1/2$, so there must exist a cluster in $S_1$ which covers at least half of the clusters in $G_r \setminus S_1$. In this case, this cluster is $M(S_2)$ and its neighbours in $G_r \setminus S_1$ are $L(S_2)$. In addition, in both cases we add some more clusters to $L(S_2)$. We select randomly a subset $L'(S_2)$ of size $d|L(S_2)|$ from $N_{G_r}(M(S_2)) \setminus L'(S_1)$. Note that $L'(S_2)$ may contain clusters from $L(S_1)$. Again, by the law of large numbers and (4.5), with high probability, we have

$$|N_{G_r}(C) \cap L'(S_2)| \geqslant \frac{\delta}{2}|L'(S_2)| = \frac{\delta}{2} d|L(S_2)| \gg d^2|L(S_2)| \tag{4.7}$$

for every cluster $C$, since, from (4.5),

$$|N_{G_r}(C) \cap \big(N_{G_r}(M(S_2)) \setminus L'(S_1)\big)| \geqslant \delta k - |L'(S_1)| \geqslant \frac{3}{4}\delta k.$$

Fix a subset $L'(S_2)$ for which (4.7) is true for every cluster $C$ in $G_r$. Remove the clusters in $L'(S_2) \cap L(S_1)$ from $L(S_1)$ and add them to $L(S_2)$ as well, so now $L'(S_2) \subset L(S_2)$.

We continue in this fashion. Assume that $S_1, \ldots, S_{i-1}$ are already defined. To get $S_i$, $M(S_i)$ is a cluster that covers at least half of the clusters in $G_r \setminus \cup_{j=1}^{i-1} S_j$, and $L(S_i)$ is the set of its neighbours in $G_r \setminus \cup_{j=1}^{i-1} S_j$. In addition, we select randomly a subset $L'(S_i)$ of size $d|L(S_i)|$ from $N_{G_r}(M(S_i)) \setminus \cup_{j=1}^{i-1} L'(S_j)$. By the law of large numbers and (4.5), with high probability, we have

$$|N_{G_r}(C) \cap L'(S_i)| \geqslant \frac{\delta}{2}|L'(S_i)| = \frac{\delta}{2} d|L(S_i)| \gg d^2|L(S_i)| \tag{4.8}$$

for every cluster $C$, since, from (4.5),

$$|N_{G_r}(C) \cap \big(N_{G_r}(M(S_i)) \setminus \cup_{j=1}^{i-1} L'(S_j)\big)| \geqslant \delta k - |\cup_{j=1}^{i-1} L'(S_j)| \geqslant \frac{3}{4}\delta k.$$

https://doi.org/10.1017/S0963548301004849 Published online by Cambridge University Press

Fix a subset $L'(S_i)$ for which (4.8) is true for every cluster $C$ in $G_r$. Remove the clusters in $L'(S_i) \cap (\cup_{j=1}^{i-1} L(S_j))$ from $\cup_{j=1}^{i-1} L(S_j)$, and add them to $L(S_i)$, so now $L'(S_i) \subset L(S_i)$.

We continue in this fashion until $n - \sum_{j=1}^{i} |L_G(S_j)| \leqslant \varepsilon n$. Since every star covers at least half of the remaining clusters, we have $s \leqslant \lceil \log \frac{1}{\varepsilon} \rceil$. Properties (1) and (3) are satisfied by the construction. For property (2), observe that during the whole process we removed at most $d|L(S_i)|$ clusters from $L(S_i) \setminus L'(S_i)$ when we constructed the stars $S_j, i < j \leqslant s$. This fact, $L'(S_i) \subset L(S_i)$ and (4.8) applied to $C = M(S_j)$ show that property (2) of the star covering is satisfied as well.

**Step 4:** Setting buffer vertices aside.
The buffer vertices in $T$ are the leaves in the $T_i$s. The buffer zones ($\mathrm{buf}(C)$) are simply subsets of every leaf cluster in the stars (so in $\cup_{i=1}^{s} L(S_i)$) such that their sizes are as equal as possible and the total number of vertices in them is the same as the number of buffer vertices in $T$. Thus all the buffer zones have size at least $\frac{d}{2}\frac{n}{k}$. We select the sets $\mathrm{buf}(C)$ at random, uniformly from among all subsets of $C$ of the given size. Denote $\mathrm{buf}(G) = \cup_C \mathrm{buf}(C)$, and, in general, for a collection of clusters $K$ let $\mathrm{buf}(K)$ denote $= \cup_{C \in K} \mathrm{buf}(C)$. This selection guarantees, by the law of large numbers (as in Lemma 2.5), that these buffers have the following property:

- $\deg(v, \mathrm{buf}(C)) \geqslant (d - 2\varepsilon)|\mathrm{buf}(C)|$ for all clusters $C$, and for all $v \in V$ such that $\deg(v, C) \geqslant (d - \varepsilon)|C|$.

**Step 5:** Decomposing $F$ into two subforests $F_1$ and $F_2$, assigning the $T_i$s in $F_1$ to stars.
In this step, for technical reasons we decompose $F$ into two subforests $F = F_1 \cup F_2$, where $F_2$ will consist of the last few levels of the $T_i$s. We assign only the $T_i$s in $F_1$ to stars; $F_2$ will be used later (Step 9) for the handling of the various exceptional vertices.

