On the Largest Common Subgraph Problem

Oleg Verbitsky

Nr. 95-01
Ulmer Informatik-Berichte
Januar 1995
On the Largest Common Subgraph Problem

Oleg Verbitsky*
Department of Mathematics, Lviv State University
Universytetska 1, 290602 Lviv, Ukraine

Abstract

Given two graphs \( G_1 = (V_1, E_1) \) and \( G_2 = (V_2, E_2) \), \(|V_1| = |V_2| = n\), to determine whether they have a size-\( k \) common subgraph is one of the earliest examples of an \( \text{NP} \)-complete problem (by a trivial reduction from the Maximum Clique problem). We show that this problem for equally sized \( G_1 \) and \( G_2 \), i.e. when \(|E_1| = |E_2| = m\), remains \( \text{NP} \)-complete. Moreover, the restriction to the case \( k = m - \sqrt{n} \), \( c > 1 \), is also \( \text{NP} \)-complete. In this result \( k \) and \( m \) can hardly be made tighter because the largest Common Subgraph problem for equally sized graphs is reducible to the Graph Isomorphism problem in time \( n^{O(m-k)} \).

Further, we consider the optimization problem of computing the maximum common subgraph size. It is only known that this problem is not harder than computing the maximum clique size (V.Kann, STACS’92), and that it is approximable within factor \( O\left(\frac{n}{\log n}\right) \) (M.Hall-dórsson, 1994). For some \( \epsilon \in (0,1) \), we prove that the largest common subgraph size is not approximable within addend \( n^\epsilon \) unless \( \text{NP} = \text{P} \).

The techniques used are reductions from the problem of distinguishing between graphs with large and small clique size.

*Supported in part by grant No. MGT 000 from the International Science Foundation. Part of this work was done while visiting the Ulm university.
1 Introduction

We address the complexity properties of the Largest Common Subgraph problem (further on LCS), one of the earliest examples of an NP-complete problem. Given two graphs \( G_1 = (V_1, E_1) \) and \( G_2 = (V_2, E_2) \), we denote the size of the largest isomorphic subgraphs of \( G_1 \) and \( G_2 \) by \( \sigma(G_1, G_2) \). The problem is, given \( G_1 \) and \( G_2 \), \( |V_1| = |V_2| = n \), and a natural \( k \), to determine whether \( \sigma(G_1, G_2) \geq k \). This problem is NP-complete by a trivial reduction from the Maximum Clique problem.

In this paper we consider a natural version of LCS for equally sized graphs, i.e. graphs with \( |E_1| = |E_2| = m \). The reduction from Maximum Clique does not work now. To show that LCS OF EQUALLY SIZED GRAPHS remains NP-complete, we suggest a reduction from a generalization of Maximum Clique, namely, from the problem of distinguishing between graphs with large and small maximum clique size. The last problem was shown to be NP-hard owing to the progress in the area of the probabilistically checkable proofs [8, 2, 1].

Furthermore, we consider the restriction of LCS to the case \( k = m - l(n) \) where \( l \) is a prespecified function of \( n \). This problem denoted by LCS\([l(n)]\) is of interest due to its connections to the Graph Isomorphism problem. We show that whereas for superpolynomially small \( l \) the problem LCS\([l(n)]\) is closely related to Graph Isomorphism (see Proposition 3.1), for polynomially small \( l \) it is NP-complete.

Finally, we address the problem of approximating \( \sigma(G_1, G_2) \). M.Halldórsson showed that the maximum common subgraph size is approximable within factor \( O \left( \frac{n}{\log n} \right) \) [6, GT36]. It is known nothing about the hardness of approximating this problem within a constant factor. The method of the present paper allows us to prove that for some \( \epsilon \in (0, 1) \), the value \( \sigma(G_1, G_2) \) is not approximable within addend \( n^\epsilon \) unless \( \text{NP} = \text{P} \). In this connection it should be remarked that some NP-hard optimization problems, e.g. Maximum Packing in Two Containers, are approximable within a constant addend ([11], see also [10]).

Related work. NP-completeness of some problems related to Graph Isomorphism was established by A.Lubiw [15].

Provided \( \text{NP} \neq \text{P} \), non-approximability within a constant addend was shown for some problems admitting FPTAS, e.g. for Knapsack [10].

