A Measure For The Lexicographically First Maximal Independent Set Problem and Its Limits

Ryuhei UEHARA

Faculty of Natural Sciences, Komazawa University
uehara@komazawa-u.ac.jp

Abstract

We introduce a structural parameter of a graph, the longest directed path length (LDPL), and express the boundary of the difficulty of the problems along this parameter. The parallel complexity of the lexicographically first maximal independent set (LFMIS) problem gradually increases as the value of LDPL grows; the LFMIS problem on a graph of LDPL $O(\log^k n)$ can be solved in $O(\log^k n)$ time in parallel; and the problem is P-complete on a graph of LDPL $\Theta(n^\epsilon)$. Computing the LDPL itself is in NC^2. This is important in the sense that a "measure" is valid only if measuring the complexity of a problem is easier than solving it. On the other hand, we also show the limits of the measure. We reduce the LFMIS problem to a kind of the lexicographically first maximal subgraph (LFMS) problems on a graph of LDPL 1. This implies that, even on a graph of LDPL 1, the LFMS problem is P-complete. Finally we discuss the probability that a random graph has LDPL l and show that a random graph of which each edge exists with probability p has LDPL $\Theta(np)$ with high probability. This implies the limit of the LDPL on the average case, because the LFMIS problem on a random graph can be efficiently solved in parallel.

Keywords: Analysis of algorithms, NC algorithms, P-completeness, the lexicographically first maximal subgraph problems, threshold function of a random graph.

1 Introduction

The parallel complexity of the problems to find a maximal vertex-induced subgraph that satisfies a specified graph property has been widely investigated. Karp and Wigderson first showed that the typical maximality problem, the maximal independent set (MIS) problem, is in the class NC [1]. Since then, much work has been devoted to the study of parallel complexity of maximality problems (see e.g. [2, 3, 4]). On the other hand, the lexicographically first maximal independent set (LFMIS) problem is a typical P-complete problem [5], and P-completeness of the lexicographically first maximal subgraph (LFMS) problems for some graph properties π has been shown [6, 3] (see also [7] for a comprehensive reference).

As noticed by Iwama and Iwamoto [8], one of the approaches to make clear the boundary between the classes NC and P is to find (sufficient) conditions for problems to be in NC or to be P-complete. They produced a new problem on graphs, called α-connectivity, whose complexity gradually increases as the value of α grows [8]. Our approach is slightly different from theirs. We provide a "measure" of sequentiality for the LFMIS problem. That is, the complexity of the problem on a graph gradually increases as the value measured on the graph grows. The measure is the *longest directed path length* (LDPL) of G, which is defined by the length of the longest directed path on the directed acyclic graph of G. The problem to compute the LDPL is in NC^2. This result is important since a "measure" is valid only if measuring the complexity of a problem is easier than solving the problem. The results for the LFMIS problem are the following: The LFMIS problem on a graph of LDPL $O(\log^k n)$ can be solved in $O(\log^k n)$ time in parallel, and the problem on a graph of LDPL $\Theta(n^\epsilon)$ is P-complete.

We next show a limit of the measure. In an early draft [9], Uehara claimed that the LDPL also measures the sequentiality of the LFMS problems for a local graph property. However, his proof is incomplete. As a counterexample, we show P-completeness of a kind of the LFMS problems even on a graph of LDPL 1.

Finally we show another limit of the measure on the average case. We consider the average value of the LDPL on a random graph. (For the detail of a random graph, see [10, 11] for example.) Let $Q(l)$ be the graph property that the LDPL of the graph is at least l. For a random graph $G(n,p)$, the threshold function for the property $Q(l)$ is $\frac{l}{n}$. The result implies that the LDPL of a random graph $G(n,p)$ is $\Theta(np)$ with high probability. Consequently, for any fixed positive real number $0 < p < 1$, $G(n,p)$ has LDPL $\Theta(n)$ with high probability. On the other hand, the LFMIS problem on the random graph $G(n,p)$ can be solved in $O(\log n)$ time (on average) in parallel [12, 13]. That is, on a random graph, there seems to exist a gap between the measure LDPL and the time required to solve the LFMIS.

