On the Approximability of the Maximum Common Subgraph Problem

Viggo Kann
Department of Numerical Analysis and Computing Science
Royal Institute of Technology
S-100 44 Stockholm
Sweden
viggo@nada.kth.se

Abstract

Some versions of the maximum common subgraph problem are studied and approximation algorithms are given. The maximum bounded common induced subgraph problem is shown to be Max SNP-hard and the maximum unbounded common induced subgraph problem is shown to be as hard to approximate as the maximum independent set problem. The maximum common induced connected subgraph problem is still harder to approximate and is shown to be NPO PB-complete, i.e. complete in the class of optimization problems with optimal value bounded by a polynomial.
Key words: Approximation, graph problems, computational complexity.
AMS(MOS) subject classification: 68R10, 68Q25, 41A99.
Abbreviated title: Approximability of the Max Common Subgraph Problem

1 Introduction

The Subgraph isomorphism problem is a famous NP-complete problem. It is one of the first problems mentioned in *Computers and Intractability* by Garey and Johnson [11]. Given two graphs the problem is to decide whether the second graph is isomorphic to any subgraph of the first graph. The problem is shown to be NP-complete by the following simple reduction from the Clique problem. Let the first input graph to Subgraph isomorphism be the input graph to Clique and let the second input graph be a K-clique where K is the bound in the Clique problem. Now the K-clique is isomorphic to a subgraph of the first graph if and only if there is a clique of size K or more in the graph.

A related optimization problem is called Maximum common subgraph. In this problem we are given two graphs and we want to find the largest subgraphs which are isomorphic. The corresponding decision problem was shown to be NP-complete by Garey and Johnson using the same reduction as above [11]. The approximation properties of various versions of this problem are studied in this paper.

NP problems, like Maximum common subgraph, which are in fact optimization problems are called NPO problems (NP optimization problems). Provided that P $\neq$ NP there is no algorithm which finds the optimal solution to an NP-complete optimization problem in polynomial time. Still there can exist *polynomial time approximation algorithms* for the problem. However the approximability of different NPO problems differs enormously.

For example the TSP (Travelling Salesperson Problem) with triangular inequality can be solved approximately within a factor 3/2, i.e. one can in polynomial time find a trip of length at most 3/2 times the shortest trip possible [8]. The general TSP cannot be approximated within any constant factor if P $\neq$ NP [11]. Another example is the knapsack problem, which is NP-complete but can be approximated within every constant in polynomial time [12]. Such a scheme for approximating within every constant is called a PTAS (polynomial time approximation scheme).

In 1988 Papadimitriou and Yannakakis defined, using Fagin's logical classification of NP [10], the classes MAX NP and MAX SNP together with a concept of reduction, called *L-reduction*, which preserves approximability within constants [23]. All problems in MAX NP and MAX SNP can be approximated within a constant in polynomial time. Several maximization problems were shown to be complete in MAX SNP under L-reductions, for example maximum 3-satisfiability, maximum cut and maximum 3-set packing in a graph with bounded degree [4, 5, 14, 17, 16, 23, 24]. Minimization problems can also be placed in MAX SNP through L-reductions to maximization problems.

All attempts to construct a PTAS for a MAX SNP-complete problem have failed and hence it seems reasonable to conjecture that no such scheme exists, in particular since if one problem had a PTAS then every problem in MAX SNP would admit a PTAS. Recently Arora, Lund, Motwani, Sudan and Szegedy confirmed this conjection using proof verification in interactive proofs [1]. Thus showing a problem to be MAX SNP-complete describes the approximability of the problem very well: it can be approximated within a constant, but it cannot be approximated within every constant.

The last years more work has been done on logical definability of NPO problems [19, 18, 22]. New classes over MAX SNP and MAX NP and analogous minimization classes have been defined. Not many of these classes seem to capture approximation properties, however. The value of classifying problems using logical definability can be discussed because the same problem may or may not be included in a class depending on how it is formulated [15].

Krentel has defined a class of optimization problems called OPTP[$\log n$], which consists of all NPO problems which are polynomially bounded, that is all problems satisfying $opt(I) \leq p(|I|)$ for all problem instances I, where p is a polynomial [20]. In this paper we shall call this class NPO PB. Some problems, for example the LONGEST INDUCED PATH problem, are NPO PB-complete under L-reductions [3]. An NPO PB-complete problem cannot be approximated within $O(n^{\varepsilon})$ for any $\varepsilon > 0$, unless P = NP. Note that this is a different reduction from the one Krentel used when he defined OPTP[$\log n$]-complete problems.

Several other attempts have been made to find a theory explaining why a problem enjoys particular approximation properties. See [7] and [16] for surveys of the field.

2 Definitions

Definition 1 [9] An NPO problem (over an alphabet Σ) is a tuple $F = (\mathcal{I}_F, S_F, m_F, opt_F)$ where

- $\mathcal{I}_{\mathcal{F}} \subseteq \Sigma^*$ is the space of *input instances*. It is recognizable in polynomial time.
- $S_F(x) \subseteq \Sigma^*$ is the space of *feasible solutions* on input $x \in \mathcal{I}_F$. The only requirement on S_F is that there exist a polynomial q and a polynomial time computable predicate π such that for all x in $\mathcal{I}_F$, S_F can be expressed as $S_F(x) = \{y : |y| \leq q(|x|) \wedge \pi(x, y)\}$ where q and π only depend on F.

- $m_F : \mathcal{I}_F \times \Sigma^* \to \mathbb{N}$, the *objective function*, is a polynomial time computable function. $m_F(x, y)$ is defined only when $y \in S_F(x)$.
- $opt_F \in \{\max, \min\}$ tells if F is a *maximization* or a *minimization* problem.

Solving an optimization problem F given the input $x \in \mathcal{I}_F$ means finding a $y \in S_F(x)$ such that $m_F(x, y)$ is optimum, that is as big as possible if $opt_F = \max$ and as small as possible if $opt_F = \min$. Let $opt_F(x)$ denote this optimal value of m_F. Approximating an optimization problem F given the input $x \in \mathcal{I}_F$ means finding any $y' \in S_F(x)$. How good the approximation is depends on the relation between $m_F(x, y')$ and $opt_F(x)$.

Definition 2 The *relative error* of a feasible solution with respect to the optimum of an NPO problem F is defined as

$$\mathcal{E}_F^r(x, y) = \frac{|opt_F(x) - m_F(x, y)|}{opt_F(x)}$$

where $y \in S_F(x)$.

Definition 3 We say that a maximization problem F *can be approximated within* p if there exists a polynomial time algorithm A such that for all instances $I \in \mathcal{I}_F, A(I) \in S_F(I) \wedge opt_F(I)/m_F(A(I)) \leq p$.