The developments in the probabilistically checkable proofs allowed one
to prove that many of the well-known optimization problems are hard to approximate (see survey [3]). First this technique was applied to the optimization problem Max Clique. V.Kann [13] showed that Maximum Common Subgraph is reducible to Maximum Clique with respect to an approximation preserving reduction. Some approximation problems reducible to Maximum Clique was shown to be hard in [4] and [17].

The approximation complexity of some variations of Maximum Common Subgraph was investigated in [13].

The paper is organized as follows. Section 2 contains the needed definitions and some auxiliary propositions along with well-known results cited in the paper. We give an account of our results consequently in Theorems 3.1, 3.2, 3.3 of Section 3. A proof of every theorem extends a proof of the previous one. The paper is concluded by the discussions of related open questions in Section 4.

2 Preliminaries

By \( G = (V, E) \) we denote a graph \( G \) whose set of vertices is \( V \) and set of edges is \( E \). The number of vertices \( |V| \) will be denoted by \( n \), \( \nu \) or \( N \). \( G' = (V', E') \) is a subgraph of \( G = (V, E) \) if \( E' \subseteq E \) and \( V' \) is the vertex set of all the edges from \( E' \). Sometimes we will identify a graph with its set of edges. Under this convention, a subgraph is just an arbitrary subset of edges.

Two graphs \( G_1 = (V_1, E_1) \) and \( G_2 = (V_2, E_2) \) are isomorphic iff there exists a bijection \( \phi : V_1 \longrightarrow V_2 \) preserving the adjacent relation. Given two graphs \( G_1 = (V_1, E_1) \) and \( G_2 = (V_2, E_2) \), by \( \sigma(G_1, G_2) \) we denote the maximum size of a common subgraph of \( G_1 \) and \( G_2 \), that is, the maximum \( |E'| \) over isomorphic \( E'_1 \subseteq E_1 \) and \( E'_2 \subseteq E_2 \).

Given two graphs \( G_1 = (V_1, E_1) \) and \( G_2 = (V_2, E_2) \) with \( V_1 \cap V_2 = \emptyset \), we define their (disjoint) union by \( G_1 \cup G_2 = (V_1 \cup V_2, E_1 \cup E_2). \)

Lemma 2.1 For any graphs \( G_1, G_2, G = (V, E) \) it is true

\[
\sigma(G_1 \cup G, G_2 \cup G) = \sigma(G_1, G_2) + |E|.
\]

The proof of Lemma 2.1 will appear in the full version of the paper.
Definition 2.1  Definitions of the problems that occur in the paper.

**LCS—LARGEST COMMON SUBGRAPH**

INPUT: graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, $|V_1| = |V_2|$, a natural number $k$

QUESTION: $\sigma(G_1, G_2) \geq k$?

**LCS OF EQUALLY SIZED GRAPHS**

INPUT: graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, $|V_1| = |V_2|$, $|E_1| = |E_2|$, a natural number $k$

QUESTION: $\sigma(G_1, G_2) \geq k$?

**LCS[l(n)]** (the problem is specified by a function $l : N \rightarrow R$ such that $[l(n)]$ is computable in polynomial time)

INPUT: graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, $|V_1| = |V_2|$, $|E_1| = |E_2|$, a natural number $k$

QUESTION: $\sigma(G_1, G_2) \geq |E_1| - l(|V_1|)$?

**GI—GRAPH ISOMORPHISM** is the problem LCS[0]

$K_n$ is a complete graph on $n$ vertices. $E_n$ stands for a graph on $n$ vertices with the empty set of edges. $\omega(G)$ denotes the size of the largest clique in a graph $G$, that is, the maximal $s$ such that $G$ has a subgraph $K_s$.

Given functions $a$ and $b$, we define the problem of distinguishing between graphs with large and small clique size CLIQUE[$a(n)$, $b(n)$] as follows. Given a graph $G$ on $n$ vertices, it is required to output

- $1$ if $\omega(G) \geq a(n)$,
- $0$ if $\omega(G) < b(n)$;

otherwise the output does not matter.