2 Preliminaries

We will deal only with graphs and digraphs without loops or multiple edges. Throughout the paper, unless stated otherwise, G always denotes the input (undirected) graph, $V = \{0, 1, \cdots, n-1\}$ and E denote the set of vertices and edges in G, respectively. Without loss of generality, we assume that G is connected. The *neighborhood* of a vertex v in G, denoted by $N_G(v)$, is the set of vertices in G adjacent to v. The *degree* of a vertex v in G is $|N_G(v)|$, and denoted by $\mathrm{d}_G(v)$. Vertices of degree 0 are called *isolated vertices*. For $U \subseteq V$, $N_G(U)$ is $\cup_{u \in U} N_G(u)$, and $G[U]$ is the graph (U, F), where $F = \{\{u,v\} \mid u, v \in U \text{ and } \{u,v\} \in E\}$. For a graph G, $|G|$ denotes the number of vertices in G.

Let $X = \{x_1, x_2, \cdots, x_k\}$ and $Y = \{y_1, y_2, \cdots, y_h\}$ be any subsets of V. (We assume that sets are always sorted, that is, $x_i < x_j$ and $y_i < y_j$ for each $1 \le i < j \le k, h$.) Let $<$ be the total ordering on the sets X and Y defined as follows: $X < Y$ if and only if $x_i = y_i$ for every i with $1 \le i \le k$, or there is an index $i' \ge 1$ such that $x_j = y_j$ for every $1 \le j < i'$, and $x_{i'} < y_{i'}$.

A subset U of V is called an *independent set* if $G[U]$ only contains isolated vertices. A *maximal independent set* (MIS) in G is an independent set that is not properly contained in any other independent set. The MIS problem is to find, given a graph G, an MIS in G. The *lexicographically first maximal independent set* (LFMIS) in G is the MIS I in G such that $I < J$ for every MIS J in G. The LFMIS problem is to find, given a graph G, the LFMIS in G. The *distance* of two vertices in G is the length of a shortest path joining the vertices. The *diameter* of a graph is the greatest distance between any two vertices in the graph.

For $G = (V, E)$, $\vec{G} = (V, \vec{E})$ is defined by the directed acyclic graph (DAG) obtained from G by replacing each edge $\{u, v\}$ by the arc $(\min\{u,v\}, \max\{u,v\})$. (It is easy to see that the resulting graph $\vec{G}$ is acyclic for any graph G.) The *neighborhood* of a vertex v in $\vec{G}$, denoted $N_{\vec{G}}(v)$, is the set of vertices u in $\vec{G}$ such that $(v, u) \in \vec{E}$. The *outdegree* of a vertex v in $\vec{G}$, denoted by $\mathrm{d}^+{}_{\vec{G}}(v)$, is $|N_{\vec{G}}(v)|$, and the *indegree* of a vertex v in $\vec{G}$, denoted by $\mathrm{d}^-{}_{\vec{G}}(v)$, is $|N_G(v)| - |N_{\vec{G}}(v)|$. We mention that any DAG $\vec{G}$ has at least one vertex v with $\mathrm{d}^-{}_{\vec{G}}(v) = 0$, and at least one vertex u with $\mathrm{d}^+{}_{\vec{G}}(u) = 0$ [14, Theorems 16.2 and 16.2']. Especially, $\mathrm{d}^-{}_{\vec{G}}(0) = 0$ and $\mathrm{d}^+{}_{\vec{G}}(n-1) = 0$ for any DAG $\vec{G}$.

Now, we define the measure of the sequentiality for the lexicographically first maximal problems. For $v_1, v_2, \cdots, v_k \in V$, $\alpha = (v_1, v_2, \cdots, v_k)$ is a *directed path* in $\vec{G} = (V, \vec{E})$ if $(v_i, v_{i+1}) \in \vec{E}$ for each $1 \le i < k$. The *length* of α, denoted by $|\alpha|$, is defined by the number of edges in α. (In this case, $|\alpha| = k - 1$.) The *longest directed path length* (LDPL) of G, denoted by LDPL(G), is defined by the maximum length of directed paths in $\vec{G}$.

