AN ALGORITHMIC VERSION OF THE BLOW-UP LEMMA

JÁNOS KOMLÓS, GABOR N. SARKOZY, AND ENDRE SZEMERÉDI

Abstract. Recently we have developed a new method in graph theory based on the Regularity Lemma. The method is applied to find certain spanning subgraphs in dense graphs. The other main general tool of the method, beside the Regularity Lemma, is the so-called Blow-up Lemma ([24]). This lemma helps to find bounded degree spanning subgraphs in ε-regular graphs. Our original proof of the lemma is not algorithmic, it applies probabilistic methods. In this paper we provide an algorithmic version of the Blow-up Lemma. The desired subgraph, for an n-vertex graph, can be found in time $O(nM(n))$, where $M(n) = O(n^{2.376})$ is the time needed to multiply two n by n matrices with 0,1 entries over the integers. We show that the algorithm can be parallelized and implemented in NC^5.

1. Introduction

1.1. **Notations and definitions.** All graphs are simple, that is, they have no loops or multiple edges. $v(G)$ is the number of vertices in G (order), $e(G)$ is the number of edges in G (size). $deg(v)$ (or $deg_G(v)$) is the degree of vertex v (within the graph G), and $deg(v, Y)$ (or $deg_G(v, Y)$) is the number of neighbours of v in Y. $\delta(G)$ and $\Delta(G)$ are the minimum degree and the maximum degree of G. $N(x)$ (or $N_G(x)$) is the set of neighbours of the vertex x, and $e(X, Y)$ is the number of edges between X and Y. A bipartite graph G with color-classes A and B and edges E will sometimes be written as $G = (A, B, E)$. For disjoint X, Y, we define the **density**

$$d(X, Y) = \frac{e(X, Y)}{|X| \cdot |Y|}.$$

The density of a bipartite graph $G = (A, B, E)$ is the number

$$d(G) = d(A, B) = \frac{|E|}{|A| \cdot |B|}.$$

For two disjoint subsets A, B of $V(G)$, the bipartite graph with vertex set $A \cup B$ which has all the edges of G with one endpoint in A and the other in B is called the pair (A, B).

A pair (A, B) is ε-**regular** if for every $X \subset A$ and $Y \subset B$ satisfying

$$|X| > \varepsilon |A| \quad \text{and} \quad |Y| > \varepsilon |B|$$

Part of this paper was written while Sarkozy was visiting MSRI Berkeley, as part of the Combinatorics Program. Research at MSRI is supported in part by NSF grant DMS-9022140.

we have

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

A pair (A, B) is (ε, δ)**-super-regular** if it is ε-regular and furthermore,

$$deg(a) \geq \delta|B| \quad \text{for all} \quad a \in A,$$

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

H is **embeddable** into G if G has a subgraph isomorphic to H, that is, if there is a one-to-one map (injection) $\varphi : V(H) \to V(G)$ such that $\{x, y\} \in E(H)$ implies $\{\varphi(x), \varphi(y)\} \in E(G)$.

As the model of computation we choose the weakest possible version of a *PRAM*, in which concurrent reads or writes of the same location are not allowed (*EREW*, see [21] for a discussion of the various *PRAM* models.) When researchers investigate the parallel complexity of a problem, the main question is whether a polylogarithmic running time is achievable on a *PRAM* containing a polynomial number of processors. If the answer is positive then the problem and the corresponding algorithm are said to belong to class NC introduced in [32] (see also [10],[38]). When the running time is $O((\log n)^i)$, the algorithm is in NC^i.

1.2. **An Algorithmic Version of the Blow-up Lemma.** In recent years the interaction between combinatorics and the theory of algorithms is getting stronger and stronger. It is therefore not surprising that there has been a significant interest in converting existence proofs into efficient algorithms. Many examples of this type can be found in [1, 31]. Some of these are general methods, so algorithmic versions of these methods immediately imply efficient algorithms for several problems. One example is the Lovász Local Lemma whose algorithmic aspects have been studied recently in [2, 7]. Another example is the Regularity Lemma [36]. The basic content of this lemma could be described by saying that every graph can, in some sense, be well approximated by random graphs. Since random graphs of a given edge density are much easier to treat than all graphs of the same edge-density, the Regularity Lemma helps us to carry over results that are trivial for random graphs to the class of all graphs with a given number of edges. The lemma has numerous applications in various areas including combinatorial number theory [13, 37], computational complexity [19] and extremal graph theory [5, 8, 9, 11, 12, 14, 15, 16, 33, 35]. Recently an NC^1-algorithmic version was given in [3].