We use $\leq_{m}^{P}$ to denote the many-one polynomial time reducibility. We say that a language $L$ is many-one reducible to CLIQUE[$a(n)$, $b(n)$] in polynomial time iff there is a polynomial time computable transformation $r$ mapping every $w$ to a graph $G$ on $n$ vertices with $\omega(G) \geq a(n)$ if $w \in L$, and with $\omega(G) < b(n)$ if $w \not\in L$. We say the problem CLIQUE[$a(n)$, $b(n)$] is NP-hard if any NP-language is $\leq_{m}^{P}$-reducible to it.

We say that the problem CLIQUE[$a(n)$, $b(n)$] is many-one reducible to a language $L$ in polynomial time iff there is a polynomial time computable transformation $r$ such that for any graph $G$ on $n$ vertices if $\omega(G) \geq a(n)$ then $r(G) \in L$, and if $\omega(G) < b(n)$ then $r(G) \not\in L$.  

4
Obviously, if $L_1 \leq^P_m \text{CLIQUE}[a(n), b(n)]$ and $\text{CLIQUE}[a(n), b(n)] \leq^P_m L_2$ then $L_1 \leq^P_m L_2$.

**Proposition 2.1** ([8, 2, 1]) i) For some $Q \geq 8$, the problem $\text{CLIQUE}[n/Q, n/4Q]$ is NP-hard.

ii) There are $q \geq 3$ and $C > 1$ such that for

$$\alpha = \frac{C + 1}{(q + 1)C + 1}, \quad \beta = \frac{C}{(q + 1)C + 1}$$  \hspace{1cm} \hspace{1cm} (1)

the problem $\text{CLIQUE}[n^\alpha, n^\beta]$ is NP-hard.

**Remark 2.1** i) is a corollary from the equality $\text{NP} = \text{PCP}(\log n, 1)$ [2, 1]. $Q = 2^q$ where $q$ is a number of oracle queries of an PCP machine recognizing an NP-complete language. More efficient using random sources by the PCP machine [12] allows one to improve i) to ii). The constant $C$ depends on the expanders used.

As a matter of fact, the restrictions of $\text{CLIQUE}[n/Q, n/4Q]$ and $\text{CLIQUE}[n^\alpha, n^\beta]$ to graphs of order $n$ such that $n/Q, n/4Q$ and $n^\alpha, n^\beta$ are integer remain NP-complete. Due to this, we can simplify the exposition assuming $n/Q, n/4Q$ and $n^\alpha, n^\beta$ integer throughout the paper.

**Proposition 2.2** (Turán's theorem, see e.g. [7, 5]) Let $f(a, b)$ denote the maximal possible number of edges in a graph on $a$ vertices without an size-$b$ clique. Then

$$f(a, b) = a(a - 1)/2 - (b - 1)d(d - 1)/2 - dr$$

where

$$a = (b - 1)d + r, \quad 0 \leq r < b - 1.$$ 

The extremal graph $G = (V, E)$ with $|V| = a, |E| = f(a, b), \omega(G) < b$ can be effectively constructed in time $(a^2)$.

**Corollary 2.1** A graph obtained from $K_a$ by deleting arbitrary $a$ edges contains a subgraph $K_{[a/3]}$.

By $F_m$ we denote a graph consisting of $m$ independent edges. $\sigma(F_{[n/2]}, G)$ is the size of the maximal matching in a graph $G$ of order $n$. It is well known that $\sigma(F_{[n/2]}, G)$ can be computed in time $(n^3)$ (see e.g. [9]).
3 The Complexity of the Largest Common Subgraph problem

Theorem 3.1 LCS of equally sized graphs is NP-complete.

Theorem 3.1 is a direct corollary from Proposition 2.1 and the following lemma.

Lemma 3.1 Let a and b be polynomial time computable functions, and $a(n) \geq 3b(n)$ for sufficiently large $n$. Then

$$\text{CLIQUE}[a(n), b(n)] \leq^P \text{LCS of equally sized graphs}.$$  

Proof: We describe a reduction transforming an graph $G = (V, E)$, $|V| = n$, $|E| = m$, into a pair of graphs $G_1, G_2$ and a natural $k$ so that

$$\omega(G) \geq a(n) \implies \sigma(G_1, G_2) \geq k, \quad (2)$$

$$\omega(G) < b(n) \implies \sigma(G_1, G_2) < k. \quad (3)$$

For the brevity we will sometimes denote $a(n)$ just by $a$, and $b(n)$ by $b$. Of course, we can consider only $G$ without isolated vertices. We assume $m \geq a(a - 1)/2 + n/2$, otherwise it is clear that $\omega(G) < a$. Given such $G$, we take $G_1 = K_a \cup E_{m-a(a-1)/2}$ and $G_2 = G \cup E_{2m-a(a-2)-n}$. Clearly, $G_1$ and $G_2$ are equally sized.

