

Computing the largest bond of a graph

Original title: Computando o maior bond de um grafo

Gabriel Lagoa Duarte

Advisor: Uéverton dos Santos Souza

Instituto de Computação, Universidade Federal Fluminense
Niterói - RJ, Brasil

`gabrielld@id.uff.br`, `ueverton@ic.uff.br`

RESUMO

Um bond de um grafo G é um conjunto desconectante minimal de arestas de G , isto é, bonds são conjuntos de corte que correspondem a cortes $[S, V \setminus S]$ de G onde $G[S]$ e $G[V \setminus S]$ são ambos conexos. Dado $s, t \in V(G)$, um st -bond de G é um bond cuja remoção desconecta s e t . Nesse trabalho, nós investigamos a complexidade de se computar o maior bond e maior st -bond de um grafo. Apesar de cortes e bonds serem correlacionados, observamos que computar o maior bond de um grafo tende a ser mais difícil do que computar o seu corte máximo. Nós mostramos que MAIOR BOND se mantém NP-difícil mesmo para grafos planares bipartidos. Além disso, apresentamos um framework para reduzir CORTE MÁXIMO a MAIOR BOND, o qual nos permite indentificar diversas classes onde o problema é NP-difícil. Em relação a complexidade parametrizada, demonstramos que MAIOR BOND e MAIOR st -BOND em grafos com clique-width w não podem ser resolvidos em tempo $f(w) \times n^{o(w)}$, a menos que ETH falhe, porém podem ser resolvidos em tempo $f(w) \times n^{O(w)}$. Utilizando minors, também mostramos que os mesmos são FPT quando parametrizados pelo tamanho da solução ou pela treewidth. Por fim, observamos que tais problemas parametrizados não admitem núcleos polinomiais, a menos que $NP \subseteq coNP/poly$.

PALAVRAS CHAVE. bond, corte, FPT, treewidth, clique-width

Tópicos (Teoria e Algoritmos em Grafos)

ABSTRACT

A bond of a graph G is an inclusion-wise minimal disconnecting set of G , i.e., bonds are cut-sets corresponding to cuts $[S, V \setminus S]$ of G such that $G[S]$ and $G[V \setminus S]$ are both connected. Given $s, t \in V(G)$, an st -bond of G is a bond whose removal disconnects s and t . In this work, we investigate the complexity of computing the largest bond and the largest st -bond of a graph. Although cuts and bonds are similar, we remark that computing the largest bond of a graph tends to be harder than computing its maximum cut. We show that LARGEST BOND remains NP-hard even for planar bipartite graphs. Besides, we present a general framework for reducing MAXIMUM CUT to LARGEST BOND, which allows us to recognize several classes for which the problem remains NP-hard. Regarding parameterized complexity, we show that LARGEST BOND and LARGEST st -BOND on graphs of clique-width w cannot be solved in time $f(w) \times n^{o(w)}$ unless the Exponential Time Hypothesis fails, but they can be solved in time $f(w) \times n^{O(w)}$. Using the notion of graph minors, we also show that both problems are FPT when parameterized by the size of the solution or the treewidth of the input graph. Finally, we remark that such parameterized problems do not admit polynomial kernels unless $NP \subseteq coNP/poly$.

KEYWORDS. bond, cut, FPT, treewidth, clique-width

Paper topics (Theory and Algorithms in Graphs)

1. Introduction

Let $G = (V, E)$ be a simple, connected, undirected graph. A *disconnecting set* of G is a set of edges $F \subseteq E(G)$ whose removal disconnects G . The edge-connectivity of G is $\kappa'(G) = \min\{|F| : F \text{ is a disconnecting set of } G\}$. A cut $[S, T]$ of G is a partition of V into two subsets S and $T = V \setminus S$. The cut-set $\partial(S)$ of a cut $[S, T]$ is the set of edges that have one endpoint in S and the other endpoint in T ; these edges are said to cross the cut. In a connected graph, each cut-set determines a unique cut. Note that every cut-set is a disconnecting set, but the converse is not true. An inclusion-wise minimal disconnecting set of a graph is called a *bond*. It is easy to see that every bond is a cut-set, but there are cut-sets that are not bonds. More precisely, a nonempty set of edges F of G is a bond if and only if F determines a cut $[S, T]$ of G such that $G[S]$ and $G[T]$ are both connected. Let $s, t \in V(G)$. An *st*-bond of G is a bond whose removal disconnects s and t .

