Acta Informatica 40, 637–655 (2004)
Digital Object Identifier (DOI) 10.1007/s00236-004-0144-0

Acta Informatica

On best transitive approximations to simple graphs

Steven Delvaux, Leon Horsten

University of Leuven, Department of Computer Science / Department of Philosophy, 3000 Leuven, Belgium

Received: 16 April 2003 / 9 May 2004
Published online: 8 July 2004 –

Abstract. In this paper, we investigate both combinatorial and complexity aspects of the problem of finding best transitive approximations to simple graphs. These problems are addressed in an interlocked way. We provide new and simple proofs of known results and in addition prove some new theorems.

1 Introduction

Given any finite graph, which transitive graphs approximate it most closely and how fast can we find them?

The answer to this question depends on the concept of "closest approximation" involved. In [8,9] a *qualitative* concept of best approximation is formulated. Roughly, a qualitatively best transitive approximation of a graph is a transitive graph which cannot be "improved" without also going against the original graph. A *quantitative* concept of best approximation goes back at least to [10]. A quantitatively best transitive approximation is a transitive graph that makes the minimal number of mistakes against the original graph. In other words, the sum of the edges that are removed from and are added to the original graph is minimal.

On both the qualitative and the quantitative conception, there usually exists more than one best transitive approximation. In [7], partial results are obtained for the number of quantitatively best transitive approximations. And in [5] it is shown that finding a best approximation in the quantitative sense is an NP-complete problem.

In this paper, we investigate both combinatorial and complexity aspects of the problem of finding best approximations, and we investigate these

aspects in an interlocked way. We give a new and simpler proof of the important NP-completeness result of [5], but also prove some additional results. We also extend the combinatorial results of [7].

Unless explicitly mentioned otherwise, when in the sequel we say "for any G,...", we mean "for any reflexive symmetrical graph G,...". For the rest, our notation is fairly standard. The notation uv stands for the edge between vertex u and vertex v; by $\overline{G}$ we denote the graph with same set vertex set as G, but with the complementary set of edges; by $\#A$ we denote the cardinal number of a set A.

The problem that is addressed in this paper has applications in all situations in which an interconnected structure must be partitioned in a way which reflects the actual interconnections as well as possible.[1] Concrete applications then follow from the actual structures under investigation.

2 The difference between graphs and transitive graphs

The definition of best equivalence-approximation of a graph G can be expressed as follows.

Definition 1 *Let G, H be graphs with the same set of vertices. We define the* difference graph $D(G,H)$ *of G and H as the graph consisting of those edges uv such that either $uv \in G$ and $uv \notin H$, or $uv \in H$ and $uv \notin G$.*

Definition 2 *For any G, the collection* $\mathbf{BA}(G)$ *(the collection of* best transitive approximations *to G) consists of all transitive graphs H such that $\#D(G,H)$ is minimal.*

In words, the idea can be expressed as follows. We consider G as a fixed graph and the transitive graph H as an approximation of G. We see then that H is obtained from G by removing and adding edges to/from G (*cutting* and *pasting*). Every action of adding and of removing an edge is counted as a mistake by H. A best transitive approximation to G is an approximation which makes a minimal number of mistakes.[2] It is a quantitative definition because the number of mistakes is *counted*.

Definition 3 *For any G and H, we define the graph $D^-(G,H)$ to be the graph consisting of those edges which are in G but not in H. Similarly we define $D^+(G,H)$ to be the graph consisting of the edges which are not in G but are in H.*

Definition 4 *For any G, the collection* $\mathbf{BA}^-(G)$ *(best transitive approximations from below) consists of all transitive graphs H such that $\#D^+(G,H) = 0$ and $\#D^-(G,H)$ is minimal.*

[1] See [10, p. 840], [2].

[2] This definition goes back at least to [10, p. 840].

In other words, we say that the best transitive approximations from below are best equivalence-approximations that only remove edges from G (but do not add edges). Every action of removing an edge is counted as a mistake.

Of course we can also define the set $\mathbf{BA}^+(G)$ consisting of the best transitive approximations *from above*. But this is a rather trivial notion: for any graph G, the set $\mathbf{BA}^+(G)$ will only contain one element, and this is the graph which is the transitive closure of G. In the sequel, this notion will be disregarded.

We will now investigate some combinatorial properties of the quantitative cut-and-paste approach. Some of these results will also be used later in our complexity calculations.

A simple upper bound for the number of modifications that need to be made in order to make a graph transitive is $\frac{1}{2}\binom{n}{2}$.[3] The reason is the following. Either the graph has $\leq \frac{1}{2}\binom{n}{2}$ edges and we remove them all, or it has $> \frac{1}{2}\binom{n}{2}$ edges and we complete the graph using $< \frac{1}{2}\binom{n}{2}$ edges. We will now give the *exact* upper bound, and investigate when this boundary is reached.

First we introduce a systematic series of definitions that includes the definition of transitive graphs.

Definition 5 *A graph is transitive if and only if there is no triple of vertices which are connected by means of exactly 2 edges, i.e. if and only if every triple of vertices is connected by 0,1, or 3 edges. Such graphs can be called 013-graphs. Similarly, we define the class of 02-graphs, 012-graphs, and so on.*

We will now consider a subclass of the transitive graphs (or 013-graphs). Namely, we focus on the 13-graphs.

Proposition 1 *Let n vertices be given, on which a graph is defined in the following way: (a) every vertex is given a sign $+$ or $-$; (b) an edge is drawn between two vertices if and only if they have the same sign. Then (1) the result is always a 13-graph, and (2) every 13-graph can be so generated.*

Proof. (1) This is immediate: to an arbitrary triple of vertices we have assigned either one sign only (this yields 3 edges) or two signs (this yields one edge).
(2) Let a 13-graph G be given. We must find an assignment of signs that yields G. A first vertex v_0 is assigned $+$, and any other vertex v is assigned $+$ if the edge v_0v belongs to G, $-$ otherwise. Let G' be the 13-graph thus generated. We claim that $G = G'$. This can be seen by considering an arbitrary triple of vertices v_0, v, w. We know that G and G' agree on 2 of the 3 possible edges between v_0, v, w, and since their number of edges is equal modulo 2, they must also coincide on the third possible edge vw.