Define $s_1'$ by

$$\sum_{i=1}^{s_1'} |L(T_i)| \leqslant (1 - \varepsilon^{1/4})|\mathrm{buf}(L(S_1))|, \quad \text{but} \quad \sum_{i=1}^{s_1'+1} |L(T_i)| > (1 - \varepsilon^{1/4})|\mathrm{buf}(L(S_1))|.$$

Let $s_1$ be the largest integer for which $s_1 \leqslant s_1'$, and $r_{s_1}$ is the last root on a level. Relations (4.2) and (4.3) imply that

$$(1 - \varepsilon^{1/4})|\mathrm{buf}(L(S_1))| \geqslant \sum_{i=1}^{s_1} |L(T_i)| \geqslant (1 - 2\varepsilon^{1/4})|\mathrm{buf}(L(S_1))|.$$

Then $T_1, \ldots, T_{s_1}$ are assigned to $S_1$, so the children of $r_1, \ldots, r_{s_1}$ in $T_1, \ldots, T_{s_1}$ (and actually all other non-leaf vertices at this level) are assigned and will be embedded into $M(S_1)$, and the leaves in $T_1, \ldots, T_{s_1}$ are assigned and will be embedded into the leaf clusters of $S_1$. Define $s_2'$ by

$$\sum_{i=s_1+1}^{s_2'} |L(T_i)| \leqslant (1 - \varepsilon^{1/4})|\mathrm{buf}(L(S_2))|, \quad \text{but} \quad \sum_{i=s_1+1}^{s_2'+1} |L(T_i)| > (1 - \varepsilon^{1/4})|\mathrm{buf}(L(S_2))|,$$

and similarly $s_2$ is the largest integer for which $s_2 \leqslant s_2'$, and $r_{s_2}$ is the last root on a level. The trees $T_{s_1+1}, \ldots, T_{s_2}$ are assigned to $S_2$, so the children of $r_{s_1+1}, \ldots, r_{s_2}$ in these trees

https://doi.org/10.1017/S0963548301004849 Published online by Cambridge University Press

are assigned and will be embedded into $M(S_2)$, and the leaves are assigned and will be embedded into the leaf clusters of $S_2$.

We continue in this fashion until we have assigned $T_i$s to every star. $F_1$ denotes the subforest of the $T_i$s that are assigned to stars and $F_2 = F \setminus F_1$. Note that there are still at least $\varepsilon^{1/3}n$ leaves in $F_2$, and these leaves will be used to handle the various exceptional vertices in Step 9.

Before proceeding further, let us point out two important facts here. First, from (4.2), for any cluster $C$,

$$|T \setminus L(T)| \leqslant \sqrt{\alpha}n \ll \varepsilon|C|. \tag{4.9}$$

Second, from (4.2) and (4.3), for any cluster $C$,

$$|V_i| < \gamma n \ll \varepsilon|C| \quad \text{for all } i. \tag{4.10}$$

Thus the total number of non-leaf vertices in $T$ and the size of one level of $T$ are both very small compared to the size of a cluster.

**Step 6:** Assigning most of the non-buffer vertices in $T$ to specific clusters.
In this step, before we start the actual embedding, we assign most of the non-buffer vertices in $T$ to specific clusters. The only non-buffer vertices in $T$ which will not be assigned in this step are the children of the roots in $F_2$. The assignment means that later these vertices will be embedded into the clusters where they were assigned. We emphasize here that, at this point, we only assign these vertices to clusters: they will be actually embedded into these clusters only later. We only assign vertices to the leaf clusters in the stars.

Denote the last region by $R_m$, where the $T_i$s at the bottom of the region are in $F_1$. Write $R'_i = R_i \setminus V_{\frac{i}{d}+j}$; thus we get $R'_i$ by removing the last level in $R_i$. In each region $R'_i, 1 \leqslant i \leqslant m$, we consider the level (denoted by $V(R_i)$) with the smallest number of leaves. Clearly, for the number of leaves in $\cup_{i=1}^m V(R_i)$ we have

$$\sum_{i=1}^{m} |L(V(R_i))| \leqslant dn. \tag{4.11}$$

This level $V(R_i)$ breaks up $R'_i$ into two parts (one possibly empty): a leaf-light part ($LL(R_i)$), which contains fewer leaves, and a leaf-heavy part ($LH(R_i)$), which contains more leaves. Thus, if $V(R_i) = V_t$ for some $\frac{i-1}{d} + j + 1 \leqslant t \leqslant \frac{i}{d} + j - 1$, and the upper half is the lighter part, then

$$\begin{aligned} LL(R_i) &= \cup_{t'}\left\{V_{t'} : \frac{i-1}{d} + j + 1 \leqslant t' \leqslant t - 1\right\}, \\ LH(R_i) &= \cup_{t'}\left\{V_{t'} : t + 1 \leqslant t' \leqslant \frac{i}{d} + j - 1\right\} \quad \text{and} \\ R'_i &= LL(R_i) \cup LH(R_i) \cup V(R_i). \end{aligned}$$

For $R_0$ we just put $LL(R_0) = R_0$, and for $i > m$, $LH(R_i) = R'_i$.

First we assign only the leaves in $\cup_{i=0}^m LL(R_i)$ to clusters. This is done with a simple greedy strategy. As we go top-down in $T$, we assign the leaves at the next level in $\cup_{i=0}^m LL(R_i)$ to the cluster with the smallest number of leaves assigned to it so far. Note

https://doi.org/10.1017/S0963548301004849 Published online by Cambridge University Press

that when we assigned all the leaves in $\cup_{i=0}^{m} LL(R_i)$ with this procedure, the difference between the number of assigned leaves in two arbitrary clusters $C_1, C_2 \in \cup_{i=1}^{s} L(S_i)$ is at most the size of the level $V_i$ in $T$ with the most leaves, which is $\ll \varepsilon|C_1| = \varepsilon|C_2|$ by (4.10). We have to make sure that this assignment is realizable; thus we have to assign the non-leaf vertices in $\cup_{i=0}^{m} LL(R_i)$ carefully. Furthermore, we are also going to assign the leaves on level $V(R_i)$.

