



HAL
open science

Groupage sur le chemin pour borner la largeur de coupe

Jean-Claude Bermond, Michel Cosnard, David Coudert, Frédéric Havet

► To cite this version:

Jean-Claude Bermond, Michel Cosnard, David Coudert, Frédéric Havet. Groupage sur le chemin pour borner la largeur de coupe. AlgoTel 2023 - 25èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications, May 2023, Cargese, France. pp.4. hal-04076999v2

HAL Id: hal-04076999

<https://inria.hal.science/hal-04076999v2>

Submitted on 26 Apr 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Groupage sur le chemin pour borner la largeur de coupe[†]

J-C. Bermond¹ et M. Cosnard¹ et D. Coudert¹ et F. Havet¹

¹ Université Côte d'Azur, Inria, CNRS, I3S, Sophia Antipolis, France

La largeur de coupe (cutwidth) est un paramètre utilisé dans de nombreux problèmes d'ordonnement linéaire (layout). Calculer la largeur de coupe est un problème NP-complet, mais il est possible de concevoir des algorithmes de branch-and-bound efficaces si on dispose de bonnes bornes inférieures pour couper des branches lors de l'exploration. Savoir évaluer rapidement de bonnes bornes dans chaque nœud de l'arbre de recherche est donc crucial. Dans cet article, nous donnons de nouvelles bornes basées sur différents paramètres de densité du graphe. Nous donnons aussi une borne inférieure basée sur le groupage de requêtes sur le chemin.

Mots-clefs : largeur de coupe (cutwidth), groupage, densité

1 Introduction

Graph layout problems are a particular class of combinatorial optimization problems whose goal is to find a layout of an input graph in such way that a certain objective cost is optimized. A *layout* of a graph G is a linear ordering $\sigma = (v_1, \dots, v_n)$ of the vertices of G . A large amount of relevant problems in different areas can be formulated as graph layout problems (see the survey [DPS02] and references therein).

The *edge-cut* of a set $X \subseteq V$ is the set of edges with one endvertex in X and one endvertex in $\bar{X} = V \setminus X$. Its size is denoted by $ec(X, \bar{X})$. Let $X_i = \{v_1, \dots, v_i\}$. The *width* $w(G, \sigma)$ of a graph G with respect to the layout σ is the maximum of $ec(X_i, \bar{X}_i)$ over all $i \in \{1, \dots, n-1\}$. The *cutwidth* of a graph G , denoted by $cw(G)$, is the minimum width of a layout : $cw(G) = \min\{w(G, \sigma) \mid \sigma \text{ layout of } G\}$.

The cutwidth was first used as a theoretical model for the number of channels in an optimal layout of a circuit in the seventies. In general, the cutwidth of a graph times the order of the graph gives a measure of the area needed to represent the graph in a VLSI layout when vertices are laid out in a row. More recent applications of this problem include network reliability, automatic graph drawing, information retrieval and as a subroutine for the cutting plane algorithm to solve the traveling salesman problem. Due to its natural definition, the cutwidth has various applications in computer science : whenever data is expected to be roughly linearly ordered and dependencies or connections are local, the cutwidth of the corresponding graph is expected to be small. However, computing the cutwidth of graphs is NP-hard [DPS02] in general. Fixed-parameter algorithms have also been designed (see [GPR⁺19]). Furthermore efficient branch-and-bound algorithms have been designed (see [MPDP13]) but they need to have good lower bounds for the cutwidth, and in particular bounds that can be quickly evaluated at each node of the branch-and-bound tree. The quality of these bounds have a strong impact on the performances of the branch-and-bound algorithms.

In this paper, we first review previous lower bounds on the cutwidth of a graph G . Next, we introduce new lower bounds based on the maximum, minimum or maximum average degree of G . Note that all these parameters can be computed in polynomial time. Moreover, the bounds are mostly incomparable and hence are complementary. Finally, we give a new lower bound based on the results obtained for the *Maximum All request Path Grooming problem* [BCCP06], a problem also known as the *Call Control Problem in Path Networks* in [AAAE07].

