The Maximum Cut and Minimum Cut Into Bounded Sets problems on Cographs

Hans L. Bodlaender

RUU-CS-87-12
August 1987

The MAXIMUM CUT and MINIMUM CUT INTO BOUNDED SETS problems on Cographs

Hans L. Bodlaender

Technical Report RUU-CS-87-12
August 1987

Department of Computer Science
University of Utrecht
P.O. Box 80.012
3508 TA Utrecht
the Netherlands

The MAXIMUM CUT and MINIMUM CUT INTO BOUNDED SETS problems on Cographs*

Hans Bodlaender
Dept. of Computer Science, University of Utrecht
P.O. Box 80.012, 3508 TA Utrecht, the Netherlands

August, 1987

Abstract

In this note we give simple $\mathcal{O}(n^2)$ algorithms for the unweighted MAX CUT problem and the unweighted MINIMUM CUT INTO BOUNDED SETS problem on cographs. The algorithms can easily be modified such that they actually yield the partitions with the desired characteristics. The weighted variants of the problems are shown to be NP-complete, when restricted to cographs (and even complete graphs), even when all weights are either 1 or 2.

1 Introduction

As it is generally believed that NP-complete problems are not solvable in polynomial time, much research has been done on the complexity of NP-complete problems. In [4] an overview is given of the known complexity of some well-known NP-complete problems when restricted to a number of important classes of graphs. It appears that some classes will usually render a problem to be solvable in polynomial time (e.g., the classes of trees, partial k-trees, or interval graphs), while other classes will in general not help to 'ease' the problem (e.g., the classes of degree k-graphs or planar graphs).

One of the classes which seems to make most NP-complete graph problems 'easy', i.e., solvable in polynomial time, is the class of cographs. A large

*This research was done, while the author was visiting the Lab. for Computer Science of the Massachusetts Institute of Technology, with financial support by the Netherlands Organization for the Advancement of Pure Research Z.W.O.

number of NP-complete graph problems has been shown to be solvable efficiently, (i.e., in polynomial time with a very small degree of the polynomial), when restricted to cographs in [1].

In this note we focus on the MAX CUT and the MINIMUM CUT INTO BOUNDED SETS problems on cographs. We give some simple quadratic algorithms that solve the unweighted variants of these problems, (which are NP-complete for general graphs) and prove the weighted variant of the problems to be NP-complete even if all weights are either 1 or 2.

2 Definitions

The MAX CUT problem is the problem where given a graph $G = (V, E)$, a weight $w(e) \in Z^+$ for each edge $e \in E$, and a positive integer K, one asks whether there exists a partition of V into disjoint sets V_1, V_2, such that the sum of the weights of the edges from E that have one endpoint in V_1 and one endpoint in V_2 is at least K.

The SIMPLE MAX CUT problem is the variant of the problem where all weights are 1, i.e., we ask for a partition of V into disjoint sets V_1, V_2, such that the number of edges in E that have one endpoint in V_1 and one endpoint in V_2 is at least K. The SIMPLE MAX CUT problem is NP-complete [3].

The MINIMUM CUT INTO BOUNDED SETS problem is the problem where given a graph $G = (V, E)$, a weight $w(e) \in Z^+$ for each edge $e \in E$, specified vertices s, $t \in V$, a positive integer $B \leq |V|$, and a positive integer K, one asks whether there exists a partition of V into disjoint sets V_1, V_2, such that $s \in V_1$, $t \in V_2$, $|V_1| \leq B$, $|V_2| \leq B$, and the sum of the weights of the edges from E that have one endpoint in V_1 and one endpoint in V_2 is at most K.

The SIMPLE MINIMUM CUT INTO BOUNDED SETS problem is the variant of the problem where all weights are 1, i.e., we ask for a partition of V into disjoint sets V_1, V_2, such that $s \in V_1$, $t \in V_2$, $|V_1| \leq B$, $|V_2| \leq B$, and the number of edges in E that have one endpoint in V_1 and one endpoint in V_2 is at most K. The SIMPLE MINIMUM CUT INTO BOUNDED SETS problem is NP-complete, even if $B = |V|/2$ [3].

We use the following well-known characterization of the class of cographs.

DEFINITION 2.1 *A graph $G = (V, E)$ is a co-graph, iff one of the following conditions hold.*

1. $|V| = 1$ *(and hence $E = \emptyset$)*

2. *The complement of G*, $\bar{G} = (V, \bar{E})$, *with* $\bar{E} = \{(v,w) \mid v,w \in V, v \neq w \wedge (v,w) \notin E\}$ *is a co-graph.*

3. *There are disjoint cographs* $G_1 = (V_1, E_1), \ldots, G_k = (V_k, E_k)$, *(i.e.,* $i \neq j \Rightarrow V_i \cap V_j = \emptyset$ *) and G is the union of* $G_1, \ldots, G_k$, *i.e.,* $G = G_1 \cup \cdots \cup G_k = (V_1 \cup \cdots \cup V_k, E_1 \cup \cdots \cup E_k)$.