Let n be a positive integer, and p be a positive real number with $0 < p < 1$ (and p may depend on n). The random graph $G(n,p)$ is a probability space over the set of graphs on the vertex set $\{0, 1, \cdots, n-1\}$ determined by $\Pr[\{i,j\} \text{ is an edge in } G] = p$, with these events mutually independent. (For the detail of a random graph, see [10, 11] for example.) Let Q be a graph property that is not destroyed by the addition of edges to a graph. A function $r(n)$ is called a *threshold function for* Q if $\lim_{n\to\infty} p(n)/r(n) = 0$ implies that almost no graph has property Q, and $\lim_{n\to\infty} p(n)/r(n) = \infty$ implies that almost every graph has property Q.

We use P to denote the class of all polynomial time computable problems, NL to denote the class of all decision problems solvable by nondeterministic Turing machines that use space bounded by $O(\log n)$, and NC^k to denote the class of all problems computable by a uniform polynomial size circuit family of depth $O(\log^k n)$, where n is the size of input, and k is some positive constant. The class NC is defined by $\cup NC^k$ (for further details, refer to [5, 15]). Although the P-completeness is defined via NC^1-reducibility in [5], we use the log space reducibility simply as in [6]: A problem F_0 is said to be *P-complete* if F_0 is in P and for each F in P there are log space computable functions f and g such that $F(x) = g(F_0(f(x)))$ for all inputs. It is well known that the MIS problem is in NC [1, 16], and the LFMIS problem is one of the fundamental P-complete problems [5, 6, 7].

Lemma 1 The problem to compute LDPL(G) is in NC^2.

Proof. Since any language in NL is also in NC^2 (see [15, Figure 7]), we show that the problem is in NL. A nondeterministic Turing machine can compute LDPL(G) guessing a path of a certain length in logarithmic space storing intermediately the length of the initial path segment and the actually visited vertex. (Remark that it is sufficient to store the actually visited vertex since $\vec{G}$ is acyclic.) This establishes the lemma.

Recall that the EREW PRAM is the parallel model where the processors operate synchronously and share a common memory, but no two of them are allowed simultaneous access to a memory cell (whether the access is for reading or for writing in that cell). The CRCW PRAM differs from the EREW PRAM in that both simultaneous reading and simultaneous writing to the same cell are allowed; in case of simultaneous writing, the processor with lowest index succeeds. It is well known that problems solvable by logspace uniform CRCW or EREW PRAM algorithms in time $O(\log^k n)$ using a polynomial-bounded number of processors are in NC^{k+1} (see [17, Section 3.4]).

3 Lexicographically first maximal independent set problem

Let $G = (V, E)$ be a given graph with $|V| = n$ and $|E| = m$. The main theorem in this section is the following:

Theorem 2 (1) The LFMIS problem on the graph of LDPL bounded by $t(n)$ can be solved in $O(t(n))$ time using $O\left(\frac{n+m}{t(n)}\right)$ processors on a CRCW PRAM. (2) The LFMIS problem on the graph of LDPL cn^ϵ is P-complete under log space reductions for any fixed positive constants c and ϵ.

Proof. (1) The following parallel greedy algorithm solves the LFMIS problem on G (see [18, 12]):

1: Set $G' := G$, and $I := \emptyset$.

2: While G' has at least one vertex, do: In parallel, for each vertex v with $\mathrm{d}^-_{\vec{G}'}(v) = 0$, add v into I, and delete v and every vertex in $N_{\vec{G}'}(v)$ from G'.

3: Output I.

We consider the time complexity of the algorithm. The algorithm essentially puts each vertex with $\mathrm{d}^-_{\vec{G}'}(v) = 0$ into I; deletes each vertex whose at least one previous vertex is in I; and puts each vertex into I whose all previous vertices are deleted. From this observation, we can get an unbounded fan-in NOR switching circuit of depth $t(n)$ and of size $n + m$ for solving the problem. It is folklore to evaluate the circuit in $O(t(n))$ time by a CRCW PRAM with the total workload $O(n + m)$.
(2) It is well known that the LFMIS problem is P-complete [6, 7]. In other words, the LFMIS problem on a graph of LDPL $n - 1$ is P-complete. We reduce the general LFMIS problem to the restricted one. For a given graph G with n vertices, we make $f(n)$ copies of the original graph with the same orders, say G', where $f(n) = \left\lceil \left(\frac{n}{c}\right)^{\frac{1}{\epsilon}} \right\rceil$. It is clear that this is a log space reduction, and the LFMIS of G' consists of $f(n)$ copies of the LFMIS of the original graph. Now, LDPL(G') is at most $n - 1$ and the number of vertices of G' is $n' = nf(n)$. Thus, for the resulting graph G' with n' vertices, LDPL$(G') < n \leq c(f(n))^\epsilon < c(nf(n))^\epsilon = cn'^\epsilon$. This completes the proof.

