LCNS, Vol 3059, pp. 87–99, Springer 2004

Simple Max-Cut for Split-Indifference Graphs and Graphs with Few P_4's

Hans L. Bodlaender[1], Celina M. H. de Figueiredo[2], Marisa Gutierrez[3], Ton Kloks[4], and Rolf Niedermeier[5]

[1] Institute of Information and Computing Sciences, Utrecht University, Padualaan 14, 3584 CH Utrecht, The Netherlands. hansb@cs.uu.nl
[2] Instituto de Matemática and COPPE, Universidade Federal do Rio de Janeiro, Caixa Postal 68530, 21945-970 Rio de Janeiro, RJ, Brazil. celina@cos.ufrj.br
[3] Departamento de Matemática, Universidad Nacional de La Plata, C. C. 172, (1900) La Plata, Argentina. marisa@mate.unlp.edu.ar
[4] klokskloks@zonnet.nl
[5] Wilhelm-Schickard-Institut für Informatik, Universität Tübingen, Sand 13, D-72076 Tübingen, Fed. Rep. of Germany. niedermr@informatik.uni-tuebingen.de.

Abstract. The SIMPLE MAX-CUT problem is as follows: given a graph, find a partition of its vertex set into two disjoint sets, such that the number of edges having one endpoint in each set is as large as possible. A split graph is a graph whose vertex set admits a partition into a stable set and a clique. The SIMPLE MAX-CUT decision problem is known to be NP-complete for split graphs. An indifference graph is the intersection graph of a set of unit intervals of the real line. We show that the SIMPLE MAX-CUT problem can be solved in linear time for a graph that is both split and indifference. Moreover, we also show that for each constant q, the SIMPLE MAX-CUT problem can be solved in polynomial time for $(q, q-4)$-graphs. These are graphs for which no set of at most q vertices induces more than $q-4$ distinct P_4's.

AMS classification: 68Q25, 05C85, 05C17.

Keywords: analysis of algorithms and problem complexity, efficient algorithms, graph decomposition algorithms, max-cut problem.

1 Introduction

The MAXIMUM CUT problem (or the MAXIMUM BIPARTITE SUBGRAPH problem) asks for a bipartition of the graph (with edge weights) with a total weight as large as possible. In this paper we consider only the simple case, i.e., all edges in the graph have weight one. Then the objective of this SIMPLE MAX-CUT problem is to delete a minimum number of edges such that the resulting graph is bipartite. Making a graph bipartite with few edge deletions has many applications [26]. A very recent one is found in the emerging field of SNP (single nucleotide polymorphism) analysis in computational molecular biology, e.g., see [11, 27]. Aiming for efficient algorithms, we only consider the unweighted case since the classes of graphs we consider in this paper contain all complete graphs and the (weighted) MAXIMUM CUT problem is NP-complete for every class of graphs containing all complete graphs [21, 26].

As SIMPLE MAX-CUT is NP-complete in general, there are basically two lines of research to cope with its computational hardness. First, one may study polynomial-time approximation algorithms (it is known to be approximable within 1.1383, see [13]) or try to develop exact (exponential-time) algorithms (see [15] for an algorithm running in time $2^{m/3} \cdot m^{O(1)}$, where m is the number of edges in the graph). Approximation and exact algorithms both have their drawbacks, i.e., non-optimality of the gained solution or poor running time even for relatively small problem instance sizes. Hence, the second line of research—as pursued in this paper—is to determine and analyze special graph structures that make it possible to solve the problem efficiently *and* optimally. This leads to the study of special graph classes. (Have a look at the classics [14, 9] for general information on numerous graph classes.) For example, it was shown that the SIMPLE MAX-CUT problem remains NP-complete for cobipartite graphs, split graphs, and graphs with chromatic number three [6]. On the positive side, the problem can be efficiently solved for cographs [6], linegraphs [1], planar graphs [24, 16], and for graphs with bounded treewidth [29].

In this paper we consider two classes of graphs, both of which possess nice decomposition properties which we make use of in the algorithms for SIMPLE MAX-CUT to be described. Also, both graph classes we study are related to split graphs. An indifference graph is the intersection graph of a set of unit intervals of the real line. (See [23] for more information on intersection graphs and their applications in biology and other fields.) A split graph is a graph whose vertex set admits a partition into a stable set and a clique. Ortiz, Maculan, and Szwarcfiter [25] characterized graphs that are both split and indifference in terms of their maximal cliques, and used this characterization to edge-colour those graphs in polynomial time. First, we show that this characterization also leads to a linear-time solution for the SIMPLE MAX-CUT problem for graphs that are both split and indifference.

Second, we study the class of $(q, q-4)$-graphs (also known as graphs with few P_4's [4] and introduced in [2]). These are graphs for which no set of at most q vertices induces more than $q-4$ distinct P_4's. (A P_4 is a path with four vertices.) In this terminology, the cographs are exactly the $(4, 0)$-graphs. The class of $(5, 1)$-graphs are called P_4-sparse graphs. Jamison and Olariu [20] showed that $(q, q-4)$-graphs allow a nice decomposition tree similar to cographs [20]. This decomposition can be used to find fast solutions for several in general NP-complete problems (see, e.g., [3, 22]). Also using this decomposition, we show that the SIMPLE MAX-CUT problem can be solved in polynomial time for $(q, q-4)$-graphs for every constant q.