[3] See [6].

Definition 6 *If we have a 13-graph in which p nodes have been assigned $+$, and q nodes have been assigned $-$, then we call p and q the* structure numbers *of the 13-graph.*

From Proposition 1, we see that a 13-graph is just a transitive graph where the vertices are distributed over 2 cliques. The sizes of these two cliques are the structure numbers p and q (one of which may be zero). By passing to the complementary graph $\overline{G}$, we see that a 02-graph is a bipartite graph with p vertices on the left and q vertices on the right, such that every of the p vertices is connected with every of the q vertices. Such a graph is usually called a complete bipartite graph and denoted as $K_{p,q}$. In this case, we still call p and q the *structure numbers* of G.

We will define now a function φ which will be important in the sequel.

Definition 7 *Let n be a positive integer, then we define $\varphi(n)$ to be*

- $\binom{\frac{n}{2}}{2} + \binom{\frac{n}{2}}{2} = \frac{n}{2}\left(\frac{n}{2} - 1\right)$ *for n even, and*
- $\binom{\frac{n-1}{2}}{2} + \binom{\frac{n+1}{2}}{2} = \frac{n+1}{2} \cdot \frac{n-1}{2}$ *for n odd.*

A small calculation shows that $\varphi(n)$ can also be written as

$$\varphi(n) = \frac{1}{2}\left(\binom{n}{2} - \left\lfloor \frac{n}{2} \right\rfloor\right)$$

for every n. It is clear that, as $n \to \infty$, the function $\varphi(n)$ becomes asymptotically equivalent to $\frac{1}{2}\binom{n}{2}$, i.e. to half of the possible edges on n points.

Now in order to gain information about the best transitive approximations of a graph G, we will first restrict ourselves to the best "13-graph"-approximations of G. We first calculate the minimal number of edges that need to be modified in order to transform a graph G of order n into a 13-graph.

Lemma 2 *The minimal number of edges that needs to be modified in order to transform a graph G of order n into a 13-graph is*

(a) at most equal to $\varphi(n)$, and (b) exactly equal to $\varphi(n)$ if and only if G is a 02-graph.

Proof. First we will prove (a).
(a1) Suppose first that n is even. Let H be an arbitrary 13-graph and consider the *difference graph* $D(G, H)$. If u is a vertex which has at least $\geq \frac{n}{2}$ neighbors in $D(G, H)$, we can replace H by the graph H' which is obtained by reversing the sign of u. We can iterate this operation until every point is connected to $\leq \frac{n}{2} - 1$ edges in $D(G, H)$. The number of mistakes of the resulting 13-graph is then at most $\frac{n}{2}\left(\frac{n}{2} - 1\right) = \varphi(n)$.
(a2) Suppose that n is odd. We can then lower the degree of the vertices in $D(G, H)$ to $\frac{n-1}{2}$, by using the same method. This does not yet suffice.

But suppose that there are two vertices which have both degree $\frac{n-1}{2}$ in $D(G,H)$ and which are not connected. Then if we reverse the sign of these two vertices, the total number of edges in $D(G,H)$ will decrease. (This is not hard to see.) We see from this that all the vertices of degree $\frac{n-1}{2}$ must be connected with each other. Hence there are at most $\frac{n+1}{2}$ vertices which have degree $\frac{n-1}{2}$, and other vertices have lower degrees. The number of edges in $D(G,H)$ is then at most

$$\frac{1}{2}\left(\frac{n+1}{2}\cdot\frac{n-1}{2}+\frac{n-1}{2}\cdot\frac{n-3}{2}\right)=\frac{n-1}{2}\cdot\frac{n-1}{2}=\varphi(n)$$

Now we will prove (b).
(b1) First we do the $\Rightarrow$-direction. (b1.1) Suppose that n is odd. From the proof of (a), we see that the maximal number of mistakes can only be reached when $D(G,H)$ is the union of 2 disjoint cliques, of size $\frac{n+1}{2}$, $\frac{n-1}{2}$, respectively. Such a $D(G,H)$ is a 13-graph. The original graph G is then the "superposition" of two 13-graphs, hence a 02-graph.
(b1.2) Suppose that n is even. In (a) we have lowered the degree to $\frac{n}{2}-1$. Now we need the additional fact that if we have 3 vertices which are pairwise unconnected in $D(G,H)$ and all have degree $\frac{n}{2}-1$ in it, then changing the sign of all these vertices will decrease the number of edges in $D(G,H)$. This implies that the maximum number $\varphi(n)$ of mistakes is only reached when $D(G,H)$ consists of 2 cliques, both containing $\frac{n}{2}$ vertices, which leads to the same situation as in (b1). (b2) Now we do the $\Leftarrow$-direction.
If G is a 02-graph, then $D(G,H)$ must be a superposition of a 02-graph and a 13-graph and hence a 13-graph. The number of edges of such a graph is at least $\binom{\frac{n}{2}}{2}+\binom{\frac{n}{2}}{2}=\varphi(n)$ for n even and $\binom{\frac{n+1}{2}}{2}+\binom{\frac{n-1}{2}}{2}=\varphi(n)$ for n odd.

This proposition generalizes a result of [7], where it is shown that in order to transform a graph of order n into a transitive graph with 2 components, at most $\varphi(n)$ mistakes have to be made, and this maximum value is reached on a graph $K_{p,q}$.

We will now investigate the best transitive approximations of a graph G which is a 012-graph (equivalently, a graph with maximum clique number at most equal to 2). First we prove the following lemma for the complementary graph $\overline{G}$, which is a 123-graph.