4 Lexicographically first maximal subgraph problem for a local property

Miyano proved the P-completeness of the lexicographically first maximal subgraph problem for a hereditary property [6], and Shoudai and Miyano showed that the maximal subgraph problem for a local property is in NC [4]. Using their techniques, Uehara claimed that we can generalize the results in the previous section in early drafts [18, 9]. However, his proofs are incomplete. We show a counterexample in this section.

For a given graph $G = (V, E)$, a subset U of V is called a *star* if $G[U]$ is acyclic and its diameter is at most 2. The vertex v in a star set U is a *center* of U if $N_{G[U]}(v) \cup \{v\} = U$. A subset U of V is called a *star set* if $G[U]$ only contains disjoint stars. A *maximal star set* (MSS) in G is a star set that is not properly contained in any other star set. The MSS problem is to find, given a graph G, the MSS in G. The MSS problem is in NC; Chen and Kasai showed the NC algorithm that finds a maximal vertex set U such that $G[U]$ is acyclic and its diameter is bounded above by a given integer [19]. The *lexicographically first maximal star set* (LFMSS) in G is the MSS I in G such that $I < J$ for every MSS J in G. The LFMSS problem is to find, given a graph G, the LFMSS in G. A star set satisfies the following properties stated in [6, 4, 18]; infinitely many graphs are star set, and some graph is not a star set (nontrivial); vertex induced subgraph of a star set is also a star set (hereditary); and independent edge set is a star set. Moreover, for a given graph $G = (V, E)$ and a subset U of V, testing if U is a star set in G can be solved in constant time using $n + m$ processors (assigned to each vertex and edge) on a CRCW PRAM by the following algorithm (its implement is not difficult and omitted here):

1. In parallel, each edge of which both endpoints are in U writes its index to both endpoints;
2. In parallel, each edge whose endpoints are both in U checks each endpoint, and labels it with "center" if it has different index from the edge;
3. If there exists an edge of which both endpoints are "center" then the answer is "No", else the answer is "Yes."

We remark that an independent edge is a star and its endpoints are both centers. However, the edge could not be checked in step 3 since both endpoints were not labeled with "center" in step 2.

Theorem 3 The LFMSS problem is P-complete even on a graph of LDPL 1.

Proof. We reduce the (general) LFMIS problem to the LFMSS problem on a graph of LDPL 1. For a

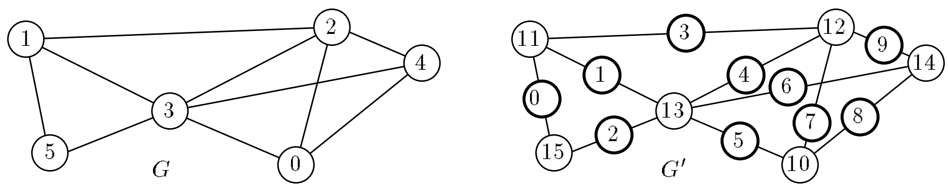

Figure 1: Reduction from the LFMIS to the LFMSS

given graph $G = (V, E)$ with n vertices and m edges, we construct a new graph $G' = (V', E')$ with $n + m$ vertices and $2m$ edges. We first replace each edge in E by a path of length 2. The replacement adds the graph m new vertices. The original vertices i in V are renumbered by $i + m$, and the new vertices are numbered by 0 to $m - 1$ in arbitrary order. Figure 1 depicts an example. Clearly, this is a log space reduction.

We first show that LDPL$(G') = 1$. In G', every path of length 2 forms either (new vertex, original vertex, new vertex) or (original vertex, new vertex, original vertex). Each original vertex is greater than any new vertex. Thus $\vec{G'}$ contains no directed paths of length 2, or LDPL$(G') = 1$.