2 Preliminaries

In this paper, G denotes a simple, undirected, finite, connected graph, and $V(G)$ and $E(G)$ are respectively the vertex and edge sets of G. The vertex-set size is denoted by $|V(G)| = N$, and K_N denotes the complete graph on N vertices. A *stable set* (or *independent set*) is a set of vertices pairwise non-adjacent in G. A *clique* is a set of vertices pairwise adjacent in G. A *maximal clique* of G is a clique not properly contained in any other clique. A *subgraph* of G is a graph H with $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. For $X \subseteq V(G)$, we denote by $G[X]$ the *subgraph induced by* X, that is, $V(G[X]) = X$ and $E(G[X])$ consists of those edges of $E(G)$ having both ends in X.

Given nonempty subsets X and Y of $V(G)$, the symbol (X, Y) denotes the subset $\{xy \in E(G) : x \in X, y \in Y\}$ of $E(G)$. A *cut* $\mathcal{K}$ of a graph G is the set of edges $(S, V(G) \setminus S)$, defined by a subset $S \subseteq V(G)$. We often write $\overline{S}$ instead of $V(G) \setminus S$. We also write $\delta(S)$ for the set of edges with exactly one endpoint in S (and the other endpoint in $V(G) \setminus S$). By $|\mathcal{K}|$ we denote the number of edges in the cut $\mathcal{K}$ and $\ell(\mathcal{K})$ is the number of edges in $E(G) \setminus \mathcal{K}$, i.e., the number of edges that are *lost* by the cut $\mathcal{K}$. A *max-cut* $\mathcal{K}$ is a cut such that $|\mathcal{K}|$ is as large as possible. The (simple) max-cut problem considers the computation of two complementary parameters of a graph G: $mc(G) = \max\{|\mathcal{K}| : \mathcal{K} \text{ is a cut of } G\} = \max_{S \subset V} |\delta(S)|$, the maximum number of edges in a cut of G; and $\ell(G) = |E(G)| - mc(G)$, the minimum number of edges lost by a cut of G (making the remaining graph bipartite). Instead of calculating $mc(G)$ directly it is sometimes more convenient to calculate first, for $i = 1, \ldots, n$, the values $mc(G, i) = \max_{S \subset V, |S| = i} |\delta(S)|$.

In the sequel, the following observations will be helpful.

Remark 1 *For K_N, the complete graph on N vertices, we have:*

- *If $(S, \overline{S})$ is a max-cut of K_N, then $|S| = \lfloor \frac{N}{2} \rfloor$;*
- *$mc(K_N) = \lfloor \frac{N}{2} \rfloor \cdot \lceil \frac{N}{2} \rceil$.*

We say that a max-cut in a complete graph is a *balanced* cut.

Remark 2 *Let H be a subgraph of a graph G and let $\mathcal{K}$ be a cut of G. If $\ell(\mathcal{K}) = \ell(H)$, then $\mathcal{K}$ is a max-cut of G.*

Proof. Since H is a subgraph of G, any cut $\mathcal{N}$ of G satisfies $\ell(\mathcal{N}) \geq \ell(H) = \ell(\mathcal{K})$. Hence $\mathcal{K}$ is a cut of minimum loss in G, in other words, $\mathcal{K}$ is a max-cut of G. ■

Remark 3 *Let $|V(G)| = N$ and let S be a subset of $V(G)$ satisfying:*

- *$|S| = \lfloor \frac{N}{2} \rfloor$;*
- *every vertex of S is adjacent to every vertex of $\overline{S}$.*

Then $(S, \overline{S})$ is a max-cut of G.

Proof. Clearly the cut $(S, \overline{S})$ has $\lfloor \frac{N}{2} \rfloor \cdot \lceil \frac{N}{2} \rceil$ edges, the maximum possible size of a cut in G. ∎

The *union* of two graphs G_1 and G_2, denoted by $G_1 \cup G_2$, is the graph such that $V(G_1 \cup G_2) = V(G_1) \cup V(G_2)$ and $E(G_1 \cup G_2) = E(G_1) \cup E(G_2)$. By way of contrast, $G_1 \setminus G_2$ denotes the subgraph of G_1 induced by $V(G_1) \setminus V(G_2)$. The (disjoint) *sum* of two graphs G_1 and G_2 makes every vertex of G_1 adjacent to every vertex of G_2.

3 Linear-time solution for split-indifference graphs

Some preliminaries

An *interval graph* is the intersection graph of a set of intervals of the real line (cf. [9, 23] for general expositions). In case of unit intervals the graph is called *unit interval*, *proper interval*, or *indifference graph.* We shall adopt the latter name, to be consistent with the terminology of indifference *orders*, defined next. (For a recent proof that the class of unit interval graphs coincides with that of the proper interval graphs, see [8].) Indifference graphs can be characterized as those interval graphs without an induced claw, (i.e., a $K_{1,3}$). Indifference graphs can also be characterized by a linear order: their vertices can be linearly ordered so that the vertices contained in the same maximal cliques are consecutive [28]. We call such an order an *indifference order.*

A *split graph* is a graph whose vertex set can be partitioned into a stable set and a clique. A *split-indifference graph* is a graph that is both split and indifference. We shall use the following characterization of split-indifference graphs in terms of their maximal cliques due to [25].