Definition 4 [23] Given two NPO problems F and G and a polynomial time transformation $f : \mathcal{I}_F \to \mathcal{I}_G$. f is an *L-reduction* from F to G if there are positive constants α and β such that for every instance $I \in \mathcal{I}_F$

i) $opt_G(f(I)) \leq \alpha \cdot opt_F(I)$,

ii) for every solution of $f(I)$ with measure c_2 we can in polynomial time find a solution of I with measure c_1 such that $|opt_F(I) - c_1| \leq \beta\ |opt_G(f(I)) - c_2|$.

Papadimitriou and Yannakakis have shown that the composition of L-reductions is an L-reduction and that if F L-reduces to G with constants α and β and there is a polynomial time approximation algorithm for G with worst-case relative error ε, then there is a polynomial time approximation algorithm for F with worst-case relative error $\alpha\beta\varepsilon$ [23]. Thus the L-reduction preserves approximability within constants.

When analyzing approximation algorithms for problems which cannot be approximated within a constant one usually specifies the approximability using a one variable function where the parameter concerns the size of the input instance. For example the maximum independent set problem can be approximated within $O\left(n/(\log n)^2\right)$ where the parameter n is the number of nodes in the input graph [6]. When reducing between two such problems, say from F to G, the L-reduction is not perfect. The trouble is that it may transform an input instance of F to a much larger input instance of G. One purpose of a reduction is to be able to use an approximation algorithm for G to construct an equally good (within a constant) approximation algorithm for F. Because of the size amplification the constructed algorithm will not be as good as the original algorithm.

In order to tell how the approximability, when given as a function, will be changed by a reduction, we have to specify how the size of the input instance will be amplified. For the problems in this article we will use the number of nodes as the measure of input size. We say that an L-reduction has *node amplification* $f(n)$ if it transforms a graph with n nodes into a graph with $f(n)$ nodes. If the node amplification is $O(n)$, i.e. if the size of the constructed structure

is a constant times the size of the original structure, we say that the reduction is *without node amplification.*

We can see that for constant and polylogarithmic approximable problems the L-reduction preserves approximability within a constant for any polynomial node amplification, since $c\log^k(n^p) = p^k c\log^k n = O(\log^k n)$. For n^c approximable problems the L-reduction preserves approximability within a constant just for node amplification $O(n)$, i.e. without node amplification.

Definition 5 Definition of the problems mentioned in the text.

MAX CIS *Maximum common induced subgraph.*
$\mathcal{I} = \{G_1 = \langle V_1, E_1\rangle, G_2 = \langle V_2, E_2\rangle \text{ graphs}\}$
$S(\langle G_1, G_2\rangle) = \{V_1' \subseteq V_1, V_2' \subseteq V_2, f : V_1' \to V_2'$ bijective function such that $G_1|_{V_1'}$ and $G_2|_{V_2'}$ are f-isomorphic, that is
$\forall v_1, v_2 \in V_1, (v_1, v_2) \in E_1 \Leftrightarrow (f(v_1), f(v_2)) \in E_2\}$
$m(\langle G_1, G_2\rangle, \langle V_1', V_2'\rangle) = |V_1'| = |V_2'|$
$opt = \max$
We say that v *is matched with* $f(v)$ and that $f(v)$ *is matched with* v.

MAX CES *Maximum common edge subgraph.*
$\mathcal{I} = \{G_1 = \langle V_1, E_1\rangle, G_2 = \langle V_2, E_2\rangle \text{ graphs}\}$
$S(\langle G_1, G_2\rangle) = \{E_1' \subseteq E_1, E_2' \subseteq E_2, f : V_1' \to V_2'$ bijective function from the nodes in the subgraph $G_1|_{E_1'}$ to the nodes in $G_2|_{E_2'}$ $suchthat G_1|_{E_1'}$ and $G_2|_{E_2'}$ are f-isomorphic, that is $\forall v_1, v_2 \in V_1', (v_1, v_2) \in E_1' \Leftrightarrow (f(v_1), f(v_2)) \in E_2'\}$
$m(\langle G_1, G_2\rangle, \langle E_1', E_2'\rangle) = |E_1'| = |E_2'|$
$opt = \max$

MAX CIS $-B$ *Maximum bounded common induced subgraph.* This is the same problem as MAX CIS but the degree of the graphs G_1 and G_2 is bounded by the constant B.

MAX CES $-B$ *Maximum bounded common edge subgraph.* This is the same problem as MAX CES but the degree of the graphs G_1 and G_2 is bounded by the constant B.

MAX CICS *Maximum common induced connected subgraph.* This is the same problem as MAX CIS but the only valid solutions are connected subgraphs.

MAX 3SAT $-B$ *Maximum bounded 3-satisfiability.*
$\mathcal{I} = \{U$ set of variables, C set of disjunctive clauses, each involving at most three literals (a variable or a negated variable) and such that the total number of occurrences of each variable is bounded by the constant $B\}$
$S(\langle U, C\rangle) = \{C' \subseteq C :$ there is a truth assignment for U such that every clause in C' is satisfied$\}$
$m(\langle U, C\rangle, C') = |C'|$
$opt = \max$

MAX CLIQUE *Maximum clique in a graph.*
$\mathcal{I} = \{G = \langle V, E\rangle : G \text{ is a graph}\}$
$S(\langle V, E\rangle) = \{V' \subseteq V : v_1, v_2 \in V' \wedge v_1 \neq v_2 \Rightarrow (v_1, v_2) \in E\}$
$m(\langle V, E\rangle, V') = |V'|$
$opt = \max$

LIP *Longest induced path in a graph.*
$\mathcal{I} = \{G = \langle V, E\rangle : G \text{ is a graph}\}$
$S(\langle V, E\rangle) = \{V' \subseteq V : G|_{V'}$ is a simple path$\}$
$m(\langle V, E\rangle, V') = |V'|$
$opt = \max$

3 Approximation algorithms

Theorem 1 *Maximum bounded common induced subgraph* (MAX CIS $-B$) *can be approximated within* $B+1$.

PROOF We use that independent sets of the same size are always isomorphic. The following trivial algorithm finds an independent set V_1' in the graph $G_1 = \langle V_1, E_1 \rangle$.

- $V_1' \leftarrow \emptyset$
- Pick nodes from V_1 in any order and add each node to V_1' if none of its neighbours are already added to V_1'.

This algorithm will create a set of size $|V_1'| \geq |V_1| / (B+1)$ because for each node in V_1 either the node itself or one of its at most B neighbour nodes must be in V_1'.