A minimum (maximum) cut of a graph G is a cut with cut-set of minimum (maximum) size. Every minimum cut is a bond, thus a minimum bond is also a minimum cut of G , and it can be found in polynomial time using the classical Edmonds–Karp algorithm [Edmonds e Karp, 1972]. Besides that, minimum *st*-bonds are well-known structures, since they are precisely the *st*-cuts involved in the Gomory–Hu trees [Gomory e Hu, 1961].

Regarding bonds on planar graphs, a folklore theorem states that if G is a connected planar graph, then a set of edges is a cycle in G if and only if it corresponds to a bond in the dual graph of G , see Godsil e Royle [2001]. Note that each cycle separates the faces of G into the faces in the interior of the cycle and the faces of the exterior of the cycle, and the duals of the cycle edges are exactly the edges that cross from the interior to the exterior [Oxley, 2006]. Consequently, the girth of a planar graph equals the edge connectivity of its dual [Cho et al., 2007].

In this work, we are interested in the complexity aspects of the following problem.

LARGEST BOND

Instance: A graph $G = (V, E)$; a positive integer k .

Question: Is there a proper subset $S \subset V(G)$ such that $G[S]$ and $G[V \setminus S]$ are connected and $|\partial(S)| \geq k$?

We also consider LARGEST *st*-BOND, where given a graph G , vertices $s, t \in V(G)$, and an integer k , we are asked whether G has an *st*-bond of size at least k .

Although cuts and bonds are similar, computing the largest bond of a graph seems to be harder than computing its maximum cut. MAXIMUM CUT is NP-hard in general [Garey e Johnson, 1979], but becomes polynomial for planar graphs [Hadlock, 1975]. On the other hand, finding a longest cycle in a planar graph is NP-hard, implying that finding a largest bond of a planar multi-graph (or simple edge-weighted) is NP-hard. In addition, it is well-known that if a simple planar graph is 3-vertex-connected, then its dual is a simple planar graph. In 1976, Garey, Johnson, and Tarjan [Garey et al., 1976] proved that the problem of establishing whether a 3-vertex-connected planar graph is Hamiltonian is NP-complete, thus, as also noted by Haglin e Venkatesan [1991], finding the largest bond of a simple planar graph is also NP-hard, contrasting with the solvability of MAXIMUM CUT on such a class.

Regarding parameterized complexity, it is well known that MAXIMUM CUT can be solved in FPT time when parametrized by the size of the solution [Mahajan e Raman, 1999], and since every graph has a cut with at least half the edges [Erdős, 1965], it follows that it has a linear kernel. However, until this work, there was no algorithmic study regarding the parameterized complexity of computing the largest bond of a graph to the best of our knowledge. Therefore, in this dissertation, we complement the state of the art on the problem of computing the largest bond of a graph.

1.1. Our contributions

In this dissertation, we complement the state of the art on the problems of computing the largest bond and the largest st -bond of a graph. Preliminarily, we present general reductions that allows us to observe that LARGEST BOND is NP-hard for several graph classes for which MAXIMUM CUT is NP-hard. Using this framework, we are able to show that LARGEST BOND on graphs of clique-width w cannot be solved in time $f(w)n^{o(w)}$ unless the ETH fails. We also prove that LARGEST BOND is NP-complete even on planar bipartite graphs. Interestingly, although MAXIMUM CUT can be solved in polynomial time on planar graphs [Hadlock, 1975; Orlova e Dorfman, 1972] and it is trivial on bipartite graphs, our problems are intractable even on the intersection of these classes.

To tackle this difficulty, we study both problems from the perspective of the parameterized complexity. Using win/win approaches, we consider the strategy of preprocessing the input in order to bound the treewidth of the resulting instance. After that, we give $O^*(2^{O(tw \log tw)})$ -time algorithms for both problems¹, where tw is the treewidth of the input graph. Moreover, we can improve this running time using the rank-based approach [Bodlaender et al., 2015] to $O^*(c^{tw})$ for some constant c . Also, let us note that our result generalizes the polynomial time algorithms for LARGEST BOND on series parallel graphs and graphs without $K_5 \setminus e$ as a minor due to Chaourar [2017, 2019] since such graphs are tree-width bounded, see Robertson e Seymour [1986]. Based on these algorithms, we give $O^*(2^{O(k)})$ -time algorithms for the problems. In addition, we remark that LARGEST BOND do not admit polynomial kernels, unless $\text{NP} \subseteq \text{coNP/poly}$.