Theorem 1. *Let G be a connected graph. Then G is a split-indifference graph if and only if*

- $G = K_N$, *or*
- $G = K_m \cup K_n$, *where* $n \geq m > 1$, *and* $K_m \setminus K_n = K_1$, *or*
- $G = K_m \cup K_n \cup K_l$, *where* $n \geq m > 1$, $n \geq l > 1$, *and* $K_m \setminus K_n = K_1$, $K_l \setminus K_n = K_1$. *Moreover,* $V(K_m) \cap V(K_l) = \emptyset$ *or* $V(K_m) \cup V(K_l) = V(G)$.

This characterization was applied to obtain a polynomial-time algorithm to edge colour split-indifference graphs [25]. In the sequel, we show how to apply this characterization to obtain a linear-time algorithm to solve the max-cut problem for split-indifference graphs.

The balanced cut is not always maximal

A natural approach [7] for solving max-cut for indifference graphs is the following. Let $v_1, v_2, \ldots, v_t$ be an indifference order for G and define $\mathcal{K} = (S, \overline{S})$ as follows: Place in S all vertices with odd labels and place in $\overline{S}$ the remaining vertices (i.e., those with even labels). By definition of $\mathcal{K}$ and by Remark 1,

$\mathcal{K} \cap E(\mathcal{M})$ is a max-cut of $\mathcal{M}$, for every graph $\mathcal{M}$ induced by a maximal clique of G. This natural approach defines a cut that is locally balanced, i.e., it gives a cut that is a max-cut with respect to each maximal clique. The following example shows that $\mathcal{K}$ is not necessarily a max-cut of G. Consider the indifference graph G with five (ordered) vertices v_1, v_2, v_3, v_4, v_5, where $\{v_1, v_2, v_3, v_4\}$ induce a K_4, and $\{v_3, v_4, v_5\}$ induce a K_3. Note that the cut $(\{v_1, v_3, v_5\}, \{v_2, v_4\})$ has 5 edges, whereas the cut $(\{v_1, v_2, v_5\}, \{v_3, v_4\})$ has 6 edges. Therefore, this approach works only when the indifference graph G has only one maximal clique, i.e., when G is a complete graph which covers the first point in Theorem 1.

Let $G = K_n \cup K_m$, where $|V(K_n) \cap V(K_m)| = i$. Call K_i the graph induced by the vertices of the *intersection*. We say that a cut $\mathcal{K}$ of G is *compatible* if:

a) $\mathcal{K} \cap E(K_n)$ is a max-cut of K_n and $\mathcal{K} \cap E(K_m)$ is a max-cut of K_m;
b) Among all cuts $\mathcal{K}$ of G satisfying condition a), $|\mathcal{K} \cap E(K_i)|$ is minimal.

Clearly, the cut proposed by the natural approach satisfies condition a) but not necessarily condition b) of the definition of compatible cut. Clearly, for the example above the compatible cut gives the maximum cut. However, our subsequent study of the max-cut problem for graphs with two maximal cliques shows that it is not always possible to define a max-cut which is a compatible cut for the graph. We actually show that there are graphs for which the max-cut is not balanced with respect to any maximal clique of the graph.

In the sequel, we show how to use this approach—considering cuts $\mathcal{K}$ such that locally $\mathcal{K} \cap E(\mathcal{M})$ is a max-cut of $\mathcal{M}$, for every graph $\mathcal{M}$ induced by a maximal clique—to find first a max-cut in a graph with two maximal cliques (which covers the second point in Theorem 1) and then to find a max-cut in a split-indifference graph (by dealing with the third point in Theorem 1).

Graphs with two maximal cliques

In this section we consider general graphs with precisely two maximal cliques. Note that a graph with precisely two maximal cliques is necessarily an indifference graph but not necessarily a split graph.

Lemma 1. *Let $G = K_n \cup K_m$ with $n \geq m > i \geq 1$, where $|V(K_n) \cap V(K_m)| = i$. Call K_i the graph induced by the vertices of the intersection. Let $(S, \overline{S})$ be a cut of G. Let $x = |S \cap V(K_i)|$. Suppose $x \leq \lfloor \frac{i}{2} \rfloor$. Then, the maximum value of a cut $(S, \overline{S})$ having x vertices in $S \cap V(K_i)$ is obtained by placing the vertices outside the intersection K_i as follows:*

- *Place in S the largest possible number that is less than or equal to $\lceil \frac{n}{2} \rceil - x$ of vertices of $K_n \setminus K_i$;*
- *Place in S the largest possible number that is less than or equal to $\lceil \frac{m}{2} \rceil - x$ of vertices of $K_m \setminus K_i$.*

Proof. Let $\mathcal{N} = (S, \overline{S})$ be a cut of G. Since G contains two maximal cliques, i.e., $G = K_n \cup K_m$, with $|V(K_n) \cap V(K_m)| = i$, we may count the number of edges in the cut $\mathcal{N}$ as follows:

$$|\mathcal{N}| = |\mathcal{N} \cap E(K_n)| + |\mathcal{N} \cap E(K_m)| - |\mathcal{N} \cap E(K_i)|.$$

Now because $x = |S \cap V(K_i)|$, we have $|\mathcal{N} \cap E(K_i)| = x(i-x)$. Hence, by placing the vertices outside the intersection K_i as described, we get a cut as close as possible to the balanced cut with respect to both K_n and K_m. ■

By using the notation of Lemma 1, let $M(x)$ be the number of edges of a maximum cut of G having x vertices of K_i in S. By Lemma 1, $M(x)$ is well defined as a function of x in the interval $[0, \lfloor \frac{i}{2} \rfloor]$. We consider three cases according to the relation between i and $\lceil \frac{m}{2} \rceil$, and i and $\lceil \frac{n}{2} \rceil$. In each case, our goal is to find the values of x in the interval $[0, \lfloor \frac{i}{2} \rfloor]$ which maximize $M(x)$.

