Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Discrete Mathematics 28 (2007) 265–272
Electronic Notes in DISCRETE MATHEMATICS
www.elsevier.com/locate/endm

Linear time algorithms for finding sparsest cuts in various graph classes

Paul S. Bonsma [1]

Department of Applied Mathematics
University of Twente
Enschede, the Netherlands

Abstract

$[S, \overline{S}]$ denotes the set of edges with exactly one end vertex in S. The *density* of an edge cut $[S, \overline{S}]$ is $|[S, \overline{S}]|/(|S||\overline{S}|)$. A *sparsest cut* is an edge cut with minimum density. We characterize the sparsest cuts for unit interval graphs, complete bipartite graphs and cactus graphs. For all of these classes, the characterizations lead to linear time algorithms to find a sparsest cut.

Keywords: Sparsest cut, minimum ratio cut, linear time, graph class.

1 Introduction

For basic graph theoretic terms that are not explained here we refer the reader to [4]. Let $G = (V, E)$ be a graph. For two disjoint non-empty sets $S \subset V$ and $T \subset V$, $[S, T]$ denotes the set of edges of G with one end vertex in S and one end vertex in T. An *edge cut* of G is a set $M \subseteq E$ such that $M = [S, \overline{S}]$ for some $S \subset V$, $S \neq \emptyset$, where $\overline{S}$ denotes $V \backslash S$. Edge cuts will also be called

[1] Email: `bonsma@math.utwente.nl`

1571-0653/$ – see front matter © 2007 Published by Elsevier B.V.
doi:10.1016/j.endm.2007.01.039

cuts for short. The *density* of an edge set $[S,T]$ is defined as $d(S,T) = \frac{|[S,T]|}{|S||T|}$. An edge cut $[S,\overline{S}]$ of G with minimum density is called a *sparsest cut* of G.

Finding a sparsest cut or the density of a sparsest cut is $\mathcal{NP}$-hard [6]. Accordingly, it is unlikely that there are straightforward methods to prove that a cut is a sparsest cut. A lot of study has been done on approximation algorithms for finding sparsest cuts and generalizations thereof (see [5] for a good introduction). We ask a different question: for which graph classes can the sparsest cuts be found efficiently? Not much is known about this: in [6], a linear time algorithm is given for trees. In [1], sparsest cuts are characterized for Cartesian product graphs $G \times H$. This leads to a polynomial time algorithm when sparsest cuts of the factors G and H can be found in polynomial time.

In this paper, we characterize sparsest cuts for three graph classes: unit interval graphs (for the definition see below), cactus graphs and complete bipartite graphs. *Cactus graphs* are connected graphs in which every edge is part of at most one cycle (cactus graphs generalize trees and cycles). These results have appeared previously in [2]. These graph classes are very restricted and well-structured, but surprisingly, two of the proofs are still non-trivial. For all of these classes, the characterizations lead to linear time algorithms. Thus the result for cactus graphs generalizes the result for trees in [6].

In Section 2 we study unit interval graphs and cactus graphs. A graph G is a *unit interval graph* (UIG) if a function $I : V(G) \to \mathbb{R}$ exists such that $uv \in E(G)$ if and only if $I(u) - 1 \leq I(v) \leq I(u) + 1$. I is called a *unit interval representation* (UIR) of G. We show that every UIG G with UIR I has a sparsest cut $[S,\overline{S}]$ such that for all $u \in S$, $v \in \overline{S}$, $I(u) \leq I(v)$. This is then used to construct a linear time algorithm to find sparsest cuts for UIGs. To prove the result, we develop some basic but useful techniques and lemmas for comparing densities in a graph.

It is easy to show that a sparsest cut in a cactus always contains one or two edges. This is used to give a linear time algorithm to find a sparsest cut for these graphs.