Suppose first that $LL(R_i)$ is the upper half of region $R_i, i \geqslant 1$. Assume further that the non-leaf vertices on the last level of $R'_{i-1}$ (*i.e.*, $V_{\frac{i-1}{d}+j-1}$) are assigned to cluster $G$ (if $i \geqslant 2$, and the $T_i$s at the bottom of $R_{i-1}$ are assigned to star $S$, then $G = M(S)$) and that the leaves on the first level of $R_i$ (*i.e.*, $V_{\frac{i-1}{d}+j+1}$) are assigned to cluster $C$. The non-leaf vertices on the last level of $R_{i-1}$ are assigned to a cluster $D$, such that $(G, D), (D, C) \in E(G_r)$ ((4.5) implies that $D$ exists). Similarly, if the non-leaf vertices on a certain level are already assigned to cluster $G$, and the leaves two levels below are assigned to cluster $C$, then the non-leaf vertices on the next level are assigned to cluster $D$, such that $(G, D), (D, C) \in E(G_r)$. For $R_0$, if the leaves on $V_1$ are assigned to cluster $C$, then the root of $T$ is assigned to a cluster $D$ with $(D, C) \in E(G_r)$, and for the remaining non-leaf vertices we follow the same strategy as above for $i \geqslant 1$.

We continue in this fashion for $i \geqslant 1$ until the assignment of the non-leaf vertices two levels before $V(R_i)$, so on the level $V_{t-2}$, if $V(R_i) = V_t$. The assignment of these non-leaf vertices requires some special care. We follow the same procedure as for the non-leaf vertices on the other levels. However, here we do not just take one arbitrary cluster among the possible $\geqslant \delta k$ clusters (guaranteed by (4.5)), but we take the cluster with the fewest leaves from $\cup_{i=1}^{m} V(R_i)$ assigned to it so far. Furthermore, we assign the leaves on level $V(R_i)$ to this cluster. But we do not assign at this point the non-leaf vertices one level before or on $V(R_i)$. These will be used to connect the embedding of $LL(R_i)$ and $LH(R_i)$.

The case when $LL(R_i), i \geqslant 1$ is the lower half of $R_i$ is similar, except that we move level by level upward until $V(R_i)$. In fact, assume that the $T_i$s at the bottom of $R_i$ are assigned to star $S$, and the leaves on the last level of $LL(R_i)$ (*i.e.*, $V_{\frac{i}{d}+j-1}$) are assigned to cluster $C$. The non-leaf vertices on $V_{\frac{i}{d}+j-2}$ (thus including the roots $r_i$) are assigned to a cluster $D$ such that $(M(S), D), (D, C) \in E(G_r)$. We continue to move upward in this fashion until the level after $V(R_i)$, so the level $V_{t+1}$. Say the leaves on this level are assigned to cluster $C$, and the non-leaf vertices on this level are assigned to cluster $G$. Again, for the assignment of the non-leaf vertices on level $V(R_i)$, we take the cluster from the at least $\delta k$ clusters in $N_{G_r}(C) \cap N_{G_r}(G)$ (using (4.5)) with the fewest leaves assigned to it so far from $\cup_{i=1}^{m} V(R_i)$. Furthermore, we assign the leaves on level $V(R_i)$ to this cluster. Thus, in this case we assign all the vertices in $LL(R_i) \cup V(R_i)$.

We follow the same procedure for all the regions $R_i, 1 \leqslant i \leqslant m$. From the above construction, (4.9) and (4.11), it follows that, when we have finished this part of the assignment, the difference between the number of assigned vertices in two arbitrary clusters $C_1, C_2 \in \cup_{i=1}^{s} L(S_i)$ is at most

$$(d/\delta + \varepsilon)|C_1| = (d/\delta + \varepsilon)|C_2|. \tag{4.12}$$

This in turn implies that the number of assigned vertices for every cluster $C$ is at most

https://doi.org/10.1017/S0963548301004849 Published online by Cambridge University Press

$(1/2+2d/\delta)|C \setminus \mathrm{buf}(C)|$, using (4.9), (4.11) and the fact that $LL(R_i)$ contained fewer leaves than $LH(R_i)$.

Next we assign the leaves in $\cup_i LH(R_i)$ with the usual greedy procedure. That is, as we go from top to bottom in $T$, we assign the leaves at the next level in $\cup_i LH(R_i)$ to the cluster with the smallest number of vertices assigned to it so far. Note that with this procedure we eliminate the somewhat larger discrepancy of (4.12), and when we assigned all the leaves in $\cup_i LH(R_i)$, the difference between the number of assigned vertices in two arbitrary clusters $C_1, C_2 \in \cup_{i=1}^{s} L(S_i)$ is $\ll \varepsilon|C_1| = \varepsilon|C_2|$ again.

We finish Step 6 by assigning the remaining non-leaf vertices (except for the children of the roots in $F_2$) to clusters. This is done the same way as in the assignment of the non-leaf vertices in $\cup_{i=1}^{m} LL(R_i)$ above. Furthermore, we assign the non-leaf vertices on the level before $V(R_i)$ in such a way that the embedding of the upper and lower halves of $R_i$ can be connected. More precisely, let us assume that the non-leaf vertices two levels above $V(R_i)$ (so on level $V_{t-2}$, if $V(R_i) = V_t$) are assigned to a cluster $C$, and the non-leaf vertices on $V(R_i)$ are assigned to cluster $G$. Then the non-leaf vertices on the level before $V(R_i)$ (on level $V_{t-1}$) are assigned to a cluster $D$, such that $(C, D), (D, G) \in E(G_r)$ (using (4.5)).