Case 1: $i \leq \lceil \frac{m}{2} \rceil \leq \lceil \frac{n}{2} \rceil$ In this case, $x \leq \lceil \frac{m}{2} \rceil$ and $i - x \leq \lceil \frac{m}{2} \rceil$. Hence, vertices outside the intersection can be placed accordingly to get balanced partitions for both K_n and K_m. Then $M(x)$ is equal to $M_1(x)$, which is defined as follows: $M_1(x) = \lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor + \lceil \frac{m}{2} \rceil \lfloor \frac{m}{2} \rfloor - x(i-x)$. We want to maximize $M_1(x)$ over the interval $[0, \lfloor \frac{i}{2} \rfloor]$. In this case, we have just one maximum, which occurs at $x = 0$.

Case 2: $\lceil \frac{m}{2} \rceil < i \leq \lceil \frac{n}{2} \rceil$ In this case, we still have $x \leq \lceil \frac{m}{2} \rceil$, but not necessarily $i - x \leq \lceil \frac{m}{2} \rceil$. If $i - x \leq \lceil \frac{m}{2} \rceil$, then the function $M(x)$ is equal to $M_1(x)$ above. Otherwise, $i - x > \lceil \frac{m}{2} \rceil$, and it is not possible to get a balanced partition with respect to K_m: By Lemma 1, the maximum cut in this case is obtained by placing all vertices of $K_m \setminus K_i$ in S. Therefore, the function $M(x)$ is

$$M(x) = \begin{cases} M_2(x) = \lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor + (m-i)(i-x) & \text{for } 0 \leq x < i - \lceil \frac{m}{2} \rceil \\ M_1(x) = \lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor + \lceil \frac{m}{2} \rceil \lfloor \frac{m}{2} \rfloor - x(i-x) & \text{for } i - \lceil \frac{m}{2} \rceil \leq x \leq \lfloor \frac{i}{2} \rfloor \end{cases}$$

It is easy to see that $M(x)$ is a function that is continuous and decreasing with maximum at $x = 0$.

Case 3: $\lceil \frac{m}{2} \rceil \leq \lceil \frac{n}{2} \rceil < i$ In this case, we distinguish three intervals for $i - x$ to be in:

If $i - x \leq \lceil \frac{m}{2} \rceil$, then vertices outside the intersection can be placed accordingly to get balanced partitions for both K_n and K_m, and $M(x) = M_1(x)$.

If $\lceil \frac{m}{2} \rceil < i - x \leq \lceil \frac{n}{2} \rceil$, then only K_n gets a balanced partition and $M(x) = M_2(x)$.

Finally, if $i - x > \lceil \frac{n}{2} \rceil$, then a maximum cut is obtained by placing all vertices outside the intersection in S and we get a new function $M_3(x)$.

Therefore, a complete description of the function $M(x)$ is

$$M(x) = \begin{cases} M_3(x) = (i-x)(n+m-2i+x) & \text{for } 0 \leq x < i - \lceil \frac{n}{2} \rceil \\ M_2(x) = \lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor + (m-i)(i-x) & \text{for } i - \lceil \frac{n}{2} \rceil \leq x < i - \lceil \frac{m}{2} \rceil \\ M_1(x) = \lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor + \lceil \frac{m}{2} \rceil \lfloor \frac{m}{2} \rfloor - x(i-x) & \text{for } i - \lceil \frac{m}{2} \rceil \leq x \leq \lfloor \frac{i}{2} \rfloor \end{cases}$$

Observe that this function is also continuous but not always decreasing. The function $M_3(x)$ is a parabola with apex at $x = \frac{2i-N}{2}$, where $N = m + n - i$ is the total number of vertices of G. For this reason, we distinguish two cases, according to the relation between i and N, as follows: $M(x)$ has maximum at $x = 0$ when $i \leq \lfloor \frac{N}{2} \rfloor$, and $M(x)$ has maximum at $x = \frac{2i-N}{2}$ when $i > \lfloor \frac{N}{2} \rfloor$. Since x takes values on the interval $[0, \lfloor \frac{i}{2} \rfloor]$, we have two possible values for x in this case: the maximum cut has either $i - \lfloor \frac{N}{2} \rfloor$ or $i - \lceil \frac{N}{2} \rceil$ vertices of K_i in S.

In summary, we have shown:

Theorem 2. *Let $G = K_n \cup K_m$ with $n \geq m > i \geq 1$, where $|V(K_n) \cap V(K_m)| = i$. Call K_i the graph induced by the vertices of the intersection. Let $v_1, v_2, \ldots, v_N$ be an indifference order of G such that vertices $v_1, v_2, \ldots, v_n$ induce a K_n, vertices $v_{n-i}, v_{n-i+1}, \ldots, v_n$ induce a K_i containing the vertices of the intersection, and $v_{n-i}, v_{n-i+1}, \ldots, v_N$ induce a K_m. A maximum cut of G is obtained as follows:*

