Research Article

A Linear Time Algorithm for a Variant of the MAX CUT Problem in Series Parallel Graphs

Brahim Chaourar

Department of Mathematics and Statistics, Al Imam Mohammad Ibn Saud Islamic University (IMSIU), P.O. Box 90950, Riyadh 11623, Saudi Arabia

Correspondence should be addressed to Brahim Chaourar; bchaourar@hotmail.com

Received 31 August 2017; Accepted 15 November 2017; Published 10 December 2017

Academic Editor: Yi-Kuei Lin

Copyright © 2017 Brahim Chaourar. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Given a graph $G = (V, E)$, a connected sides cut $(U, V \setminus U)$ or $\delta(U)$ is the set of edges of E linking all vertices of U to all vertices of $V \setminus U$ such that the induced subgraphs $G[U]$ and $G[V \setminus U]$ are connected. Given a positive weight function w defined on E , the maximum connected sides cut problem (MAX CS CUT) is to find a connected sides cut Ω such that $w(\Omega)$ is maximum. MAX CS CUT is NP-hard. In this paper, we give a linear time algorithm to solve MAX CS CUT for series parallel graphs. We deduce a linear time algorithm for the minimum cut problem in the same class of graphs without computing the maximum flow.

1. Introduction

Sets and their characteristic vectors will not be distinguished. We refer to Bondy and Murty [1] about graph theory terminology and facts.

Given an undirected graph $G = (V, E)$ and positive weights $w_{ij} = w_{ji}$ on the edges $(i, j) \in E$, the maximum cut problem (MAX CUT) is that of finding the set of vertices S that maximizes the weight of the edges in the cut $(S, V \setminus S)$ or $\delta(S)$ or $\delta(V \setminus S)$, that is, the weight of the edges with one endpoint in S and the other in $V \setminus S$. The (decision variant of the) MAX CUT is one of the Karp's original NP-complete problems [2] and has long been known to be NP-complete even if the problem is unweighted, that is, if $w_{ij} = 1$ for all $(i, j) \in E$ [3]. This motivates the research to solve the MAX CUT problem in special classes of graphs. The MAX CUT problem is solvable in polynomial time for the following special classes of graphs: planar graphs [4–6], line graphs [7], graphs with bounded treewidth, or cographs [8]. But the problem remains NP-complete for chordal graphs, undirected path graphs, split graphs, tripartite graphs, graphs that are the complement of a bipartite graph [8], and planar graphs if the weights are of arbitrary sign [9]. Besides its theoretical importance, the MAX CUT problem has applications in circuit layout design and statistical physics [10]. For a comprehensive survey of the MAX CUT problem, the

reader is referred to Poljak and Tuza [11] and Ben-Ameur et al. [12]. The best known algorithm for MAX CUT in planar graphs has running time complexity $O(n^{3/2} \log n)$, where n is the number of vertices of the given graph [13]. This algorithm is not a linear time one. The main result of this paper is to exhibit a linear time algorithm for a special variant of MAX CUT in series parallel graphs. Series parallel graphs can be used to model series and parallel electric circuits [14]. Another motivation is that there is not a known linear time algorithm for the MAX CUT problem or any one of its NP-hard versions in any special class of graphs till today. This motivates the research on linear time algorithms in special classes of graphs for our problem which is an NP-hard variant of MAX CUT. Finally, many NP-hard problems have been studied in series parallel graphs [15–31] but not all outputted a linear time algorithm for the considered problem.

Let us give some definitions. Given an undirected graph $G = (V, E)$ and a subset of vertices U , a connected sides cut $\delta(U)$ is a cut where both induced subgraphs $G[U]$ and $G[V \setminus U]$ are connected. Special connected sides cuts are trivial cuts, that is, cuts with one single vertex in one side (when this vertex is not a disconnecting vertex). The corresponding weighted variant of MAX CUT for connected sides cuts is called MAX CONNECTED SIDES CUT problem (MAX CS CUT). It is clear that MAX CUT and MAX CS CUT are the same problem for complete graphs. Since MAX CUT is

NP-hard for complete graphs (see [2]), then MAX CS CUT is NP-hard in the general case. Another motivation is that MAX CS CUT gives a lower bound for MAX CUT.