This finishes the assignment procedure: we have assigned every non-buffer vertex of $T$ except for the children of the roots in $F_2$. The difference between the number of assigned vertices in two arbitrary clusters $C_1, C_2 \in \cup_{i=1}^{s} L(S_i)$ is $\ll \varepsilon|C_1| = \varepsilon|C_2|$. This fact implies that, for every cluster $C \in \cup_{i=1}^{s} L(S_i)$, the difference between the number of assigned non-buffer vertices and the available non-buffer zone vertices in $C$ is $\ll \varepsilon|C|$.

Now we are in a position to start the actual embedding process. As we move down in $T$ region by region we execute the following two tasks in a given region.

- We embed the vertices in $T \setminus F$ level by level in the region into non-buffer zone vertices of the cluster where the vertex is assigned (see Step 7).
- If the $T_i$s at the bottom of the region belong to $F_1$, and they are assigned to star $S_i$, then with a special procedure (described in Step 8) we first embed the roots $r_i$ into their assigned cluster, then their children into $M(S_i)$, and finally we assign the leaves in the $T_i$s to $L(S_i)$. These buffer vertices will be embedded only at the end by a König–Hall procedure in Step 11. Otherwise, if the $T_i$s belong to $F_2$, then we embed them by another special procedure (described in Step 9) to handle the various exceptional vertices.

Even though these two tasks are executed together for each region, we separate their discussion for better understanding.

**Step 7:** Embedding $T \setminus F$.

We embed the vertices in $T \setminus F$ level by level from top to bottom with a simple greedy procedure into the non-buffer zone vertices of the clusters where they are assigned. Namely, we always embed a non-leaf vertex of $T$ into a non-buffer zone vertex $v$ of the given cluster, such that $v$ has high degree (at least $(d-\varepsilon)$-portion) to the remaining non-buffer zone vertices of the one or two clusters where the vertices at the next level of $T$ are assigned. $\varepsilon$-regularity makes this possible, and this guarantees that we can always continue the embedding. The roots $r_i$ and their children in the $T_i$s in $F_1$ are embedded with a special strategy, described in the next step, but the buffer vertices in $F_1$ are not embedded

https://doi.org/10.1017/S0963548301004849 Published online by Cambridge University Press

at this point. $F_2$ is embeddded by another special strategy, described in Step 9. When there are only a few non-buffer zone vertices left ($< \sqrt{\varepsilon}|C|$) in a cluster $C \in \cup_{i=1}^{s} L(S_i)$, we unite the buffer zone with the non-buffer zone and continue embedding in this set.

**Step 8:** Embedding the roots and their children in $F_1$.
Consider the first $S_1$ and the trees $T_1, T_2, \dots, T_{s_1}$ assigned to it. Because of the above construction, when we are embedding an $r_j, 1 \leqslant j \leqslant s_1$, it can still be embedded into a large subset ($\geqslant (d-\varepsilon)\sqrt{\varepsilon}$-portion) of the cluster where it was assigned. Denote this cluster by $Cl(r_j)$. We have $(Cl(r_j), M(S_1)) \in E(G_r)$. We are going to partition the children of these roots in $F$ into groups $G_1, G_2, \dots$ depending on the number of their children. We place those children of the roots in $G_i$ when the number of children is in the interval

$$\left(\frac{cn}{\log n}\, 2^{-i}, \frac{cn}{\log n}\, 2^{1-i}\right].$$

The number of groups (denoted by $g$) is at most $\log n$. To each root we assign two kinds of weights corresponding to the groups. For $r_j, w_i(r_j)$ is the number of $G_i$ children of $r_j$, and $lw_i(r_j)$ (leaf weight) is the number of leaves under these $G_i$ children of $r_j$. Similarly, for $v \in G_i$, $lw(v)$ is the number of leaves under $v$, and $lw(G_i) = \sum_{v \in G_i} lw(v)$.

For each $i, 1 \leqslant i \leqslant g$ we partition the roots into two classes: we say that a root $r_j, 1 \leqslant j \leqslant s_1$ is *i-heavy* if

$$w_i(r_j) \geqslant c_2 \log n, \tag{4.13}$$

where $c_2$ is a sufficiently large constant, and we say it is *i-light* if (4.13) does not hold. We denote the set of $i$-heavy roots by $iH$, and similarly $iL$ for the $i$-light roots.

We are going to embed the vertices in $G_i$ in quite large blocks, and the $i$-light roots can cause some problems. However, it is enough to restrict our attention to groups $G_i$ with

$$\sum_{r_j \in iL} w_i(r_j) \geqslant c_3 \log n, \tag{4.14}$$

where $c_3$ is a sufficiently large constant compared to $c_2$. In fact, let us assume that (4.14) does not hold for a group $G_i$. First let us embed arbitrarily the $G_i$ children of the $i$-light roots into unoccupied vertices in $M(S_1)$. Then we embed the leaves under one such $G_i$ child of an $i$-light root into the cluster $C \in N_{G_r}(M(S_1))$ with the smallest number of vertices assigned to it so far. The number of exceptional leaves we have to embed in this way is very small: it is

$$\leqslant \sum_{i=1}^{g} c_3 \log n\, \frac{cn}{2^{i-1} \log n} \ll \varepsilon |C|,$$

for every cluster $C$, if $c$ is sufficiently small. Thus, for simplicity, we assume that (4.14) holds in every group $G_i, 1 \leqslant i \leqslant g$.

For the embedding of the $G_i$ children of the $i$-light roots we will set aside a random buffer zone $\mathrm{buf}(G_i)$ in $M(S_1)$. (Note that this buffer zone has nothing to do with the buffer zones for the leaves in $F$.) We construct this random set in the following way. First we remove a small number of exceptional vertices from $M(S_1)$ denoted by $\mathrm{Exc}(M(S_1))$. These are vertices with the property that they have few ($< (d-\varepsilon)$-portion) neighbours in many

https://doi.org/10.1017/S0963548301004849 Published online by Cambridge University Press

($\geqslant \sqrt{\varepsilon}$-portion) leaf clusters of $S_1$. The number of these vertices is $\leqslant \sqrt{\varepsilon}|M(S_1)|$. In the remaining set every vertex has large degree to most of the leaf clusters. This fact will be important later for the completing König–Hall argument.