– *If $i \leq \lceil \frac{m}{2} \rceil \leq \lceil \frac{n}{2} \rceil$, then the compatible cut $(S, \overline{S})$ that places in S the first $\lceil \frac{n}{2} \rceil$ vertices, and the last $\lceil \frac{m}{2} \rceil$ vertices, contains zero edges of K_i, and is a maximum cut of G.*
– *If $\lceil \frac{m}{2} \rceil < i \leq \lceil \frac{n}{2} \rceil$, then the cut $(S, \overline{S})$ that places in S the first $\lceil \frac{n}{2} \rceil$ vertices, and the last $m - i$ vertices, contains zero edges of K_i, is not a compatible cut, and is a maximum cut of G.*
– *If $\lceil \frac{m}{2} \rceil \leq \lceil \frac{n}{2} \rceil < i$, then we distinguish two cases. If $i \leq \lfloor \frac{N}{2} \rfloor$, then the cut $(S, \overline{S})$ that places in S the first $n - i$ vertices, and the last $m - i$ vertices, contains zero edges of K_i, is not a compatible cut, and is a maximum cut of G. If $i > \lfloor \frac{N}{2} \rfloor$, then the cut $(S, \overline{S})$ that places in $\overline{S}$ $\lceil \frac{N}{2} \rceil$ vertices if the intersection is not a compatible cut, and is a maximum cut of G.*

Split-indifference graphs with three maximal cliques

In this section we consider split-indifference graphs with precisely three maximal cliques. By Theorem 1, any such graph $G = K_m \cup K_n \cup K_l$, with $n \geq m$, $n \geq l$, satisfies $K_m \backslash K_n = \{1\}$, $K_l \backslash K_n = \{t\}$, i.e., the vertex set $V(G) = V(K_n) \cup \{1, t\}$. In other words, we have $|V(G)| = N = n + 2$. In addition, there exists an indifference order for G having vertex 1 first, vertex t last, and the remaining vertices between 1 and t.

To obtain a solution for the max-cut problem for a split-indifference graph with precisely three maximal cliques, we shall consider three cases.

Case 1: vertex 1 is adjacent to at most $\lfloor \frac{n}{2} \rfloor$ vertices or vertex t is adjacent to at most $\lfloor \frac{n}{2} \rfloor$ vertices In the preceding subsection we studied the case of two maximal cliques. In particular, we got the easy case that if a graph $H = K_n \cup K_m$, with $n \geq m$ and such that $K_m \setminus K_n = \{1\}$, then there exists a max-cut of H that places on the same side the $\lceil \frac{n}{2} \rceil$ vertices that are closer to vertex 1 with respect to the indifference order of H, and places vertex 1 and the remaining $\lfloor \frac{n}{2} \rfloor$ vertices on the opposite side.

Now suppose vertex t is adjacent to at most $\lfloor\frac{n}{2}\rfloor$ vertices. Let $(S, V(H)\setminus S)$ be a max-cut of H that places all neighbours of t on the same side S. By Remark 2, $(S, V(H)\setminus S\cup\{t\})$ is a max-cut of the entire graph G, because $(S, V(H)\setminus S\cup\{t\})$ looses the same number of edges as the cut $(S, V(H)\setminus S)$.

Case 2: both vertices 1 and t are adjacent to at least $\lceil\frac{n}{2}\rceil$ vertices but there are not $\lfloor\frac{N}{2}\rfloor$ vertices adjacent to both 1 and t Note that every vertex of K_n is adjacent to 1 or to t. Let S contain vertex 1 and a set of $\lceil\frac{n}{2}\rceil$ neighbours of t that includes all nonneighbours of 1. The only "missing" edge in the cut $(S, \overline{S})$ is the edge $1t$, an edge not present in G. Since there are not $\lfloor\frac{N}{2}\rfloor$ vertices adjacent to both 1 and t, it is not possible to define a cut for G larger than $(S, \overline{S})$ by placing vertices 1 and t on the same side.

Case 3: there exist $\lfloor\frac{N}{2}\rfloor$ vertices adjacent to both 1 and t Let S be a set of $\lfloor\frac{N}{2}\rfloor$ vertices adjacent to both 1 and t. Remark 3 justifies $(S, \overline{S})$ to be a max-cut of G.

Theorem 3. *Let G be a split-indifference graph with three maximal cliques K_m, K_n, and K_l, with $n \geq m$, $n \geq l$, and satisfying $K_m \setminus K_n = \{1\}$, $K_l \setminus K_n = \{t\}$. Let $v_1, v_2, \ldots, v_N$ be an indifference order of G having vertex 1 first, vertex t last. A maximum cut of G is obtained as follows:*

- *If vertex t is adjacent to at most $\lfloor\frac{n}{2}\rfloor$ vertices, then the cut $(S, \overline{S})$ that places in S vertex 1 and the $\lfloor\frac{n}{2}\rfloor$ vertices that are closer to t with respect to the indifference order is a maximum cut of G. An analogous result follows if vertex 1 is adjacent to at most $\lfloor\frac{n}{2}\rfloor$ vertices.*
- *If both vertices 1 and t are adjacent to at least $\lceil\frac{n}{2}\rceil$ vertices but there are not $\lfloor\frac{N}{2}\rfloor$ vertices adjacent to both 1 and t, then the cut $(S, \overline{S})$ that places in S vertex 1 and the $\lceil\frac{n}{2}\rceil$ vertices that are closer to t with respect to the indifference order is a maximum cut of G.*
- *If there exist $\lfloor\frac{N}{2}\rfloor$ vertices adjacent to both 1 and t, then the cut $(S, \overline{S})$ that places in S a set of $\lfloor\frac{N}{2}\rfloor$ vertices adjacent to both 1 and t is a maximum cut of G.*

Altogether, we obtain the following main result.