To each cograph G one can associate a corresponding rooted tree T, called the "cotree" of G, in the following manner. Non-leaf nodes in the tree are labeled either with $\cup$ ("union") or with $\bar{\cdot}$ ("complement"). A complement node has exactly one child-node in the cotree. To each node of the cotree one can associate a cograph in the following way. A leaf-node corresponds to a cotree with one vertex and no edges. A union-node corresponds to the (disjoint) union of the cographs corresponding to its children in the tree. A complement-node corresponds to the complement of the cograph corresponding to its child in the tree. The root-node of the cotree now corresponds to the cograph G which is represented by this cotree.

In [2] it is shown that in linear time one can recognize whether a graph is a cograph, and for cographs find the corresponding co-tree.

By noting that $G_1 \cup \cdots \cup G_{k-1} \cup G_k = (G_1 \cup \cdots \cup G_{k-1}) \cup G_k$ it easily follows that we can restrict ourselves to binary tree, i.e., we may assume that each union-node has exactly 2 children. By removing all pairs of adjacent complement-nodes from the tree, one obtains a cotree, with $\mathcal{O}(|V|)$ nodes.

3 A quadratic algorithm for SIMPLE MAX CUT on cotrees

We now will give an informal description of our algorithm for the SIMPLE MAX CUT problem on cotrees. The first step of the algorithm is to calculate the cotree of G, as indicated in the previous section. Next, recursively, we will calculate for each node α in the cotree T a table. Let $H = (W, F)$ be the cograph, corresponding to H. The table will contain for each i, $0 \leq i \leq |W|$, two numbers $\text{maxc}(i, H)$, and $\text{minc}(i, H)$, with $\text{maxc}(i, H)$ denoting the "maximum cut of H with $|V_1| = i$" and $\text{minc}(i, H)$ denoting the "minimum cut of H with $|V_1| = i$", i.e.,

- $\text{maxc}(i, H) = \max \{|\{(v,w) \mid v \in V_1, w \in V_2\}| \mid V_1 \cup V_2 = W, V_1 \cap V_2 = \emptyset, |V_1| = i\}$.
- $\text{minc}(i, H) = \min \{|\{(v,w) \mid v \in V_1, w \in V_2\}| \mid V_1 \cup V_2 = W, V_1 \cap V_2 = \emptyset, |V_1| = i\}$.

It is obvious that one can calculate the maximum cut of a graph G in time linear in $|V|$, after the table with all values of $\mathrm{maxc}(i, G)$ has been calculated. So it remains to show that the tables can be calculated effectively for each node in the cotree.

It is obvious that the following table must be used for a leaf of the cotree.

i	maxc	minc
0	0	0
1	0	0

For complement-nodes in the cotree, one can use the following lemma in order to calculate the maxc- and minc-tables.

Lemma 3.1 *Let $G = (V, E)$ be a graph with $|V| = n$. Then*

1. $\mathrm{maxc}(i, \bar{G}) = i(n - i) - \mathrm{minc}(i, G)$.
2. $\mathrm{minc}(i, \bar{G}) = i(n - i) - \mathrm{maxc}(i, G)$.

Proof. Consider a partition V_1, V_2 of V with $|V_1| = i$. Note that $|\{(v, w) \in \bar{E} \mid v \in V_1 \wedge w \in V_2\}| + |\{(v, w) \in E \mid v \in V_1 \wedge w \in V_2\}| = |V_1| \cdot |V_2| = i(n - i)$. Hence the lemma follows. **Q.E.D.**

The lemma suggests a procedure to calculate in linear time the table for a complement-node from the table for its child-node. As there are at most $\mathcal{O}(|V|)$ complement nodes in the cotree, the total work of calculating the tables for complement nodes is bounded by $\mathcal{O}(|V|^2)$.

For union-nodes we use the following lemma.

Lemma 3.2 *Let $G = (V, E)$, $H = (W, F)$ be disjoint graphs. Then:*

1. $\mathrm{maxc}(i, G \cup H) = \max_{0 \le j \le i}(\mathrm{maxc}(j, G) + \mathrm{maxc}(i - j, H))$.
2. $\mathrm{minc}(i, G \cup H) = \min_{0 \le j \le i}(\mathrm{minc}(j, G) + \mathrm{minc}(i - j, H))$.

Proof.