During the past couple of years we have developed a new method in graph theory based on the Regularity Lemma. We usually apply this method to find certain spanning subgraphs in dense graphs. Typical examples are spanning trees (Bollobás-conjecture, see [23]), Hamiltonian cycles or powers of Hamiltonian cycles (Pósa-Seymour conjecture, see [25, 26]) or H-factors for a fixed graph H (Alon-Yuster conjecture, see [27]). The other main general tool in the method, beside the Regularity Lemma, is the so-called Blow-up Lemma ([24]). This lemma helps to find bounded degree spanning subgraphs in ε-regular graphs. The rough idea of the original proof of this lemma was the following: we use a randomized greedy algorithm to embed most of the vertices of the bounded degree graph, and then finish the embedding by a König-Hall argument. Given the recent algorithmic version of the Regularity Lemma, the obvious question arises whether there is an algorithmic version also for the Blow-up Lemma. In this paper we give an affirmative answer to this question.

Theorem 1. (An Algorithmic Version of the Blow-up Lemma) *Given a graph R of order r and positive parameters δ, Δ, there exists an $\varepsilon > 0$ such that the following holds. Let N be an arbitrary positive integer, and let us replace the vertices of R with pairwise disjoint N-sets $V_1, V_2, \dots, V_r$ (blowing up). We construct two graphs on the same vertex-set $V = \cup V_i$. The graph $R(N)$ is obtained by replacing all edges of R with copies of the complete bipartite graph $K_{N,N}$, and a sparser graph G is constructed by replacing the edges of R with some (ε, δ)-super-regular pairs. If a graph H with $\Delta(H) \leq \Delta$ is embeddable into $R(N)$ then it is already embeddable into G. We can construct a copy of H in G in $O(nM(n))$ sequential time, where $M(n) = O(n^{2.376})$ is the time needed to multiply two n by n matrices with 0,1 entries over the integers. Furthermore, the algorithm can be parallelized and implemented in NC^5.*

Remark. For some very special cases of this theorem (e.g. a Hamiltonian path in a super-regular pair) NC algorithms can be found in [34].

When using the Blow-up Lemma, we typically also need the following strengthened version: Given $c > 0$, there are positive functions $\varepsilon = \varepsilon(\delta, \Delta, r, c)$ and $\alpha = \alpha(\delta, \Delta, r, c)$ such that the Blow-up Lemma remains true if for every i there are certain vertices x to be embedded into V_i whose images are *a priori* restricted to certain sets $C_x \subset V_i$ provided that

(i) each C_x within a V_i is of size at least $c|V_i|$,

(ii) the number of such restrictions within a V_i is not more than $\alpha|V_i|$.

Our proof is going to be similar to our original probabilistic proof, but we have to replace the probabilistic arguments with deterministic ones. In Section 2 we give a deterministic sequential algorithm for the embedding problem without considering implementation and time complexity issues. In Section 3 we show that the algorithm is correct. Implementation is discussed in Section 4. Finally, Section 5 contains various algorithmic applications.

2. The algorithm

The main idea of the algorithm is the following. We embed the vertices of H one-by-one by following a greedy algorithm, which works smoothly until there is only a small proportion of H left, and then it may get stuck hopelessly. To avoid that, we will set aside a positive proportion of the vertices of H as buffer vertices. Most of these buffer vertices will be embedded only at the very end by using a König-Hall argument.

2.1. **Preprocessing.** We will assume that $|V(H)| = |V(G)| = |\cup_i V_i| = n = rN$. We can also assume, without restricting generality, that $N \geq N_0(\delta, \Delta, r)$, for Theorem 1 is trivial for small N since ε-regularity with a small enough ε implies $G = R(N)$. Finally, we will assume for simplicity, that the density of every super-regular pair in G is exactly δ. This is not a significant restriction, otherwise we just have to put everywhere the actual density instead of δ.

We will use the following parameters:

$$\varepsilon \ll \varepsilon' \ll \varepsilon'' \ll \delta''' \ll \delta'' \ll \delta' \ll \delta,$$

where $a \ll b$ means that a is small enough compared to b.

For easier reading, we will mostly use the letter x for vertices of H, and the letter v for vertices of the host graph G.