Let $s$ be the size of the maximal matching in $G$. We take $k = a(a - 1)/2 + (s - a)$.

As easily seen, if $\omega(G) \geq a$ then $G_1$ and $G_2$ have a common subgraph $K_a \cup E_{s-a}$ of size $k$. Thus, $\sigma(G_1, G_2) \geq k$ and (2) is true.

Suppose now that $\sigma(G_1, G_2) \geq k$. This means there is a subgraph $H$ of $G_1$ that can be embedded into $G$. It is clear that $|H \cap K_a| \geq k - s = a(a - 1)/2 - a$. By Corollary 2.1, $H \cap K_a$ contains a subgraph $K_b$. Thus, $\omega(G) \geq b$ and (3) follows.

The following proposition shows the relations between the problems LCS[$l(n)$] and GI.

Proposition 3.1 i) GI $\leq^P$ LCS[$l(n)$] provided $l(n) \leq \sqrt{n}/3$.

ii) LCS[$l(n)$] is many-one reducible to GI in time $n^{O}(l(n))$.  

6
Proof: The proof of i) will appear in the full version of the paper. As to ii), the straightforward Turing reduction can be turned into a many-one reduction by techniques from [14] involving OR-function for GI.

Theorem 3.2 For any $c > 1$, the problem $LCS[\sqrt{\gamma}]$ is NP-complete.

Theorem 3.2 follows from the first part of Proposition 2.1 and the following lemma by trivial padding arguments.

Lemma 3.2 For any rational $Q$, $CLIQUE[n/Q, n/4Q] \leq^P LCS[\nu/4]$.

Proof: We have to describe a reduction transforming a graph $G = (V, E)$, $|V| = n$, $|E| = m$, into a pair of graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, $|V_1| = |V_2| = \nu$, $|E_1| = |E_2| = \mu$, so that

1. $\omega(G) \geq n/Q$ implies $\sigma(G_1, G_2) \geq \mu - \nu/4$, (4)
2. $\omega(G) < n/4Q$ implies $\sigma(G_1, G_2) < \mu - \nu/4$. (5)

Denote $a = n/Q$, $b = n/4Q$. We can assume $m \leq f(n, b)$; on all the other instances $CLIQUE[a, b]$ can be easily solved because then $\omega(G) \geq b$.

First we take an extremal graph on $n$ vertices without $K_4$-subgraph and delete $m$ edges from it. Denote the obtained graph by $G$. Take the disjoint union of $G$ and $\hat{G}$ and add $n$ more edges to it so that every vertex of one graph is matched to a vertex of the other graph. Denote the obtained graph by $G'$. Obviously, $G'$ has $2n$ vertices, $f(n, b) + n$ edges and a size-$n$ matching. Clearly, $\omega(G) \geq a$ implies $\omega(G') \geq a$, and $\omega(G) < b$ implies $\omega(G') < b$.

Now we apply to $G'$ the reduction from the proof of Lemma 3.1, obtaining two graphs $G'_1 = (V'_1, E'_1)$ and $G'_2 = (V'_2, E'_2)$, $|V'_1| = |V'_2| = N$, $|E'_1| = |E'_2| = M$. We have

$$M = f(n, b) + n, \quad N = 2M - a(a - 2).$$ (6)

For $K = a(a - 1)/2 + (n - a)$, (2) and (3) give

1. $\omega(G) \geq a$ implies $\sigma(G'_1, G'_2) \geq K$, (7)
2. $\omega(G) < b$ implies $\sigma(G'_1, G'_2) < K$. (8)
By Turán's theorem, \( f(n, b) = \frac{1}{2}n^2 - 2Qn - 2Q \). Therefore,

\[
M = \frac{1}{2}n^2 - (2Q - 1)n - 2Q, \\
N = (1 - Q^{-2})n^2 - 2(2Q - 1 - Q^{-1})n - 4Q, \\
K = Q^{-2}n^2 + (1 - \frac{3}{2}Q^{-1})n.
\]