Applying the algorithm to G_1 and G_2 gives us two independent sets V_1' and V_2'. If they are of different size, remove nodes from the largest set until they have got the same size. These sets are a legal solution of the problem of size at least $\min(|V_1|, |V_2|)/(B+1)$. Since the optimal solution has size at most $\min(|V_1|, |V_2|)$ the algorithm approximates the problem within the constant $B+1$. □

Lemma 2 *A maximum matching of a graph* $G = \langle V, E \rangle$ *with degree at most* B *contains at least* $|E| / (B+1)$ *edges.*

PROOF Let ν be the number of edges in the maximum matching and p be the number of nodes in the graph. If there exists a perfect matching, then $p = 2\nu$ and

$$\frac{|E|}{B+1} \leq \frac{p \cdot B/2}{B+1} < \frac{p}{2} = \nu.$$

If $p \geq 2\nu + 1$ the inequality $|E| \leq (B+1)\nu$ follows from [21, theorem 3.4.6]. □

Theorem 3 *Maximum bounded common edge subgraph* (MAX CES $-B$) *can be approximated within* $B+1$.

PROOF We use the same idea as in the proof of Theorem 1, but create an independent set of *edges* instead. In polynomial time we can find a maximum matching of the graphs. The size of the smallest maximum matching is by Lemma 2 at least $\min(|E_1|, |E_2|)/(B+1)$ and the size of the optimal solution is at most $\min(|E_1|, |E_2|)$, so we can approximate this problem too within the constant $B+1$. □

4 Reductions between the problems

Theorem 4 *There is a reduction from* MAX CIS *to* MAX CLIQUE *which is an L-reduction with node amplification* n^2.

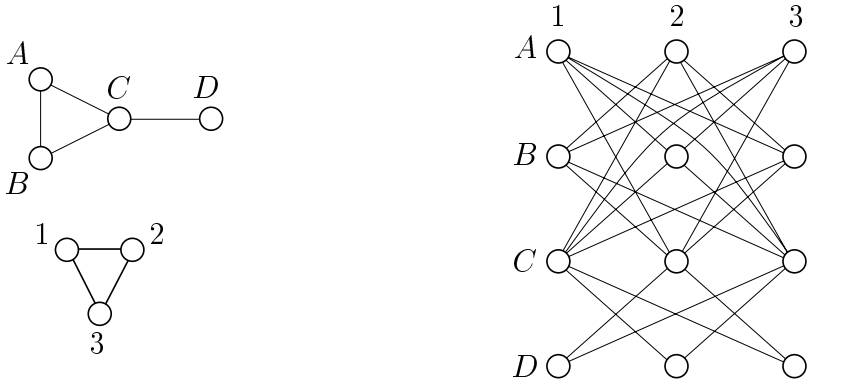

Figure 1: *An example of the transformation from MaxCIS to MaxClique in Theorem 4. The maximum size cliques in the right graph have size three, e.g. (A1,B3,C2) which corresponds to a common subgraph matching between node A and 1, B and 3, C and 2 in the left graphs.*

PROOF From the input graphs G_1 and G_2 we form a *derived graph* $G = \langle V, E \rangle$ in the following way (due to Barrow and Burstall [2]).

Let $V = V_1 \times V_2$ and call V a set of *pairs*. Call two pairs $\langle v_1, v_2 \rangle$ and $\langle w_1, w_2 \rangle$ *compatible* if $v_1 \neq w_1$ and $v_2 \neq w_2$ and if they preserve the edge relation, that is there is an edge between v_1 and w_1 if and only if there is an edge between v_2 and w_2. Let E be the set of compatible pairs. See Figure 1.

A k-clique in the derived graph G can be interpreted as a matching between two induced k-node subgraphs. The subgraphs are isomorphic since the compatible pairs preserve the edge relation.

Thus, if we have a polynomial time approximation algorithm for the MAX CLIQUE problem which finds a solution within p we can apply it to the derived graph and use the answer to get an approximate solution for the MAX CIS problem of exactly the same size. Since the maximum clique corresponds to the maximum common induced subgraph this yields a solution within p for the MAX CIS problem and we have an L-reduction from MAX CIS to MAX CLIQUE with $\alpha = \beta = 1$.

For example we can use the MAX CLIQUE approximation algorithm by Boppana and Halldórsson [6]. This algorithm will, for a graph of size n, find a clique of size at least $O((\log n)^2/n)$ times the size of the maximum clique.

Note that in spite of the size of the optimal solution being preserved by the reduction, the size of the problem instance is increased. If the two input graphs of the MAX CIS problem contain m_1 and m_2 nodes respectively, the constructed graph will contain $m_1 m_2$ nodes and the algorithm will only guarantee a common induced subgraph of size $O((\log m_1 m_2)^2/(m_1 m_2))$ times the size of the maximum common induced subgraph.

Thus the reduction is an L-reduction with node amplification n^2. □

Theorem 5 *There is a reduction from* MAX CES *to* MAX CLIQUE *which is an L-reduction with node amplification* n^2.

PROOF We use the same idea as in the preceding proof but the pairs are now pairs of directed edges instead of pairs of nodes. Let $V = A_1 \times A_2$, where A_i consists of two directed edges, $\overleftarrow{e}$ and $\overrightarrow{e}$, for each edge $e \in E_i$. We say that two pairs $\langle \overrightarrow{m_1}, \overrightarrow{m_2} \rangle$ and $\langle \overrightarrow{n_1}, \overrightarrow{n_2} \rangle$ are compatible if $\overrightarrow{m_1} \neq \overrightarrow{n_1}$, $\overrightarrow{m_1} \neq \overleftarrow{n_1}$, $\overrightarrow{m_2} \neq \overrightarrow{n_2}$, $\overrightarrow{m_2} \neq \overleftarrow{n_2}$ and they preserve the node relation, that is $\overrightarrow{m_1}$ and $\overrightarrow{n_1}$ are incident

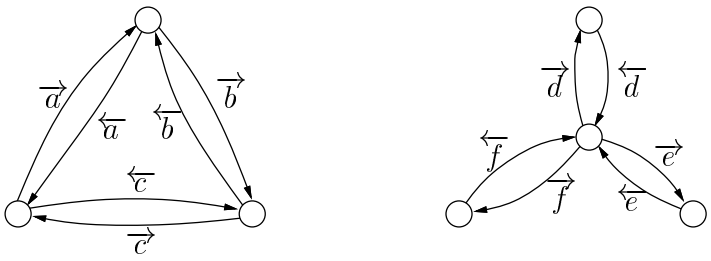

Figure 2: *The digraphs resulting from a triangle and a 3-star in Theorem 5.*

to the same node if and only if $\overrightarrow{m_2}$ and $\overrightarrow{n_2}$ are incident to the same node *in the same way*. For example, in Figure 2 $\langle \overrightarrow{a}, \overleftarrow{d} \rangle$ is compatible with $\langle \overrightarrow{b}, \overrightarrow{e} \rangle$, $\langle \overrightarrow{b}, \overrightarrow{f} \rangle$, $\langle \overleftarrow{b}, \overleftarrow{e} \rangle$ and $\langle \overleftarrow{b}, \overleftarrow{f} \rangle$ but not with e.g. $\langle \overleftarrow{b}, \overrightarrow{e} \rangle$ or $\langle \overrightarrow{b}, \overleftarrow{e} \rangle$.