In Section 3 we study complete bipartite graphs $K_{m,n}$ with $m \leq n$ and $n \geq 2$. We show that $d(K_{m,n}) = \min\{\frac{1}{2}, \frac{m}{m+n-1}\}$, and construct the corresponding cuts. This obviously gives a linear time algorithm that either returns a sparsest cut, or shows that the input graph is not complete bipartite.

For complete graphs and disconnected graphs finding a sparsest cut is trivial. To summarize, the current results and the results in [1] show that sparsest cuts can be found efficiently for complete graphs, disconnected graphs, complete bipartite graphs, cactus graphs, UIGs, and products of (products of) these.

An interesting question for future research is for which other classes the sparsest cuts can efficiently be found, and for which classes the problem is $\mathcal{NP}$-hard.

2 Sparsest cuts in unit interval graphs and cactus graphs

For the proofs in this section, we need to express a density $d(S,T)$ as a weighted average of densities between subsets of S and T.

Proposition 2.1 *If A, B and C are disjoint non-empty subsets of $V(G)$, then $d(A, B \cup C) = \frac{d(A,B)|B|+d(A,C)|C|}{|B|+|C|}$.*

Definition 2.2 Let G be a graph with sparsest cut $[S, \overline{S}]$. Let $A, B \subseteq V(G)$ with $A \cap B = \emptyset$. The *normalized density* between A and B is $e(A,B) = d(A,B) - d(S, \overline{S})$.

Normalized densities can also be expressed as weighted averages:

Proposition 2.3 *If A, B and C are disjoint non-empty subsets of $V(G)$, then $e(A, B \cup C) = \frac{e(A,B)|B|+e(A,C)|C|}{|B|+|C|}$.*

The following simple lemma is the key to our approach.

Lemma 2.4 *If $[A \cup B, C]$ is a sparsest cut of G, with A and B disjoint and non-empty, then $e(A,B) \geq 0$. If $e(A,B) = 0$, then $[A, B \cup C]$ or $[B, A \cup C]$ is also a sparsest cut of G.*

Proof. $[A \cup B, C]$ is a sparsest cut so $e(A \cup B, C) = 0$. Since this is a weighted average of $e(A,C)$ and $e(B,C)$ (Observation 2.3), we may assume $e(A,C) \leq 0$ and $e(B,C) \geq 0$. If $e(A,B) < 0$, then

$$e(A, B \cup C) = \frac{e(A,B)|B| + e(A,C)|C|}{|B| + |C|} < 0,$$

a contradiction. This shows that $e(A,B) \geq 0$. If $e(A,B) = 0$, then similarly we find that $e(A, B \cup C) = 0$, so $[A, B \cup C]$ is a sparsest cut. □

Corollary 2.5 *If $[S,T]$ is a sparsest cut in a connected graph G, then $G[S]$ and $G[T]$ are connected.*

Cactus graphs From this corollary, we obtain a characterization of the sparsest cuts for cactus graphs.

Proposition 2.6 *If M is a sparsest cut of a cactus G, then $|M| = 1$, or M contains exactly two edges, which are part of the same cycle.*

Obviously it follows that for cactus graphs, a sparsest cut can be found in polynomial time. We shortly sketch a linear time algorithm for a graph G with $|V(G)| = n$. The blocks of G can be found in linear time [7], so we can decide in linear time if it is a cactus. For every block B and vertex $v \in V(B)$, let $w(B, v)$ be the number of vertices in the component of $G - E(B)$ that contains v. These values can be calculated in linear time: first assign $w'(v) := 1$ for every vertex v. Now repeatedly consider a block B that contains at most one cut vertex u of G. For all $v \in V(B)$ with $v \neq u$, $w(B, v) = w'(v)$. $w(B, u)$ follows from $\sum_{v \in V(B)} w(B, v) = n$. Contract B into a single vertex u, and set $w'(u) := \sum_{v \in V(B)} w'(v)$. Repeat this procedure until no blocks remain. Now for every cycle C, it can be checked that the edge cut with the smallest density, that cuts C, can be calculated in time $O(|V(C)|)$, using the values $w(C, v)$. In addition, for every bridge the corresponding density can now be calculated in constant time. By Proposition 2.6, one of these cuts is a sparsest cut.