We select this set $\mathrm{buf}(G_i)$ randomly (and disjointly for different $1 \leqslant i \leqslant g$) from the remaining non-exceptional vertices with size

$$(1+\sqrt{\varepsilon}) \sum_{r_j \in iL} w_i(r_j) \ \ (\geqslant c_3 \log n). \tag{4.15}$$

Note that Lemma 2.5 and (4.2) imply that with high probability $(\mathrm{buf}(G_i), C)$ is a $\sqrt{\varepsilon}$-regular pair for every $1 \leqslant i \leqslant g$ and cluster $C \in N_{G_r}(M(S_1))$, and furthermore, if for a $v \in C$ we have $\deg(v, M(S_1)) \geqslant (d-\varepsilon)|M(S_1)|$ (most vertices are such in $C$), then $\deg(v, \mathrm{buf}(G_i)) \geqslant (d-2\varepsilon)|\mathrm{buf}(G_i)|$.

We are going to embed the $G_i$ children of the $i$-light roots into $\mathrm{buf}(G_i)$. These $G_i$ vertices embedded into $\mathrm{buf}(G_i)$ will form the first (and the only light) block $B_{i,1}$ for $G_i$. The remaining blocks $B_{i,2}, B_{i,3}, \ldots$ will be heavy blocks defined below.

We start the embedding with $r_1$. Consider those $i$s for which $r_1$ is $i$-light, denoted by $\mathrm{Light}(r_1)$. Similarly $\mathrm{Heavy}(r_1)$ can be defined. $r_1$ can still be embedded into a large subset of $Cl(r_1)$. From this subset we remove the few exceptional vertices $v$ for which $\deg(v, M(S_1)) < (d-\varepsilon)|M(S_1)|$. We embed $r_1$ into a random vertex of the remaining set. For an $i \in \mathrm{Heavy}(r_1)$ we embed the $G_i$ children of $r_1$ into a random subset (with the given size $w_i(r_1)$) of the remaining unoccupied vertices in $N_{M(S_1)}(r_1) \setminus \mathrm{Exc}(M(S_1))$. At the same time we assign the leaf children of these vertices to $L(S_1)$. For these $i$s this is a heavy block, so it is $B_{i,j}$ for some $j \geqslant 2$. The heavy blocks behave the same way as the light blocks, so, for every block $B_{i,j}$, the pair $(B_{i,j}, C)$ is a $\sqrt{\varepsilon}$-regular pair for every cluster $C \in N_{G_r}(M(S_1))$.

For $i \in \mathrm{Light}(r_1)$, by the above remark with high probability the (embedded) $r_1$ has many neighbours in $\mathrm{buf}(G_i)$ for every $1 \leqslant i \leqslant g$. The $G_i$ children of $r_1$ are embedded into these neighbours and we assign the leaf children of these vertices to $L(S_1)$.

Next we move to $r_2$ and we follow the same procedure. We update $\mathrm{buf}(G_i)$ by removing the vertices occupied in the previous step. We embed $r_2$ into a random vertex in $Cl(r_2)$ that has many neighbours in $M(S_1)$. For $i \in \mathrm{Heavy}(r_2)$ we embed the $G_i$ children of $r_2$ into a random subset of the remaining unoccupied vertices in $N_{M(S_1)}(r_2) \setminus \mathrm{Exc}(M(S_1))$ of the appropriate size and we assign the leaf children to $L(S_1)$. For $i \in \mathrm{Light}(r_2)$ we embed the $G_i$ children of $r_2$ into $\mathrm{buf}(G_i)$.

We continue with this procedure for all the roots assigned to $S_1$. Lemma 2.5 and (4.15) imply that we never get stuck, we are able to embed all the $G_i$ children of $i$-light roots into $\mathrm{buf}(G_i)$. Lemma 2.5 also implies that with high probability we have the following: for every $B_{i,j}$, $C \in L(S_1)$ and $v \in C$, if

$$\deg(v, M(S_1)) \geqslant (d-\varepsilon)|M(S_1)|, \tag{4.16}$$

then we have

$$\deg(v, B_{i,j}) \geqslant (d-2\varepsilon)|B_{i,j}|. \tag{4.17}$$

Note that, apart from $\leqslant \varepsilon|C|$ exceptional vertices, (4.16) holds for every vertex $v \in C$.

We follow the same procedure for all the stars, and the $T_i$s assigned to them. At this

https://doi.org/10.1017/S0963548301004849 Published online by Cambridge University Press

point the only $T_i$s remaining are in $F_2$. They will be used to handle the various exceptional vertices in the next step.

**Step 9:** Embedding $F_2$ and handling the various exceptional vertices.
It is time to take care of the various exceptional vertices, a common task in applications of the Regularity Lemma. First of all we have the remaining unoccupied vertices in $C_0$ and in the exceptional clusters not covered by the leaves of the stars. Denote their set by $E$. We add some more vertices to $E$ from each star $S$. These are unoccupied vertices $v$ in the leaf clusters of $S$ such that, for the exceptional blocks $B_{i,j}$ for which (4.17) does not hold for $v$, we have

$$\sum lw(B_{i,j}) \geqslant \sqrt{\varepsilon}|L_G(S)|. \tag{4.18}$$

The number of these exceptional vertices is at most $\sqrt{\varepsilon}|C|$ in every $C \in L(S)$. We add these vertices to $E$ as well. The size of $E$ is still at most $2\sqrt{\varepsilon}n \ll \varepsilon^{1/3}n$.