Given an embedding of H into $R(N)$, it defines an *assignment*

$$\psi : V(H) \to \{V_1, V_2, \dots, V_r\},$$

and we want to find an *embedding*

$$\varphi : V(H) \to V(G), \quad \varphi \text{ is one-to-one}$$

such that $\varphi(x) \in \psi(x)$ for all $x \in V(H)$. We will write $X_i = \psi^{-1}(V_i)$ for $i = 1, 2, \dots, r$.

Before we start the algorithm, we order the vertices of H into a sequence $S = (x_1, x_2, \dots, x_n)$ which is more or less, but not exactly, the order in which the vertices will be embedded. Let $m = r\lceil \delta' N \rceil$. For each i, choose a set B_i of $\lceil \delta' N \rceil$ vertices in X_i such that any two of these vertices are at a distance at least four in H. (This is possible, for H is a bounded degree graph.) These vertices $b_1, \dots, b_m$ will be called the **buffer vertices** and they will be the last vertices in S.

The order S starts with the neighbourhoods $N_H(b_1), N_H(b_2), \dots, N_H(b_m)$. The length of this initial segment of S will be denoted by T_0. Thus $T_0 = \sum_{i=1}^{m} |N_H(b_i)| \leq \Delta m$.

The rest of S is an arbitrary ordering of the leftover vertices of H.

(When certain images are *a priori* restricted, as in the remark after the theorem, we also list the restricted vertices at the beginning of S right after the neighbours of the buffer vertices.)

2.2. **Sketch of the algorithm.** In *Phase 1* of the algorithm we will embed the vertices in S one-by-one into G until all non-buffer vertices are embedded. For each x_j not embedded yet (including the buffer vertices) we keep track of an ever shrinking host set H_{t,x_j} that x_j is confined to at time t, and we only make a final choice for the location of x_j from H_{t,x_j} at time j. At time 0, H_{0,x_j} is the cluster that x_j is assigned to. For technical reasons we will also maintain another similar set, C_{t,x_j}, where we will ignore the possibility that some vertices are occupied already.

In *Phase 2*, we embed the leftover vertices by using a König-Hall type argument.

2.3. **Embedding Algorithm.** At time 0, set $C_{0,x} = H_{0,x} = \psi(x)$ for all $x \in V(H)$. Put $T_1 = \lfloor \delta'' n \rfloor$.

Phase 1

For $t \geq 1$, repeat the following steps.

Step 1 - Extending the embedding. We embed x_t. Consider the vertices in H_{t-1,x_t}. We will pick one of these vertices as the image $\varphi(x_t)$ by using the Selection Algorithm (described below in Section 2.4).

Step 2 - Updating. For each unembedded vertex y (i.e. the set of vertices $x_j, t < j \leq n$), set

$$C_{t,y} = \begin{cases} C_{t-1,y} \cap N_G(\varphi(x_t)) & \text{if } \{x_t, y\} \in E(H) \\ C_{t-1,y} & \text{otherwise,} \end{cases}$$

and

$$H_{t,y} = \begin{cases} H_{t-1,y} \cap N_G(\varphi(x_t)) & \text{if } \{x_t, y\} \in E(H) \\ H_{t-1,y} \setminus \{\varphi(x)\} & \text{otherwise.} \end{cases}$$

Step 3 - Exceptional vertices in H.

1. If T_1 does not divide t, then go to Step 4.

2. If T_1 divides t, then we do the following. We find all exceptional unembedded vertices $y \in H$ such that $|H_{t,y}| \leq (\delta')^2 n$. At this point we slightly change the order of the remaining unembedded vertices in S. We bring these exceptional vertices forward (even if they are buffer vertices), followed by the non-exceptional vertices in the same relative order as before. For simplicity we still use the notation $(x_1, x_2, \ldots, x_n)$ for the new order.

Step 4 - Exceptional vertices in G.