Corollary 1. SIMPLE MAX-CUT *can be solved in linear time for split-indifference graphs.*

Proof. The result directly follows from combining Theorem 1 with Remark 1, Theorem 2, and Theorem 3. ■

4 Polynomial-time solution for $(q, q-4)$-graphs

Some preliminaries A graph is a (q,t)-*graph* if no set of at most q vertices induces more than t distinct P_4's. The class of cographs are exactly the $(4,0)$-graphs, i.e., cographs are graphs without induced P_4. The class of so-called P_4-sparse graphs coincides with the $(5,1)$-graphs. The class of P_4-sparse graphs was extensively studied in [17–19, 12].

It was shown in [3] that many problems can be solved efficiently for $(q, q-4)$-graphs for each constant q. These results make use of a decomposition theorem which we state below. In this section we show that this decomposition can also be used to solve the SIMPLE MAX-CUT problem. In order to state the decomposition theorem for $(q, q-4)$-graphs we need some preliminaries.

Recall that a *split graph* is a graph of which the vertex set can be split into two sets K and I such that K induces a clique and I induces an independent set in G. A *spider* is a split graph consisting of a clique and an independent set of equal size (at least two) such that each vertex of the independent set has precisely one neighbor in the clique and each vertex of the clique has precisely one neighbor in the independent set, or it is the complement of such a graph. We call a spider *thin* if every vertex of the independent set has precisely one neighbor in the clique. A spider is *thick* if every vertex of the independent set is non-adjacent to precisely one vertex of the clique. The smallest spider is a path with four vertices (i.e., a P_4) and this spider is at the same time both thick and thin.

The SIMPLE MAX-CUT problem is easy to solve for spiders:

Remark 4 *Let G be a thin spider with $2n$ vertices where $n \geq 3$. Then $mc(G) = \lfloor \frac{n^2}{4} \rfloor + n$. If G is a thick spider then $mc(G) = n(n-1)$.*

A graph G is *p-connected* if for every partition into two non-empty sets there is a crossing P_4, that is a P_4 with vertices in both sets of the partition. The *p-connected components* of a graph are the maximal induced subgraphs which are p-connected. A p-connected graph is *separable* if there is a partition (V_1, V_2) such that every crossing P_4 has its midpoints in V_1 and its endpoints in V_2.

Recall that a *module* is a non-trivial (i.e., not $\emptyset$ or V) set of vertices which have equivalent neighborhoods outside the set. The *characteristic* of a graph is obtained by shrinking the non-trivial modules to single vertices. It can be shown (see [2, 20]) that a p-connected graph is separable if and only if its characteristic is a split graph.

Our main algorithmic tool is the following structural theorem due to [20].

Theorem 4. *For an arbitrary graph G exactly one of the following holds:*

- *G or $\overline{G}$ is disconnected.*
- *There is a unique proper separable p-connected component H of G with separation (V_1, V_2) such that every vertex outside H is adjacent to all vertices of V_1 and to none of V_2.*
- *G is p-connected.*

Furthermore, the following characterization of p-connectedness for $(q,q-4)$-graphs was obtained in [2] (also see [4]).

Theorem 5. *Let $G=(V,E)$ be a $(q,q-4)$-graph which is p-connected. Then either $|V|<q$ or G is a spider.*

Theorem 4 and Theorem 5 lead to a binary *decomposition tree* for $(q,q-4)$-graphs (also see [3] for more details). This decomposition tree can be found in linear time [5]. The leaves of this tree correspond with spiders or graphs with less than q vertices (this reflects the last point of Theorem 4 and Theorem 5). The internal nodes of this tree have one of three possible labels. If the label of an internal node is 0 or 1, then the graph corresponding with this node is the disjoint union or the sum of the graphs corresponding with the children of the node (this reflects the first point of Theorem 4). If the label of the node is 2 (this reflects the second point of Theorem 4), one of the graphs, w.l.o.g. G_1, has a separation (V_1^1,V_1^2) and it is either a spider or a graph with less than q vertices of which the characteristic is a split graph (Theorems 4 and 5), and G_2 is arbitrary. If G_1 is a spider, all vertices of G_2 are made adjacent exactly to all vertices of the clique (induced by V_1^1) of G_1. If G_1 is a graph of which the characteristic is a split graph, all vertices of G_2 are made adjacent exactly to all vertices (i.e., V_1^1) of every clique module of G_1.

In the following subsections we briefly describe the method to compute the simple max-cut for graphs with few P_4's. The main idea of the algorithm is that we compute for each node of the decomposition tree all relevant values of $mc(G',i)$, G' being the graph corresponding with this node. The table of values for such a node is computed, given the tables of the children of the node. In the subsequent paragraphs, we discuss the methods to do this, for each of the types of nodes in the decomposition tree. Once we have the table of the root node, i.e., all values $mc(G,i)$, we are done.

Cographs We review the algorithm for the SIMPLE MAX-CUT problem for cographs (i.e., $(4,0)$-graphs) which was published in [6]. A cograph which is not a single vertex is either the sum or the union of two (smaller) cographs. In other words: cographs have a decomposition tree with all internal nodes labelled 0 or 1.