Finally, we consider a different structural graph parameter. We design tight (assuming ETH) XP-time algorithms for both problems when parameterized by clique-width cw . Our algorithm is tight assuming ETH due to our framework for intractability.

The results of this dissertation were published in the *Algorithmica* journal, one of the main and most prestigious journals in the area of *Algorithms*:

- Gabriel L. Duarte et al. *Computing the Largest Bond and the Maximum Connected Cut of a Graph*. **Algorithmica** (2021). <https://doi.org/10.1007/s00453-020-00789-1>.

In addition, preliminary results of this dissertation were also presented in 2019 at *IPEC*, the main international conference specializing in Parameterized Complexity:

- Gabriel L. Duarte et al. *Computing the largest bond of a graph*. **International Symposium on Parameterized and Exact Computation (IPEC)**. Munich, Germany, 2019, <http://dx.doi.org/10.4230/LIPIcs.IPEC.2019.12>.

The full dissertation can be found **here** in the official repository of the Universidade Federal Fluminense. <http://www.ic.uff.br/PosGraduacao/frontend-tesesdissertacoes/download.php>

Next, we briefly present the main results of the dissertation. Due to the space, for more details and specialized definitions, please see the dissertation.

2. Intractability results

In this section, we discuss aspects of the hardness of computing the largest bond. Notice that LARGEST BOND is Turing reducible to LARGEST st -BOND. Therefore, the intractability results presented in this section also holds for LARGEST st -BOND.

¹The $O^*(\cdot)$ notation suppresses polynomial factors in the input size.

Although MAXIMUM CUT is trivial for bipartite graphs, we first observe that the same does not apply to compute the largest bond. Since a connected planar graph is Eulerian if and only if its dual graph is bipartite; subdivision of edges does not increase the size of the largest bond; and to decide whether a 4-regular planar graph has a Hamiltonian cycle is NP-complete [Picouleau, 1994], the following holds.

Theorem 1. LARGEST BOND is NP-complete for planar bipartite graphs.

Since LARGEST BOND is NP-complete even for bipartite graphs, it is natural to ask for parameterizations for which the problem becomes fixed-parameter tractable. The next result shows that the size of the set S is not a useful single parameter to design an FPT algorithm.

Theorem 2. Let G be a simple bipartite graph and $\ell \in \mathbb{N}$. To determine the largest bond $\partial(S)$ of G with $|S| = \ell$ is $W[1]$ -hard with respect to ℓ .

Next, we present a general framework for reducibility from MAXIMUM CUT to LARGEST BOND, by defining a special graph operator ψ such that MAXIMUM CUT on a graph class $\mathcal{F}$ is reducible to LARGEST BOND on the image of $\mathcal{F}$ via ψ . An interesting particular case occurs when $\mathcal{F}$ is closed under ψ .

Definition 3. Let G be a graph and let $n = V(G)$. The graph $\psi(G)$ is constructed as follows: (i) create n disjoint copies $G_1, G_2, \dots, G_n$ of G ; (ii) add vertices v_a and v_b ; (iii) add an edge between v_a and v_b ; (iv) add all possible edges between $V(G_1 \cup G_2 \cup \dots \cup G_n)$ and $\{v_a, v_b\}$.

Definition 4. A set of graphs $\mathcal{G}$ is closed under operator ψ if whenever $G \in \mathcal{G}$, then $\psi(G) \in \mathcal{G}$.

From the fact that a graph G has a cut $[S, V(G) \setminus S]$ of size k if and only if $\psi(G)$ has a bond $\partial(S')$ of size at least $nk + n^2 + 1$, the following theorem holds.

Theorem 5. LARGEST BOND is NP-complete for any graph class $\mathcal{G}$ such that: $\mathcal{G}$ is closed under operator ψ ; and MAXCUT is NP-complete for graphs in $\mathcal{G}$.

At this point, we obtain the following corollary.

Corollary 6. LARGEST BOND is NP-complete for the following classes:

1. chordal graphs;
2. co-comparability graphs;
3. P_5 -free graphs;
4. AT-free graphs.