A parallel closure of a graph is an induced subgraph on two vertices. A series extension of the graph $G = (V, E)$ based on the edge $e \in E$ is adding a vertex v of degree 2 in the middle of e in order to have two edges instead of e . A parallel extension of G based on the edge e is adding an edge f having the same incident vertices as e . Series parallel graphs are graphs obtained by applying recursively series and/or parallel extensions starting from one edge. A series degree of a vertex v in a graph G is the degree of v after replacing every parallel closure of G by one single edge. A series labeling of the vertices of a series parallel graph is a labeling of the vertices from 0 to $n - 1 = |V| - 1$ starting from the first two vertices v_0 and v_1 and so on to the last added vertex. Any series parallel graph contains at least one vertex of series degree 2. So, given a vertex v of series degree 2 with the two parallel closures P_0 and P_1 incident to v , and the two adjacent vertices u_0 and u_1 to v , we can contract all edges of P_0 (or P_1) and replace v by u_0 (or u_1), and we obtain a new series parallel graph with a new vertex of series degree 2. Each involved graph in any step of this process is labeled G_j , $0 \leq j \leq n - 1$, with $G_{n-1} = G$, and G_1 is the induced subgraph on the two vertices v_0 and v_1 .

Let G_1 and G_2 be two graphs with e_j an edge of G_j , $j = 1, 2$. The 2-sum of G_1 and G_2 , denoted by $G_1 \oplus_e G_2$ or $G_1 \oplus_2 G_2$, based on the edges e_1 and e_2 is the graph obtained by identifying e_1 and e_2 on an edge e and keeping $G_j \setminus e_j$, $j = 1, 2$, as they are.

We say that a problem is linear for a class of graphs if there is a linear time algorithm to solve it in such class.

The remaining of the paper is organized as follows: in Section 2, we give a linear time algorithm for MAX CS CUT in series parallel graphs; in Section 3, we prove that 2-sums preserve the linearity of MAX CS CUT. We deduce a linear time algorithm for MIN CUT in series parallel graphs in Section 4, and we conclude in Section 5.

2. MAX CS CUT Is Linear for Series Parallel Graphs

MAXCSCUTSP Algorithm

Input. A series parallel graph $G = (V, E)$ with a series labeling of V , a positive weight function w defined on E .

Output. A w -maximum connected sides cut Ω in G .

- (0) Begin
- (1) $j := n - 1$;
- (2) While $j > 1$ do
- (3) Begin
- (4) Let P_0 and P_1 be the two parallel closures incident to v_j in G_j ;
- (5) If $w(P_0) > w(P_1)$ then contract P_1 ;
- (6) Else: contract P_0 ;

- (7) $j := j - 1$;
- (8) End of While
- (9) $j := 2$;
- (10) $\Omega := E(G_1)$;
- (11) While $j \leq n - 1$ do
- (12) Begin
- (13) Let P_0 and P_1 the two parallel closures incident to v_j in G_j ;
- (14) If $w(P_0) + w(P_1) > w(\Omega)$ then $\Omega := P_0 \cup P_1$;
- (15) $j := j + 1$;
- (16) End of While
- (17) End of MAXCSCUTSP algorithm.

This algorithm has two phases: Phase I (steps (1)–(6)) and Phase II (steps (9)–(16)). In each step, we do roughly n operations, so the complexity of MAXCSCUTSP is $O(n)$, where $n = |V|$.

Theorem 1. MAXCSCUTSP algorithm solves MAX CS CUT in series parallel graphs.