Unit interval graphs In order to give a short expression for the form of a sparsest cut of UIGs, we use the following definition.

Definition 2.7 Let $I : V(G) \to \mathbb{R}$ be a UIR for the graph G. If A and B are disjoint non-empty subsets of $V(G)$, we write $A \prec_I B$ if for all $u \in A$ and $v \in B$, $I(u) \leq I(v)$ holds.

Let G be a UIG with UIR I. We show that G has a sparsest cut $[S, \overline{S}]$ such that $S \prec_I \overline{S}$. This is done by considering an arbitrary cut $[S, T]$, and partitioning S and T into $S_1, \ldots, S_k$ resp. $T_1, \ldots, T_l$ with $k - 1 \leq l \leq k$ such that $S_1 \prec_I T_1 \prec_I S_2 \prec_I \ldots \prec_I T_{k-1} \prec_I S_k$, and in addition $S_k \prec_I T_k$ if $l = k$. For a given cut $[S, T]$ and UIR I of G, a partition of S and T into non-empty subsets with this property is called an *I-partition* of S and T. Observe that w.l.o.g. we can always find an I-partition.

We show that if $k > 1$, then we can reassign these subsets into disjoint non-empty subsets S' and T' with $S' \cup T' = V(G)$, such that $d(S', T') \leq d(S, T)$.

Lemma 2.8 *Let $I : V(G) \to \mathbb{R}$ be a UIR for the graph G. If $A, B, C \subseteq V(G)$ with $A \prec_I B \prec_I C$, then $d(B, A \cup C) \geq d(A, C)$.*

Theorem 2.9 *Let $I : V(G) \to \mathbb{R}$ be a UIR for the graph G. G has a sparsest cut $[S, T]$ such that $S \prec_I T$.*

Proof. Consider a sparsest cut $[S,T]$ and I-partition $\{S_1,\ldots,S_k\}$ and $\{T_1,\ldots,T_l\}$ of S and T, that has $k+l$ minimum, among all such sparsest cuts and I-partitions. We use the following shorthand notation: $S_{i\ldots j} = S_i \cup \ldots \cup S_j$, and $T_{i\ldots j} = T_i \cup \ldots \cup T_j$.

If $k=1$, then $S = S_1 \prec_I T_1 = T$, and we have found the desired sparsest cut. Otherwise, we distinguish two cases: $l = k-1$, and $l = k \geq 2$.

Case 1: $l = k-1$.
For all $1 \leq t \leq k-1$, $e(S_{1\ldots t}, S_{t+1\ldots k}) \geq 0$ (Lemma 2.4). If this is an equality, then $[S_{1\ldots t}, S_{t+1\ldots k} \cup T_{1\ldots k-1}]$ or $[S_{t+1\ldots k}, S_{1\ldots t} \cup T_{1\ldots k-1}]$ is also a sparsest cut (Lemma 2.4). The first cut has an I-partition with $2t < 2k-1 = k+l$ classes, and the second cut has an I-partition with $2(k-t) < 2k-1 = k+l$ classes, both contradictions with our choice of $[S,T]$. We conclude that $e(S_{1\ldots t}, S_{t+1\ldots k}) > 0$ for every $1 \leq t \leq k-1$.

Since $S_{1\ldots t} \prec_I T_t \prec_I S_{t+1\ldots k}$ and $S_{1\ldots t} \cup S_{t+1\ldots k} = S$, it follows from Lemma 2.8 that $e(T_t, S) \geq e(S_{1\ldots t}, S_{t+1\ldots k}) > 0$, for all $1 \leq t \leq k-1$. Since

$$e(S,T) = \frac{e(T_1,S)|T_1| + \ldots + e(T_{k-1},S)|T_{k-1}|}{|T_1| + \ldots + |T_{k-1}|},$$

we have $e(S,T) > 0$, a contradiction with the fact that $[S,T]$ is a sparsest cut. This concludes the case $l = k-1$.

