Approximation algorithms for simple maxcut of split graphs

Rubens Sucupira[1] **Luerbio Faria**[1] **Sulamita Klein**[2]

1- IME/UERJ
UERJ, Rio de Janeiro–RJ, Brasil
rasucupira@oi.com.br, luerbio@cos.ufrj.br

2 - Instituto de Matemática e COPPE/Sistemas
UFRJ, Rio de Janeiro–RJ, Brasil
sula@cos.ufrj.br

Abstract

Given a graph, the SIMPLE MAX-CUT problem asks to 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. It is known that the SIMPLE MAX-CUT decision problem is NP-complete for general graphs and there is a polynomial time (1/2)-approximation algorithm to solve this problem. In particular, Bodlaender and Jansen proved that this problem remains NP-complete for split graphs. A split graph is a graph whose vertex set admits a partition into a stable set and a clique. Goemans and Williamson developed a semidefinite programming approximation algorithm with approximation ratio of 0,87856 to solve the SIMPLE MAX-CUT problem for general graphs. In this paper we show a polynomial time (2/3)-approximation algorithm for simple maxcut of split graphs and deterministic algorithms for simple maxcut of full (k,n)-split graphs using only simple combinatorial methods.

KEYWORDS: Simple Max Cut. Split Graphs. Approximation Algorithms.
Main Area: Algorithms and Graph Theory

1 Introduction

The MAX-CUT problem is a combinatorial graph problem in which we have a weighted graph $G = (V, E)$ and we look for a partition of the vertices set V of G into two disjoint sets A and B such that the sum of the weights of the edges with one endpoint in A and the other in B is as large as possible. In the SIMPLE MAX-CUT problem, we consider the variant where all edges have weight one. The subset of the edges with one endpoint in A an the other in B is denoted by $[A, B]$ and is called an *edge cut* of G and the edges of $[A, B]$ are called *cut edges*. The *size* of the cut $[A, B]$ is the number of cut edges of $[A, B]$ which is represented by $|[A, B]|$.

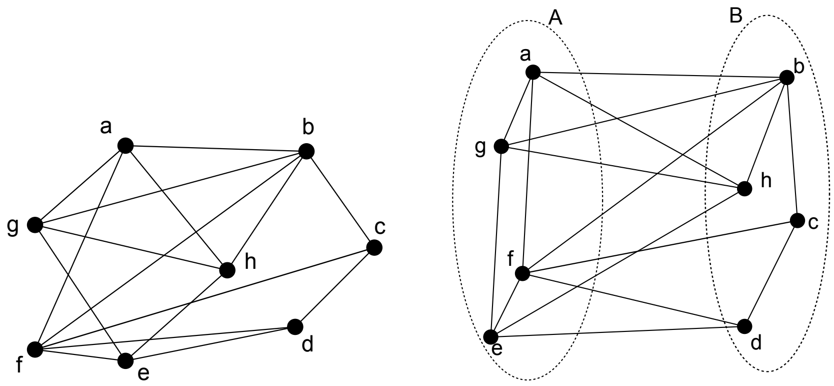
We can see in figure 1 an example of a graph G and an edge cut of G.

The SIMPLE MAX-CUT problem can be formulated as a decision problem as stated bel-low:

SIMPLE MAX-CUT PROBLEM
Instance: Undirected graph $G = (V, E)$, $k \in \mathbb{N}$.
Question: Does there exist a set $S \subset V$, such that the number of edges with one end point in S and the other one in $V \backslash S$ is greater than or equal to k ?

Karp (1972) proved that this decision problem is NP-complete for general graphs using a reduction from PARTITION problem. Karp proved in his article "Reducibility among combinatorial problems" that 21 specified combinatorial problems are NP-complete starting from SATISFIABILITY. Goemans & Williamson (1995) developed the first semidefinite programming approximation algorithm to solve the SIMPLE MAX-CUT problem for general graphs with approximation ratio of 0,87856. Later, Bodlaender, Jansen & Forschungsgemeinschaft (2000) proved that the SIMPLE MAX-CUT problem is NP-complete for some special graph classes such as chordal graphs, undirected path graphs, split graphs, tripartite graphs, and graphs that are the complement of a bipartite graph.