Proof. The summary of the algorithm is as follows: MAXCSCUT chooses a vertex v with series degree 2 (step (4)) and contracts the less weighted parallel closure incident to v (steps (5) and (6)), and so on, until the resulting graph becomes G_1 , which is the starting single parallel closure (Phase I). In G_1 , the w -maximum connected sides cut is $E(G_1)$ (step (10)). After that, it goes in the reverse path (Phase II): the w -maximum connected sides cut is either the trivial cut based on the current vertex v_j with series degree 2 or the current computed connected sides cut (step (14)). Let v_j be the chosen vertex with series degree 2 in G_j , and let P_0 and P_1 be the two parallel closures incident to v_j . Without loss of generality, we can suppose that $w(P_0) < w(P_1)$ and $G_{j-1} = G_j/P_0$. Let Ω_j be the w -maximum connected sides cut in G_j , $1 \leq j \leq n - 1$. It suffices to prove that $(\Omega_j) = \max\{w(\Omega_{j-1}), w(P_0 \cup P_1)\}$.

Let Ω be a connected sides cut in G_j distinct from $P_0 \cup P_1$. Since $w(P_0) < w(P_1)$, we have only two cases.

Case 1. $P_1 \subseteq \Omega$, then Ω is a connected sides cut in $G_{j-1} = G_j/P_0$ containing P_1 . And vice versa, any connected sides cut in $G_{j-1} = G_j/P_0$ containing P_1 is a connected sides cut in G_j containing P_1 .

Case 2. $P_1 \not\subseteq \Omega$, then Ω is a connected sides cut in $G_{j-1} = G_j/P_0$ not containing P_1 . And vice versa, any connected sides cut in $G_{j-1} = G_j/P_0$ not containing P_1 is a connected sides cut in G_j not containing P_1 .

So the connected sides cuts candidates for the w -maximum connected sides cut in G_j and G_{j-1} are the same, except $P_0 \cup P_1$. $\square$

Note that MAXCSCUTSP algorithm solves MAX CS CUT in series parallel graphs even for arbitrary sign weight functions.

3. 2-Sums Preserve Linearity of MAX CS CUT

Let $\mathcal{C}(G)$ be the class of connected sides cuts of G . We need the following lemma.

Lemma 2. *One has $\mathcal{C}(G_1 \oplus_2 G_2) = \{\Omega_j \in \mathcal{C}(G_j) : e_j \notin \Omega_j, j = 1, 2\} \cup \{\Omega_1 \oplus_2 \Omega_2 : \Omega_j \in \mathcal{C}(G_j) \text{ and } e_j \in \Omega_j, j = 1, 2\}$.*

We can now state one consequence of this as follows.

Theorem 3. *If MAX CS CUT is linear for two given graphs G_1 and G_2 , then it is also linear for $G_1 \oplus_2 G_2$.*

Proof. It follows from Lemma 2 that a w -maximum connected sides cut in $G_1 \oplus_2 G_2$ is one of the three following connected sides cuts:

(cases 1-2) one of the two w -maximum connected sides cuts in G_j which does not contain $e_j, j = 1, 2$, or

(case 3) the 2-sum of the w -maximum connected sides cuts containing $e_j, j = 1, 2$.

To find a w -maximum connected sides cut in G_j which does not contain $e_j, j = 1, 2$ (case 2), we have to contract e_j . We need then to perform at most $c_{j1}(n_j - 1)$ operations, where $c_{j1}, j = 1, 2$, is the linearity coefficient of the algorithm solving MAX CS CUT in $G_j/e_j, j = 1, 2$, and $n_j, j = 1, 2$, is the number of vertices of $G_j, j = 1, 2$ (by induction).

To find $\Omega_1 \oplus_2 \Omega_2$ (case 3), we have to put $w(e_j), j = 1, 2$, as big as possible, for example, sum of the positive weights of all edges, and find $\Omega_j, j = 1, 2$. In this case, we need to perform at most $c_{12}n_1 + c_{22}n_2$ operations (by induction), where $c_{j2}, j = 1, 2$, is the linearity coefficient of the algorithm solving MAX CS CUT in $G_j, j = 1, 2$.

So we have to compute MAX CS CUT twice in each graph and compare three cuts. The total number of operations is then bounded by $(c_{11} + c_{12})n_1 + (c_{21} + c_{22})n_2 - (c_{11} + c_{21}) \leq c_{\max}(n + 1)$, where n is the number of vertices of $G_1 \oplus_2 G_2$ and $c_{\max} = \max\{c_{11} + c_{12}, c_{21} + c_{22}\}$. So linearity of the problem is preserved. $\square$

The same property holds for K_4 , the unique excluded minor for series parallel graphs, which means that the class of graphs for which MAX CS CUT is linear is larger than series parallel graphs.