Case 2: $l = k \geq 2$.
We again have $e(T_t, S) > 0$ for all $1 \leq t \leq k-1$ (see the previous case), but it is possible that $e(T_k, S) < 0$, so we cannot immediately obtain a contradiction this way.

First we show that for every $2 \leq t \leq k$,

$$e(S_t, T) > e(T_{1\ldots t-1}, S)\frac{|S|}{|T|}, \tag{1}$$

and for every $1 \leq t \leq k-1$,

$$e(T_t, S) > e(S_{t+1\ldots k}, T)\frac{|T|}{|S|}. \tag{2}$$

For a fixed t with $2 \leq t \leq k$, we denote $T_L = T_{1\ldots t-1}$ and $T_H = T_{t\ldots k}$. We showed that $e(T_i, S) > 0$ for all $i < k$, so we have $e(T_L, S) = \alpha > 0$ ($e(T_L,S)$ is a weighted average of $e(T_i,S)$ for $i = 1,\ldots,t-1$). Since $0 = e(T_L \cup T_H, S) = \frac{\alpha|T_L| + e(T_H,S)|T_H|}{|T_L|+|T_H|}$, we have $e(T_H,S) = -\alpha\frac{|T_L|}{|T_H|}$. Now we consider the cut $[T_H, V(G)\backslash T_H]$. Since $0 \leq e(T_H, V(G)\backslash T_H) = \frac{e(T_H,S)|S| + e(T_H,T_L)|T_L|}{|S|+|T_L|}$, we have $e(T_H,T_L) \geq -e(T_H,S)\frac{|S|}{|T_L|} = \alpha\frac{|S|}{|T_H|}$. Finally, using Lemma 2.8 and $T_L \prec_I S_t \prec_I T_H$ we obtain $e(S_t,T) \geq e(T_L,T_H) \geq \alpha\frac{|S|}{|T_H|} > e(T_L,S)\frac{|S|}{|T|}$. By symmetry (since $l = k$), (2) can be proved the same way.

Using (1) and (2), we now prove by induction on i that for all $1 \le i \le k-1$,

$$e(T_{1\dots i}, S) > e(S_{i+1\dots k}, T)\frac{|T|}{|S|}. \tag{3}$$

If $i = 1$, then (3) is equal to (2) for $t = 1$. If $i > 1$ then our induction hypothesis is that $e(T_{1\dots i-1}, S) > e(S_{i\dots k}, T)\frac{|T|}{|S|}$. When we combine this with (1) we get $e(S_i, T) > e(T_{1\dots i-1}, S)\frac{|S|}{|T|} > e(S_{i\dots k}, T)$. Since $e(S_{i\dots k}, T)$ is a weighted average of $e(S_i, T)$ and $e(S_{i+1\dots k}, T)$, it follows that $e(S_{i\dots k}, T) > e(S_{i+1\dots k}, T)$. We combine this with the induction hypothesis: $e(T_{1\dots i-1}, S) > e(S_{i\dots k}, T)\frac{|T|}{|S|} > e(S_{i+1\dots k}, T)\frac{|T|}{|S|}$. From (2) we see that $e(S_{i+1\dots k}, T)\frac{|T|}{|S|}$ is also a lower bound for $e(T_i, S)$. Since $e(T_{1\dots i}, S)$ is a weighted average of $e(T_{1\dots i-1}, S)$ and $e(T_i, S)$, it follows that $e(T_{1\dots i}, S) > e(S_{i+1\dots k}, T)\frac{|T|}{|S|}$, which concludes the induction proof. Using (3) resp. (1), we obtain a contradiction for the case $l = k$: $e(T_{1\dots k-1}, S) > e(S_k, T)\frac{|T|}{|S|} > e(T_{1\dots k-1}, S)$.