Lemma 2. *Let $G=(V,E)$ be the union of $G_1=(V_1,E_1)$ and $G_2=(V_2,E_2)$. Then*

$$mc(G,i)=\max\{mc(G_1,j)+mc(G_2,i-j)\ :\ 0\le j\le i\wedge |V_1|\ge j\wedge |V_2|\ge i-j\}$$

Let $G=(V,E)$ be the sum of $G_1=(V_1,E_1)$ and $G_2=(V_2,E_2)$. Then

$$mc(G,i)=\max\{mc(G_1,j)+mc(G_2,i-j)+j(|V_2|-(i-j))+(|V_1|-j)(i-j)\ :\ 0\le j\le i\wedge |V_1|\ge j\wedge |V_2|\ge i-j\}$$

Corollary 2. *There exists an $O(N^2)$ time algorithm to compute the simple max-cut of a cograph.*

***P_4*-sparse graphs** The decomposition tree (as defined above) for graphs that are P_4-sparse has nodes with label 0, 1, or 2 [17]. Note that in the case of label 2 we can assume here that the graph G_1 is a spider (see the discussion after Theorems 4 and 5, and [18]). In the lemma below, we assume G is obtained from G_1 and G_2 as described above by the operation of a 2-labeled node. Let K be the clique and S be the independent set of G_1. Let n_i denote the number of vertices of G_i. Note that every vertex of G_2 is adjacent to every vertex of K.

Lemma 3. *Let G, G_1, G_2, S, and K be as above. Let G_1 be a thick spider. Then*

$$\begin{aligned} mc(G,i) = \max\{&mc(G_2,j) + j(|K| - j') + j'(n_2 - j) + \\ &(i - j - j')(|K| - 1) \; : \; 0 \le j, j' \le i\} \end{aligned}$$

Let G_1 be a thin spider. Then

$$\begin{aligned} mc(G,i) = \max\{&mc(G_2,j) + j(|K| - j') + j'(n_2 - j) + \\ &(i - j - j') \; : \; 0 \le j, j' \le i\} \end{aligned}$$

For P_4-sparse graphs (i.e., $(5,1)$-graphs), Lemmas 2 and 3 are sufficient to compute all the values $mc(G', i)$ for all graphs associated with nodes in the decomposition tree. Thus, we obtain:

Corollary 3. *There exists an $O(N^3)$ time algorithm to compute the simple max-cut for a P_4-sparse graph.*

$|V_1| < q$ and the characteristic of G_1 is a split graph If we have a decomposition tree of $(q, q-4)$-graphs, then there is one remaining case: G is obtained from G_1 and G_2 by the operation corresponding to a 2-labeled node, and G_1 has less than q vertices. In this case the vertex set of G_2 acts as a module, i.e., every vertex of G_2 has exactly the same set of neighbors in G_1. Let K be the set of vertices of G_1 which are adjacent to all vertices of G_2.

Let $mc(G_1, j, j')$ be the maximum cut in G_1 with exactly j vertices in K and j' vertices in $V_1 - K$. Since G_1 is constant size the numbers $mc(G_1, j, j')$ can easily be computed in constant time.

Lemma 4. *Let G, G_1, G_2, K be as above. Suppose that $|V_1| < q$ and the characteristic of G_1 is a split graph. Then*

$$\begin{aligned} mc(G,i) = \max\{&mc(G_2,j) + mc(G_1, j', i - j - j') + \\ &j(|K| - j') + j'(n_2 - j) + (i - j - j') \; : \; 0 \le j, j' \le i\} \end{aligned}$$

Now, with Lemma 4, and Lemmas 2 and 3, we obtain:

Theorem 6. *There exists an $O(N^4)$ time algorithm for the* SIMPLE MAX-CUT *problem on $(q, q-4)$-graphs for each constant q.*

5 Concluding remarks

This paper considers two classes of graphs: indifference graphs and $(q, q-4)$-graphs. Both classes possess nice decomposition properties which we make use of in the described algorithms for SIMPLE MAX-CUT. Also, both graph classes we study are related to split graphs, a class of graphs for which SIMPLE MAX-CUT is known to be hard.

A linear-time algorithm for the recognition of indifference graphs was presented by de Figueiredo et al. [10]. The algorithm partitions in linear time the vertex set of an indifference graph into sets of *twin* vertices, i.e., vertices of the graph that belong to the same set of maximal cliques.

Given a graph G with a bounded number of maximal cliques, the partition of G into sets of twins contains a bounded number k of sets. Hence, we can compute $mc(G)$ in polynomial time, by maximizing a function on k variables x, that assume integer values in a limited region of the space, i.e., on a finite domain. This simple argument establishes the polynomial upper bound $O(N^k)$ for the max-cut problem for a class of graphs with a bounded number of maximal cliques.

One goal of this paper was to establish a linear time upper bound for the computation of $mc(G)$ for a split-indifference graph G, by computing the value of $mc(G)$ in constant time, given that we can in linear time determine which case of the computation we are in. We leave it as an open problem to extend the proposed solution to the whole class of indifference graphs.

Another goal reached by this paper was to extend to the whole class of $(q, q-4)$-graphs the known solution of SIMPLE MAX-CUT for cographs. We leave it as an open problem to find a more efficient polynomial-time algorithm for the computation of $mc(G)$ for a $(q, q-4)$-graph G.