1. $\mathrm{maxc}(i, G \cup H) = \max_{0 \le j \le i} \{|\{(v, w) \mid v \in V_1 \wedge w \in V_2\}| \mid V_1 \cup V_2 = V \cup W,\ V_1 \cap V_2 = \emptyset,\ |V_1| = i,\ |V_1 \cap V_2| = j\} = \max_{0 \le j \le i}\{|\{(v, w) \mid v \in V_1 \cap V,\ w \in V_2 \cap V\}| + \{|\{(v, w) \mid v \in V_1 \cap W,\ w \in V_2 \cap W\}| \mid V_1 \cup V_2 = V \cup W,\ V_1 \cap V_2 = \emptyset,\ |V_1| = i,\ |V_1 \cap V_2| = j\} = \max_{0 \le j \le i}(\mathrm{maxc}(j, G) + \mathrm{maxc}(i - j, H))$.

2. Similar. **Q.E.D.**

It follows that one can calculate the table for $G \cup H$ in $\mathcal{O}(|V| \cdot |W|)$ time from the tables for $G = (V, E)$ and $H = (W, F)$. Let $u(G)$ denote the total time needed for the calculations of tables in all *union nodes* in the cotree of the cograph G. Let $u(n)$ be the maximum of $u(G)$ over all cographs $G = (V, E)$ with $|V| = n$. For all cographs $G = (V, E)$, either $G = H_1 \cup H_2$ or $\bar{G} = H_1 \cup H_2$ or $|V| = 1$. (H_1, H_2 disjoint graphs.) In the latter case $u(G) = 0$. In the former two cases, let $H_1 = (W_1, F_1)$, $H_2 = (W_2, E_2)$, $|W_1| = m_1$, $|W_2| = m_2$. It follows that $u(G) \leq u(H_1) + u(H_2) + c \cdot m_1 \cdot m_2$, for some constant c. Hence for all $n > 1$, $u(n) \leq \max_{1 \leq i \leq n} i \cdot (n - i) + u(i) + u(n - i)$; and $u(1) = 0$. With induction it follows that $\forall n : u(n) \leq 2c \cdot n^2$. Hence the total work needed to calculate the tables for all union nodes takes time, quadratic in $|V|$.

Theorem 3.3 *There exists an $\mathcal{O}(n^2)$ algorithm for* SIMPLE MAX CUT *on cographs.*

We remark that it is not difficult to modify the algorithm, such that it actually will yield a partition which gives the maximum cut without increasing the time by more than a small constant factor. We leave the (easy) details to the reader.

4 Unweighted MINIMUM CUT INTO BOUNDED SETS on cographs

In this section we show how the algorithm of section 3 can be modified, such that it solves the unweighted MINIMUM CUT INTO BOUNDED SETS problem on cographs. We use the following variant of maxc and minc, for some given vertices $s, t \in V$.

- $\text{maxc}'(i, H) = \max \{|\{(v, w) \mid v \in V_1,\ w \in V_2\}| \mid V_1 \cup V_2 = W, V_1 \cap V_2 = \emptyset, |V_1| = i, s \in V \rightarrow s \in V_1, t \in V \rightarrow t \in V_2\}$.
- $\text{minc}'(i, H) = \min \{|\{(v, w) \mid v \in V_1,\ w \in V_2\}| \mid V_1 \cup V_2 = W, V_1 \cap V_2 = \emptyset, |V_1| = i, s \in V \rightarrow s \in V_1, t \in V \rightarrow t \in V_2\}$.

The algorithm is very similar to the algorithm of section 3. In fact, the *only* changes that have to be made, are in the procedure to calculate the tables for leaf-nodes, and in the procedure to look up the answer from the table of the root-node. For union-nodes and complement nodes the same procedures can be used, as shown by the following lemmas.

Lemma 4.1 *Let $G = (V, E)$ be a graph with $|V| = n$. Then*

1. $\mathrm{maxc}'(i, \bar{G}) = i(n-i) - \mathrm{minc}'(i, G)$.
2. $\mathrm{minc}'(i, \bar{G}) = i(n-i) - \mathrm{maxc}'(i, G)$.

Lemma 4.2 *Let $G = (V, E)$, $H = (W, F)$ be disjoint graphs. Then:*

1. $\mathrm{maxc}'(i, G \cup H) = \max_{0 \le j \le i} (\mathrm{maxc}'(j, G) + \mathrm{maxc}'(i-j, H))$.
2. $\mathrm{minc}'(i, G \cup H) = \min_{0 \le j \le i} (\mathrm{minc}'(j, G) + \mathrm{minc}'(i-j, H))$.