We showed that both cases with $k > 1$ lead to a contradiction, so with our choice of S and T, $k = 1$, and thus $S \prec_I T$. □

UIGs can be recognized in linear time, and a UIR can be found in linear time [3]. It follows that for UIGs, sparsest cuts can be found in linear time: Number the vertices $v_1, \dots, v_n$, according to the linear order given by the UIR (w.l.o.g. $I(u) \neq I(v)$ for all u, v). Now we only have to evaluate the densities of $[\{v_1, \dots, v_i\}, \{v_{i+1}, \dots, v_n\}]$, for $i = 1, \dots, n-1$ (Theorem 2.9). Note that the number of edges in the i-th cut can be deduced in time $O(d(v_i))$ from the number of edges in the $(i-1)$-th cut. Therefore the algorithm has complexity $O(|V| + |E|)$.

3 Sparsest cuts in complete bipartite graphs

In this section we give an explicit expression for the density of a sparsest cut of $K_{m,n}$, and in the proof of Theorem 3.1 construct all corresponding cuts. A completely different proof of this theorem (by induction over $n - m$) appears in [8].

Theorem 3.1 *If $[S, \overline{S}]$ is a sparsest cut of $K_{m,n}$ with $m \le n$ and $n \ge 2$, then $d(S, \overline{S}) = \min\{\frac{1}{2}, \frac{m}{n+m-1}\}$.*

Proof. First we give an expression for the density of an arbitrary edge cut $[S, \overline{S}]$ of $K_{m,n}$. Let $\{A, B\}$ be a bipartition of the vertices, with $|A| = m$,

$|B| = n$. If $|S \cap A| = x$ and $|S \cap B| = y$, then

$$d(S, \overline{S}) = \frac{|[S \cap A, \overline{S} \cap B]| + |[S \cap B, \overline{S} \cap A]|}{|S||\overline{S}|} = \frac{x(n-y) + y(m-x)}{(x+y)(n+m-x-y)}.$$

We denote this function by $d(x, y)$. So we want to find the minimum of $d(x, y)$ over all integer values of x and y with $0 \leq x \leq m$, $0 \leq y \leq n$, and $1 \leq x + y \leq n + m - 1$. Because of the symmetry, we only have to consider values of x and y with $1 \leq x+y \leq (n+m)/2$. First we analyze $d(x, y)$ without considering the integrality constraints for x and y.

Consider combinations of x and y with $x + y = c$ for a constant c. For fixed values of n, m and c we will denote this function as $d_c(x) = d(x, c - x)$. The denominator of $d_c(x)$ is a constant, so $d_c(x)$ is minimum when the numerator is minimum. Substituting $y = c - x$ gives

$$x(n - c + x) + (c - x)(m - x) = 2x^2 + (n - m - 2c)x + mc.$$

This is minimum when $x = \frac{c}{2} - \frac{n-m}{4}$, and thus $y = \frac{c}{2} + \frac{n-m}{4}$. If $\frac{n-m}{4} \geq \frac{c}{2}$, then this value for x is negative, so within our range the minimum of $d_c(x)$ occurs at $x = 0$ and $y = c$.

We substitute these values of x and y into $d(x, y)$, to find the value of c with $1 \leq c \leq (n + m)/2$ for which the minimum is attained. First suppose $(n-m) \leq 2$, and write $e = \frac{n-m}{4}$. so for all values of c with $1 \leq c \leq (n+m)/2$, we can substitute $x = \frac{c}{2} - e$ and $y = \frac{c}{2} + e$.

$$d(x, y) = \frac{(\frac{c}{2} - e)(n - \frac{c}{2} - e) + (\frac{c}{2} + e)(m - \frac{c}{2} + e)}{c(n + m - c)} =$$

$$\frac{\frac{c}{2}(n + m - c) - e(n - m) + 2e^2}{c(n + m - c)} = \frac{z - 2e^2}{2z},$$

with $z = c(n + m - c)/2$. Note that z is strictly positive for our choices of c. If $(n - m)/4 = e = 0$, then the minimum of $d(x, y)$ is $\frac{1}{2}$, which occurs for all $x = \frac{c}{2}$ and $y = \frac{c}{2}$, with $1 \leq c \leq (n + m)/2$.