1. If $t \neq T_0$, then go to Step 5.

2. If $t = T_0$, then we do the following. We find the set (denoted by E_i) of those exceptional vertices $v \in V_i$, $1 \leq i \leq r$ for which v is not covered yet in the embedding and

$$|\{b : b \in B_i, v \in C_{t,b}\}| < \delta'' |B_i|.$$

Once again we are going to change slightly the order of the remaining unembedded vertices in S. We choose a set E of vertices $x \in H$ of size $\sum_{i=1}^{r} |E_i|$ (more precisely $|E_i|$ vertices from X_i for all $1 \leq i \leq r$) with

$$H_{t,x} = H_{0,x} \setminus \{\varphi(x_j) : j \leq t\} = \psi(x) \setminus \{\varphi(x_j) : j \leq t\}.$$

Thus in particular, if $x \in X_i$, then $E_i \subset H_{t,x}$. For example, we may choose the vertices in E as vertices in H that are at a distance at least four from each other and any of the vertices embedded so far. We are going to show later in the proof of correctness that this is possible since H is a bounded degree graph and $\sum_{i=1}^{r} |E_i|$ is very small. We bring the vertices in E forward, followed by the remaining unembedded vertices in the same relative order as before. Again, for simplicity we keep the notation $(x_1, x_2, \ldots, x_n)$ for the resulting order. After this we set $t \leftarrow t + 1$ and go back to

Step 1. However, when we perform Step 1 for a vertex $x \in E$ we pick $\varphi(x)$ from the appropriate E_i.

Step 5 - If there are no more unembedded non-buffer vertices left, then set $T = t$ and go to Phase 2, otherwise set $t \leftarrow t+1$ and go back to Step 1.

Phase 2

Find a system of distinct representatives of the sets $H_{T,y}$ for all unembedded y (i.e. the set of vertices x_j, $T < j \leq n$).

2.4. **Selection Algorithm.** We choose a vertex $v \in H_{t-1,x_t}$ as the image $\varphi(x_t)$ for which the following hold for all unembedded y with $\{x_t, y\} \in E(H)$

$$(\delta-\varepsilon)|H_{t-1,y}| \leq deg_G(v, H_{t-1,y}) \leq (\delta+\varepsilon)|H_{t-1,y}|,$$

$$(\delta-\varepsilon)|C_{t-1,y}| \leq deg_G(v, C_{t-1,y}) \leq (\delta+\varepsilon)|C_{t-1,y}|$$

and

$$(\delta-\varepsilon)|C_{t-1,y} \cap C_{t-1,y'}| \leq deg_G(v, C_{t-1,y} \cap C_{t-1,y'}) \leq (\delta+\varepsilon)|C_{t-1,y} \cap C_{t-1,y'}|,$$

for at least $(1-\varepsilon')$ proportion of the unembedded vertices y' with $\psi(y') = \psi(y)$.

3. Proof of correctness

The following claims state that our algorithm finds a good embedding of P into G.

Claim 1. *Phase 1 always succeeds.*

Claim 2. *Phase 2 always succeeds.*

If at time t, S is a set of unembedded vertices $x \in H$ with $\psi(x) = V_i$, then we define the bipartite graph U_t as follows. One color class is S, the other is V_i, and we have an edge between a $x \in S$ and a $v \in V_i$ whenever $v \in C_{t,x}$.

In the proofs of the above claims the following lemma will play a major role.

Lemma 2. *We are given integers $1 \leq i \leq r$, $1 \leq t \leq T$ and a set $S \subset X_i$ of unembedded vertices at time t with $|S| \geq (\delta''')^2|X_i| = (\delta''')^2 N$. Then apart from an exceptional set F of size at most $\varepsilon'' N$, for every vertex $v \in V_i$ we have the following*

$$deg_{U_t}(v) = |\{x : x \in S, v \in C_{t,x}\}| \geq (1-\varepsilon'')d(U_t)|S| \quad \left(\geq \frac{\delta^\Delta}{2}|S|\right).$$

Proof. In the proof of this lemma we will use the "defect form" of the Cauchy-Schwarz inequality (just as in the original proof of the Regularity Lemma): if

$$\sum_{k=1}^{m} X_k = \frac{m}{n}\sum_{k=1}^{n} X_k + \Delta \qquad (m \leq n)$$

then

$$\sum_{k=1}^{n} X_k^2 \geq \frac{1}{n}\left(\sum_{k=1}^{n} X_k\right)^2 + \frac{\Delta^2 n}{m(n-m)}.$$

Assume indirectly that the statement in Lemma 2 is not true, that is, $|F| > \varepsilon'' N$. Let us write $\nu(t,x)$ for the number of neighbors (in H) of x embedded by time t. Then in U_t we have the following.

$$e(U_t) = d(U_t)|S||V_i| = \sum_{v\in V_i} deg_{U_t}(v) = \sum_{x\in S} deg_{U_t}(x) = \sum_{x\in S} |C_{t,x}| \geq \sum_{x\in S}(\delta-\varepsilon)^{\nu(t,x)} N. \tag{1}$$