Finally, take a graph \( H \) on \( 4(M - K) - N \) vertices with \( M - 2K \) edges and construct the desired graphs \( G_1 = G'_1 \cup H \) and \( G_2 = G'_2 \cup H \). As easily seen,

\[
\nu = 4(M - K), \quad \mu = 2(M - K).
\]

By Lemma 2.1,

\[
\sigma(G_1, G_2) = \sigma(G'_1, G'_2) + (M - 2K).
\]

Thus,

\[
\sigma(G'_1, G'_2) \geq K \quad \text{implies} \quad \sigma(G_1, G_2) \geq M - K, \\
\sigma(G'_1, G'_2) < K \quad \text{implies} \quad \sigma(G_1, G_2) < M - K.
\]

Summing up this with (7), (8), and (9), we get (4) and (5). 

In the rest of this section we consider an optimization problem of computing the value \( \sigma(G_1, G_2) \), given a pair of graphs \( G_1 = (V_1, E_1) \) and \( G_2 = (V_2, E_2) \), \(|V_1| = |V_2| = n|.

**Definition 3.1** \( \sigma \) is approximable within addend \( g(n) \) iff there is a polynomial time algorithm \( A \) such that

\[
\sigma(G_1, G_2) - g(n) \leq A(G_1, G_2) \leq \sigma(G_1, G_2)
\]

**Theorem 3.3** For some \( \epsilon \in (0, 1) \), \( \sigma(G_1, G_2) \) is not approximable within addend \( n^\epsilon \) unless \( \text{NP} = \text{P} \).

Theorem 3.3 immediately follows from the second part of Proposition 2.1 and the following lemma.
Lemma 3.3 For any \( \varepsilon < \alpha - \beta / 2 \) where \( \alpha \) and \( \beta \) are defined by (1), \( q, C > 1 \), the problem CLIQUE\([n^\alpha, n^\beta]\) is reducible to approximation of \( \sigma(G_1, G_2) \) within addend \( N^\varepsilon \), that is, a polynomial time algorithm approximating the value \( \sigma(G_1, G_2) \) within addend \( N^\varepsilon \) can be transformed into a polynomial time algorithm solving CLIQUE\([n^\alpha, n^\beta]\).

Proof: Denote \( a = n^\alpha \) and \( b = n^\beta \). Consider the reduction from the proof of Lemma 3.2 transforming a graph \( G = (V, E) \), \(|V| = n, |E| = m\), into a pair of graphs \( G_1 = (V_1, E_1) \) and \( G_2 = (V_2, E_2) \), \(|V_1| = |V_2| = N, |E_1| = |E_2| = M\). By (7),

\[
\omega(G) \geq a \implies \sigma(G_1', G_2') \geq n + \frac{1}{2} n^{2\alpha} - \frac{3}{2} n^\alpha. \quad (11)
\]

By Turán's theorem,

\[
f(a, b) = \frac{1}{2} n^{1\alpha} - \frac{1}{2} n^{2\alpha - \beta} - \frac{1}{2} n^{2\alpha - 2\beta} - \frac{1}{2} n^{\alpha - \beta}.
\]

Assume \( \sigma(G_1', G_2') \geq n + \frac{1}{2} n^{2\alpha} - \frac{1}{2} n^{2\alpha - \beta} \). Then for \( G' \) (see the description of the reduction) we have \( \sigma(K_a, G') \geq \frac{1}{2} n^{2\alpha} - \frac{1}{2} n^{2\alpha - \beta} > f(a, b) \). Therefore,

\[
\omega(G') \geq b \text{ and } \omega(G) \geq b. \text{ It is proved that}
\]

\[
\omega(G) < b \implies \sigma(G_1', G_2') < n + \frac{1}{2} n^{2\alpha} - \frac{1}{2} n^{2\alpha - \beta}. \quad (12)
\]

(11) and (12) show that an algorithm approximating \( \sigma(G_1, G_2) \) within addend \( n^\varepsilon \), for any \( \gamma < 2\alpha - \beta \), allows one to distinguish between the cases \( \omega(G) \geq a \) and \( \omega(G) < b \). To conclude the proof it suffices to notice that \( N = n^2(1 - o(1)) \). This follows from (6) and the simple estimates

\[
f(n, b) = n(n - 1)/2 - (n^\beta - 1)d(d - 1)/2 - dr
\]

where

\[
d = \sum_{i=1}^{r+1} n^{1 - i\beta} = \frac{n^{1 - \beta}(1 - n^{-(r+1)\beta})}{1 - n^{-\beta}} < n^{1 - \beta}(1 + 2n^{-\beta}), \quad r = n^{\alpha - \beta},
\]

obtained by Turán's theorem.