We next show that the algorithm for the LFMSS on G' also finds the LFMIS on G. Without loss of generality, we assume that G contains at least three vertices. Let U be the LFMSS on G'. Since U is lexicographically first, U must contain independent set $\{0, 1, \cdots, m - 1\}$. That is, U contains all new vertices. Let U' be $U - \{0, 1, \cdots, m - 1\}$. Then it is sufficient to show that U is the LFMSS on G' if and only if U' is the LFMIS on G (we sometimes ignore that the index of u in G is changed to $u + m$ in G' for ease to read). We here show that U contains no pair of original vertices that are joined in G. To derive a contradiction, we suppose that U contains two vertices u and v with $\{u, v\} \in E$. Since G is connected and contains at least three vertices, u or v has another neighbor. We say that w is the other neighbor of u. Then, U is not a star set of G' since four vertices (the new vertex on the edge $\{w, u\}$, the original vertex u, the new vertex on the edge $\{u, v\}$, and the original vertex v) produce a path of length 3. Thus if U is a star set then U' is an independent set of G. Similar argument shows that if U is not a star set then U' is not an independent set of G. Thus U is a star set of G' if and only if U' is an independent set of G. It is easy to see that the maximality of the star set U on G' directly corresponds to the maximality of the independent set U' on G. From the rule of the reduction, it is also easy to see that U is the LFMSS of G' if and only if U' is the LFMIS on G.

Remark: In [9], for a given graph G, Uehara claimed that his greedy parallel algorithm solves the lexicographically first maximal subgraph problem in the time bounded by LDPL(G). We briefly view how his algorithm misses to efficiently find the LFMSS of G' constructed in the proof above, although LDPL(G') = 1. The algorithm, for given G', first finds the lexicographically first independent set $\{0, 1, \cdots, m-1\}$ in a constant time. Then, the algorithm adds auxiliary edges that produce the original graph G in this case. In the time, the algorithm faces the problem on the original graph G of large LDPL, and can not solve it efficiently.

5 LDPL on a random graph

In this section, we consider the average value of the LDPL on a random graph. We denote by $Q(l)$ the graph property that the LDPL of the graph is at least l. Without loss of generality, we assume that l is an increasing function of n with $l < n$. The property $Q(l)$ seems to be similar to the graph property that contains a path of length l, say $Q'(l)$. However, these properties are completely different, and so their threshold functions are. While the threshold function for $Q'(l)$ is $n^{-(l+1)/l}$ [11, Figure 13.1], the threshold function for $Q(l)$ is $\frac{l}{n}$ stated later. The reason is, intuitively, stated as follows: Let K_n be the complete graph with n vertices. Then while the number of the *undirected* (vertex-disjoint) paths of length l in K_n is $\binom{n}{l+1}\frac{(l+1)!}{2}$, the number of the *directed* paths on $\vec{K}_n$ is only $\binom{n}{l+1}$, since a directed path using specific $(l+1)$ vertices is uniquely determined following their order. This observation implies the following proposition.

Proposition 4 Let $N(n,p,l)$ be the expected value of the number of the directed paths of length l on $\vec{G}(n,p)$ with $l \geq 1$. Then $N(n,p,l) = \binom{n}{l+1}p^l$.

Let l_0 be the least integer that satisfies $N(n,p,l_0) \leq 1$. Intuitively speaking, l_0 gives the LDPL on $G(n,p)$ on the average case. Using the equation $\left(\frac{n}{l}\right)^l \leq \binom{n}{l}$ (see [20, Proposition B.2] for example), we get $1 \geq \binom{n}{l_0+1}p^{l_0} \geq \left(\frac{n}{l_0+1}p\right)^{l_0}$, consequently, $np - 1 \leq l_0$. On the other hand, using the equation $\binom{n}{l} \leq \left(\frac{en}{l}\right)^l$ and $N(n,p,l_0-1) > 1$, we also get $1 < \binom{n}{l_0}p^{l_0-1} \leq \left(\frac{en}{l_0}\right)^{l_0} p^{l_0-1}$. This implies that $p < \left(\frac{enp}{l_0}\right)^{l_0}$, consequently, $l_0 < \frac{enp}{p^{\frac{1}{l_0}}} = enp^{1-\frac{1}{l_0}} < enp + 1$ for sufficient large n (or l_0). Thus the LDPL on $G(n,p)$ is nearly equal to np on the average case. We show that the average case occurs with high probability.