Lemma 3 *Let G be a 123-graph of order n. Then: 1. G contains at least $\varphi(n)$ edges and 2. This minimum value is reached when G consists of 2 disjoint but equally large cliques, i.e., when G is a 13-graph with structure numbers $\left\{\frac{n}{2},\frac{n}{2}\right\}$ (n is even) or $\left\{\frac{n-1}{2},\frac{n+1}{2}\right\}$ (n is odd).*

Proof. Let u be a vertex with a minimal number $d-1$ of neighbors. There must then be $n-d$ vertices which are not connected with u (excluding u

itself). Since every triple of vertices in G must contain at least one edge (by definition of 123-graph), these $n - d$ vertices must form a clique. On the other hand, by the minimality of d, we see that every vertex must have degree at least $d - 1$. So the minimal number of edges is reached when:

a. there are no additional edges to the clique of $n - d$ vertices, and

b. each of the d vertices has degree $d - 1$.

It is clear that there is exactly one way in which these conditions can be satisfied, namely if G consists of 2 disjoint cliques. Therefore G must be a 13-graph. As in the proof of the previous lemma, we see that such a graph has at least $\varphi(n)$ edges, and that this minimum is reached when the structure numbers of the graph differ by at most 1.

Incidentally, we note that this lemma can be generalized somewhat:

Lemma 4 *Let G be a graph of order n, and μ an integer such that there is at least one edge in every collection of $\mu + 1$ vertices (or, equivalently, that its complementary graph has maximum clique number at most μ). Then the minimal number of edges of G approximately equals $\frac{1}{\mu}\binom{n}{2}$, and this minimum is reached when G consists of μ disjoint, almost equally large cliques.*

Proof. Again let u be a vertex with a minimal number $d - 1$ of neighbors. By assumption between every μ-tuple of vertices not connected with u there has to be at least one edge. We then reason inductively: the number of edges on these $n - d$ vertices can only be minimal if they consist of $\mu - 1$ disjoint cliques. As in the previous lemma, the other d vertices must then also form a clique in order to be minimal. So G consists of μ disjoint cliques, and it is easy to see that the minimal number of edges of such a graph can only be reached when the difference in size of each pair of cliques amounts to at most 1.

Now we will prove that, for a 012-graph, we can always find a best cut-and-paste-approximation that does not paste.

Theorem 5 *Let G be a graph with maximum clique number 2. Then there is an element H of $\mathbf{BA}(G)$ which belongs to $\mathbf{BA}^{-}(G)$.*

Proof. Suppose k vertices, with $k \geq 3$, which a best transitive approximation transforms into a clique. Let G' be the graph G restricted to these k vertices. By renaming, we take G to be the graph G' (which is also a 012-graph), and it will then suffice to show that there exists a best approximation from below of G which does not make more mistakes than an approximation which transforms whole of G into a clique.

(a.) First we assume that the number of disjoint edges δ of G is maximal, i.e. $\delta = \left\lfloor \frac{n}{2} \right\rfloor$. By the previous lemma, G can have at most $\binom{n}{2} - \varphi(n)$ edges. Therefore a best approximation from below of G will make at most

$$\binom{n}{2} - \varphi(n) - \delta = 2\varphi(n) - \varphi(n) = \varphi(n)$$

mistakes against G. On the other hand, the approximation which transforms G into a clique will make at least $\varphi(n)$ mistakes. This, then, is never more profitable than the best approximations from below. So there indeed exists a best cut-and-paste approximation which nowhere pastes.

(b.) Now let us assume that the maximal number δ of disjoint edges of G is smaller than $\left\lfloor \frac{n}{2} \right\rfloor$. We choose a partition of the vertices of G into two equally large sets G_1 and G_2, both containing $\frac{n}{2}$ vertices if n is even and containing $\frac{n-1}{2}$ and $\frac{n+1}{2}$ vertices respectively if n is odd. We do this in such a way that a *maximum matching* is reached between G_1 and G_2. Let $A_1 \subseteq G_1$ and $A_2 \subseteq G_2$ be the sets of end vertices of this matching, both containing δ vertices. We will then remove all edges u_1v_1 from G_1 in the following manner:

First case: both end-vertices of u_1v_1 belong to A_1. Remove this edge, and also remove the "opposite" edge u_2v_2 if there is one. In their place we add the "crossing" edges u_1v_2 and u_2v_1, which were not yet present since G has clique number 2. In this way the total number of edges of G has not diminished.

Second case: one of the end-vertices v_1 does not belong to A_1 but the other one u_1 does. Remove the edge u_1u_2 and replace it by the "crossing" edge u_2v_1. Again the total number of edges cannot be diminished.

Third case: neither of the end-vertices belong to A_1. This situation does not occur, since it would contradict the maximality of δ.

In the same way, we can remove the remaining edges between vertices of G_2 (we have to consider only the second case). In all these operations the total number of edges has not diminished. To conclude, we add all edges running from one of the c_1 vertices of G_1 which do not belong to A_1, to one of the c_2 vertices of G_2 which do not belong to A_2. This gives us c_1c_2 extra edges.

Now by construction, the resulting graph is bipartite, therefore with maximum clique number 2, and it has the maximal number of disjoint edges $\delta = \left\lfloor \frac{n}{2} \right\rfloor$. So the inequalities in the proof of part (a) of this proof are valid for this resulting graph. Also since $0 \neq c_1c_2 \geq \max\{c_1, c_2\}$, we have added more edges to G than we have increased the number δ, and so these inequalities are also valid for G itself.

Corollary 6 *For every* 02*-graph* G *with structure numbers* $\{p, q\}$ *(with* $p \leq q$*), the number of mistakes of every element of* $\mathbf{BA}(G)$ *is exactly* $p(q-1)$.

Proof. By the previous theorem we know that there exists a best cut-and-paste approximation of G which does not paste. Since our 02-graph contains pq edges and its maximal number of disjoint edges δ equals its smallest structure number p, the total number of mistakes made by a best cut-approximation is exactly $pq - p = p(q-1)$.

Corollary 7 *For every* 012*-graph* G*, there exists a polynomial time algorithm for finding an element of* $\mathbf{BA}(G)$.