Remark 3.1 The slightly updated proof of Theorem 3.3 gives also the following proposition. For any \( \lambda \in (0, 1/2) \) there exist \( \varepsilon, \delta \in (0, 1) \) such that the difference \( n^\lambda - \sigma(G_1, G_2) \), if positive, is not approximable within factor \( n^\varepsilon \) even for \( G_1 \) and \( G_2 \) with \( \frac{1}{2} n(1 - n^{-\delta}) < m < \frac{1}{2} n \) unless \( \text{NP} = \text{P} \).
4 Discussions and Open Problems

4.1 Approximation properties of Maximum Common Subgraph

In this paper we obtained the first hardness approximation result for the Maximum Common Subgraph problem saying that approximating it within addend \( n^c \) is NP-hard. The challenging problem is to show that \( \sigma(G_1, G_2) \) is hard to approximate within factor \( c \), at least for some constant \( c > 1 \). In particular, it is unclear whether Maximum Common Subgraph is Max SNP-hard.

4.2 The case of half-sized graphs

We say that \( G = (V, E) \) is a half-sized graph if \( |V| = n \) and \( |E| = n(n-1)/4 \). The Largest Common Subgraph problem restricted to half-sized graphs \( G_1 = (V_1, E_1), G_2 = (V_2, E_2), |V_1| = |V_2| \), is of interest due to the following fact.

**Proposition 4.1** For any two graphs \( G_1 = (V_1, E_1) \) and \( G_2 = (V_2, E_2) \) with \( |V_1| = |V_2| = n, |E| = n(n-1)/4 \),

\[
\sigma(G_1, G_2) \geq \frac{n(n-1)}{8}.
\]

It is not hard to show that Largest Common Subgraph of Equally Sized Graphs is equivalent to its restriction on half-sized graphs. Denote the restriction of LCS[\( l(n) \)] to half-sized graphs by LCS[\( l(n) \)] of Half-sized Graphs. Theorem 3.2 can be somewhat strengthened.

**Proposition 4.2** For any \( c > 1 \), the problem LCS[\( \sqrt{n} \)] of Half-sized Graphs is NP-complete.

In view of Propositions 4.1 and 4.2 the following question looks interesting. Is it possible to establish the NP-completeness of LCS[\( l(n) \)] of Half-sized Graphs for \( l(n) = \frac{n(n-1)}{8} + o(1) \)? I cannot solve this question even for \( l(n) = cn(n-1), c \in (0, 1/8) \).

Proposition 4.1 shows that \( \sigma(G_1, G_2) \) is trivially approximable within factor 2 for half-sized graphs. What is complexity of finding a size-\( \frac{n(n-1)}{8} \) common subgraph in this case? Can the performance ratio be improved?
4.3 A generalized version of the Graph Automorphism problem

The Graph Isomorphism and Graph Automorphism problems are of particular interest in structural complexity theory (see [14]). It is natural to regard the problem LCS[l(n)] as an edge deletion generalization of GI. Whereas Proposition 3.1 asserts that for superpolynomially small l the problem LCS[l(n)] is closely related to GI, Theorem 3.2 shows that for polynomially small l LCS[l(n)] is NP-complete.

In this subsection we introduce an NP-problem related to Graph Automorphism. The problem deals with the asymmetry measure of a graph that has been studied in combinatorics.

Given a graph $G = (V,E)$, define $A(G) = \{ |D| : E \Delta D \text{ has nontrivial automorphism} \}$ where $D$ is a set of edges on $V$. In words, $A(G)$ is the minimal number of edges to be changed in $G$ in order to obtain a symmetrical graph. A natural problem is defined as follows.

Given a graph $G$ and a natural $k$, to determine whether $A(G) \leq k$.

Is this problem NP-complete?