In this paper we present an approximation algorithm to determine the SIMPLE MAX-CUT of a split graph with approximation ratio of $\frac{2}{3}$. Although the ratio given by Goemans-Williamson algorithm is better than $\frac{2}{3}$, it is based on semidefinite programming. In this paper we give a simple combinatorial method to find an approximation of the max-cut. Considering the time complexity, Goemans-Williamson algorithm runs in polynomial time in the input and $log\frac{1}{\varepsilon}$. Given any $\varepsilon > 0$, semidefinite programs can be solved within an additive error of ε in polynomial time (ε is part of the input, so the running time dependence on ε is polynomial in $log\frac{1}{\varepsilon}$).Vega (1996) gives,for any given $0 < \alpha < 1$,$\varepsilon > 0$, a randomized algorithm which runs in a polynomial time and which, when applied to any given graph G on n vertices with minimum degree $\geq \alpha n$, outputs a cut of G that has probability greater or equal to $1 - 2^{-n}$ to have size greater or equal to $(1 - \varepsilon)|MC(G)|$, where $|MC(G)|$ is the size of the maximum cut of G.



(a) (b)

Figure 1: (a) A graph G and (b) An edge cut of G

Later, Feige, Karpinski & Langberg (2002) used the Goemans-Williamson algorithm with an improvement local step and got for graphs with degree at most Δ, an approximation ratio of at least $\alpha + \varepsilon$, where $\alpha = 0.87856$ and $\varepsilon > 0$ depends only on Δ. Using computer assisted analysis, Feige shows that for graphs of maximal degree 3 his algorithm obtains an approximation ratio of at least 0.921, and for 3-regular graphs the approximation ratio is at least 0.924. Our algorithm is deterministic with linear time and in some cases, gives exactly the maximum cut of the considered graph.

2 Preliminaries

A *split graph* $G = (S, K)$ is a graph that has the vertices set $V = S \cup K$ where S is a *stable set* and K is a *maximal clique*. If all vertices of S have the same degree k, we call G a *k-split graph*. A (k, n)*-split graph* is a k-split graph where the maximal clique K has n vertices. Note that $k < n$, otherwise the clique K is not a maximal clique since it can be increased. In figure 2 we present examples of a split graph and a $(2, 4)$-split graph.

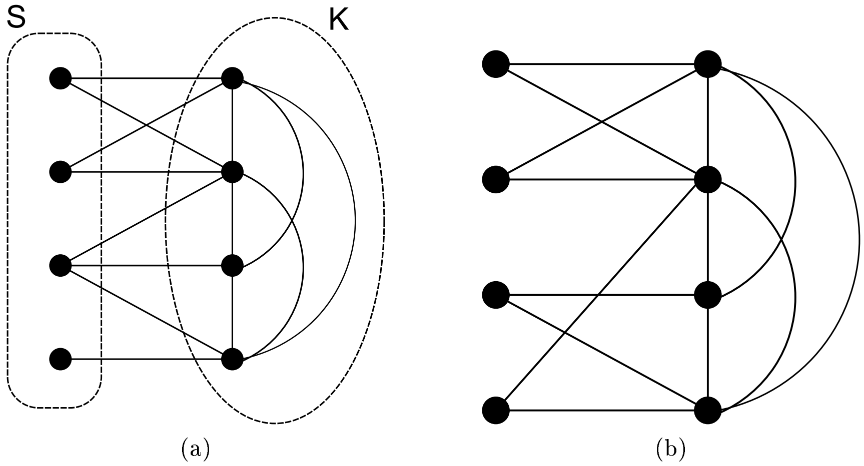


Figure 2: (a) A split graph and (b) A (2,4)-split graph

In the next section we present a polynomial time $\frac{2}{3}$-approximation algorithm to find the maxcut of a split graph. We use the next lemma due to Bodlaender *et al.* (2004):

Lemma 2.1. *In a complete graph K_n with n vertices the maximum cut has size $\lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor$ and the best partition (A, B) of V is anyone that have sizes $|A| = \lceil \frac{n}{2} \rceil$ and $|B| = \lfloor \frac{n}{2} \rfloor$ vertices.*

The following lemma gives an upper bound for the cardinality of the maxcut of a split graph $G = (S, K)$.