Proof. There exists a polynomial time algorithm which finds, given a graph G, a maximum matching of G. See for instance [1], which describes an algorithm to find a maximum matching, with running time equal to $\mathcal{O}(n^3)$. So in particular, given a graph G with maximum clique number 2, there exists a polynomial time algorithm for finding an element of $\mathbf{BA}^{-}(G)$ and hence, by Theorem 5, for finding an element of $\mathbf{BA}(G)$.

We can also obtain a characterization of the *least transitive graphs*:

Corollary 8 *For every* n *there is a graph of order* n *which makes exactly* $\varphi(n)$ *mistakes. For* n *even these are the graphs* $K_{\frac{n}{2},\frac{n}{2}}$ *and* $K_{\frac{n}{2}-1,\frac{n}{2}+1}$ *and for* n *odd this is the graph* $K_{\frac{n-1}{2},\frac{n+1}{2}}$.

Proof. By Lemma 2, we know that the value $\varphi(n)$ can only be reached in a 02-graph. If we evaluate the expression $p(q-1)$ from the previous lemma for $(p, q) = \left(\frac{n}{2}, \frac{n}{2}\right)$ and $\left(\frac{n}{2}-1, \frac{n}{2}+1\right)$ for n even, we obtain

$$\frac{n}{2}\left(\frac{n}{2}-1\right) = \left(\frac{n}{2}-1\right)\frac{n}{2} = \varphi(n).$$

If we evaluate $p(q-1)$ for $(p, q) = \left(\frac{n-1}{2}, \frac{n+1}{2}\right)$ for n odd, we obtain also

$$\frac{n-1}{2} \cdot \frac{n-1}{2} = \varphi(n).$$

Other structure numbers yield smaller values.

3 Qualitative approximations

In this section, we will work with a qualitative notion of best approximation:

Definition 8 *H is a qualitatively best transitive approximation of G ($H \in \mathbf{BA}_{\mathbf{QL}}(G)$) if for every equivalence relation H', $D(G, H') \not\subseteq D(G, H)$. H is a qualitatively best transitive approximation from below of G ($H \in \mathbf{BA}^{-}_{\mathbf{QL}}(G)$) if $H \subseteq G$ and $H \in \mathbf{BA}_{\mathbf{QL}}(G)$.*

Thus H is a qualitatively best transitive approximation of G if and only if progress can only be made (with respect to H) by also at some places going against the original graph G, and H is a qualitatively best transitive approximation *from below* of G if additionally $H \subseteq G$. [4] In fact, it is easy to see that $H \in \mathbf{BA}^{-}_{\mathbf{QL}}(G)$ is equivalent with $H \subseteq G$ and for every equivalence relation H', $D^{-}(G, H') \nsubseteq D^{-}(G, H)$. Note that these notions are qualitative because we do not *count* mistakes.

Proposition 9 *There exists a polynomial-time algorithm for generating, for any given G, elements of $\mathbf{BA}^{-}_{\mathbf{QL}}(G)$.*

Proof. It is not hard to verify that the algorithm below does the job.

Algorithm 1 *1. Wellorder the domain of G.*
2. Build the equivalence classes of G^{-} in stages as follows:
- *(a) Start with the first element u_1 in the wellordering. Assign to it an equivalence class H_1.*
- *(b) Move on to the next element (u). If it can be added to one of the already partially constructed equivalence classes H_i in such a way that all elements of $H_i \cup \{u\}$ are G-related, do so. Otherwise, start a new equivalence class containing u.*
- *(c) Repeat step b. until the domain of G is exhausted.*

In fact, it is not hard to see that *all* elements of $\mathbf{BA}^{-}_{\mathbf{QL}}(G)$ are generated by this algorithm.

Definition 9 *An element of $\mathbf{BA}^{-}_{\mathbf{QL}}(G)$ is said to meet the* minimal cell requirement [5] *if there exists no best transitive approximation of G with fewer cliques.*

One can easily check that for elements of $\mathbf{BA}^{-}_{\mathbf{QL}}(G)$, satisfying the minimal cell requirement does *not* guarantee belonging to $\mathbf{BA}^{-}(G)$. We have the following result.

Proposition 10 *The problem of finding, for any given G, an element of $\mathbf{BA}^{-}_{\mathbf{QL}}(G)$ that meets the minimal cell requirement is NP-complete.*[6]

Proof. Let G be given. Consider the complement $\overline{G}$ of G. A (minimal) coloring of $\overline{G}$ corresponds to an equivalence relation H on the set of vertices of G such that (i) $H \subseteq G$ and (ii) H consists of a minimal number of cliques.

[4] With a slightly other, but equivalent definition, this set $\mathbf{BA}^{-}_{\mathbf{QL}}(G)$ has also been considered by Timothy Williamson in [8,9].

[5] See also Williamson [9, p. 72–73]

[6] Hannes Leitgeb pointed this out to us in private communication.

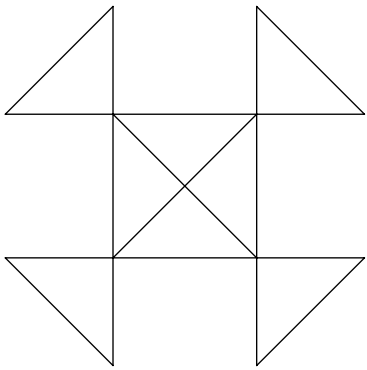

Fig. 1. Graph G of which the largest clique is split by best approximations

(i) is true because if two nodes are given the same color, then they cannot be acquainted according to $\overline{G}$, which means that they must be acquainted to G. (ii) also holds: the chromatic number of $\overline{G}$ is identical to the number of partition classes in optimal partitions of G. But generating optimal colorings is NP-complete.[7]

4 Complexity questions

In the preceding section, we investigated some questions concerning the complexity of best qualitative approximations. Now we return to the usual, quantitatively best approximations of Sect. 2, and these problems will be more difficult. We first look at approximations from below. Our strategy will be to reduce the *clique problem* to the problem of finding a quantitatively best transitive relation from below.[8] This is not completely straightforward, for there are graphs G such that every quantitatively best approximation from below breaks the largest cliques in G. For instance, consider the graph G in Fig. 1.