Consider the path $P_\sigma = v_1 \cdots v_n$ associated to σ . For each edge $e \in E(G)$, the *request associated to e by σ* , denoted by $R_\sigma(e)$, is the subpath of P_σ whose endvertices are those of e . Let $\mathcal{R}_\sigma(G)$ be the set of

[†]This work has been supported by IDEX UCA^{JEDI} (ANR-15-IDEX-01) and the ANR project MULTIMOD (ANR-17-CE22-0016).

requests associated by σ to the edges of G . Observe that the load of the edge $v_i v_{i+1}$ in P_σ under $\mathcal{R}_\sigma(G)$ denoted $l(v_i v_{i+1})$ is the number of edges $v_j v_\ell$ of G with $j \leq i$ and $i+1 \leq \ell$, that is $ec(X_i, \bar{X}_i)$. Hence $w(G, \sigma)$ is the maximum load of an edge of P_σ under $\mathcal{R}_\sigma(G)$.

The *grooming factor* C is the maximum load that is allowed on the edges of P . Given a grooming factor C , the maximum number of requests that can be satisfied (or groomed) together on P such that the load of any edge is at most C is denoted by $T(C, n)$. We show that, if G is a graph with n vertices and m edges, then $cw(G) \geq C^*(m, n) = \min\{C \mid m \leq T(C, n)\}$.

2 Previous lower bounds

Several lower bounds on the cutwidth of graphs have been proposed in [DPS02, JM92, Klo10, MPDP13].

Bound based on minimal edge-cuts. From the definition of the cutwidth, it is obvious that the minimum size of an edge-cut between any pair u, v of vertices of the graph, which we denote by $\kappa'(u, v)$, is a lower bound on its cutwidth (see [DPS02] for more details). So, we get :

$$cw(G) \geq \max\{\kappa'(u, v) \mid u, v \in V, u \neq v\} \quad (1)$$

Bound based on spectral property. Let λ_2 denote the second smallest eigenvalue of the Laplacian matrix associated with the graph. The following lower bound is due to [JM92] :

$$cw(G) \geq \frac{\lambda_2}{n} \lfloor \frac{n}{2} \rfloor \lceil \frac{n}{2} \rceil \quad (2)$$

Bound based on the degeneracy. A graph G is *k-degenerate* if every subgraph H of G has minimum degree at most k . Equivalently, G is *k-degenerate*, if there exists a layout (v_1, \dots, v_n) such that for all $i \in \{2, \dots, n\}$ every vertex v_i has at most k neighbours in $\{v_1, \dots, v_{i-1}\}$. The *degeneracy* of G , denoted by $\delta^*(G)$, is the smallest k such that G is *k-degenerate*. The following lower bound is proved in [Klo10].

$$cw(G) \geq \frac{1}{4} \delta^*(G)(\delta^*(G) + 2) \quad (3)$$

3 New lower bound based on degrees

Bound based on the maximum degree. Let $\Delta(G)$ denote the maximum degree of G . Let v_i be a vertex with degree $\Delta(G)$. One set among $\{v_1, \dots, v_{i-1}\}$ and $\{v_{i+1}, \dots, v_n\}$ contains at least half of the neighbours of v_i . Consequently,

$$cw(G) \geq \left\lceil \frac{\Delta(G)}{2} \right\rceil \quad (4)$$

Bound based on the minimum degree. Let $\delta(G)$ denote the minimum degree of G . Set $k = \lfloor \delta(G)/2 \rfloor + 1$ and let $X_k = \{v_1, \dots, v_k\}$. Every vertex in X_k has at most $k-1 = \lfloor \delta(G)/2 \rfloor$ neighbours in X_k and so at least $\lceil \delta(G)/2 \rceil$ neighbours in \bar{X}_k . Hence $ec(X_k, \bar{X}_k) \geq k \lceil \delta(G)/2 \rceil = (\lfloor \delta(G)/2 \rfloor + 1) \lceil \delta(G)/2 \rceil$. Thus

$$cw(G) \geq (\lfloor \delta(G)/2 \rfloor + 1) \lceil \delta(G)/2 \rceil \geq \frac{1}{4} \delta(G)(\delta(G) + 2) \quad (5)$$

As $\delta^*(G) \geq \delta(G)$, the bound of Eq. (3) is better than the bound of Eq. (5). However by definition of degeneracy, there is a subgraph H of G such that $\delta(H) = \delta^*(G)$. Since $cw(G) \geq cw(H)$, Eq. (5) implies the bound of Eq. (3) (with a simpler proof than the original one [Klo10]).