Lemma 5 Let d be any positive integer with $d + l_0 \leq n$. Then $N(n,p,d+l_0) \leq (1-p)^d \left(\frac{l_0+1}{l_0+2}\right)^d$.

Proof. Using $\binom{n}{l_0+1}p^{l_0} \leq 1$ and $l_0 \geq np - 1$, we have

$$\begin{aligned} N(n,p,l_0+d) &= \frac{n-l_0-1}{l_0+2}\frac{n-l_0-2}{l_0+3}\cdots\frac{n-l_0-d}{l_0+d+1}\binom{n}{l_0+1}p^{l_0}p^d \\ &\leq \frac{n-l_0-1}{l_0+2}\frac{n-l_0-2}{l_0+3}\cdots\frac{n-l_0-d}{l_0+d+1}\left(\frac{l_0+1}{n}\right)^d \\ &\leq \left(\frac{n-l_0-1}{n}\right)^d\left(\frac{l_0+1}{l_0+2}\right)^d \leq (1-p)^d\left(\frac{l_0+1}{l_0+2}\right)^d. \end{aligned}$$

Lemma 6 We assume that $\frac{n}{p} = o(n^2)$. Let c_1 be any fixed real number with $0 < c_1 < 1$, and l_1 be $c_1 np$. Let ϵ be any given real number with $0 < \epsilon < 1$. Then for sufficiently large n, the LDPL of $G(n,p)$ is at least l_1 with probability greater than $1 - \epsilon$.

Proof. We consider the following algorithm:

1: Set $v := 0$;

2: Find the least vertex u such that $u > v$ and (v,u) is an arc in $\vec{G}(n,p)$;

3: If such u does not exist or the length of the directed path produced by the computed vertices is l_1, halt;

4: Set $v := u$, and go to step 2.

We show that the algorithm finds a directed path of length l_1 with high probability. This implies that the LDPL of $G(n,p)$ is at least l_1. Let X be a random variable defined to be the number of trials required to check whether (v,u) is an arc in $\vec{G}(n,p)$ in step 2. Let $C_1, C_2, \cdots, C_X$ denote the sequence of trials, where C_i is "*success*" if (v,u) is an arc in $\vec{G}(n,p)$, or "*failure*" otherwise. We assume that the algorithm halts because it finds a directed path of length l_1. In this case, C_X is success. We divide the sequence into *epochs* as follows; epoch 1 begins with C_1 and ends with the first success; and epoch i with $i > 1$ begins with the trial following the ith success and ends with the trial on which we obtain the $(i+1)$st success. Define the random variable X_i, for $1 \le i \le l_1$, to be the number of trials in the ith epoch, so that $X = \sum_{i=1}^{l_1} X_i$. The random variable X_i is geometrically distributed with parameter p since the probability of success on any trial is p from the definition of a random graph $G(n,p)$. Thus, the expected value of X_i is $\frac{1}{p}$, and its variance is $\frac{1-p}{p^2}$. Hence, the expected value of X is $\frac{l_1}{p} = c_1 n$, and its variance is $\frac{1-p}{p} c_1 n$. Thus, using Chebyshev's Inequality, we get $\Pr\left[\,|X - c_1 n| \ge \sqrt{\frac{(1-p)c_1 n}{(1-\epsilon)p}}\,\right] \le 1-\epsilon$. Here $\sqrt{\frac{(1-p)c_1 n}{(1-\epsilon)p}} < \sqrt{\frac{c_1}{1-\epsilon}}\sqrt{\frac{n}{p}} = o(n)$. Hence $c_1 n + \sqrt{\frac{(1-p)c_1 n}{(1-\epsilon)p}} < n$ for sufficiently large n. Consequently, the algorithm success to find a path of length l_1 with probability greater than $1-\epsilon$ for sufficiently large n. This completes the proof.

Now we are ready to show the threshold function of $Q(l)$. We remind that the function $r(n)$ is the threshold function of $Q(l)$ if $\lim_{n\to\infty} p(n)/r(n) = 0$ implies that almost no graph has property $Q(l)$, and $\lim_{n\to\infty} p(n)/r(n) = \infty$ implies that almost every graph has property $Q(l)$.

Theorem 7 For $Q(l)$ with $l \ge 1$, $r(n,l) = \frac{l}{n}$ is the threshold function.