The transitive approximation H_1 which consists of the four corner triangles has $\#D(G, H_1) = 6$, whereas the transitive approximation H_2 which consists of the maximum clique, i.e. the middle square, plus 4 isolated edges has $\#D(G, H_2) = 8$. Therefore, H_1 is the (unique) best approximation from below, and it breaks the maximum clique.

However, generalizing from Fig. 1, we see that this phenomenon is bounded:

Proposition 11 *Let μ_G be the maximum clique number of a given graph G. Then the maximum clique number of every best transitive approximation from below to G is at least $\frac{\mu_G+1}{2}$.*

[7] See [4, p. 154].

[8] The clique problem is known to be NP-complete. See [4, p. 155].

Proof. Let C be a subclique of G which reaches the maximal value of μ_G vertices. Let H be a best transitive approximation to G, with maximum clique number μ_H, and let $e \leq \binom{\mu_G}{2}$ be the number of edges between vertices in C in this graph H. Now we can transform H into a new transitive graph H', by removing all the edges joining a vertex of C with a vertex outside of C, and then rejoining all the vertices of C with each other. This yields a new transitive graph H'. The number of removed edges is at most equal to $\mu_G(\mu_H - 1) - 2e$. The number of added edges is equal to $\binom{\mu_G}{2} - e$. Because of our choice of H as a best transitive approximation to G, we must then have that $\mu_G(\mu_H - 1) - 2e \geq \binom{\mu_G}{2} - e$. It follows a fortiori that $\mu_G(\mu_H - 1) \geq \binom{\mu_G}{2}$, and hence $\mu_H - 1 \geq \frac{\mu_G - 1}{2}$. So we conclude that $\mu_H \geq \frac{\mu_G + 1}{2}$.

This proposition shows that from the best transitive approximation, we can deduce information about the maximum clique number of the graph. In the next theorem we will deduce an even tighter connection, from which we will be able to prove the NP-completeness of finding the best transitive approximation from below to G. (Remark that it suffices to prove the NP-hardness, because this problem obviously belongs to NP.)

The idea of the proof is illustrated in Fig. 2. The original graph G consists of the thick solid lines and their endpoints; this is the graph of Fig. 1. The thinner edges and vertices are added to G, thereby transforming it into a graph G'. By adding one vertex to G, and connecting it with all the vertices of G, one makes it slightly less profitable to split large cliques. So by adding sufficiently many vertices to G and connecting them to each other and to all of G, as in Fig. 2, one makes the resulting graph markedly unprofitable to split all maximum cliques. So a maximum clique of G can be recovered from every cut-approximation to G'. The precise calculation is given in the proof of the theorem, to which we turn now.

Theorem 12 *The problem of finding, for any given graph G, an element of $\mathbf{BA}^-(G)$ is NP-complete.*

Proof. Let G be given, and let n be the number of vertices of G. We now construct a graph G' as follows:

1. Add a clique C of n^2 vertices to G;

2. Draw an edge from every vertex of C to every vertex of G. Now suppose that there would be a polynomial time algorithm $\mathcal{A}$ for finding a quantitatively best cut-approximation, and let the resulting graph $\mathcal{A}(G)$ be called H. Then we claim that (1) C remains intact in H, and (2) C will be connected with a clique of maximal size of the graph G. We will now prove these two assertions.

(1) Note that every clique of H is the union of a subclique of C with a clique of G. Let $C_1 \cup G_1$ and $C_2 \cup G_2$ be two such cliques of H, and suppose

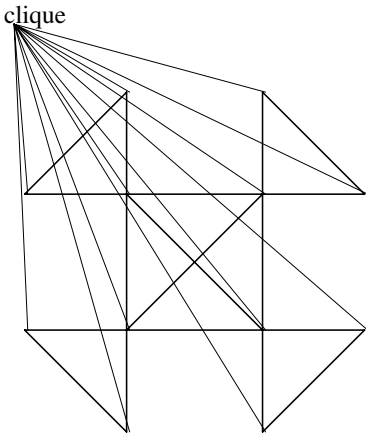

Fig. 2. Modified graph G' in which it is no longer profitable to split largest cliques

that $C_1 \neq C_2$. Assume also that n_1, n_2, c_1, c_2 are the number of elements of G_1, G_2, C_1, C_2 respectively, and that $n_1 \geq n_2$. Then we can reunite C_1 and C_2 and join the resulting clique with G_1. This yields

$$c_1c_2 + c_2n_1 - c_2n_2 > 0$$

edges, which is quantitatively better.

(2) Assume that C ends up with a clique with $< \mu$ edges, with μ the maximum clique number of G. Then this total clique contains at most $\binom{n^2+\mu-1}{2}$ edges. All other cliques of H together can contain at most $\binom{n}{2}$ edges. However, a calculation shows that

$$\binom{n^2+\mu}{2} - \binom{n^2+\mu-1}{2} = n^2 + \mu - 1 \geq n^2 > \binom{n}{2},$$

whereby C must belong to a clique which reaches the maximum size μ.

Remark 13 Consider the problem of finding, for a given graph G, a coloring of G such that

$$\sum_i \binom{\text{number of vertices of color } i}{2}$$

is maximal. This problem is equivalent to finding a best transitive approximation from below for the complementary graph $\overline{G}$ of G. Hence this problem is also NP-complete. In fact, we did not have to use the binomial function in stating this remark. By the same token, we can consider the problem of finding a coloring of G such that

$$\sum_i f(\text{number of vertices of color } i)$$

Fig. 3. Graph G

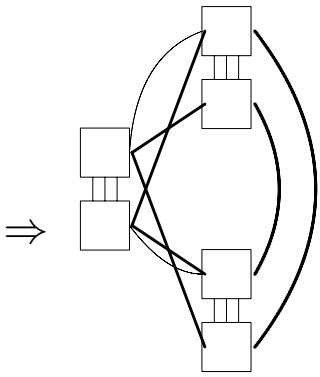

Fig. 4. Graph G'

is maximal. Then if f is such that $\lim_{n\to\infty} f(n+1) - f(n) = \infty$, this problem will also be NP-complete.