We also have

$$\sum_{x_1\in S}\sum_{x_2\in S} |N_{U_t}(x_1)\cap N_{U_t}(x_2)| \leq \sum_{x_1\in S}\sum_{x_2\in S}(\delta+\varepsilon)^{\nu(t,x_1)+\nu(t,x_2)} N + 4\Delta\varepsilon' N|S|^2. \tag{2}$$

On the other hand using (1) and the Cauchy-Schwarz inequality with $m = \lceil \varepsilon'' N \rceil$, we get

$$\begin{aligned}
&\sum_{x_1\in S}\sum_{x_2\in S} |N_{U_t}(x_1)\cap N_{U_t}(x_2)| = \sum_{v\in V_i} (deg_{U_t}(v))^2 \geq \\
&\geq \frac{1}{N}\left(\sum_{v\in V_i} deg_{U_t}(v)\right)^2 + (\varepsilon'')^3 d(U_t)^2 N|S|^2 \geq \\
&\geq \frac{1}{N}\left(\sum_{x\in S}(\delta-\varepsilon)^{\nu(t,x)}N\right)^2 + (\varepsilon'')^3\delta^{2\Delta}N|S|^2 \geq \\
&\geq \sum_{x_1\in S}\sum_{x_2\in S}(\delta-\varepsilon)^{\nu(t,x_1)+\nu(t,x_2)}N + (\varepsilon'')^3\delta^{2\Delta}N|S|^2,
\end{aligned}$$

which is a contradiction with (2), if ε'' is sufficiently large compared to ε' and ε.

An easy consequence of Lemma 2 is the following lemma.

Lemma 3. *We are given integers $1 \leq i \leq r$, $1 \leq t \leq T$, a set $S \subset X_i$ of unembedded vertices at time t with $|S| \geq \delta'''|X_i| = \delta''' N$ and a set $A \subset V_i$ with $|A| \geq \delta'''|V_i| = \delta''' N$. Then apart from an exceptional set F of size at most $(\delta''')^2 N$, for every vertex $x \in S$ we have the following*

$$|A \cap C_{t,x}| \geq \frac{|A|}{2N}|C_{t,x}|. \tag{3}$$

Proof. Assume indirectly that the statement is not true, i.e. there exists a set $S' \subset S$ with $|S'| > (\delta''')^2 N$ such that for every $x \in S'$ (3) does not hold. Once again we consider the bipartite graph $U_t = U_t(S', V_i)$. We have

$$\sum_{v\in A} deg_{U_t}(v) = \sum_{x\in S'} |A\cap C_{t,x}| < \frac{|A|}{2N}\sum_{x\in S'}|C_{t,x}| = \frac{|A|}{2N} d(U_t)|S'|N.$$

On the other hand, applying Lemma 2 for S' we get

$$\sum_{v\in A} deg_{U_t}(v) \geq (1-\varepsilon'')d(U_t)|S'|(|A|-\varepsilon'' N)$$

contradicting the previous inequality.

Finally we have

Lemma 4. *For every $1 \leq t \leq T$ and for every vertex y that is unembedded at time t, we have*

$$|H_{t,y}| > \delta'' N.$$

Proof. Indeed, Lemma 3 implies that every time option 2 is executed in Step 3 (i.e. $T_1|t$), we find at most $(\delta''')^2 N$ ($\ll \delta'' N$) exceptional vertices. This fact clearly implies Lemma 4.

Lemma 2 implies that for E_i in Step 3 of Phase 1 of the Embedding Algorithm we have $|E_i| < \varepsilon'' N$, thus

$$\sum_{i=1}^{r} |E_i| \leq r\varepsilon'' N.$$

Lemma 4 implies that the Selection Algorithm always selects from a set of size at least $\delta'' N$. Furthermore, since $|H_{t,y}| > \varepsilon N$ for every $1 \leq t \leq T$ and for every vertex y unembedded at time t, the ε-regularity and some simple computation imply that at most $\varepsilon' N$ ($\ll \delta'' N$ if ε' is small enough) vertices violate the degree requirements in the Selection Algorithm. Therefore the Selection Algorithm always succeeds in selecting an image $\varphi(x_t)$, proving Claim 1.