Next we take care of the vertices in $E$: we embed leaves from the remaining $T_i$s ($F_2$) into them by a simple greedy strategy. From the construction we know that there are still at least $\varepsilon^{1/3}n$ (much more than $|E|$) unassigned leaves in $F_2$. Denote the roots in $F_2$ by $r'_1, r'_2, \ldots$. These roots are embedded into $G$ by the greedy strategy described in Step 7. Denote the embedded image vertex in $G$ by $G(r'_i)$. From $\delta(G) \geqslant \left(1/2 + \delta\right) n$ and the random choice of the buffer zones, for every $e \in E$, the two vertices $G(r'_1)$ and $e$ have at least $\delta n$ common unoccupied neighbours in $\mathrm{buf}(G)$. Thus there is an unoccupied neighbour $u_1$ of $G(r'_1)$ which has at least $\delta|E|$ neighbours in $E$. Embed one of the children of $r'_1$ in $F_2$ into $u_1$ and embed the leaves under this child of $r'_1$ in $F_2$ into neighbours of $u_1$ in $E$. We repeat this procedure for the remaining vertices in $E$. When $|E|$ becomes small ($\leqslant \frac{cn}{\delta \log n}$), it might not be possible to embed all the leaves under a child of a root to these neighbours in $E$. In this case, let us assume that this child of a root in $F_2$ is embedded into vertex $u_i$ in $G$. We embed the remaining leaves that cannot be embedded in $E$ into the remaining buffer vertices of the cluster with the smallest number of embedded vertices, chosen from those clusters where $u_i$ has many neighbours ($\geqslant d$-portion in at least half the clusters). We handle similarly the remaining leaves in $F_2$, when there are no more unoccupied vertices left in $E$.

When we are finished with the procedure, and all the vertices are embedded except for the $F_1$-buffer leaves, for every cluster $C \in \cup_{i=1}^{s} L(S_i)$ the difference between the number of vertices that are embedded into $C$ and the non-buffer zone vertices in $C$ is $\ll \varepsilon^{\frac{1}{4}}|C|$.

**Step 10:** Adjusting the assignment.
Thus at this point there can be a small difference ($\ll \varepsilon^{1/4}|S_i|$) in a star $S_i$ between the number of remaining unoccupied vertices in the leaf clusters of the star and the number of $F_1$ leaves assigned to it. Therefore we need some adjusting. Consider a star $S'$ where we assigned more $F_1$ leaves than vertices remaining, and a star $S''$ where we assigned fewer $F_1$ leaves than vertices remaining (there has to be one such star since we are looking for a spanning subgraph). By property (2) of the star covering, in $S''$ among the leaf clusters we have many ($\geqslant d^2|L(S'')|$) clusters which are bridge clusters from $S''$ to $S'$, that is, they are neighbours of $M(S')$ in $G_r$ as well. We add one typical vertex from these common clusters from $S''$ to $S'$. Now we are one step closer to the perfect assignment, and by iterating

https://doi.org/10.1017/S0963548301004849 Published online by Cambridge University Press

this process we can achieve it. Furthermore, using $\varepsilon \ll d$ we can guarantee that, during the whole adjustment procedure, we have removed or added at most $\varepsilon^{\frac{1}{3}}|C|$ vertices from every cluster $C$.

**Step 11:** Finishing the embedding of the $F_1$-buffer leaves by a König–Hall argument.
We can treat the stars separately. Consider the bipartite graph $G(A, B)$, where $A$ is the set of the children of the roots in $F_1$ embedded into the middle cluster of the given star (say $S$) and $B$ is the set of the remaining unoccupied vertices in the leaf clusters. Let $\underline{d}$ denote $(d_a : a \in A)$, where $d_a$ is the number of leaf children of $a$ (so $d_a = lw(a)$). We are looking for a $\underline{d}$-matching from $A$ to $B$. We need to check the König–Hall criterion:

$$|N(X)| \geqslant d_X \quad \text{for all } X \subset A$$

holds with probability close to 1 (the choice of $A$ was random). From the construction

$$A = \cup_{i=1}^{g} G_i, \ G_i = \cup_{j=1}^{n_i} B_{i,j},$$

where $n_i$ is the number of blocks in $G_i$, $v \in G_i$ implies

$$\frac{cn}{2^i \log n} < d_v \leqslant \frac{cn}{2^{i-1} \log n}, \tag{4.19}$$

and

$$\sum_{i=1}^{g} d_{G_i} = \sum_{i=1}^{g} \sum_{j=1}^{n_i} d_{B_{i,j}} = |B|. \tag{4.20}$$

Furthermore, from the construction we get that, with high probability, for every $B_{i,j}$, the pair $(B_{i,j}, B)$ is $\varepsilon^{\frac{1}{6}}$-regular.

We distinguish three cases (the proof is similar to the König–Hall argument in [6]).

**Case 1:** $X_{i,j} = X \cap B_{i,j} < \varepsilon^{\frac{1}{6}}|B_{i,j}|$ for all blocks $B_{i,j}$.
Then, if $v \in X$ is arbitrary we get

$$|N(X)| \geqslant \deg(v) \geqslant \frac{d}{2}|B| = \sum_{i=1}^{g} \sum_{j=1}^{n_i} \frac{d}{2} d_{B_{i,j}} > \sum_{i=1}^{g} \sum_{j=1}^{n_i} d_{X_{i,j}} = d_X.$$

Here we used the fact that the minimum degree in $G(A, B)$ from $A$ to $B$ is at least $\frac{d}{2}|B|$, (4.19), (4.20), and that $\varepsilon$ is much smaller than $d$.