Hence we have:

Theorem 4.3 *There exists an $\mathcal{O}(n^2)$ algorithm for* Simple Minimum Cut into Bounded Sets *on cographs.*

5 NP-completeness for the weighted variants

In this section we will proof that the weighted variants of the problems (on cographs) are NP-complete, even if all weights are either 1 or 2. To be precise, we will proof NP-completeness of these problems restricted to the class of all complete graphs, which clearly is a subclass of the cographs.

Theorem 5.1 *Weighted* Max Cut *with all weights in $\{1, 2\}$ restricted to complete graphs is NP-complete.*

Proof. It is obvious that the problem is in NP. In order to proof that the problem is NP-hard, we use a transformation from Simple Max Cut (without restrictions on the graphs). Consider an unweighted graph $G = (V, E)$; let $|V| = n$. Let $w_1, \ldots, w_n$ be n new vertices, not in V. Let $W = \{w_1, \ldots, w_n\}$. Let $H = (V \cup W, F)$ be the complete graph on $V \cup W$. Now we label each edge e in H with a weight $w(e)$ that is either 1 or 2, as follows: $w(e) = 1$, if $e \notin E$, and $w(e) = 2$, if $e \in E$.

Now we claim that there is a partition of V in disjoint sets V_1, V_2, such that at least K edges are going from a vertex in V_1 to a vertex in V_2, if and only if there is a partition of $V \cup W$ in disjoint sets V_1', V_2', such that the sum of the weights of all edges going from a vertex in V_1' to a vertex in V_2' (in H) is at least $K + \frac{1}{4}n^2$.

First suppose we have a partition of V in disjoint sets V_1, V_2, such that at least K edges are going from a vertex in V_1 to a vertex in V_2. W can be partitioned into disjoint sets W_1, and W_2, such that $|V_1| = |W_2|$ and $|V_2| = |W_1|$. Let $V_1' = V_1 \cup W_1$ and $V_2' = V_2 \cup W_2$. Note that $|V_1'| = |V_2'| = \frac{1}{2}n$. It follows that the sum of the weights of all edges between V_1' and V_2' is at least $K + |V_1'| \cdot |V_2'| = K + \frac{1}{4}n^2$.

Next, suppose we have a partition of $V \cup W$ in disjoint sets V_1', V_2', such that the sum of the weights of all edges going from a vertex in V_1' to a vertex in V_2' (in H) is at least $K + \frac{1}{4}n^2$. Let $V_1 = V_1' \cap V$ and $V_2 = V_2' \cap V$. Let L be the number of edges in E going between a vertex in V_1 and a vertex in V_2. Note that $K + \frac{1}{4}n^2 \leq L + |V_1'| \cdot |V_2'|$. Because $|V_1'| + |V_2'| = n$, it follows that $|V_1'| \cdot |V_2'| \leq \frac{1}{4}n^2$ and hence $K \leq L$.

So, we have a polynomial transformation from the SIMPLE MAX CUT problem (without restrictions on the graphs), to the weighted MAX CUT problem on complete graphs with weights in $\{1, 2\}$. Hence the latter is NP-complete. **Q.E.D.**

In a similar way one obtains the same result for the MINIMUM CUT INTO BOUNDED SETS problem. Use for instance that the unweighted MINIMUM CUT INTO BOUNDED SETS problem is NP-complete, even if one requires that $K = |V|/2$.

Theorem 5.2 *Weighted* MINIMUM CUT INTO BOUNDED SETS *with all weights in* $\{1, 2\}$ *restricted to complete graphs is NP-complete.*

6 Some final remarks

In this note we determined the complexity of the weighted and unweighted variants of the MAX CUT problem and the MINIMUM CUT INTO BOUNDED SETS problem on cographs. Some open problems are whether better (i.e., $\mathcal{O}(n)$ or $\mathcal{O}(n \log n)$) algorithms exists for the unweighted variants, and whether polynomial algorithms can be found for larger classes of graphs, e.g., the permutation graphs.

The technique used in this note seems to be useful also for other problems on cographs. For instance, with similar means, one can design polynomial algorithms for DOMATIC NUMBER, MINIMUM MAXIMAL MATCHING, PARTITION INTO TRIANGLES, PARTITION INTO FORESTS, CUBIC SUBGRAPH, and probably others when restricted to the class of cographs.

References

[1] D. Corneil, H. Lerchs, and L. Stewart. Complement reducible graphs. *Disc. Applied Math.*, 3:163–174, 1981.

[2] D. G. Corneil, Y. Perl, and L. Stewart. A linear recognition algorithm for cographs. *SIAM J. Comput.*, 4:926–934, 1985.

[3] M. R. Garey and D. S. Johnson. *Computers and Intractability, A Guide to the Theory of NP-Completeness.* W.H. Freeman and Company, New York, 1979.

[4] D. S. Johnson. The NP-completeness column: an ongoing guide. *J. of Algorithms*, 6:434–451, 1985.