Bound based on the average degree. Let $\sigma = (v_1, \dots, v_n)$ be an ordering of $V(G)$ and let $X_i = \{v_1, \dots, v_i\}$. For each vertex v_i , let $d^+(v_i)$ be the number of edges (v_i, v_j) with $j > i$ (*forward edges* from v_i).

Proposition 1 $w(G, \sigma) \geq \frac{1}{n-1} \sum_{i=1}^{n-1} d^+(v_i) \left(\frac{d^+(v_i) + 1}{2} \right)$

Groupage sur le chemin pour borner la largeur de coupe

Proof. We use a classical trick in grooming. We first compute the total load of G on $P_\sigma = v_1v_2 \dots v_n$, also called *sum cut* of the layout σ [DPS02], which is $L(G, \sigma) = \sum_{i=1}^{n-1} l(v_i v_{i+1}) = \sum_{i=1}^{n-1} \text{ec}(X_i, \bar{X}_i)$. To compute $L(G, \sigma)$, we note that the load of the forward edges of G from v_i is at least $d^+(v_i)$ on the edge $v_i v_{i+1}$, at least $d^+(v_i) - 1$ on the edge $v_{i+1} v_{i+2}$ and so on. Therefore the load due to the edges from v_i is at least $\frac{1}{2} d^+(v_i)(d^+(v_i) + 1)$ and the total load is at least $\frac{1}{2} \sum_{i=1}^{n-1} d^+(v_i)(d^+(v_i) + 1)$. Then we note that, as the path has $n - 1$ edges, there exists an edge with load at least $\frac{L(G, \sigma)}{n-1}$ and so $w(G, \sigma) \geq \frac{L(G, \sigma)}{n-1}$. \square

The *average degree* of a graph G with m edges and n vertices is $\text{Ad}(G) = 2m/n$.

If a function $f(x)$ is convex and if $\sum_{i=1}^{n-1} x_i = m$, then the minimum of $\sum_{i=1}^{n-1} f(x_i)$ is attained when all the x_i are equal to $m/(n-1)$. Applying this argument to the convex function $x \mapsto x(x+1)$ with $x_i = d^+(v_i)$ and using the fact that $\sum_{i=1}^{n-1} d^+(v_i) = m$, we get :

$$\text{cw}(G) \geq \frac{1}{2} \frac{m}{n-1} \left(\frac{m}{n-1} + 1 \right) = \frac{1}{8} \left(1 + \frac{1}{n-1} \right)^2 \text{Ad}(G) \left(\text{Ad}(G) + 2 - \frac{2}{n} \right) \quad (6)$$

The *maximum average degree* of G is $\text{Mad}(G) = \max\{\text{Ad}(H) \mid H \text{ subgraph of } G\}$. Observe that this parameter can be computed in polynomial time. By definition of Mad , there is a subgraph H of G such that $\text{Ad}(H) = \text{Mad}(G)$. Since $\text{cw}(G) \geq \text{cw}(H)$, Eq. (6) directly implies

$$\text{cw}(G) \geq \frac{1}{8} \left(1 + \frac{1}{n-1} \right)^2 \text{Mad}(G) \left(\text{Mad}(G) + 2 - \frac{2}{n} \right) \quad (7)$$

The bounds given by Eq. (6) and Eq. (3) are incomparable. Indeed, on the one hand, there exist graphs with bounded average degree and degeneracy as large as we want. For example, take a complete graph of order $k+1$ and a path of length p and identify a vertex of each. Then the degeneracy is k but the average degree tends to 2 when p tends to infinity. So for p large, the bound of Eq. (3) is better. On the other hand, consider the graph G where vertex i is joined to the s vertices $i+h, 1 \leq h \leq s$ with $i+h \leq n$. Then its degeneracy is s (consider the layout (v_1, \dots, v_n)), but we have $m = sn - s(s+1)/2$ and so for n large the bound of Eq. (6) is better.