If $e \neq 0$, then we conclude from the expression above that $d(x, y)$ attains its minimum when $z = c(n + m - c)/2$ is as small as possible, so $c = 1$. We conclude that if $0 < n - m \leq 2$, $d(x, y)$ is minimum for $x = \frac{1}{2} - \frac{n-m}{4}$ and $y = \frac{1}{2} + \frac{n-m}{4}$.

Now suppose $(n - m) \geq 2$. The same reasoning as above shows that when $c \geq (n-m)/2$, $d(x, y)$ is minimum when c is minimum, so $c = (n-m)/2$, and $x = 0$ and $y = c$. When $1 \leq c \leq (n-m)/2$, we showed that $d_c(x)$ is minimum when $x = 0$ and $y = c$. For all such pairs, $d(x, y) = \frac{cm}{c(n+m-c)} = \frac{m}{n+m-c}$. This function is minimum when c is minimum, so $c = 1$. We conclude that when

$(n-m) \geq 2$, $d(x,y)$ is minimum for $x=0$, $y=1$. Now we will consider the integrality constraints for x and y.

If $n-m \geq 2$, the minimum occurs at integer values of x and y ($x=0$, $y=1$), so the best density is $m/(n+m-1)$.

If $n-m=0$, then the minimum occurs whenever $x=y$, so the best density is $\frac{1}{2}$.

Now consider the case $n-m=1$. For fixed $c=x+y$, we know that $d_c(x)$ is a degree two polynomial, with minimum at $x=\frac{c}{2}-\frac{1}{4}$. Rounding to the closest values of x and y with $x+y=c$ gives $x=\frac{c}{2}$ and $y=\frac{c}{2}$ when c is even, and $x=\frac{c}{2}-\frac{1}{2}$ and $y=\frac{c}{2}+\frac{1}{2}$ when c is odd. Since $d_c(x)$ is a polynomial of degree two, this way of rounding gives the best integer values of $d_c(x)$. Now we calculate the densities for these integer values. If $x=y$, then $d(x,y)=\frac{x(n-x)+x(n-1-x)}{2x(2n-1-2x)}=\frac{1}{2}$. If $y=x+1$, then $d(x,y)=\frac{x(n-x-1)+(x+1)(n-1-x)}{(2x+1)(2n-1-2x-1)}=\frac{1}{2}$. We conclude that $\frac{1}{2}$ is the minimum density, and that it is attained by many combinations of x and y, one for every value of $x+y$. □

References

[1] P. S. Bonsma. Sparsest cuts and concurrent flows in product graphs. *Discrete Applied Mathematics*, 136(2-3):173–182, 2004.

[2] P. S. Bonsma. *Sparse cuts, matching-cuts and leafy trees in graphs*. PhD thesis, University of Twente, Enschede, the Netherlands, 2006.

[3] D. G. Corneil. A simple 3-sweep LBFS algorithm for the recognition of unit interval graphs. *Discrete Applied Mathematics*, 138(3):371–379, 2004.

[4] R. Diestel. *Graph Theory*. Number 173 in Graduate Texts in Mathematics. Springer-Verlag, New York, 1997.

[5] T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM*, 46(6):787–832, 1999.

[6] D. W. Maluta and F. Shahrokhi. Sparsest cuts and bottlenecks in graphs. *Discrete Applied Mathematics*, 27:113–123, 1990.

[7] R. Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160, 1972.

[8] M. Visser. Finding sparsest cuts and balanced vertex cuts in different graph classes. Master's thesis, University of Twente, Enschede, the Netherlands, 2002.