Proof of Claim 2. We want to show that we can find a system of distinct representatives of the sets $H_{T,x_j}, T < j \leq n$, where the H_{T,x_j}-s belong to a given cluster V_i.

To simplify notation, let us denote by Y the set of remaining vertices in V_i, and by X the set of remaining unembedded (buffer) vertices assigned to V_i. If $x = x_j \in X$ then write H_x for its possible location H_{T,x_j} at time T. Also write $M = |X| = |Y|$.

The König-Hall condition for the existence of a system of distinct representatives obviously follows from the following three conditions:

$$|H_x| > \delta''' M \quad \text{for all} \quad x \in X, \tag{4}$$

$$|\bigcup_{x\in S} H_x| \geq (1-\delta''')M \quad \text{for all subsets} \quad S \subset X, |S| \geq \delta''' M, \tag{5}$$

$$|\bigcup_{x\in S} H_x| = M \quad \text{for all subsets} \quad S \subset X, |S| \geq (1-\delta''')M. \tag{6}$$

(4) is an immediate consequence of Lemma 4, (5) is a consequence of Lemma 2.

Finally to prove (6), we have to show that every vertex in $Y \subset V_i$ belongs to at least $\delta'''|X|$ sets H_x. But this is trivial from the construction of the embedding algorithm, in Step 3 of Phase 1 we took care of the small number of exceptional vertices for which this is not true. This finishes the proof of Claim 2 and the proof of correctness.

4. Implementation

The sequential implementation is immediate. In Phase 1 we have $\leq n$ iterations, and it is not hard to see that one iteration can be implemented in $O(M(n))$ time. Phase 2 can be implemented in $O(rN^{5/2}) = O(M(n))$ time by applying an algorithm for finding a maximum matching in a bipartite graph (see e.g. [20, 29]).

For the parallel implementation, our main tool is the NC^4 algorithm for the maximal independent set problem. A subset I of the vertices of a graph G is independent if there are no edges between any two vertices in I. An independent set I is *maximal* if it is not a proper subset of any other independent set. Karp and Wigderson ([22]) were the first to give an NC^4-algorithm for this problem. Better algorithms were later described in [4, 17, 18, 30]. We call this the MIS algorithm.

For the parallelization of the embedding algorithm, we show that if α is a small enough constant and n' is the number of remaining unembedded vertices, then we can embed $\lfloor \alpha n' \rfloor$ vertices in parallel. First we pick these vertices by running the MIS algorithm on the following auxiliary graph. The vertices are the vertices of H, and we put an edge between two vertices, if either they are at a distance less than 4, or both vertices are embedded already. If in the maximal independent set that we find, we have a vertex that is embedded already (we can have only one such vertex), then we remove this vertex from the independent set. We keep $\lfloor \alpha n' \rfloor$ vertices from the remaining vertices. These vertices are brought forward in the order in the embedding algorithm and we embed these vertices in parallel. For each such vertex we determine the set of vertices where it could be embedded by the embedding algorithm. Once again running the MIS algorithm on the appropriately defined auxiliary graph, we can choose a distinct representative from these sets. Finally we embed each vertex to its representative. We iterate this procedure until the number of remaining unembedded vertices is $\leq (\log n)^5$, and then we embed these vertices sequentially. Thus Phase 1 can be implemented in $O(\log n (\log n)^4) = O((\log n)^5)$ parallel time.

For Phase 2 it remains to show, that if the bipartite graph U_t is defined as above between X and Y (i.e. there is an edge between $x \in X$ and $v \in Y$ if and only if $v \in H_x$), then we can construct a perfect matching in U_t in NC^4. For this purpose we obtain a maximal matching by running MIS on the linegraph of U_t. Then obviously for the remaining unmatched vertices, say $Z(X)$ and $Z(Y)$, $|Z(X)| = |Z(Y)|$, and $Z(X) \cup Z(Y)$ is an independent set. From (5) $|Z(X)| \leq \delta''' N$ follows. Furthermore, Lemma 3 and Lemma 4 imply that, if we take $x \in Z(X), v \in Z(Y)$, then there are many ($\gg \delta''' N$) (internally) vertex-disjoint alternating paths of length 5 between x and v. Running again MIS on an appropriately defined auxiliary graph, we can find vertex-disjoint alternating paths of length 5 between the pairs in $Z(X) \cup Z(Y)$.

Changing the matching edges to non-matching edges on these paths we get a perfect matching.