The following tight inequalities are well-known (see e.g. Proposition 3.1 of [NDM12]) : $\delta^*(G) \leq \text{Mad}(G) < 2\delta^*(G)$. Thus the bounds given by Eq. (7) and Eq. (3) are incomparable. The lower bound of Eq. (7) is better than the bound of Eq. (3) when $\text{Mad}(G) > \sqrt{2}\delta^*(G)$ and worse in the other case.

4 A lower bound using the grooming on the path

Let σ be a layout of G such that $w(G, \sigma) = \text{cw}(G)$. As $w(G, \sigma)$ is the maximum load of an edge of P_σ under $\mathcal{R}_\sigma(G)$, we have $T(\text{cw}(G), n) \geq m$, and so

$$\text{cw}(G) \geq C^*(m, n) = \min\{C \mid m \leq T(C, n)\} \quad (8)$$

Note that the bound is reached by taking as edges of G the pairs of endvertices of $T(C^*(m, n), n)$ requests that can be groomed on a path of n vertices with grooming factor $C^*(m, n)$.

A closed formula for $T(C, n)$ as been given in [BCCP06]. We shall use it to determine $C^*(m, n)$.

Theorem 2 ([BCCP06]) *Let n and C be fixed positive integers.*

- If n is odd and $\frac{n^2-1}{8} \leq C \leq \frac{n^2-1}{4}$, then $T(C, n) = \frac{(n-1)^2}{4} + C$;
- If n is even and $\frac{n(n+2)}{8} \leq C \leq \frac{n^2}{4}$, then $T(C, n) = \frac{n(n-2)}{4} + C$;
- Otherwise, let $C = C_s - d$ with $C_s = \frac{s(s+1)}{2}$ and $0 \leq d < s$; $n = qs + r$ with $0 \leq r < s$; $r = aq + \alpha$ with $0 \leq \alpha < q$; $s - r = b(q+1) + \beta$ with $0 \leq \beta \leq q$; $T_s = sn - C_s$; $A_s = ar - \frac{a(a+1)}{2}q$ and $B_s = (b+1)(s-r) - \frac{b(b+1)}{2}(q+1)$. We have $T(C, n) = T_s - dq + \min\{A_s + d, B_s\}$.

One could think that an optimal solution can be obtained by greedily selecting all requests of length 1, then all requests of length 2, and so on until the capacity limit C is reached. But that is true only for $C \leq 9$.

In particular, one could have expected that, for $C = C_s = s(s+1)/2$, an optimal solution would be obtained by taking all the requests of size at most s in number $T_s = sn - C_s$ (example at the end of Section 3). But it is not true for $n = 11$ and $C = 10$ ($s = 4$). The solution with all the requests of size at most 4 has $T_4 = 34$ requests. The maximum load is 10, but it is reached only for the edges of the path $v_1 \cdots v_{11}$ of the form $v_i v_{i+1}$ with $4 \leq i \leq 8$. So we can delete the request of length 4 : $v_4 \cdots v_8$ and add the two requests of length 5 $v_1 \cdots v_6$ and $v_6 \cdots v_{11}$ to get a solution with $35 > T_4$ requests. In that case, we say that we have an *anomaly*.

We did not find yet a simple closed formula for $C^*(m, n)$, but in some cases it is computed easily. For example, if $n - 1 < m \leq \frac{3n-3}{2}$, then $C^*(m, n) = 2$. Similarly, if $\frac{3n-3}{2} < m \leq 2n - 3$, then $C^*(m, n) = 3$ and so on. The computation would have been easier if there were no anomalies. Fortunately, we can have good approximations by using results of [BC23]. As example, for $C = C_s = s(s+1)/2$, we have $T_s \leq T(C_s, n) \leq T_s + \text{An}^*(C_s)$ where T_s is the value obtained by taking all the requests of size s and $\text{An}^*(C_s)$ represents the number of anomalies. A good upper bound on $\text{An}^*(C_s) \leq (5 - 2\sqrt{6})C_s$ is given in [BC23].