We will now turn to the problem of the complexity of finding a best cut-and-paste approximation to a given graph. In [5] it was shown that the problem is NP-complete. Here a new and simpler proof of this theorem is given. We will give a detailed description of a construction that can also be used to prove the corollaries following our proof.

Theorem 14 *The problem of finding, for any given graph G, an element of $\mathbf{BA}(G)$ is NP-complete.*

Proof. The main idea of the proof is the following. We suppose an arbitrary algorithm $\mathcal{A}$ which, given a graph G, yields an element $\mathcal{A}(G) \in \mathbf{BA}(G)$. Then we construct a polynomial-time transformation of G into a graph G' such that, from the graph $\mathbf{BA}(G')$, we are able to find a best cut-approximation for our original graph G. But we know form the previous theorem that this latter problem is NP-complete.

We will now describe this reduction. Suppose that a graph G of order n is given. We then transform this graph into a graph G' in the following way:

1. We replace every vertex u of G by a clique C_u consisting of a huge number of vertices (say, $2n^{10}$ vertices).
2. We connect each pair C_u,C_v by exactly half of the possible edges between them (i.e., $2n^{20}$ edges). We do not do this in an arbitrary way, but in a specific manner that we describe below.
3. Finally we introduce the information contained in the original graph G.
   - If there was no connection between a pair of vertices u and v in the original graph G, then we remove n^2 edges from the connections between C_u and C_v.
   - If there was a connection between u and v in the original graph G, then we add 1 more edge between C_u and C_v.

This completes the description of the transformed graph G'. As an illustration, consider the simple graph G in Figure 3, which is transformed into the graph G' in Figure 4.

Now we formulate the following claim:

Claim We can draw the edges in step 2 of this construction in such a way that when the algorithm $\mathcal{A}$ (the cut-and-paste-algorithm) is applied to G', it leaves all the cliques C_u of G' intact.

Suppose for a moment that this claim is true. Then it is clear that the algorithm $\mathcal{A}$, when applied to G', will yield us a best approximation from below for the original graph G. For if there is no edge between a pair of vertices u and v in G, then pasting C_u and C_v in G' gives us an extra cost of $2n^2$ edges, compared with non-pasting. Cutting an edge gives us an extra cost of 2 edges, compared with non-cutting. So we see that the algorithm $\mathcal{A}$ will nowhere paste an edge in G', and that it will cut a minimal number of times. The graph $\mathcal{A}(G')$ yields us then a best approximation from below for the original graph G. This ends the proof of the theorem. $\mathcal{A}$.

The hard part in finishing the proof is then to prove the claim that we made in the proof. To this end we must describe in which way we have to draw the $2n^{20}$ edges between C_u and C_v in step 2 of the construction of G'. First, we regard each clique C_u as consisting of an upper part C_u^1 and a lower part C_u^2, each containing n^{10} vertices. Then we connect each pair C_u,C_v crosswise, i.e., we fully connect C_u^1 with C_v^2 and we fully connect C_u^2 with C_v^1. This construction is illustrated in Fugure 4. With this construction, we will be able to prove the claim. We will do this in two steps. The first step is as follows:

Lemma 15 *Suppose for the moment that for the transitive graph $\mathcal{A}(G')$, each of the cliques C_u is either preserved or split into its two components C_u^1, C_u^2. Then, neglecting the $\mathcal{O}(n^2)$ changes in step 3 of the construction of G', splitting can never be more profitable than leaving each C_u intact.*

Proof. Suppose that in the transitive graph $\mathcal{A}(G')$, k of the cliques C_u are split into their components C_u^1, C_u^2, and the other $n-k$ cliques C_v are left intact. We will try to find a reduction of the problem by gradually eliminating edges in $\mathcal{A}(G')$. First, remark that we may freely eliminate all the edges between a split clique C_u^1 (say) and an entire clique C_v. Indeed, due to the construction of G' there were originally only half of the possible edges between C_u^1 and C_v, so there will be no extra costs if we eliminate these edges instead of completing them.

The same reasoning holds for the edges between two cliques C_u, C_v which were left intact in $\mathcal{A}(G')$: by construction only half of the possible edges between these cliques were present in the graph G', so for the costs it does not matter whether we eliminate or complete them.

Then let us look at the mutual edges between the split cliques C_u^1, C_u^2, C_v^1, C_v^2. Suppose l vertices u_i with corresponding indices $x_i \in \{1, 2\}$ such that the cliques $C_{u_i}^{x_i}$ form a big clique in $\mathcal{A}(G')$. We want to measure how much mistakes were necessary to create this clique. For this, we define an auxiliary graph H on the l vertices u_i: we draw an edge between u_i, u_j in the graph H if and only if the cliques $C_{u_i}^{x_i}$, $C_{u_j}^{x_j}$ were originally connected with each other in G'. Due to the construction of G', this reduces to saying that the corresponding indices x_i, x_j are different from each other. As a consequence, using the terminology of Sect. 2 we can say that this graph H must necessarily be a 02-graph (see Proposition 1), and so its complementary graph $\overline{H}$, which will be a 13-graph, has at least $\varphi(l)$ edges.