A k-clique in the derived graph can be interpreted as a matching between two edge subgraphs with k edges in each subgraph. The subgraphs are isomorphic since the compatible pairs preserve the node relation.

Thus we get an L-reduction from MAX CES to MAX CLIQUE with $\alpha = \beta = 1$ which we can use to transform a MAX CLIQUE approximation algorithm to a MAX CES approximation algorithm. As in Theorem 4 the reduction has node amplification n^2. □

Theorem 6 MAX CIS $-B$ *is* MAX SNP*-hard when* $B \geq 25$.

PROOF The problem MAX 3SAT -6, where each variable is in at most six clauses is known to be MAX SNP-complete [23]. We assume that each variable x_i occurs both as x_i in some clause and as $\overline{x}_i$ in some other clause, that no clause is trivially satisfied (e.g. $x_i \vee \overline{x}_i$) and that there are more clauses than variables. The problem is still MAX SNP-complete under these assumptions. We will show that there is an L-reduction f_1 from this problem to MAX CIS $-B$. Let $U = \{x_1, x_2, \ldots, x_n\}$ be the variables and $C = \{c_1, c_2, \ldots, c_m\}$ be the clauses of the input instance.

f_1 takes the sets U and C and constructs a MAX CIS $-B$ instance (the graphs G_1 and G_2) in the following way. G_1 and G_2 are similar and consist of $6n$ *literal nodes* (six for each variable), $18m$ *clique nodes* (18 for each clause) and a number of *clause nodes*. G_1 has $7m$ clause nodes (seven for each clause) and G_2 has m clause nodes (one for each clause). The clique nodes are connected in 18-cliques (m in each graph). In both graphs the six literal nodes for a variable x_i are connected in two 3-cliques — one 3-clique we call the x_i *clique* and the other 3-clique we call the $\overline{x}_i$ *clique*.

In G_2 each clause node is connected with one of the clique nodes in the corresponding 18-clique and with all the literal nodes corresponding to the at most three literals which are contained in the corresponding clause in the MAX 3SAT -6 problem. This completes the description of graph G_2. G_1 has edges between each pair of literal nodes which corresponds to the same variable (i.e. building a 6-clique). Finally there are some edges from the clause nodes to the clique nodes and literal nodes in G_1. Number the seven clause nodes of clause c_j from 1 to 7 and the 18 clique nodes in the corresponding clique from 1 to 18. Now add edges between clause node i and clique node i for i from 1 to 7. Call the three literal 3-cliques corresponding to the three literals in c_i A, B and C. Add edges between clause node 1 and each node in A, 2 and B, 3 and A, 3 and B, 4 and C, 5 and A, 5 and C, 6 and B, 6 and C, 7 and A, 7 and B, 7 and C. If c_i only has two

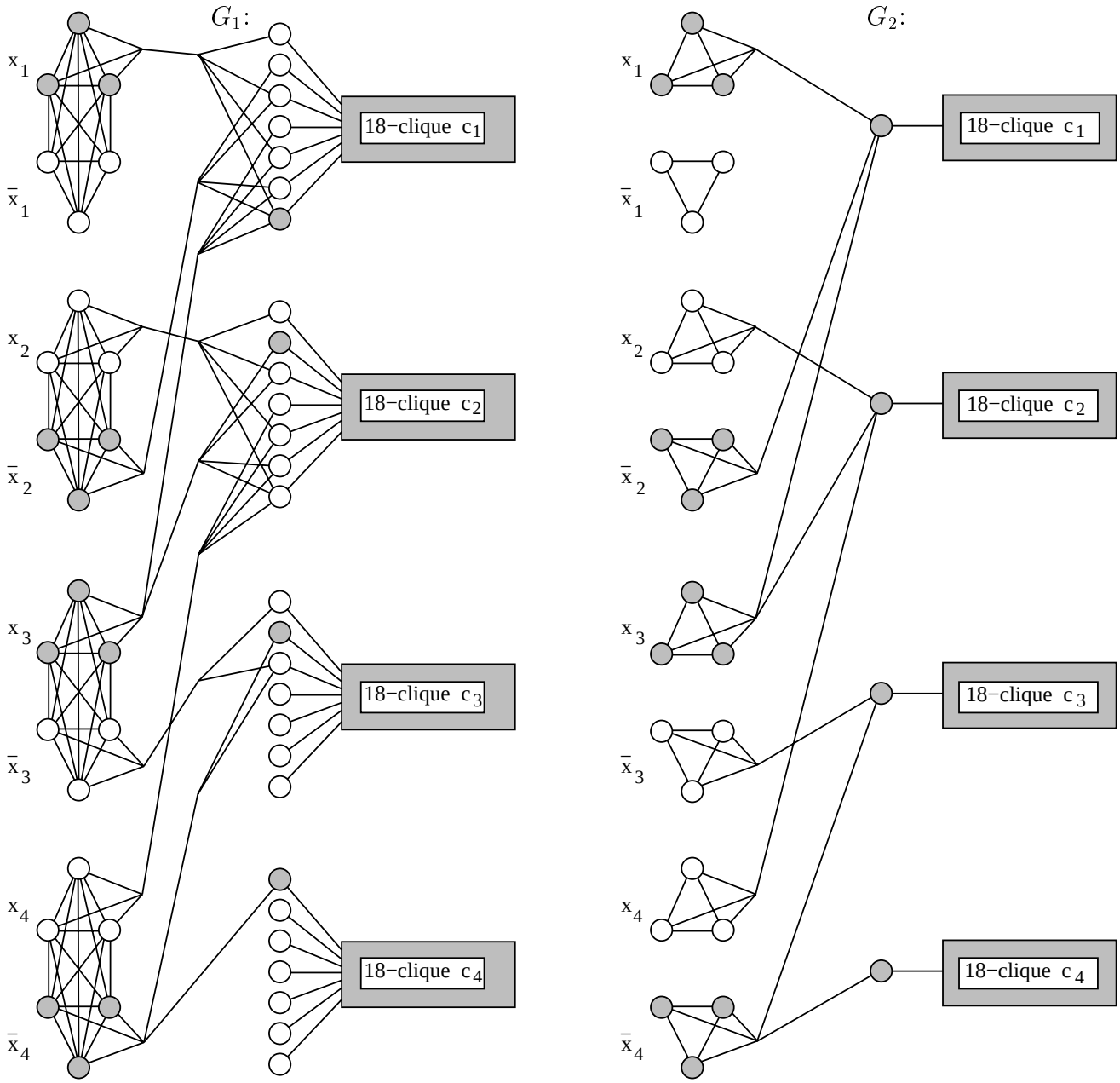

Figure 3: *The constructed instance of the* MAX CIS $-B$ *problem for the* MAX 3SAT -6 *input* $U = \{x_1, x_2, x_3, x_4\}, C = \{(x_1 \vee \overline{x}_2 \vee x_3), (x_2 \vee x_3 \vee x_4), (\overline{x}_3 \vee \overline{x}_4), (\overline{x}_4)\}$. *One of the possible maximum common subgraphs is formed by including the shaded nodes.*