Lemma 2.2. *If $G = (S, K)$ is a split graph where $|K| = n$, $k = \frac{\sum_{v \in S} d(v)}{|S|}$ is the average degree of S and $[A, B]$ is an edge cut of G, then $|[A, B]| \leq \lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor + k|S|$.*

Proof. Observe that according to lemma 2.1, the best cut possible for K has size $\lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor$ and all the edges from S to K can be added to this cut. ☐

3 The Algorithm

Let $G = (S, K)$ be a split graph and $N(S)$ the neighborhood of the stable set S. We observe that $N(S) \subset K$. We make a feasible solution to MAXCUT problem consisting of a partition of the set $V(G)$ into two disjoint subsets A and B such that the edges of the cut are those connecting vertices from A to B.

Case 1: If $|N(S)| \leq \lfloor \frac{n}{2} \rfloor$, then put all the vertices of $N(S)$ in A, and we complete this part with other vertices of $K \backslash N(S)$, if it is necessary, in order to obtain $\lfloor \frac{n}{2} \rfloor$ vertices in part A and put all the vertices of S and the remaining vertices of K in part B.

Case 2: If $|N(S)| > \lfloor \frac{n}{2} \rfloor$, then evaluate the average degree of the vertices of S: $k = \frac{\sum_{v \in S} d(v)}{|S|}$. If $k|S| \leq 2\lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor$, then choose randomly $\lfloor \frac{n}{2} \rfloor$ vertices of $N(S)$ and put them in A. The remaining vertices of K put in B. For each vertex $v \in S$, if $|N(v) \cap A| \geq |N(v) \cap B|$ put v in B, otherwise put v in A. If $k|S| > 2\lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor$ then make $A = S$ and $B = K$.

Theorem 3.1. *If $G = (K, S)$ is a split graph, the maxcut approximation algorithm has ratio $\frac{2}{3}$.*

Proof. Note that in Case 1, when $|N(S)| \leq \lfloor \frac{n}{2} \rfloor$, we obtain a maximum cut of G, because all the edges from S to K are edges in the cut and we get the maximum number $\lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor$ of edges of K in the cut placing $\lfloor \frac{n}{2} \rfloor$ vertices into A and $\lceil \frac{n}{2} \rceil$ vertices into B. Therefore there are $k|S| + \lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor$ edges in the cut.

In Case 2, when $|N(S)| > \lfloor \frac{n}{2} \rfloor$, we have two subcases: $k|S| \leq 2\lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor$ and $k|S| > 2\lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor$.

If $k|S| \leq 2\lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor$ then the construction provides us a cut with at least $\frac{k|S|}{2} + \lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor$ edges on the graph that has $k|S| + \lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor$ edges as an upper bound for the maxcut. Then we have the approximation ratio $\frac{2}{3}$ since $k|S| \leq 2\lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor$ implies $\frac{k|S|}{2} \leq \lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor$ and then $3\frac{k|S|}{2} + 3\lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor \geq 2k|S| + 2\lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor$ what gives us $\frac{\frac{k|S|}{2} + \lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor}{k|S| + \lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor} \geq \frac{2}{3}$.

If $k|S| > 2\lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor$ then the construction provides us a cut with exactly $k|S|$ edges. The approximation ratio is still $\frac{2}{3}$ since $k|S| > 2\lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor$ implies $3k|S| > 2k|S| + 2\lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor$ and then $\frac{k|S|}{k|S| + \lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor} > \frac{2}{3}$. □

4 Full (k, n)-split graphs

A special subclass of (k, n)-split graphs is the class of full (k, n)-split graphs. A *full (k, n)-split graph* can be *simple* or *multiple*.