**Case 2:** There exists a block $B_{i,j}$ with $X_{i,j} \geqslant \varepsilon^{\frac{1}{6}}|B_{i,j}|$ but $d_X \leqslant (1 - \varepsilon^{\frac{1}{6}})|B|$.
In this case we have

$$|N(X)| \geqslant (1 - \varepsilon^{\frac{1}{6}})|B| \geqslant d_X. \tag{4.21}$$

Indeed, this follows from the fact that $(B_{i,j}, B)$ is a $\varepsilon^{\frac{1}{6}}$-regular pair. Thus, if (4.21) were not true, then with $Y = B \setminus N(X)$ we would have $|X| \geqslant \varepsilon^{\frac{1}{6}}|B_{i,j}|$, $|Y| \geqslant \varepsilon^{\frac{1}{6}}|B|$, but $d(X, Y) = 0$, contradicting $\varepsilon^{\frac{1}{6}}$-regularity.

**Case 3:** $d_X > (1 - \varepsilon^{\frac{1}{6}})|B|$.
In this case, with high probability,

$$|N(X)| = |B| \geqslant d_X.$$

In fact, for an arbitrary $v \in B$, with high probability, (4.17) holds for almost all the blocks

https://doi.org/10.1017/S0963548301004849 Published online by Cambridge University Press

$B_{i,j}$; the exceptional blocks (the blocks not satisfying (4.17)) have total $d$-weight $\leqslant 2\sqrt{\varepsilon}|B|$ (the exceptional vertices were removed from $B$). Therefore, in this case, there must be a non-exceptional block $B_{i,j}$ with $|X_{i,j}| \geqslant (1-\varepsilon^{\frac{1}{5}})|B_{i,j}|$, implying $v \in N(X_{i,j}) \subset N(X)$. Since $v$ was arbitrary,

$$N(X) = B,$$

thus finishing Case 3 and the proof of the well-distributed case.

### 4.3. The concentrated case

In this case we can assume that there is a level $V_r$ with

$$|V_r| \geqslant \gamma n.$$

We are going to use ideas similar to the well-distributed case; therefore we will sometimes omit the details.

We are going to divide the proof into three (perhaps overlapping) special cases.

**Case 1:** There is an $r$ with

$$|V_r| \geqslant \gamma n \quad \text{but} \quad |V_{r-1}| < \gamma |V_r|. \tag{4.22}$$

The proof of this case is similar to (but simpler than) the well-distributed case. Consider the leaves in $V_r$, their parents and grandparents. In this case these are the $T_i$s, with roots $r_i$ on level $V_{r-2}$. The choice of buffer zones is the same. We start the embedding into non-buffer zone vertices from the top by arbitrarily embedding the root of $T$. Suppose that a non-leaf vertex $v$ on a certain level is already embedded. Then, we embed its non-leaf children in $T$ one by one into vertices with large degree to the cluster with the fewest vertices embedded into it so far, and then we embed the leaf grandchildren of $v$ into these neighbours. We move down $T$ in a top-down breadth-first manner until $V_{r-3}$, that is, one level before the roots of the $T_i$s. The handling of the $T_i$s and the construction of the stars is very similar to the well-distributed case. The only difference is that, since now we are not in the well-distributed case, a $T_i$ can be large (for example there could be only one $T_1$). We divide $T_i$ into smaller pieces $T_{i,j}$ which are disjoint apart from the common root $r_i$. The only problem could be that a $T_i$ could be assigned to several stars, and then $r_i$ would have to be connected to several stars. Thus we do the following. Take the first $T_1$ from the left. Assume that the father of $r_1$ on level $V_{r-3}$ is embedded into cluster $C$. Take an arbitrary cluster $G \in N_{G_r}(C)$ and $r_1$ will be embedded into cluster $G$. But first we start the construction of the star covering. It is similar to the well-distributed case, but for $S_1$ we take $M(S_1) \in N_{G_r}(G)$. We assign pieces $T_{1,j}$ from $T_1$ to $S_1$ as in the well-distributed case. If we have leftover pieces in $T_1$, then we move on to constructing $S_2$ with $M(S_2) \in N_{G_r}(G)$. For this purpose we pick $M(S_2)$ as the cluster in $N_{G_r}(G)$ which covers the most clusters ($\geqslant \delta$-portion) in $G_r \setminus L(S_1)$. Otherwise, if we assigned all the pieces of $T_1$ to $S_1$, then we move to $T_2$. If the father of $r_2$ is embedded into a cluster $C'$, then $r_2$ will be embedded into a cluster $G' \in N_{G_r}(C') \cap N_{G_r}(M(S_1))$. Then we start assigning pieces from $T_2$ to $S_1$. If we have leftover pieces in $T_2$, then we construct $S_2$ with $M(S_2) \in N_{G_r}(G')$. We continue in this fashion. This procedure guarantees that, if $r_i$ is embedded into a cluster $G''$ and a piece of $T_i$ is assigned to star $S_j$, then we have $(G'', M(S_j)) \in E(G_r)$. The leaf children of a

https://doi.org/10.1017/S0963548301004849 Published online by Cambridge University Press

root $r_i$ on level $V_{r-1}$ are embedded into the neighbour cluster of $G''$ in $G_r$ with the fewest vertices embedded into it so far. By (4.22) the total number of these leaves is very small ($\leqslant \gamma$-portion) compared to the number of buffer vertices; therefore they are not causing any significant discrepancies among the number of embedded vertices in the clusters. For the embedding of the part of $T$ below $V_r$, we follow the same simple greedy strategy as in the upper half. There are no further difficulties; we can follow the same steps as in the well-distributed case.