Concluding, we remark some relevant combinatorial results. It is known that $A(G) < n/2$ for any $G = (V,E)$, $|V| = n$, and $A(G) = n(1/2 - o(1))$ for some sequence of graphs [7]. The proof of these facts in [7] is non-constructive. Some constructivization was achieved in [16] where it was shown that a sequence of graphs $G_n$ with $A(G_n) \geq (1/12 - \epsilon)n$ can be encoded by boolean function computable by constant depth $\text{poly}(\log n)$-size circuits over basis $\{\&\equiv\}$.

Acknowledgement

I am grateful to Johannes Köbler and Pavel Pudlák for conversations on this subject, and to Alexander Razborov for valuable suggestions. I am indebted to Uwe Schöning for his hospitality during my research stay at the Ulm university.
References


List of technical reports published by the University of Ulm

Some of them are available by FTP from ftp.informatik.uni-ulm.de
Reports marked with * are out of print

91-01  Ker-I Ko, P. Orponen, U. Schöning, O. Watanabe
       Instance Complexity

91-02*  K. Gladitz, H. Fassbender, H. Vogler
       Compiler-Based Implementation of Syntax-Directed Functional Programming

91-03  Alfons Geser
       Relative Termination

91-04*  J. Köbler, U. Schöning, J. Toran
       Graph Isomorphism is low for PP

91-05  Johannes Köbler, Thomas Thierauf
       Complexity Restricted Advice Functions

91-06  Uwe Schöning
       Recent Highlights in Structural Complexity Theory

91-07  F. Green, J. Köbler, J. Toran
       The Power of Middle Bit

91-08*  V. Arvind, Y. Han, L. Hamachandra, J. Köbler, A. Lozano,
         M. Mundhenk, A. Ogiwara, U. Schöning, R. Silvestri, T. Thierauf
       Reductions for Sets of Low Information Content

92-01*  Vikraman Arvind, Johannes Köbler, Martin Mundhenk
       On Bounded Truth-Table and Conjunctive Reductions to Sparse and Tally Sets

92-02*  Thomas Noll, Heiko Vogler
       Top-down Parsing with Simultaneous Evaluation of Noncircular Attribute Grammars

92-03  Fakultät für Informatik
       17. Workshop über Komplexitätstheorie, effiziente Algorithmen und Datenstrukturen

92-04  V. Arvind, J. Köbler, M. Mundhenk
       Lowness and the Complexity of Sparse and Tally Descriptions

92-05  Johannes Köbler
       Locating P/poly Optimally in the Extended Low Hierarchy

92-06  Armin Kühnemann, Heiko Vogler
       Synthesized and inherited functions - a new computational model for syntax-directed semantics

92-07  Heinz Fassbender, Heiko Vogler
       A Universal Unification Algorithm Based on Unification-Driven Leftmost Outermost Narrowing
Liste der bisher erschienenen Ulmer Informatik-Berichte
Einige davon sind per FTP von ftp.informatik.uni-ulm.de erhältlich
Die mit * markierten Berichte sind vergriffen
List of technical reports published by the University of Ulm
Some of them are available by FTP from ftp.informatik.uni-ulm.de
Reports marked with * are out of print

91-01  Ker-I Ko, P. Orponen, U. Schöning, O. Watanabe
Instance Complexity

91-02*  K. Gladitz, H. Fassbender, H. Vogler
Compiler-Based Implementation of Syntax-Directed Functional Programming

91-03  Alfons Geser
Relative Termination

91-04*  J. Köbler, U. Schöning, J. Toran
Graph Isomorphism is low for PP

91-05  Johannes Köbler, Thomas Thierauf
Complexity Restricted Advice Functions

91-06  Uwe Schöning
Recent Highlights in Structural Complexity Theory

91-07  F. Green, J. Köbler, J. Toran
The Power of Middle Bit

91-08*  V. Arvind, Y. Han, L. Hamachandra, J. Köbler, A. Lozano, M. Mundhenk, A. Ogiwara, U. Schöning, R. Silvestri, T. Thierauf
Reducions for Sets of Low Information Content

92-01*  Vikraman Arvind, Johannes Köbler, Martin Mundhenk
On Bounded Truth-Table and Conjunctive Reductions to Sparse and Tally Sets

92-02*  Thomas Noll, Heiko Vogler
Top-down Parsing with Simultaneous Evaluationof Noncircular Attribute Grammars