5. Applications

In most applications of our method, the only non-constructive parts are the Regularity Lemma and the Blow-up Lemma. Therefore, the existence proofs together with the NC^1 version of the Regularity Lemma and Theorem 1, provide several immediate algorithmic applications. Let us mention here three applications. Additional applications and the details of the proofs will appear in the full version of the paper.

Theorem 5. *(Existential version in* [23], *NC-version in* [34]*) Let* Δ *and* $0 < \delta < 1/2$ *be given. Then there exists an* n_0 *with the following properties. If* $n \geq n_0$, T *is a tree of order* n *and maximum degree* Δ, *and* G *is a graph of order* n *and minimum degree at least* $((1/2) + \delta)n$, *then* T *is a subgraph of* G. *Furthermore, a copy of* T *in* G *can be found in* $O(nM(n))$ *sequential time as well as in* NC^5.

Theorem 6. *(Existential version in* [26]*) There exists an* n_0 *such that if* G *has order* n *with* $n \geq n_0$ *and* $\delta(G) \geq 2n/3$, *then* G *contains the square of a Hamiltonian cycle. Furthermore, a copy of a square of a Hamiltonian cycle can be found in* $O(nM(n))$ *sequential time as well as in* NC^5.

Theorem 7. *(Existential version in* [25]*) For any* $p > 0$ *and positive integer* k *there exists an* $n_0 = n_0(p, k)$ *such that if* G *has order* $n \geq n_0$ *and minimal degree*

$$\delta(G) \geq \left(\frac{k}{k+1} + p\right) n,$$

then G *contains the* k^{th} *power of a Hamiltonian cycle. Furthermore, a copy of the* k^{th} *power of a Hamiltonian cycle can be found in* $O(nM(n))$ *sequential time as well as in* NC^5.

References

[1] N. Alon, *Non-constructive proofs in combinatorics*, in: Proceedings of the International Congress of Mathematicians, Kyoto 1990, Japan, 1421-1429, Springer Verlag, Tokyo, 1991.
[2] N. Alon, *A parallel algorithmic version of the local lemma*, Random Struct. Algorithms 2 (1991), 367-378, in: Proc. 32nd IEEE FOCS (1991), 586-593.
[3] N. Alon, R. Duke, H. Leffman, V. Rödl, R. Yuster, *The algorithmic aspects of the regularity lemma*, Proc. 33rd IEEE FOCS (1993), 479-481, Journal of Algorithms 16 (1994), 80-109.
[4] N. Alon, L. Babai, A. Itai, *A fast randomized parallel algorithm for the maximal independent set problem*, J. Algorithms 7 (1986), 567-583.
[5] N. Alon, R. Yuster, *Almost H-factors in dense graphs*, Graphs and Combinatorics 8 (1992), 95-102.
[6] N. Alon, R. Yuster, *H-factors in dense graphs*, J. Combinatorial Theory Ser. B, to appear.
[7] J. Beck, *An algorithmic approach to the Lovász Local Lemma*, Random Struct. Algorithms 2 (1991), 343-365.

[8] B. Bollobás, P. Erdős, M. Simonovits, E. Szemerédi, *Extremal graphs without large forbidden subgraphs*, in: Advances in Graph Theory, Cambridge Combinatorial Conference, Trinity College, 1977, Ann. Discrete Math. 3 (1978), 29-41.

[9] V. Chvátal, V. Rödl, E. Szemerédi, W. T. Trotter Jr., *The Ramsey number of a graph with bounded maximum degree*, Journal of Combinatorial Theory B34 (1983), 239-243.

[10] S. A. Cook, *Taxonomy of problems with fast parallel algorithms*, Information and Control 64 (1985), 2-22.

[11] P. Erdős, P. Frankl, V. Rödl, *The asymptotic number of graphs not containing a fixed subgraph and a problem for hypergraphs having no exponent*, Graphs Combin. 2 (1986), 113-121.

[12] F. Franek, V. Rödl, *On Erdős conjecture for multiplicities of complete subgraphs*, Graphs Combin., to appear.

[13] P. Frankl, R. L. Graham, V. Rödl, *On subsets of Abelian groups with no 3-term arithmetic progression*, J. Combin. Theory A45 (1987), 157-161.

[14] P. Frankl, Z. Füredi, *Exact solution of some Turán-type problems*, J. Combin. Theory A45 (1987), 226-262.

[15] Z. Füredi, *The maximum number of edges in a minimal graph of diameter 2*, J. Graph Theory 16 (1992), 81-98.