3. Algorithmic upper and lower bounds for clique-width parameterization

The *clique-width* of a graph G , denoted by $cw(G)$, is defined as the minimum number of labels needed to construct G , using the following four operations:

1. Create a single vertex v with an integer label ℓ (denoted by $\ell(v)$);
2. Take the disjoint union (i.e., co-join) of two graphs (denoted by $\oplus$);

3. Join by an edge every vertex labeled i to every vertex labeled j for $i \neq j$ (denoted by $\eta(i, j)$);
4. Relabel all vertices with label i by label j (denoted by $\rho(i, j)$).

An algebraic term that represents such a construction of G and uses at most w labels is said to be a w -expression of G , and the clique-width of G is the minimum w for which G has a w -expression.

Graph classes with bounded clique-width include cographs, distance-hereditary graphs, graphs of bounded treewidth, graphs of bounded branchwidth, and graphs of bounded rank-width.

In the '90s, Courcelle, Makowsky, and Rotics [Courcelle et al., 2000] proved that all problems expressible in MSO_1 -logic are fixed-parameter tractable when parameterized by the clique-width of a graph and the logical expression size. The applicability of this meta-theorem has made clique-width become one of the most studied parameters in parameterized complexity. However, although several problems are MSO_1 -expressible, this is not the case with MAXIMUM CUT.

In 2014, Fomin, Golovach, Lokshtanov and Saurabh [Fomin et al., 2014] showed that MAXIMUM CUT on a graph of clique-width w cannot be solved in time $f(w) \times n^{o(w)}$ for any function f of w unless Exponential Time Hypothesis (ETH) fails. Using operator ψ , we are able to extend this result to LARGEST BOND.

Lemma 7. *LARGEST BOND on graphs of clique-width w cannot be solved in $f(w) \times n^{o(w)}$ time unless the ETH fails.*

Lemma 7 shows that LARGEST BOND on graphs of clique-width w cannot be solved in time $f(w) \times n^{o(w)}$ unless the ETH fails. Next, we show that given an expression tree of width at most w , LARGEST BOND can be solved in $f(w) \times n^{O(w)}$ time.

Theorem 8. *LARGEST BOND cannot be solved in time $f(w) \times n^{o(w)}$ unless ETH fails, where w is the clique-width of the input graph. Moreover, given an expression tree of width at most w , LARGEST BOND can be solved in time $f(w) \times n^{O(w)}$.*

In order to extend this result to LARGEST st -BOND, it is enough to observe that given a tree expression $\mathcal{T}$ of G with width w , it is easy to construct a tree expression $\mathcal{T}'$ with width equals $w + 2$, where no vertex of $V(G)$ has the same label than either s or t . Let $w + 1$ be the label of s , and let $w + 2$ be the label of t . By fixing, for each ℓ , $s_{w+1} = |L_{w+1}(X_\ell)|$ and $s_{w+2} = 0$, one can solve LARGEST st -BOND in time $f(w) \times n^{O(w)}$.

4. Bounding the treewidth of the input graph

In the remainder of this work we deal with our main problems: LARGEST BOND and LARGEST st -BOND parameterized by the size of the solution (k). Inspired by the principle of preprocessing the input to obtain a kernel, we consider the strategy of preprocessing the input in order to bound the treewidth of the resulting instance. First, we present some definitions.

Definition 9. *A tree decomposition of a undirected graph G is a pair $\mathcal{T} = (T, \{X_t\}_{t \in V(T)})$, such that T is a tree where each node t is assigned to a set of vertices $X_t \subseteq V(G)$, called bags, according to the following conditions:*

- $\bigcup_{t \in V(T)} X_t = V(G)$;
- For each $uv \in E(G)$ there is a node t such that $\{u, v\} \subseteq X_t$;

- For each $v \in V(G)$, the set $T_v = \{t \in V(T) : v \in X_t\}$ spans a subtree of T .

The *width* of a tree decomposition $\mathcal{T}$ is the size of its largest bag minus one. The treewidth of G is the minimum width among all tree decompositions of G .

Definition 10. A graph H is a minor of a graph G if H can be constructed from G by deleting vertices or edges, and contracting edges.

Now, we start our analysis with LARGEST BOND.

Definition 11. A graph H is called a minor of a graph G if H can be formed from G by deleting edges, deleting vertices, and by contracting edges. For each vertex v of H , the set of vertices of G that are contracted into v is called a branch set of H .