Using this, we see that to transform the $C_{u_i}^{x_i}$ into a big clique in $\mathcal{A}(G')$, we needed to make at least $\varphi(l)n^{20}$ mistakes, and the number of preserved edges was at most $(\binom{l}{2} - \varphi(l))n^{20}$. The difference between these two coefficients is

$$\binom{l}{2} - 2\varphi(l) = \left\lfloor \frac{l}{2} \right\rfloor .$$

In contrast with the previous cases, we see that this number can be different from zero. So there may have been an effective profit induced by the splitting. But this effect is bounded: since there are exactly $2k$ cliques $C_{u_i}^{x_i}$ in the graph $\mathcal{A}(G')$, summing over all the big cliques in $\mathcal{A}(G')$ which are consisting of these $C_{u_i}^{x_i}$, we have that their sizes l must satisfy

$$\sum_{l, \sum l = 2k} \left\lfloor \frac{l}{2} \right\rfloor \leq k.$$

So there will be at most a profit of kn^{20} in the splitting compared to the non-splitting case. However, there is still a cost that we did not take into consideration: of course the *splitting itself* of these k cliques C_u into C_u^1, C_u^2 will induce kn^{20} extra costs compared to non-splitting! So, globally, we see that there can be no netto profit in the splitting case.

We see however that this lemma was very "close", in the sense that it can be exactly as profitable to split as not to split, and because we neglected the $\mathcal{O}(n^2)$ edges from step 3 of the construction of G'. Therefore we modify the construction of G' slightly, to make absolutely sure that it is not profitable to split a clique in two. To this end, in the construction of G' we remove each time n^5 (say) of the "crossed" edges between C_u^1 and C_v^2 and replace them by "straight" edges, i.e., n^5 edges between C_u^1 and C_v^1. We modify the connections between C_u^2 with C_v^1 in a similar way.

Note that after this modification there are still exactly $2n^{20}$ edges between each pair of cliques C_u and C_v, and all our previous results remain valid. These modifications make it non-profitable to split because, in order to

make the splitting case as profitable as the non-splitting case, the number of preserved edges between cliques $C_{u_i}^{x_i}$, $C_{u_j}^{x_j}$ must surely have been greater then the number of them which is not preserved.

Now the second and final step in proving the claim will be the following, technical lemma.

Lemma 16 *In the transitive graph $\mathcal{A}(G')$, each of the cliques C_u is either preserved or split into its two components C_u^1, C_u^2.*

Proof. Suppose some clique C_{u_0} which is divided into a number of subcliques D_i, $i = 1, 2, \ldots, N$, where N is a certain integer. Define $D_i^x = D_i \cap C_{u0}^x$, $x \in \{1, 2\}$, and let d_i^x be the number of vertices of D_i^x.

We define the *profit* $W_i^x \in \mathbb{Z}$ which an "average" vertex w in D_i^x makes, i.e., W_i^x is the number of preserved edges starting from w, minus the number of modified edges. Hereby we do not take into account the connections inside the clique C_{u_0} itself, nor do we take into account the $\mathcal{O}(n^5)$ "extra modified" edges between each pair C_{u_0}, C_u, arising from the remark before this lemma. (Remark that an analogous notion of profit already appeared in the proof of the previous lemma). Also, let $W_{\max}^x = \max\{W_i^x\}_{i=1}^N$, $x \in \{1, 2\}$.

Now consider the first scenario: we destroy all the D_i and build up the two subcliques $C_{u_0}^1$, $C_{u_0}^2$. For the connections with vertices in the other cliques C_u, we first destroy all these edges and then rebuild them, following the pattern encountered in $W_{\max}^x$. Then the cost of splitting the cliques D_i yields each time $d_i^1 d_i^2$ mistakes. On the other hand, the profit inside D_i^1 of reconnecting $C_{u_0}^1$ is $\frac{1}{2} d_i^1 \left(n^{10} - d_i^1\right)$. (The factor $\frac{1}{2}$ is there because each of these edges is counted for two i-values.) Moreover, the "better" connections with the other cliques C_u yield also a profit of $d_i^1 \Delta W_i^1$, where $\Delta W_i^1 = W_{\max}^1 - W_i^1 \geq 0$ (neglecting the $n\mathcal{O}(n^5) = \mathcal{O}(n^6)$ "modified" edges.) So to be non-profitable, at least for one block D_i we must have that

$$d_i^1 d_i^2 \geq \frac{1}{2} d_i^1 \left(n^{10} - d_i^1\right) + d_i^1 \Delta W_i^1 \quad \text{(plus } \mathcal{O}(n^6)\text{)}$$

and thus

$$\Delta W_i^1 \leq \frac{1}{2} d_i^1 + d_i^2 - \frac{1}{2} n^{10} \quad \text{(plus } \mathcal{O}(n^6)\text{)} \tag{1}$$

Now suppose that this inequality is satisfied for some block D_i. Let us then consider the second scenario: joining all of C_{u_0}, i.e., adding to D_i all of the $n^{10} - d_i^1$ other vertices in the upper part. (The adding of the lower part can be handled separately, in exactly the same way). For the connections with the other cliques C_u, we just choose the pattern that was present in D_i. Due to the reconnection of D_i, we obtain a profit of $\left(n^{10} - d_i^1\right)\left(d_i^1 + d_i^2\right)$ edges. On the other hand, the possible loss by a "worse" connection with

the other cliques C_u is at most $\left(n^{10} - d_i^1\right) \Delta W_i^1$. So for this operation to be non-profitable, we must have

$$\Delta W_i^1 \geq d_i^1 + d_i^2 \quad (\text{plus } \mathcal{O}(n^6)) \tag{2}$$

But then it is clear that (1) and (2) can not be satisfied simultaneously, because this would imply that $d_i^1 \leq -n^{10}$ plus $\mathcal{O}(n^6)$, which is a contradiction.

This establishes the claim. Therefore we have completed the proof of Theorem 15. We will now list some corollaries of the construction that was used in the proof of this theorem.

Corollary 17 *The problem of finding, for any given G, a best transitive approximation of G consisting of at most 3 components is NP-complete.*

Proof. Suppose, for a reduction, that there exists a polynomial time algorithm $\mathcal{A}$ which yields, for any G, such a transitive approximation $\mathcal{A}(G)$. Using the same construction as in the main theorem, we can find in polynomial time a best approximation which adds a minimal number of edges. In particular, we will be able to know whether it is possible to find such a transitive approximation *without* "pasting" edges. Equivalently, we would know in polynomial time for every graph G whether its complement $\overline{G}$ allows a 3-coloring. But the problem 3-color is known to be NP-complete,[9] so we have reached the desired contradiction.