[16] Z. Füredi, M. Goemans, D. J. Kleitman, *On the maximum number of triangles in wheel-free graphs*, to appear.

[17] M. Goldberg, T. Spencer, *A new parallel algorithm for the maximal independent set problem*, SIAM J. Computing 18 (1989), 419-427.

[18] M. Goldberg, T. Spencer, *Constructing a maximal independent set in parallel*, SIAM J. Disc. Math. 2 (1989), 322-328.

[19] A. Hajnal, W. Maass, G. Turán, *On the communication complexity of graph properties*, in: Proc. 20th ACM STOC (1988), 186-191.

[20] J. E. Hopcroft, R. M. Karp, *An* $n^{5/2}$ *algorithm for maximum matchings in bipartite graphs*, SIAM J. Comput. 2 (1973), 225-231.

[21] R. Karp, V. Ramachandran, *Parallel algorithms for shared memory machines*, in: Handbook of Theoretical Computer Science, (J. Van Leeuven ed.) North Holland (1990), 869-941.

[22] R. Karp, A. Wigderson, *A fast parallel algorithm for the maximal independent set problem*, JACM 32 (1985), 762-773.

[23] J. Komlós, G. N. Sárközy, E. Szemerédi, *Proof of a packing conjecture of Bollobás*, Combinatorics, Probability and Computing 4 (1995), 241-255.

[24] J. Komlós, G. N. Sárközy, E. Szemerédi, *The Blow-up Lemma*, to appear in Combinatorica.

[25] J. Komlós, G. N. Sárközy, E. Szemerédi, *On the Pósa-Seymour conjecture*, to appear in the Journal of Graph Theory.

[26] J. Komlós, G. N. Sárközy, E. Szemerédi, *On the square of a Hamiltonian cycle in dense graphs*, Random Structures and Algorithms 9 (1996), 193-211.

[27] J. Komlós, G. N. Sárközy, E. Szemerédi, *Proof of the Alon-Yuster conjecture*, in preparation.

[28] J. Komlós, M. Simonovits, *Szemerédi's Regularity Lemma and its applications in graph theory*, in: Paul Erdős is 80, Vol.2., Proc. Colloq. Math. Soc. János Bolyai (1996), 295-352.

[29] L. Lovász, M. D. Plummer, *Matching Theory*, (Annals Discr. Math. 29), North Holland, 1986.

[30] M. Luby, *A simple parallel algorithm for the maximal independent set problem*, SIAM J. Computing 15 (1986), 1036-1053.

[31] C. H. Papadimitriou, *On graph theoretic lemmata and complexity classes*, Proc. 31st IEEE FOCS (1990), 794-801, *On the complexity of the parity argument and other inefficient proofs of existence*, J. Comput. System Sci. 47 (1993).

[32] N. J. Pippenger, *On simultaneous resource bounds*, Proc. 20th IEEE FOCS (1979), 307-311.
[33] V. Rödl, *On universality of graphs with uniformly distributed edges*, Disc. Math. 59 (1986), 125-134.
[34] G. N. Sárközy, *Fast parallel algorithms for finding Hamiltonian cycles and trees in graphs*, Technical Report 93-81, DIMACS, Rutgers University.
[35] M. Simonovits, V. T. Sós, *Szemerédi's partition and quasirandomness*, Random Sruct. Algorithms 2 (1991), 1-10.
[36] E. Szemerédi, *Regular partitions of graphs*, in: Colloques Internationaux C.N.R.S. (J.-C. Bermond, J.-C. Fournier, M. Las Vergnas, D. Sotteau, Eds.), (1978), 399-401.
[37] E. Szemerédi, *On a set containing no k elements in arithmetic progression*, Acta Arithmetica XXVII (1975), 199-245.
[38] L. G. Valiant, *Parallel computation*, Proc. 7th IBM Symp. on Math. Found. of Comp. Science, 1982.

János Komlós, Mathematics Department, Rutgers University, New Brunswick, NJ 08903

E-mail address: `komlos@math.rutgers.edu`

Gabor N. Sarkozy, Computer Science Department, Worcester Polytechnic Institute, Worcester, MA 01609

E-mail address: `gsarkozy@cs.wpi.edu`

Endre Szemerédi, Computer Science Depratment, Rutgers University, New Brunswick, NJ 08903

E-mail address: `szemered@cs.rutgers.edu`