literals just add the edges from clause nodes 1, 2 and 3. If c_i only has one literal just add three edges, between node 1 and the literal 3-clique. See Figure 3 for an example.

The idea is that a truth assignment shall be encoded in the subgraph problem by including the corresponding literal 3-cliques in the subgraphs. For example, if x_4 is true then the literal 3-clique x_4 is in the subgraphs, and if x_4 is false then the literal 3-clique $\overline{x}_4$ is in the subgraphs. The included literal 3-cliques of G_1 and G_2 are matched with each other. A clause node in graph G_2 is included in the subgraph iff it is satisfied by the truth assignment. If a clause node in G_2 is included then it is matched with one of the corresponding seven clause nodes in G_2, namely with the node which is connected with exactly those literals in the clause which are true in the truth assignment. All the clique nodes are included in the subgraphs and are matched with each other clause-wise.

A solution of the MAX 3SAT -6 problem with k satisfied clauses will be encoded as two subgraphs (of G_1 and G_2), each with k clause nodes, $3n$ literal nodes and $18m$ clique nodes. Since a literal can be contained in at most five clauses at the same time, a literal node in the graph G_1 has degree at most $5 \cdot 4 + 5 = 25$, a clause node has degree at most 4 and a clique node has degree 17 or 18. In G_2 a literal node has degree at most 7, a clause node at most 4 and a clique node at most 18. Thus $B \geq 25$.

We will now prove that the maximum solution of the MAX 3SAT -6 problem will be encoded as a maximum solution of the MAX CIS $-B$ problem, and that given a solution of the MAX CIS $-B$ problem we can in polynomial time find an at least as good solution which is a legal encoding of a MAX 3SAT -6 solution.

Suppose we have any solution of the MAX CIS $-B$ problem, that is an induced subgraph of G_1, an induced subgraph of G_2 and a matching between each node in the G_1 subgraph and the corresponding node in the isomorphic G_2 subgraph.

- First we would like to include all clique nodes in the subgraphs and match each 18-clique in the first graph with some 18-clique in the second. Observe that, besides the clique nodes, no node in the graph G_2 is in a clique of size five or more. This means that if five or more clique nodes in the same 18-clique are included in the subgraph of G_1, then they must be matched with clique nodes in the other subgraph. In the same way we see that besides the clique nodes, no node in the graph G_1 is in a clique of size seven or more, so if seven or more clique nodes in the same 18-clique are included in the subgraph of G_2, then they must be matched with clique nodes in the subgraph of G_1.

  In each 18-clique in G_1 which has $5 \leq p < 18$ nodes included in the subgraph, we add the rest of the 18-clique to the subgraph and remove every clause node which is connected to an added clique node. We have added $18 - p$ clique nodes and removed at most the same number of clause nodes.

  The matched 18-clique in G_2 must also have p clique nodes in the subgraph. Add the remaining $18 - p$ clique nodes and remove the nodes which are matched with the removed clause nodes in G_1. It is easy to see that we now can match the two 18-cliques with each other without problems.

  Perform the same operation for each 18-clique in G_1 which has at least five but not all nodes included in the subgraph. The resulting subgraphs are at least as large as the original subgraphs, and they are still isomorphic.

  Now every 18-clique in G_1 either has all nodes in the subgraph or at most four nodes in the subgraph. For each 18-clique in G_1 with $0 \leq p \leq 4$ nodes we do the following.

  We add $18 - p$ clique nodes to the subgraph of G_1 and remove every clause node which is connected to a clique node in the current 18-clique. We have added $18 - p$ nodes and removed at most 7. In G_2 we choose one 18-clique with $0 \leq q \leq 6$ nodes in the subgraph and add the remaining $18 - q$ clique nodes. We remove the p nodes which are matched with the old clique nodes in the first subgraph and the at most 7 nodes which are matched with the removed nodes in the first subgraph. Furthermore we have to remove the q nodes in G_1 which are matched with the q old clique nodes in G_2. If the clause node in G_2 (which is a neighbour to the current 18-clique in G_2) is included in the second subgraph we remove it and its matched node in G_1. We have now added $18 - p \geq 14$ nodes to the first subgraph and removed at most $7 + q + 1 \leq 14$. We have added $18 - q \geq 12$ nodes to the second subgraph and removed at most $7 + p + 1 \leq 12$. As before, since the 18-cliques are now separate connected components we can match them with each other without problems.

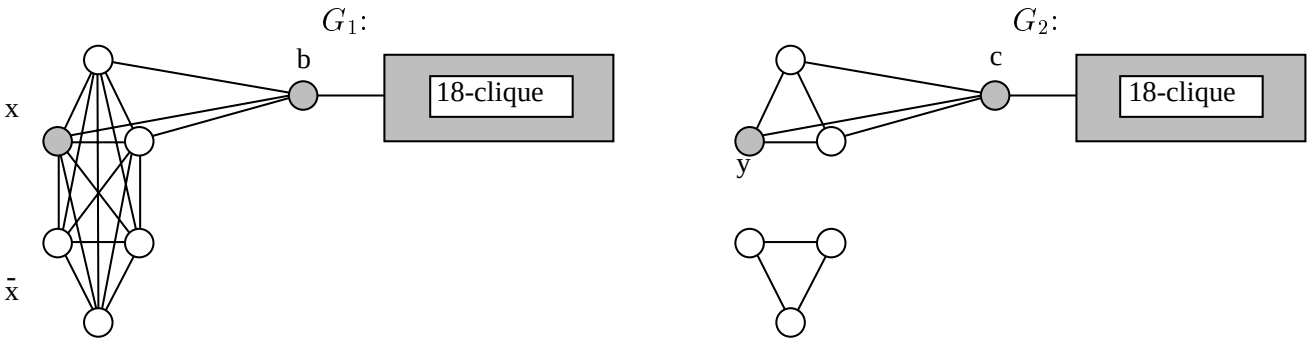

Figure 4: *The structure in case 1.*

The two subgraphs are still isomorphic, and thus a solution of the problem. They are at least as large as before, but now all clique nodes are included in the subgraphs and are matched with each other (but they are not necessarily matched in order yet).