A graph is a *simple full (k, n)-split graph* if for each subset of k vertices of K there is a vertex $u \in S$ adjacent to all these k vertices. Thus $|S| = \binom{n}{k}$ and $\sum_{v \in S} d(v) = k\binom{n}{k}$. For $k \geq 2$ the number of edges with one endpoint in S and the other in K is at least twice the number of edges in K, that is, according to the algorithm, the cut obtained is $A = S$ and $B = K$. We show that this is the best cut possible, i.e., the maximum cut.

A graph is a *multiple full (k,n)-split graph* if for each subset of k vertices of K there is at least a vertex $u \in S$ adjacent to all these k vertices. For each subset $W = \{v_{i_1}, v_{i_2}, \cdots, v_{i_k}\}$ of K with $i_1 < i_2 < \cdots < i_k$ there is a set of vertices $U = \{u^1_{i_1 i_2 \cdots i_k}, \cdots, u^m_{i_1 i_2 \cdots i_k}\}$ in S such that $N(u^j_{i_1 i_2 \cdots i_k}) = W$ for all $u^j_{i_1 i_2 \cdots i_k} \in U$. The integer m is the *multiplicity* of the set W. Call $p = \binom{n}{k}$ the number of subsets of K with k vertices. Let $W_1, W_2, \cdots, W_p$ be all these subsets and $m_1, m_2, \cdots, m_p$ their respective multiplicities. We have $|S| = \sum_{1 \leq i \leq p} m_i$ and $\sum_{v \in S} d(v) = k|S|$. As we show below, the best cut possible is $[S, K]$.

Theorem 4.1. *For a multiple full (k, n)-split graph with $k \geq 2$ the partition (S, K) gives a maximum cut of size $k|S|$.*

Proof. Let $G = (S, K)$ be a multiple full (k,n)-split graph. Call $v_1, v_2, \cdots, v_n$ the vertices of K. For each subset $W = \{v_{i_1}, v_{i_2}, \cdots, v_{i_k}\}$ of K with $i_1 < i_2 < \cdots < i_k$ there is a vertex set $U = \{u^1_{i_1 i_2 \cdots i_k}, \cdots, u^m_{i_1 i_2 \cdots i_k}\}$ in S such that $N(u^j_{i_1 i_2 \cdots i_k}) = W$ for all $u^j_{i_1 i_2 \cdots i_k} \in U$. All the vertices of S have degree k and the number of edges with one endpoint in S and the other in K is $k|S|$. For each vertex $v_i \in K$ the number of edges connecting v_i with vertices of S is the sum $T_i = \sum_{1 \leq j \leq q} m_{i_j}$ where $q = \binom{n-1}{k-1}$ is the number of subsets of K with exactly k vertices such that v_i belongs to these subsets and m_{i_j} is the multiplicity of the set W_{i_j} which contains v_i. Obviously $T_i \geq q$ for each $v_i \in K$. Consider the original partition (S, K). If we place a vertex of S in $B = K$ we miss k edges in the cut. On the other hand,