Acknowledgments We are grateful to J. Spinrad for pointing out that the max-cut problem is solvable in polynomial time for any class of graphs with a bounded number of maximal cliques. Parts of this research were done while Celina M. H. de Figueiredo visited Universidad Nacional de La Plata supported by FAPERJ, and Marisa Gutierrez visited Universidade Federal do Rio de Janeiro supported by FOMEC and FAPERJ. Rolf Niedermeier is supported by the DFG, Emmy Noether research group "PIAF" (fixed-parameter algorithms), NI 369/4.

References

1. C. Arbib. A polynomial characterization of some graph partitioning problem. *Inform. Process. Lett.*, 26:223–230, 1987/1988.
2. L. Babel. On the P_4-structure of graphs, *Habilitationsschrift*, Zentrum für Mathematik, Technische Universität München, 1997.
3. L. Babel, T. Kloks, J. Kratochvíl, D. Kratsch, H. Müller, and S. Olariu. Efficient algorithms for graphs with few P_4's. Combinatorics (Prague, 1998). *Discrete Math.*, 235:29–51, 2001.

4. L. Babel and S. Olariu. On the structure of graphs with few P_4s. *Discrete Appl. Math.*, 84:1–13, 1998.
5. S. Baumann. A linear algorithm for the homogeneous decomposition of graphs, Report No. M-9615, Zentrum für Mathematik, Technische Universität München, 1996.
6. H. L. Bodlaender and K. Jansen. On the complexity of the maximum cut problem. In *STACS 94*, Lecture Notes in Computer Science 775, 769–780, Springer, Berlin, 1994. Also in *Nordic J. Comput.*, 7(1):14–31, 2000.
7. H. L. Bodlaender, T. Kloks, and R. Niedermeier. Simple max-cut for unit interval graphs and graphs with few P_4's. In *Extended abstracts of the 6th Twente Workshop on Graphs and Combinatorial Optimization* 12–19, 1999. Also in *Electronic Notes in Discrete Mathematics* **3**, 1999.
8. K. P. Bogart and D. B. West. A short proof that 'proper = unit', *Discrete Math.*, 201:21–23, 1999.
9. A. Brandstädt, V. B. Le, and J. P. Spinrad, *Graph Classes: a Survey.* SIAM Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1999.
10. C. M. H. de Figueiredo, J. Meidanis, and C. P. de Mello. A linear-time algorithm for proper interval graph recognition. *Inform. Process. Lett.*, 56:179–184, 1995.
11. E. Eskin, E. Halperin, and R. M. Karp. Large scale reconstruction of haplotypes from genotype data. In *RECOMB 2003*, pp. 104–113, ACM Press, 2003.
12. V. Giakoumakis, F. Roussel, and H. Thuillier. On P_4-tidy graphs. *Discrete Mathematics and Theoretical Computer Science*, 1:17–41, 1997.
13. M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42:1115–1145, 1995.
14. M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.
15. J. Gramm, E. A. Hirsch, R. Niedermeier, and P. Rossmanith. Worst-case upper bounds for MAX-2-SAT with application to MAX-CUT. *Discrete Appl. Math.*, 130(2):139–155, 2003.
16. F. O. Hadlock. Finding a maximum cut of a planar graph in polynomial time. *SIAM J. Comput.*, 4:221–225, 1975.
17. B. Jamison and S. Olariu. A tree representation for P_4-sparse graphs. *Discrete Appl. Math.*, 35:115–129, 1992.
18. B. Jamison and S. Olariu. Recognizing P_4-sparse graphs in linear time. *SIAM J. Comput.*, 21:381–406, 1992.
19. B. Jamison and S. Olariu. Linear time optimization algorithms for P_4-sparse graphs. *Discrete Appl. Math.*, 61:155–175, 1995.
20. B. Jamison and S. Olariu. p-components and the homogeneous decomposition of graphs. *SIAM J. Discrete Math.*, 8:448–463, 1995.
21. R. M. Karp. Reducibility among combinatorial problems. *Complexity of computation* (R. E. Miller and J. W. Thather eds.), pp. 85–103, 1972.
22. T. Kloks and R. B. Tan. Bandwidth and topological bandwidth of graphs with few P_4's. In 1st Japanese-Hungarian Symposium for Discrete Mathematics and its Applications (Kyoto, 1999). *Discrete Appl. Math.*, 115(1–3):117–133, 2001.
23. T. A. McKee and F. R. Morris. *Topics in Intersection Graph Theory.* SIAM Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1999
24. G. I. Orlova and Y. G. Dorfman, Finding the maximal cut in a graph, *Engrg. Cybernetics*, 10:502–504, 1972.

25. C. Ortiz Z., N. Maculan, and J. L. Szwarcfiter. Characterizing and edge-colouring split-indifference graphs. *Discrete Appl. Math.*, 82(1–3):209–217, 1998.
26. S. Poljak and Z. Tuza. Maximum cuts and large bipartite subgraphs. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science* (W. Cook, L. Lovász, and P. Seymour eds.), 20:181–244, Amer. Math. Soc., Providence, RI, 1995.
27. R. Rizzi, V. Bafna, S. Istrail, and G. Lancia. Practical algorithms and fixed-parameter tractability for the single individual SNP haplotypying problem. In *WABI 2002*, Lecture Notes in Computer Science 2452, pp. 29–43, Springer, Berlin, 2002.
28. F. S. Roberts. On the compatibility between a graph and a simple order. *J. Combinatorial Theory Ser. B*, 11:28–38, 1971.
29. T. V. Wimer. *Linear algorithms on k-terminal graphs*, PhD Thesis, Department of Computer Science, Clemson University, South Carolina, 1987.