This corollary can easily be extended for a number of components $k \geq 3$. We use the following reduction: given a graph G, we construct a graph G' by adding $k-3$ disjoint cliques to G (so that the resulting graph is disconnected). Then an algorithm for finding a best transitive approximation consisting of at most k components for the graph G', would also yield us a best transitive approximation for G consisting of at most 3 components.

We will now prove the corollary for the value $k = 2$.

Corollary 18 *The problem of finding, for any given G, a best transitive approximation of G consisting of at most 2 components (i.e., a best approximating 13-graph) is NP-complete.*

Proof. Suppose that there exists a polynomial algorithm $\mathcal{A}$ which yields, for any G, such a best approximation $\mathcal{A}(G)$. Using the construction of the proof of the main theorem again, we can give a huge relative "weight" to each paste-operation. Thus we will be able to find a division of G into 2 disjoint cliques, such that a minimal number of edges has to be added. Passing to the complement $\overline{G}$, this means that we have a partition of the vertex set into two disjoint parts such that there is a minimal number of edges lying entirely in

[9] See [4, p. 154].

one of them. Equivalently, we have a partition such that there is a maximal number of edges running from the first to the second part. But it is known that the latter problem is NP-complete.[10]

In our earlier applications of the best transitive approximations of a graph, we assigned to both cutting and pasting a penalty of 1. But we can also use *weighted* penalties. For every $a, b \in \mathbb{R}^+$, we can define the notion of *best a-b-approximation* by stipulating that every paste-action carries a cost of a and every cut-action carries a cost of b. The construction of our main theorem then shows that these notions of best transitive approximation also lead to NP-completeness, provided that $b \neq 0$. We will prove this in the following corollary.

Corollary 19 *Let $a \in \mathbb{R}^+, b \in \mathbb{R}_0^+$. The problem of finding for any G a best a-b-approximation is an NP-complete problem.*

Proof. This follows by using the same construction as in the proof of the main theorem. There are, however, some slight differences in the construction of G'. In step 3 of the construction, in the case where there is no edge between u and v in the graph G, we must remove $\frac{a}{b}n^2$ more edges between C_u and C_v, because of the weights. To be sure that this number $\frac{a}{b}n^2$ can be neglected when compared with the number of vertices in each C_u, we substitute in step 1 each point u by a clique C_u consisting of $\frac{a}{b}n^{10}$ vertices and in the construction of step 2 we modify each time $\frac{a}{b}n^5$ "straight" edges. (Of course the number $\frac{a}{b}$ does not have to be an integer, but it suffices to round it up.)

We do not have to assign constant weighs to cutting and pasting edges. We can allow each vertex u to have a different penalty $a(u)$ for pasting an edge and a penalty $b(u)$ for cutting an edge. Consider the problem of finding a best "weighted" transitive approximation for a graph G, using the weights $a(u)$ and $b(u)$. When these weights are such that the quotient $\frac{a(u)}{b(u)}$ of the largest value of $a(u)$ by the smallest value of $b(u)$ increases at most as a polynomial function of n, this problem is NP-complete. This follows immediately by using the same technique as in the previous corollary.

5 Conclusion

In this paper, we have investigated complexity and combinatorial questions concerning best approximations to arbitrary graphs. From our research, a fairly complete picture emerges.

[10] See [3], where this problem is called *Max Cut*.

For different kinds of graphs, we have calculated the minimal number of edges that need to be removed or added in order to obtain a transitive graph. We have seen how this number can be expressed as a function of the order of the graph and we have investigated the structure of the particular graphs for which this minimal value is reached.

The combinatorial facts which were thus obtained were subsequently used in complexity calculations. It was shown that there is a polynomial-time algorithm for finding a best approximation for graphs with maximal clique number 2. But most of the natural complexity problems that can be posed turn out to be NP-complete. The task of finding a best transitive approximation which only removes edges is NP-complete. The task of finding a best transitive cut-and-paste approximation is also NP-complete. Even when we look for a best cut-and-paste approximation consisting of n components (with $n > 1$), the task is NP-complete. And also when the cost of removing and adding edges is (possibly non-uniformly) weighed, we face an NP-complete task. For qualitative approximations, the task becomes NP-complete when we look for an approximation with a minimal number of equivalence-classes.

Acknowledgements. We are indebted to Rafael De Clercq, Hannes Leitgeb and Bruno Leclerc for helpful discussions and suggestions. We have a second reason for being grateful to Rafael De Clercq: he has generously assisted us with drawing the figures in Latex.

References

1. Blum, N. (1990) A new approach to maximum matchings in general graphs. In: Paterson, M. (ed.) ICALP 90: Automata, Languages and Programming (LNCS 443) 17: 586–597
2. De Clercq, R., Horsten, L. Closer. Synthese (to appear)
3. Garey, M.R., Johnson, D.S., Stockmeyer, L. (1976) Some simplified NP-complete graph problems. Theoretical Computer Science 1: 237–267
4. Krantz, S. (2002) Logic and proof techniques for computer science. Birkhäuser
5. Krivanek, M., Moravek, J. (1986) NP-hard problems in hierarchical tree-clustering. Acta Informatica 23: 311–323
6. Moon, J.W. (1966) A note on approximating symmetric relations by equivalence classes. SIAM Journal of Applied Mathematics 14: 226–227
7. Tomescu, I. (1974) La réduction minimale d'un graphe à une réunion de cliques. Discrete Mathematics 10: 173–179
8. Williamson, T. (1986) Criteria of identity and the axiom of choice. Journal of Philosophy 83: 380–394
9. Williamson, T. (1990) Identity and discrimination. Blackwell
10. Zahn, C.T., Jr. (1964) Approximating symmetric relations by equivalence relations. SIAM Journal of Applied Mathematics 12: 840–847