Proof. Let l_2 be the least integer that satisfies $N(n,r,l_2) \le 1$ on a random graph $G(n,r)$. Then $enr + 1 \ge l_2 \ge nr - 1$.

We first consider a random graph $G(n,p)$ such that $\lim_{n\to\infty} p(n,l)/r(n,l) = \infty$. Then for any positive fixed constant c_3 with $0 < c_3 < 1$, $c_3 np > enr + 1 \ge l_2$ for sufficiently large n. This implies that $\frac{n}{p} < \frac{n^2}{l_2} = \frac{n^2}{c_2 l} = o(n^2)$. Thus, from Lemma 6, the LDPL of $G(n,p)$ is at least l_2 with probability greater than $1-\epsilon$ for any fixed positive real number ϵ.

Next assume that $\lim_{n\to\infty} p(n,l)/r(n,l) = 0$. Let l_3 be the least integer that satisfies $N(n,p,l_3) \le 1$ on a random graph $G(n,p)$. Since $enp + 1 \ge l_3$, for any fixed positive integer d, $(1+d)l_3 \le (1+d)(enp+1) < nr - 1 \le l_2 < n$ for sufficiently large n. From Lemma 5, $N(n,p,l_3 + dl_3) \le (1-p)^{dl_3}\left(\frac{l_3+1}{l_3+2}\right)^{dl_3}$. When p is a positive constant, $N(n,p,l_3+dl_3)$ converges to 0 since $N(n,p,l_3+dl_3) \le (1-p)^{dl_3}$. When $p = o(1)$, consequently, $l_3 = o(n)$, we have $N(n,p,l_3+dl_3) \le \left(\frac{l_3+1}{l_3+2}\right)^{dl_3} = \left(1 - \frac{1}{l_3+2}\right)^{d(l_3+2)}\left(\frac{l_3+2}{l_3+1}\right)^{2d} \le \left(\left(1+\frac{1}{l_3+1}\right)^2 \frac{1}{e}\right)^d$, which also converges to 0 for large d and n (or l_3). From their definitions, $Q(l)$ is satisfied with probability at most $N(n,p,l)$. Thus the probability that the LDPL of $G(n,p)$ is at least l_3 converges to 0. Since $l_3 < l_2$, the LDPL of $G(n,p)$ is at least l_2 with probability less than ϵ for any fixed positive real number ϵ.

6 Concluding remarks

In an early draft [18], as a measure, Uehara defined the *maximum directed tree size* (MDTS) as follows: A *directed tree* T in G is the edge-induced subgraph of G such that T is a tree and every path from the root to each leaf is a directed path on $\vec{G}$. The *maximum directed tree size* (MDTS) of G is defined by the maximum number of vertices of a directed tree in G. The MDTS equals to $\max\{\,|R(v)|\,\}$, where $R(v)$ is the set of vertices that are reachable from v on $\vec{G}$. Thus, the graph of LDPL l has the MDTS at least l. On the other hand, a complete binary tree of size n has the MDTS n and the LDPL $\lceil \log n \rceil$. Hence, the graph family of LDPL $O(\log n)$ *properly* contains the graph family of MDTS $O(\log n)$. That is, there is a graph such that (1) using the measure LDPL, we can know that the LFMIS can be found in $O(\log n)$ time in parallel unless computing itself, and (2) we cannot know the fact using the measure

MDTS since its MDTS is n. From the viewpoint of preciseness, the LDPL is a better measure than the MDTS. On the other hand, the graph G' in the proof of Theorem 3 has MDTS 3. That is, the proof of Theorem 3 also states that the LFMSS problem is P-complete even on a graph of MDTS 3. Hence we cannot use the LDPL and MDTS as measures of the sequentiality of the general lexicographically first maximal subgraph problem.

Moreover, the LDPL does not completely characterize the parallel complexity of the LFMIS problem. For any fixed positive real number $0 < p < 1$, while the LDPL of a random graph $G(n,p)$ is $\Theta(n)$ with high probability, the LFMIS problem on the random graph can be solved in $O(\log n)$ time (on average) in parallel [12, 13]. That is, on a random graph, there seems to exist a gap between the measure LDPL and the time required to solve the LFMIS.

However, it is worth remarking again that computing the LDPL is in NC^2 for any given graph. In the sense, using the LDPL, we can efficiently measure the sequentiality of the LFMIS problem without solving the problem.