One can show that the bound given by Eq. (8) is better than that of Eq. (6).

Proposition 3 $C^*(m, n) \geq \frac{1}{2} \frac{m}{n-1} \left(\frac{m}{n-1} + 1 \right)$

Proof. According to the definition of $C^*(m, n)$, it suffices to prove that, for $C = C_b = \frac{1}{2} \frac{m}{n-1} \left(\frac{m}{n-1} + 1 \right)$, we have $T(C_b, n) < m$. We give here the proof when $m/(n-1)$ is an integer s . In that case, $C_b = C_s = \frac{s(s+1)}{2}$ and so $T(C_b, n) = sn - C_s$. On the other side $m = sn - s$. As $C_s > s$ for $s \geq 2$, we get $T(C_b, n) < m$. The proof is a more involved when $m/(n-1)$ is not an integer but works by taking $s = \lceil m/(n-1) \rceil$. \square

The gap between the bounds can be significant. For example, $C^*(21, 8) = 9$ while Eq. (6) gives 6. If we take $s = 10$ ($C_{10} = 55$) and n multiple of 21, according to [BCCP06] there are no anomalies and so $T(C_{10}, n) = 10n - 55$. For $n = 42$, $T(C_{10}, 42) = 365$. But for $m = 365$, Eq. (6) gives a value of $44.07 \ll 55$.

Lower bounds in branch-and-bound algorithms can be computed quickly by using $C^*(m, n)$ (or a good approximation) or the bounds of Eq. (6) or Eq. (5). We can also use the better bound of Eq. (7) but it needs the computation of $\text{Mad}(G)$ which is polynomial but long in practice. We can also use an improvement of Eq. (8) by noting that $\text{cw}(G) \geq \text{cw}(H)$ for any subgraph H of G and so

$$\text{cw}(G) \geq \max\{C^*(m_H, n_H) \mid H \text{ subgraph of } G\} \quad (9)$$

In particular, we can compute the bound only for a subgraph H_0 such that $\text{Ad}(H_0) = \text{Mad}(G)$ and so $\text{cw}(G) \geq C^*(m_{H_0}, n_{H_0})$. By Proposition 3, this bound is better than that of Eq. (7).

In conclusion, it appears that the bounds given in this section are better than the bounds based on degrees. Our next step will be to design fast algorithms for computing the bounds of Eq. (8) and Eq. (9).

Références

- [AAAE07] U. Adamy, C. Ambühl, R. S. Anand, and T. Erlebach. Call control in rings. *Algorithmica*, 47(3) :217–238, 2007.
- [BC23] J-C. Bermond and M. Cosnard. Grooming on 3 or 4 paths with a common vertex. Technical report, 2023.
- [BCCP06] J-C. Bermond, M. Cosnard, D. Coudert, and S. Pérennes. Optimal Solution of the Maximum All Request Path Grooming Problem. In *AICT*. IEEE, 2006.
- [DPS02] J. Díaz, J. Petit, and M. Serna. A survey on graph layout problems. *ACM Computing Surveys*, 34(3) :313–356, 2002.
- [GPR⁺19] A.C. Giannopoulou, M. Pilipczuk, J-F. Raymond, D.M Thilikos, and M. Wrochna. Cutwidth : Obstructions and algorithmic aspects. *Algorithmica*, 81 :557–588, 2019.
- [JM92] M. Juvan and B. Mohar. Optimal linear labelings and eigenvalues of graphs. *Discrete Applied Mathematics*, 36(2) :153 – 168, 1992.
- [Klo10] B. Kloeckner. Cutwidth and degeneracy of graphs. Technical report, 2010. hal-00408210v2.
- [MPDP13] R. Martí, J. J. Pantrigo, A. Duarte, and E. G. Pardo. Branch and bound for the cutwidth minimization problem. *Computers & Operations Research*, 40(1) :137 – 149, 2013.
- [NDM12] J. Nešetřil and P. O. De Mendez. *Sparsity : graphs, structures, and algorithms*, volume 28. Springer Science & Business Media, 2012.