**Case 2:** $r$ is small, $r \leqslant \frac{1}{\gamma^2}$ (for simplicity we assume that $\frac{1}{\gamma^2}$ is an integer).
We may assume that $|V_{r-1}| \geqslant \gamma |V_r|$; otherwise Case 1 holds. Similarly we may assume that $|V_{r-2}| \geqslant \gamma |V_{r-1}|$, since otherwise we have

$$|V_{r-1}| \geqslant \gamma^2 n \quad \text{but} \quad |V_{r-2}| < \gamma |V_{r-1}|;$$

the situation is similar to Case 1. Iterating this we get

$$c \frac{n}{\log n} \geqslant |V_1| \geqslant \gamma^r n \geqslant \gamma^{\frac{1}{\gamma^2}} n,$$

a contradiction if $n$ is sufficiently large.

**Case 3:** $r > \frac{1}{\gamma^2}$.
Consider the levels between $V_r$ and $V_{r-\frac{1}{\gamma^2}}$, and let $V_s$ denote the level with the smallest number of leaves ($\leqslant \gamma^2 n$) in it. Take the non-leaf vertices on $V_{s-1}$, and the subtrees of $T$ below them between levels $V_{s-1}$ and $V_r$ (including $V_r$). These subtrees (denoted by $T_1', T_2', \ldots$) provide a partition of the region between $V_{s-1}$ and $V_r$. If for one of these subtrees $T_i'$ we have

$$|T_i'| \geqslant \gamma^2 n,$$

then the proof is similar to Case 2. Therefore we may assume that

$$|T_i'| < \gamma^2 n \tag{4.23}$$

for every subtree.

For the embedding of the part of $T$ above $V_{s-1}$, we follow the same greedy strategy as in Case 1. For the region between $V_{s-1}$ and $V_r$ we do the following. First of all, the $T_i$s again consist of the leaves on level $V_r$, their parents and their grandparents. We take the first subtree $T_1'$ in this region. We embed the leaves in $T_i'$ at one level (except for the leaves on level $V_s$) into the cluster with the smallest number of vertices embedded into it so far. This is done by going upward (with the assignment procedure, and then downward with the actual embedding) in $T_1'$ starting with the leaves on level $V_{r-1}$ and following the same procedure as in the well-distributed case. We keep going until $V_s$; we embed the root of $T_1'$ so we can connect the embedding of $T_i'$ and the part of $T$ above $V_s$. Then the leaves on level $V_s$ are evenly distributed among the clusters where the embedded root has many neighbours.

We continue the procedure for the other $T_i'$s in this fashion. Relation (4.23), and the fact that on $V_s$ there are $\leqslant \gamma^2 n$ leaves, implies that we again fill up the non-buffer zones in a balanced way, and there are no further complications in this case, thus completing the proof of Theorem 1.2.

https://doi.org/10.1017/S0963548301004849 Published online by Cambridge University Press

**Acknowledgement**

We are grateful to one of the referees for many valuable comments on an earlier version of this paper.

## References

[1] Alon, N., Duke, R., Lefmann, H., Rödl, V. and Yuster, R. (1994) The algorithmic aspects of the Regularity Lemma. *J. Algorithms* **16** 80–109.
[2] Bollobás, B. (1978) *Extremal Graph Theory*, Academic Press, London.
[3] Bollobás, B. and Eldridge, S. E. (1978) Packings of graphs and applications to computational complexity. *J. Combin. Theory Ser. B* **25** 105–124.
[4] Hajnal, A. and Szemerédi, E. (1970) Proof of a conjecture of Erdős. In *Combinatorial Theory and its Applications*, Vol. II (P. Erdős, A. Rényi and V. T. Sós, eds), *Colloq. Math. Soc. J. Bolyai* **4**, North-Holland, Amsterdam, pp. 601–623.
[5] Kohayakawa, Y. (1997) Szemerédi's Regularity Lemma for sparse graphs. In *Foundations of Computational Mathematics* (F. Cucker and M. Schub, eds), Springer.
[6] Komlós, J., Sárközy, G. N. and Szemerédi, E. (1995) Proof of a packing conjecture of Bollobás. *Combin. Probab. Comput.* **4** 241–255.
[7] Komlós, J., Sárközy, G. N. and Szemerédi, E. (1996) On the square of a Hamiltonian cycle in dense graphs. *Random Struct. Alg.* **9** 193–211.
[8] Komlós, J., Sárközy, G. N. and Szemerédi, E. (1997) Blow-up Lemma. *Combinatorica* **17** 109–123.
[9] Komlós, J., Sárközy, G. N. and Szemerédi, E. (1998) An algorithmic version of the Blow-up Lemma. *Random Struct. Alg.* **12** 297–312.
[10] Komlós, J., Sárközy, G. N. and Szemerédi, E. (1998) Proof of the Seymour conjecture for large graphs. *Ann. Combin.* **2** 43–60.
[11] Komlós, J., Sárközy, G. N. and Szemerédi, E. Proof of the Alon–Yuster conjecture. To appear.
[12] Lovász, L. (1979) *Combinatorial Problems and Exercises*, Akadémiai Kiadó, Budapest.
[13] Sauer, N. and Spencer, J. (1978) Edge disjoint placement of graphs. *J. Combin. Theory Ser. B* **25** 295–302.
[14] Simonovits, M. and Sós, V. T. (1991) Szemerédi's partition and quasirandomness, *Random Struct. Alg.* **2** 1–10.
[15] Szemerédi, E. (1976) Regular partitions of graphs. *Colloques Internationaux C.N.R.S.* Nº 260: *Problèmes Combinatoires et Théorie des Graphes*, Orsay, pp. 399–401.
[16] Szemerédi, E. (1975) On a set containing no $k$ elements in arithmetic progression. *Acta Arithmetica* **XXVII** 199–245.

https://doi.org/10.1017/S0963548301004849 Published online by Cambridge University Press