These facts motivate us to find a better measure that characterizes the parallel complexity of P-complete problems more strictly and/or generally.

Acknowledgements

I thank Professor Osamu Watanabe, who noticed an error for the parallel complexity of the LDPL problem in an early draft. I indebted to Professor Seinosuke Toda and Professor Zhi-Zhong Chen for their many helpful comments. Professor Seinosuke Toda is the first man who noticed that we can use the LDPL as another measure. Professor Zhi-Zhong Chen gave many fruitful suggestion for the complexity of computing the measure. My thanks also to Dr. Koichi Yamazaki for discussions and encouragement.

References

[1] R.M. Karp and A. Wigderson. A Fast Parallel Algorithm for the Maximal Independent Set Problem. *Journal of the ACM*, 32(4):762–773, 1985.

[2] K. Diks, O. Garrido, and A. Lingas. Parallel algorithms for finding maximal k-dependent sets and maximal f-matchings. In *ISA '91 Algorithms*, pages 385–395. Lecture Notes in Computer Science Vol. 557, Springer-Verlag, 1991.

[3] D. Pearson and V.V. Vazirani. Efficient Sequential and Parallel Algorithms for Maximal Bipartite Sets. *Journal of Algorithms*, 14:171–179, 1993.

[4] T. Shoudai and S. Miyano. Using Maximal Independent Sets to Solve Problems in Parallel. *Theoretical Computer Science*, 148:57–65, 1995.

[5] S.A. Cook. A Taxonomy of Problems with Fast Parallel Algorithms. *Information and Control*, 64:2–22, 1985.

[6] S. Miyano. The Lexicographically First Maximal Subgraph Problems: P-Completeness and NC Algorithms. *Mathematical Systems Theory*, 22:47–73, 1989.

[7] R. Greenlaw, H.J. Hoover, and W.L. Ruzzo. *Limits to Parallel Computation*. Oxford University Press, 1995.

[8] K. Iwama and C. Iwamoto. α-Connectivity: A Gradually Nonparallel Graph Problem. *Journal of Algorithms*, 20:526–544, 1996.

[9] R. Uehara. Parallel Complexity of the Lexicographically First Maximal Subgraph Problems on Restricted Graphs. Technical Report COMP97-68, IEICE Technical Report, 11/14 1997.

[10] B. Bollobás. *Random Graphs*. Academic Press, 1985.

[11] G. Chartrand and L. Lesniak. *Graphs & Digraphs*. Chapman & Hall, 3rd edition, 1996.

[12] D. Coppersmith, P. Raghavan, and M. Tompa. Parallel Graph Algorithms That Are Efficient on Average. *Information and Computation*, 81:318–333, 1989.

[13] N. Calkin and A. Frieze. Probabilistic Analysis of a Parallel Algorithm for Finding Maximal Independent Sets. *Random Structure and Algorithms*, 1(1):39–50, 1990.

[14] F. Harary. *Graph Theory.* Addison-Wesley, 1972.

[15] D.S. Johnson. A Catalog of Complexity Classes. In J. van Leeuwen, editor, *The Handbook of Theoretical Computer Science, Vol. I: Algorithms and Complexity*, pages 69–161. Elsevier, 1990.

[16] D.C. Kozen. *The Design and Analysis of Algorithms.* Springer-Verlag, 1992.

[17] R.M. Karp and V. Ramachandran. Parallel Algorithms for Shared-Memory Machines. In J. van Leeuwen, editor, *The Handbook of Theoretical Computer Science, Vol. I: Algorithms and Complexity*, pages 870–941. Elsevier, 1990.

[18] R. Uehara. A Measure of Parallelization for the Lexicographically First Maximal Subgraph Problem. In *23rd International Workshop on Graph-Theoretic Concepts in Computer Science (WG '97)*, pages 333–341. Lecture Notes in Computer Science Vol. 1335, Springer-Verlag, 1997.

[19] Z.-Z. Chen and T. Kasai. Parallel Algorithms for the Maximal Tree Cover Problems. *IEICE Trans. Inf. and Syst.*, E75-D(1):30–34, 1992.

[20] R. Motwani and P. Raghavan. *Randomized Algorithms.* Cambridge, 1995.