92-03  Fakultät für Informatik
17. Workshop über Komplexitätstheorie, effiziente Algorithmen und Datenstrukturen

92-04  V. Arvind, J. Köbler, M. Mundhenk
Lowness and the Complexity of Sparse and Tally Descriptions

92-05  Johannes Köbler
Locating P/poly Optimally in the Extended Low Hierarchy

92-06  Armin Kühnemann, Heiko Vogler
Synthesized and inherited functions -a new computational model for syntax-directed semantics

92-07  Heinz Fassbender, Heiko Vogler
A Universal Unification Algorithm Based on Unification-Driven Leftmost Outermost Narrowing
92-08  Uwe Schöning  
On Random Reductions from Sparse Sets to Tally Sets

92-09  Hermann von Hasseln, Laura Martignon  
Consistency in Stochastic Network

92-10  Michael Schmitt  
A Slightly Improved Upper Bound on the Size of Weights Sufficient to Represent Any Linearly Separable Boolean Function

92-11  Johannes Köbler, Seinosuke Toda  
On the Power of Generalized MOD-Classes

92-12  V. Arvind, J. Köbler, M. Mundhenk  
Reliable Reductions, High Sets and Low Sets

92-13  Alfons Geser  
On a monotonic semantic path ordering

92-14*  Joost Engelfriet, Heiko Vogler  
The Translation Power of Top-Down Tree-To-Graph Transducers

93-01  Alfred Lupper, Konrad Froitzheim  
AppleTalk Link Access Protocol basierend auf dem Abstract Personal Communications Manager

The COCOON Object Model

93-03  Thomas Thierauf, Seinosuke Toda, Osamu Watanabe  
On Sets Bounded Truth-Table Reducible to P-selective Sets

93-04  Jin-Yi Cai, Frederic Green, Thomas Thierauf  
On the Correlation of Symmetric Functions

93-05  K. Kuhn, M. Reichert, M. Nathe, T. Beuter, C. Heinlein, P. Dadam  
A Conceptual Approach to an Open Hospital Information System

93-06  Klaus Gaßner  
Rechnerunterstützung für die konzeptionelle Modellierung

93-07  Ulrich Kessler, Peter Dadam  
Towards Customizable, Flexible Storage Structures for Complex Objects

94-01  Michael Schmitt  
On the Complexity of Consistency Problems for Neurons with Binary Weights

94-02  Armin Kühnemann, Heiko Vogler  
A Pumping Lemma for Output Languages of Attributed Tree Transducers

94-03  Harry Buhrman, Jim Kadin, Thomas Thierauf  
On Functions Computable with Nonadaptive Queries to NP

94-04  Heinz Faßbender, Heiko Vogler, Andrea Wedel  
Implementation of a Deterministic Partial E-Unification Algorithm for Macro Tree Transducers
94-05  V. Arvind, J. Köbler, R. Schuler  
On Helping and Interactive Proof Systems

94-06  Christian Kalus, Peter Dadam  
Incorporating record subtyping into a relational data model

94-07  Markus Tresch, Marc H. Scholl  
A Classification of Multi-Database Languages

94-08  Friedrich von Henke, Harald Rueß  
Arbeitstreffen Typtheorie: Zusammenfassung der Beiträge

Construction and Deduction Methods for the Formal Development of Software

94-10  Axel Dold  
Formalisierung schematischer Algorithmen

94-11  Johannes Köbler, Osamu Watanabe  
New Collapse Consequences of NP Having Small Circuits

94-12  Rainer Schuler  
On Average Polynomial Time

94-13  Rainer Schuler, Osamu Watanabe  
Towards Average-Case Complexity Analysis of NP Optimization Problems

94-14  Wolfram Schulte, Ton Vullinghs  
Linking Reactive Software to the X-Window System

94-15  Alfred Lupper  
Namensverwaltung und Adressierung in Distributed Shared Memory-Systemen

94-16  Robert Regn  
Verteilte Unix-Betriebssysteme

94-17  Helmuth Partsch  
Again on Recognition and Parsin of Context-Free Grammars: Two Exercises in Transformational Programming

94-18  Helmuth Partsch  
Transformational Development of Data-Parallel Algorithms: an Example

95-01  Oleg Verbitsky  
On the Largest Common Problem