- We observe that in each 7-group of clause nodes in G_1 at most one node is in the subgraph. The explanation is that every clause node is in contact with an 18-clique, which is completely in the subgraph, but in the subgraph of G_2 only one node can be in contact with each 18-clique (namely the corresponding clause node). Hence a structure with two or more nodes connected to an 18-clique cannot be isomorphic with any structure in G_2. Furthermore we can see that clause nodes in one of the subgraphs can only be matched with clause nodes in the other subgraph and, since all clique nodes are matched with clique nodes, literal nodes must be matched with literal nodes.

- We would like to change the subgraphs so that each literal 3-clique is either totally included in a subgraph or is not included at all. Furthermore at most one of the two literal 3-cliques in G_1 corresponding to the same variable may be included in the subgraph. Suppose that there is (at least) one node in the literal 3-clique x which is included in the subgraph of G_1. Let y be the literal node in the subgraph of G_2 with which this node is matched. We examine two cases.

Case 1. At least one of the clause nodes connected to x is included in the subgraph of G_1. Let b be one of these clause nodes and let c be the clause node in G_2 with which b is matched. See Figure 4. First we shall see that no node in the $\overline{x}$ literal 3-clique can be in the subgraph. This is because the nodes in the $\overline{x}$ clique are connected to the nodes in the x clique but not to b (since we have assumed that x and $\overline{x}$ cannot be in the same clause), but in G_2 there are no literal nodes which are connected to y but not to c. Since all three nodes in the x clique have the same connections to the environment in G_1 and all three nodes in the literal 3-clique containing y have the same environment in G_2 we still have isomorphic subgraphs if we include the whole x 3-clique in the subgraph of G_1 and the whole 3-clique containing y in the subgraph of G_2.

Case 2. None of the clause nodes connected to x are in the subgraph of G_1. If one or more nodes in $\overline{x}$ are in the subgraph of G_1 then none of the clause nodes which are connected to $\overline{x}$ can be in the subgraph, since we in that case would be in case 1 with the clause node as b and $\overline{x}$ as x. Thus we have a separate k-clique (with $1 \leq k \leq 6$) of literal nodes which by the above must be matched with a separate k-clique of literal nodes in G_2. In G_2 the largest possible clique of literal nodes is of size 3. Therefore the only possible cases are $1 \leq k \leq 3$. We remove those k nodes and instead include the whole

x 3-clique in the subgraph of G_1 and the whole 3-clique containing one of the matched nodes in the subgraph of G_2.

In both cases we get a subgraph of G_1 where each literal 3-clique is either totally included in a subgraph or is not included at all and where both of the literal 3-cliques corresponding to the same variable are never included in the subgraph of G_1 at the same time.

- We now forget about the subgraph of G_2 and concentrate on the subgraph of G_1. It contains all clique nodes, at most one of each 7-group of clause nodes and at most one of each pair of literal 3-cliques. First we will include literal nodes so that every pair of literal 3-cliques has exactly one of the 3-cliques in the subgraph. We will have to remove some of the clause nodes, but the subgraph should contain at least as many nodes as before. Then we reorder the clause nodes to form a legal encoding of a MAX 3SAT -6 solution.

  1. Suppose there are k variables which do not have any of their literal 3-cliques in the subgraph and that there are j clauses which contain these variables. We know that each variable can occur in at most six clauses, thus $j \leq 6k$. Using a simple algorithm (see for example [13]) we can give values to the k variables so that at least half of the j clauses are satisfied. We first remove all of the j clause nodes which are in the subgraph from the subgraph and then include one clause node for each clause which was satisfied by the algorithm and the literal 3-cliques corresponding to the k variable values. We will then have removed at most j nodes and included at least $3k + j/2 \geq j/2 + j/2 = j$ nodes.
  2. In order to create a subgraph of G_1 which is a legal encoding of a MAX 3SAT -6 solution we may have to substitute some clause nodes in the subgraph for other clause nodes in the same 7-groups. Every clause node in the resulting subgraph should have connections to exactly those literal 3-cliques which are included in the subgraph of G_1 and correspond to literals in the corresponding clause. It is easy to see that this operation is always possible.

- As the subgraph of G_2 choose nodes as shown in the description of the encoding above. This is possible since the subgraph of G_1 is a legal encoding of a MAX 3SAT -6 solution. The isomorphic matching is then trivial.

We have now shown that every solution of the MAX CIS $-B$ problem can be transformed to an at least as large solution which is a legal encoding of a MAX 3SAT -6 solution. Moreover this transformation can be done in polynomial time.

If the optimal number of satisfied clauses is r and we do not have more variables than clauses then the optimal number of nodes in a subgraph is $3n+18m+r \leq (3+18)m+r \leq 21 \cdot 2r+r = 43r$, since we can always satisfy at least half of the clauses. Thus the transformation f_1 of a MAX 3SAT -6 problem to a MAX CIS $-B$ problem, where $B \geq 25$, is an L-reduction with $\alpha = 43$ and $\beta = 1$. □

Theorem 7 *There is a reduction from* MAX CLIQUE *to* MAX CIS *which is an L-reduction without node amplification.*

PROOF The reduction is the same as the one Garey and Johnson used to prove that SUBGRAPH ISOMORPHISM is NP-complete. The MAX CLIQUE input is given as a graph G. Let the first of the MAX CIS graphs G_1 be this graph. Let G_2 be a $|V_1|$-clique, that is a complete graph with

the same number of nodes as G_1. The constructed graphs have the same number of nodes as the input graph.

Every induced subgraph in G_2 is a clique. Thus each common induced subgraph is a clique. The optimal solution of the MAX CLIQUE problem is a clique of size at most $|V|$, and this clique is also the largest clique in G_1 and is therefore the largest common induced subgraph.

In order to prove that this is an L-reduction we also have to show that given a solution of the MAX CIS problem we in polynomial time can find an at least as good solution of the MAX CLIQUE problem. But since every common subgraph is a clique we can directly use the subgraph as a solution to the MAX CLIQUE problem. The solutions have the same size. □

The MAX CLIQUE problem is hard to approximate. Recently Arora, Lund, Motwani, Sudan and Szegedy showed that for some constant $c > 0$ it is impossible to approximate MAX CLIQUE within n^c [1]. Theorem 7 shows that this result is valid for the unbounded MAX CIS problem as well.