Lemma 12. Let G be a simple connected undirected graph, and k be a positive integer. If G contains $K_{2,k}$ as a minor then G has a bond of size at least k .

Combined with Lemma 12, the following results show that, without loss of generality, our study on k -bonds can be reduced to graphs of treewidth $O(k)$.

Lemma 13. [Bodlaender et al., 1997] Every graph $G = (V, E)$ contains $K_{2,k}$ as a minor or has treewidth at most $2k - 2$.

Lemma 14. [Bodlaender et al., 1997] There is a polynomial-time algorithm that either concludes that the input graph G contains $K_{2,k}$ as a minor, or outputs a tree-decomposition of G of width at most $2k - 2$.

From Lemma 12 and Lemma 14 it follows that there is a polynomial-time algorithm that either concludes that the input graph G has a bond of size at least k , or outputs a tree-decomposition of G of width at most $2k - 2$.

4.1. The st-bond case

Let $S \subseteq V(G)$ and let $\partial(S)$ be a bond of a connected graph G . Recall that a block is a 2-vertex-connected subgraph of G which is inclusion-wise maximal, and a block-cut tree of G is a tree whose vertices represent the blocks and the cut vertices of G , and there is an edge in the block-cut tree for each pair of a block and a cut vertex that belongs to that block. Then, $\partial(S)$ intersects at most one block of G . More precisely, for any two distinct blocks B_1 and B_2 of G , if $S \cap V(B_1) \neq \emptyset$ and $S \cap V(B_1) \neq V(B_1)$, then either $V(B_2) \subseteq S$, or $V(B_2) \subseteq V \setminus S$. Indeed, if this is not the case, then either $G[S]$ or $G[V \setminus S]$ would be disconnected. Thus, to solve LARGEST *st*-BOND, it is enough to consider, individually, each block on the path between s and t in the block-cut tree of G . Also, if a block is composed of a single edge, then it is a bridge in G , which is not a solution for the problem unless $k = 1$. Thus, we may assume without loss of generality that G is 2-vertex-connected.

Lemma 15. Let G be a 2-vertex-connected graph. If G contains $K_{2,2k}$ as a minor, then there exists $S \subseteq V(G)$ such that $\partial(S)$ is a bond of size at least k .

Corollary 16. Given a graph G , vertices $s, t \in V(G)$, and an integer k , there exists a polynomial-time algorithm that either concludes that G has an *st*-bond of size at least k or outputs a subgraph G' of G together with a tree decomposition of G' of width equals $O(k)$, such that G' has an *st*-bond of size at least k if and only if G has an *st*-bond of size at least k .

5. Taking the treewidth as parameter

In the following, given a tree decomposition $\mathcal{T}$, we denote by ℓ one node of $\mathcal{T}$ and by X_ℓ the vertices contained in the *bag* of ℓ . We assume w.l.o.g that $\mathcal{T}$ is a extended version of a *nice* tree decomposition (see Cygan et al. [2015]), that is, we assume that there is a special root node r such that $X_r = \emptyset$ and all edges of the tree are directed towards r and each node ℓ has one of the following five types: *Leaf*; *Introduce vertex*; *Introduce edge*; *Forget vertex*; and *Join*.

Moreover, define G_ℓ to be the subgraph of G which contains only vertices and edges that have been introduced in ℓ or in a descendant of ℓ .

The number of partitions of a set of k elements is the k -th *Bell number*, which we denote by $B(k)$ ($B(k) \leq k!$, see Odlyzko [1995]).

Theorem 17. *Given a nice tree decomposition of G with width tw , one can find a bond of maximum size in time $2^{O(tw \log tw)} \times n$ where n is the number of vertices of G .*

We can extend the results of Theorem 17 to resolve the LARGEST st -BOND by adding s and t in all the nodes and fixing $s \in S$ and $t \notin S$. Then, we can use the same solution presents in the proof.

At this point, to solve the problems in FPT time when parameterized by k , it is enough to present FPT algorithms for both problems parameterized by the treewidth.