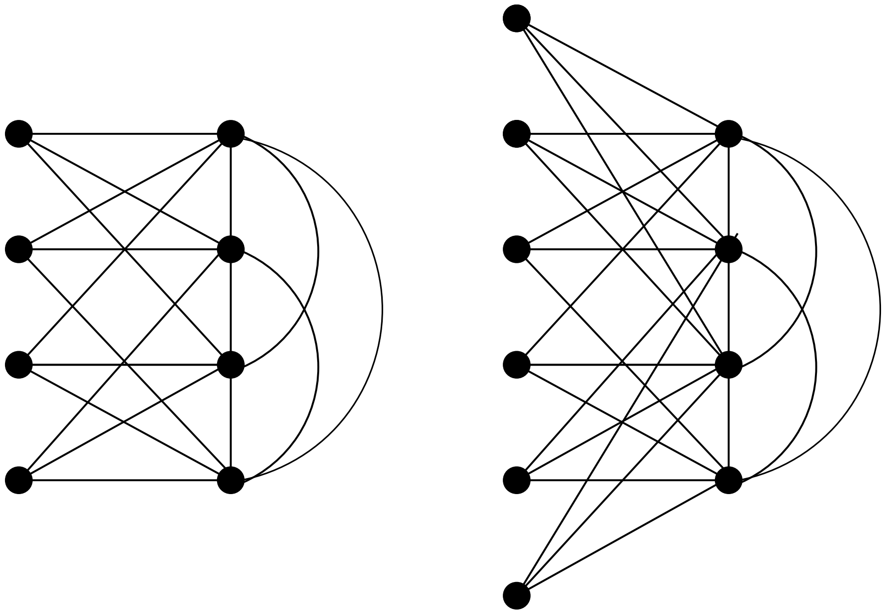


Figure 3: (a) A simple full (3,4)-split graph and (b) A multiple full (3,4)-split graph

if we place a vertex of K in $A = S$ we miss at least $\binom{n-1}{k-1} - n + 1$ edges in the cut. So any change in the partition (S, K) yields a new partition with a smaller number of edges in the cut. □

Corollary 4.2. *For a simple full (k, n)-split graph with $k \geq 2$, the partition (S, K) gives a maximum cut of size $k\binom{n}{k}$.*

Proof. Let $G = (S, K)$ be a simple full (k, n)-split graph. Call $v_1, v_2, \cdots, v_n$ the vertices of K. For each subset $W = \{v_{i_1}, v_{i_2}, \cdots, v_{i_k}\}$ of K with $i_1 < i_2 < \cdots < i_k$ there is a vertex $u_{i_1 i_2 \cdots i_k}$ in S such that $N(u_{i_1 i_2 \cdots i_k}) = W$. Of course all the vertices of S have degree k and the number of edges with one endpoint in S and the other in K is $k\binom{n}{k}$. For each vertex $v_i \in K$ the number of edges connecting v_i with vertices of S is the number of subsets of K with exactly k vertices such that v_i belongs to these subsets, that is $\binom{n-1}{k-1}$. So the degree of v_i is $d(v_i) = \binom{n-1}{k-1} + n - 1$ for all $i \in \{1, 2, \cdots, n\}$. Consider the original partition (S, K). If we place a vertex of S in $B = K$, then we miss k edges in the cut. On the other hand, if we place a vertex of K in $A = S$, then we miss $\binom{n-1}{k-1} - n + 1$ edges in the cut. So any change in the partition (S, K) yields a new partition with a smaller number of edges in the cut. □

References

BODLAENDER, H. L.; FIGUEIREDO, C. M. D.; GUTIERREZ, M.; KLOKS, T.; NIEDERMEIER, R. Simple max-cut for split-indifference graphs and graphs with few p

4's. In: **Experimental and Efficient Algorithms**. [S.l.]: Springer, 2004. p. 87–99.

BODLAENDER, H. L.; JANSEN, K.; FORSCHUNGSGEMEINSCHAFT, D. On the complexity of the maximum cut problem. **Nord. J. Comput.**, v. 7, n. 1, p. 14–31, 2000.

FEIGE, U.; KARPINSKI, M.; LANGBERG, M. Improved approximation of max-cut on graphs of bounded degree. **J. Algorithms**, Academic Press, Inc., Duluth, MN, USA, v. 43, n. 2, p. 201–219, may 2002. ISSN 0196-6774. Disponível em: <http://dx.doi.org/10.1016/S0196-6774(02)00005-6>.

GOEMANS, M. X.; WILLIAMSON, D. P. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. **Journal of the ACM (JACM)**, ACM, v. 42, n. 6, p. 1115–1145, 1995.

KARP, R. M. **Reducibility among combinatorial problems**. [S.l.]: Springer, 1972.

VEGA, W. F. de la. Max-cut has a randomized approximation scheme in dense graphs. **Random Struct. Algorithms**, John Wiley & Sons, Inc., New York, NY, USA, v. 8, n. 3, p. 187–198, may 1996. ISSN 1042-9832. Disponível em: <http://dx.doi.org/10.1002/(SICI)1098-2418(199605)8:3<187::AID-RSA3>3.0.CO;2-U>.