When we gave an approximation algorithm for the MAX CIS $-B$ problem in Theorem 1 we constructed a maximal independent set to use as the common subgraph. Somehow this feels like cheating, because an independent set of nodes is usually not the type of common subgraph we want. Perhaps we would rather like to find a big common *connected* subgraph. Unfortunately the MAX CIS problem, where we demand that the common subgraph is connected, is *provably* hard to approximate. We will prove that this problem is NPO PB-complete under L-reductions, which means that every NPO problem with polynomially bounded optimal value can be L-reduced to it.

In general, for similar graph problems, the demand that the solution subgraph is connected seems to lead to harder problems [25].

Theorem 8 MAXIMUM COMMON INDUCED CONNECTED SUBGRAPH (MAX CICS) *is* NPO PB-*complete under L-reductions.*

PROOF NPO PB consists of all NPO problems which are polynomially bounded, that is all problems F satisfying $opt_F(I) \leq p(|I|)$ for all problem instances $I \in \mathcal{I}_F$ where p is a polynomial. It is obvious that MAX CICS satisfies this and therefore is in NPO PB.

We know that LONGEST INDUCED PATH in a graph G is NPO PB-complete under L-reductions [3] so we will L-reduce this problem to MAX CICS in order to show that the latter problem is NPO PB-complete.

Choose G as the first graph G_1 and choose a simple path with $|V|$ nodes as the second graph G_2. We observe that every induced connected subgraph in G_2 must be a simple path. The maximum induced connected subgraph is the longest induced path that can be found in G_1, that is the optimal solution of the LIP problem with input G. Since every non-optimal solution of size c immediately gives a solution of the LIP problem of size c the transformation is an L-reduction with $\alpha = \beta = 1$ and without node amplification. □

Finally we return to the edge subgraph problem and show that if the degrees of the graphs are bounded then the MAX CIS problem is at least as hard to approximate as the MAX CES problem.

Theorem 9 *There is an L-reduction from* MAX CES $-B$ *to* MAX CIS$-(2B+3)$.

PROOF Let f_3 be the following transformation from MAX CES $-B$ to MAX CIS $-(2B+3)$. An input instance $\{{G_1}^E, {G_2}^E\}$ of MAX CES $-B$ shall be transformed to an instance $\{{G_1}^I, {G_2}^I\}$

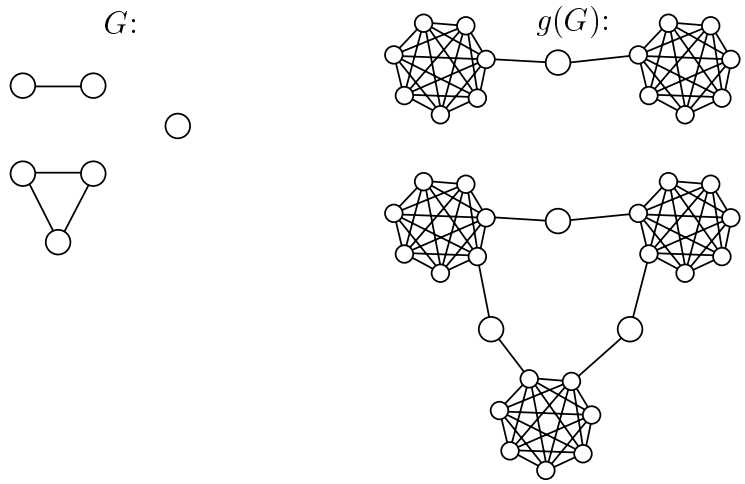

Figure 5: *An example of the transformation g in Theorem 9.*

of MAX CIS $-(2B+3)$. Let ${G_1}^I = g({G_1}^E)$ and ${G_2}^I = g({G_2}^E)$ where g transforms each node with degree greater than zero to a $(2B+3)$-clique and each edge to two edges connected with an *edge node*. The two edges are connected to one node in each of the two $(2B+3)$-cliques corresponding to the end points of the edge in the original graph. This shall be done so that every clique node is connected to at most one edge node. The constructed graphs have degree $2B+3$. See Figure 5.

Solutions $\{{E_1}', {E_2}'\}$ to the MAX CES $-B$ problem is encoded as solutions $\{{V_1}', {V_2}'\}$ to the MAX CIS problem where an edge node is in V_i' iff the corresponding edge is in E_i' and where $\min(|V_1'|, |V_2'|)$ $(2B+3)$-cliques, among them all cliques corresponding to nodes adjacent to edges in E_1' and E_2', are included in each subgraph.

In the same way as in the proof of Theorem 6 we will show that given a solution of the MAX CIS problem, which is d smaller than the optimal solution, we can in polynomial time find a solution of the MAX CES $-B$ problem which is at most d smaller than the optimal solution.

Given solutions V_1' and V_2' we first modify them so that all clique nodes are included. Observe that cliques of size three or more only can be found in the $(2B+3)$-cliques in ${G_1}^I$ and ${G_2}^I$ and thus that a $(2B+3)$-clique in ${G_1}^I$ with k nodes $(k \geq 3)$ in V_1' only can be matched with a $(2B+3)$-clique in ${G_2}^I$ with exactly k nodes in V_2' (and vice versa). For each $(2B+3)$-clique in ${G_1}^I$ with k nodes $(3 \leq k < 2B+3)$ in V_1' we can add the remaining $2B+3-k$ nodes if we remove all edge nodes in V_1' which are connected to added nodes and perform the same operations on the matched clique in the other graph.

Now every clique either has all nodes in the subgraph or at most two nodes in the subgraph. For each clique in ${G_1}^{I'}$ with $0 \leq p \leq 2$ nodes we do the following (until there are no nonfull cliques left in one of the subgraphs).

We add $2B+3-p$ clique nodes to the subgraph of ${G_1}^I$ and remove every edge node which is connected to a clique node in the current clique (at most B nodes). In ${G_2}^I$ we choose one clique with $0 \leq q \leq p$ nodes in the subgraph. If one of the p nodes in the first subgraph is matched with a clique node in the second subgraph (which is always the case if $p = 2$ because two edge nodes never are connected) we choose this clique. We add the remaining $2B+3-q$ clique nodes and remove every edge node which is connected to a clique node in the current clique (at most B nodes).

If one of the q nodes in the second subgraph is matched with an edge node in the first subgraph we have to remove this edge node from the first subgraph. If one of the p nodes in the first subgraph is matched with an edge node in the second subgraph we have to remove this edge

$$\begin{array}{ccccc} \textsc{Max 3Sat} -B & < & \textsc{Max Clique} & < & \textsc{LIP} \\ \wedge & & \wedge\vee & & \wedge\vee \\ \textsc{Max CIS} -B & < & \textsc{Max CIS} & < & \textsc{Max CICS} \\ \vee & & \vee & & \\ \textsc{Max CES} -B & < & \textsc{Max CES} & & \end{array}$$

Figure 6: *Summary of how the common subgraph problems are related. Here $P_1 < P_2$ means that there is an L-reduction from problem P_1 to problem P_2.*

node from the second subgraph.