Theorem 18. *LARGEST BOND and LARGEST st -BOND are FPT when parameterized either by the treewidth or by the size of the solution, k .*

As seen previously, any bond $\partial(S)$ of a graph G intersects at most one of its block. Thus, an or-composition for LARGEST BOND parameterized by k can be done from the disjoint union of ℓ inputs, by selecting exactly one vertex of each input graph and contracting them into a single vertex. Now, let $(G_1, k, s_1, t_1), (G_2, k, s_2, t_2), \dots, (G_\ell, k, s_\ell, t_\ell)$ be ℓ instances of LARGEST st -BOND parameterized by k . An or-composition for LARGEST st -BOND parameterized by k can be done from the disjoint union of $G_1, G_2, \dots, G_\ell$, by contracting t_i, s_{i+1} into a single vertex, $1 \leq i \leq \ell - 1$, and setting $s = s_1$ and $t = t_\ell$. Therefore, the following holds.

Theorem 19. *LARGEST BOND and LARGEST st -BOND do not admit polynomial kernel, unless $NP \subseteq coNP/poly$, when parameterized by k .*

References

- Bodlaender, H. L., Cygan, M., Kratsch, S., e Nederlof, J. (2015). Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. *Information and Computation*, 243:86–111.
- Bodlaender, H. L., Van Leeuwen, J., Tan, R., e Thilikos, D. M. (1997). On interval routing schemes and treewidth. *Information and Computation*, 139(1):92–109.
- Chaourar, B. (2017). A linear time algorithm for a variant of the MAX CUT problem in series parallel graphs. *Advances in Operations Research*, p. 1267108:1–1267108:4.
- Chaourar, B. (2019). Connected max cut is polynomial for graphs without $K_5 \setminus e$ as a minor. *CoRR*, abs/1903.12641.
- Cho, J. J., Chen, Y., e Ding, Y. (2007). On the (co)girth of a connected matroid. *Discrete Applied Mathematics*, 155(18):2456 – 2470. ISSN 0166-218X.

- Courcelle, B., Makowsky, J. A., e Rotics, U. (2000). Linear time solvable optimization problems on graphs of bounded clique-width. *Theory of Computing Systems*, 33(2):125–150.
- Cygan, M., Fomin, F. V., Kowalik, Ł., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., e Saurabh, S. (2015). *Parameterized algorithms*, volume 4. Springer.
- Edmonds, J. e Karp, R. M. (1972). Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM*, 19(2):248–264. ISSN 0004-5411.
- Erdős, P. (1965). On some extremal problems in graph theory. *Isr. J. Math.*, 3(2):113–116.
- Fomin, F. V., Golovach, P. A., Lokshtanov, D., e Saurabh, S. (2014). Almost optimal lower bounds for problems parameterized by clique-width. *SIAM Journal on Computing*, 43(5):1541–1563.
- Garey, M. R. e Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman.
- Garey, M. R., Johnson, D. S., e Tarjan, R. E. (1976). The planar hamiltonian circuit problem is NP-complete. *SIAM Journal on Computing*, 5(4):704–714.
- Godsil, C. D. e Royle, G. F. (2001). *Algebraic Graph Theory*. Graduate texts in mathematics. Springer. ISBN 978-0-387-95220-8.
- Gomory, R. E. e Hu, T. C. (1961). Multi-terminal network flows. *Journal of the Society for Industrial and Applied Mathematics*, 9(4):551–570.
- Hadlock, F. (1975). Finding a maximum cut of a planar graph in polynomial time. *SIAM Journal on Computing*, 4(3):221–225.
- Haglin, D. J. e Venkatesan, S. M. (1991). Approximation and intractability results for the maximum cut problem and its variants. *IEEE Transactions on Computers*, 40(1):110–113. ISSN 0018-9340.
- Mahajan, M. e Raman, V. (1999). Parameterizing above guaranteed values: MaxSat and MaxCut. *Journal of Algorithms*, 31(2):335–354.
- Odlyzko, A. M. (1995). Asymptotic enumeration methods. *Handbook of combinatorics*, 2(1063):1229.
- Orlova, G. I. e Dorfman, Y. G. (1972). Finding the maximal cut in a graph. *Engineering Cybernetics*, 10(3):502–506.
- Oxley, J. G. (2006). *Matroid theory*, volume 3. Oxford University Press, USA.
- Picouleau, C. (1994). Complexity of the hamiltonian cycle in regular graph problem. *Theoretical Computer Science*, 131(2):463 – 473. ISSN 0304-3975.
- Robertson, N. e Seymour, P. D. (1986). Graph minors. V. excluding a planar graph. *Journal of Combinatorial Theory, Series B*, 41(1):92–114.