Furthermore we have to remove the at most B nodes in the first subgraph which are matched with edge nodes which are connected with nodes in the current clique in the second subgraph. And symmetrically we have to remove the at most B nodes in the second subgraph which are matched with edge nodes which are connected with nodes in the current clique in the first subgraph.

We have now added $2B+3-p \geq 2B+1$ nodes to the first subgraph and $2B+3-q \geq 2B+1$ nodes from the second subgraph and removed at most $B+1+B = 2B+1$ nodes from the first subgraph and $B+1+B = 2B+1$ nodes from the second. If we match the cliques with each other the two subgraphs are still isomorphic.

Now every clique node in at least one of the graphs, say in ${G_1}^I$, is included in the corresponding subgraph and are matched with clique nodes in the other subgraph. Therefore the edge nodes in the second subgraph must be matched with edge nodes in the first subgraph. Every edge node in the first subgraph must be matched with an edge node in the second subgraph, because it is adjacent to a $(2B+3)$-clique in the first subgraph, and no clique node in the second subgraph is adjacent to a $(2B+3)$-clique. Thus we have subgraphs which are an encoding of a Max CES $-B$ solution, where an edge node is in the Max CIS subgraph if and only if the corresponding edge is in the Max CES $-B$ subgraph.

If the subgraphs in an optimal solution of the Max CES $-B$ problem contain k edges then the number of nodes in the subgraphs in the optimal solution to the Max CIS problem is

$$k + (2B+3)\cdot 2 \cdot \min\left(\left|{E_1}^E\right|, \left|{E_2}^E\right|\right) \leq k + (2B+3)\cdot 2\cdot (B+1)k = (4B^2+10B+7)k$$

Thus the reduction f_3 is an L-reduction with $\alpha = 4B^2+10B+7$ and $\beta = 1$. □

5 Discussion

We have studied the approximability of some maximum common subgraph problems. Figure 6 illustrates the situation.

Yannakakis observed in [26] that problems on edges tend to be easier to solve than their node-analogues. We have seen that this is valid for the approximability of the maximum common subgraph problem as well.

Acknowledgements

I would like to thank my advisor Johan Håstad for valuable support.

References

[1] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and hardness of approximation problems. In *Proc. 33rd Ann. IEEE Symp. on Foundations of Comput. Sci.*, pages 14–23. IEEE Computer Society, 1992.

[2] H. Barrow and R. Burstall. Subgraph isomorphism, matching relational structures and maximal cliques. *Inform. Process. Lett.*, 4:83–84, 1976.

[3] P. Berman and G. Schnitger. On the complexity of approximating the independent set problem. In *Proc. 6th Ann. Symp. on Theoretical Aspects of Comput. Sci.*, pages 256–268. Springer-Verlag, 1989. Lecture Notes in Comput. Sci. 349.

[4] M. Bern and P. Plassmann. The Steiner problem with edge lengths 1 and 2. *Inform. Process. Lett.*, 32:171–176, 1989.

[5] A. Blum, T. Jiang, M. Li, J. Tromp, and M. Yannakakis. Linear approximation of shortest superstrings. In *Proc. Twenty third Ann. ACM Symp. on Theory of Comp.*, pages 328–336. ACM, 1991.

[6] R. Boppana and M. M. Halldórsson. Approximating maximum independent sets by excluding subgraphs. *Bit*, 32(2):180–196, 1992.

[7] D. Bruschi, D. Joseph, and P. Young. A structural overview of NP optimization problems. In *Proc. Optimal Algorithms*, pages 205–231. Springer-Verlag, 1989. Lecture Notes in Comput. Sci. 401.

[8] N. Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical report, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, 1976.

[9] P. Crescenzi and A. Panconesi. Completeness in approximation classes. *Inform. and Comput.*, 93(2):241–262, 1991.

[10] R. Fagin. Generalized first-order spectra, and polynomial-time recognizable sets. In R. Karp, editor, *Complexity and Computations*. AMS, 1974.

[11] M. R. Garey and D. S. Johnson. *Computers and Intractability: a guide to the theory of NP-completeness*. W. H. Freeman and Company, San Francisco, 1979.

[12] O. H. Ibarra and C. E. Kim. Fast approximation for the knapsack and sum of subset problems. *J. ACM*, 22(4):463–468, 1975.

[13] D. S. Johnson. Approximation algorithms for combinatorial problems. *J. Comput. System Sci.*, 9:256–278, 1974.

[14] V. Kann. Maximum bounded 3-dimensional matching is MAX SNP-complete. *Inform. Process. Lett.*, 37:27–35, 1991.

[15] V. Kann. Which definition of MAX SNP is the best? Manuscript, 1991.

[16] V. Kann. *On the Approximability of NP-complete Optimization Problems*. PhD thesis, Department of Numerical Analysis and Computer Science, Royal Institute of Technology, Stockholm, 1992. NADA report TRITA-NA-9206.

[17] V. Kann. Maximum bounded H-matching is MAX SNP-complete. *Inform. Process. Lett.*, 49:309–318, 1994.

[18] P. G. Kolaitis and M. N. Thakur. Approximation properties of NP minimization classes. In *Proc. 6th Ann. Conf. on Structures in Computer Science*, pages 353–366, 1991.

[19] P. G. Kolaitis and M. N. Thakur. Logical definability of NP optimization problems. Technical Report UCSC-CRL-93-10, Board of Studies in Computer and Information Sciences, University of California at Santa Cruz, 1993.

[20] M. W. Krentel. The complexity of optimization problems. *J. Comput. System Sci.*, 36:490–509, 1988.

[21] L. Lovász and M. D. Plummer. *Matching Theory*, volume 29 of *Annals of Discrete Mathematics*, page 114. Elsevier science publishing company, Amsterdam, 1986.

[22] A. Panconesi and D. Ranjan. Quantifiers and approximation. In *Proc. Twenty second Ann. ACM Symp. on Theory of Comp.*, pages 446–456. ACM, 1990.

[23] C. H. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *J. Comput. System Sci.*, 43:425–440, 1991.

[24] C. H. Papadimitriou and M. Yannakakis. The traveling salesman problem with distances one and two. *Math. Oper. Res.*, 1992. To appear.

[25] M. Yannakakis. The effect of a connectivity requirement on the complexity of maximum subgraph problems. *J. ACM*, 26:618–630, 1979.

[26] M. Yannakakis. Edge deletion problems. *SIAM J. Computing*, 10:297–309, 1981.