Lemma 4. *MAX CS CUT is linear for K_4 .*

Proof. It is not difficult to see that $\mathcal{C}(K_4) = 7$: for any perfect matching of K_4 (there are 3 perfect matchings in K_4), we have a connected sides cut (by removing the perfect matching), and for any vertex of K_4 (there are 4 vertices), the corresponding trivial cut is a connected sides one. We can reduce the number of cuts to be considered as follows.

Let $e = uv \in E(K_4)$ and $f = ab \in E(K_4)$ such that $\{e, f\}$ is a perfect matching of K_4 . We have the following property.

$$2w(e) \geq w(\delta(u)) \quad \text{iff} \quad w(\delta(v)) \geq w(\delta\{u, v\}). \quad (1)$$

So by such comparisons (for each perfect matching of K_4), we eliminate step-by-step cuts to be considered for comparison between them. $\square$

We have then the following corollary directly from Theorems 1 and 3 and Lemma 4.

Corollary 5. *MAX CS CUT is linear for 2-sums of series parallel graphs and copies of K_4 .*

4. MIN CUT Is Linear for Series Parallel Graphs

MINCUTSP Algorithm

Input. A series parallel graph $G = (V, E)$ with a series labeling of V , a positive weight function w defined on E .

Output. A w -minimum connected sides cut Ω in G .

We keep the same steps as MAXCSCUTSP algorithm except the following changes in two steps:

(5) If $w(P_0) < w(P_1)$ then contract P_1 ;

(14) If $w(P_0) + w(P_1) < w(\Omega)$ then $\Omega := P_0 \cup P_1$;

Since this algorithm is similar to MAXCSCUTSP, then its complexity is $O(n)$, where $n = |V|$.

And it is not difficult to see, similarly to MAXCSCUTSP, that MINCUTSP gives the minimum weighted connected sides cut in a series parallel graph without computing the maximum flow.

We can conclude with the following result.

Theorem 6. *Given a connected graph $G = (V, E)$ and a positive weight function w defined on E , then any w -minimum cut is a connected sides cut of G .*

Proof. Let $\delta(U)$ be a cut with $G[U]$ disconnected. It suffices to prove that $\delta(U)$ is not a w -minimum cut. Let $G[U_1]$ be one connected component of $G[U]$. Since G is connected, then $w(V \setminus U, U_1) > 0$ (i.e., there are edges between $V \setminus U$ and U_1). It follows that $w(\delta(U \setminus U_1)) = w(\delta(U)) - w(V \setminus U, U_1) < w(\delta(U))$. $\square$

Other consequences of Lemma 2 and Theorem 6 are the following corollaries.

Corollary 7. *2-sums preserve the linearity of MIN CUT.*

Corollary 8. *MIN CUT is linear for 2-sums of series parallel graphs and copies of K_4 .*

5. Conclusion

We have introduced a new variant of MAX CUT: MAX CS CUT, which is also NP-hard. We have provided two linear time algorithms for MAX CS CUT and MIN CUT, respectively, in series parallel graphs. We have proved that 2-sums preserve the linearity of MAX CS CUT and MIN CUT. Further directions are to study MAX CS CUT in larger classes of graphs than series parallel graphs.

Conflicts of Interest

The author declares that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The author is grateful to the deanship of Scientific Research at Al Imam Mohammad Ibn Saud Islamic University (IMSIU) for supporting financially this research under the Grant no. 331203.

References

- [1] J. A. Bondy and U. S. R. Murty, *Graph Theory with Application*, Elsevier, NY, USA, 1976.
- [2] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*, pp. 85–103, Springer, New York, NY, USA, 1972.
- [3] M. R. Garey, D. S. Johnson, and L. Stockmeyer, "Some simplified NP-complete graph problems," *Theoretical Computer Science*, vol. 1, no. 3, pp. 237–267, 1976.
- [4] F. Barahona, "Planar multicommodity flows, max cut, and the Chinese postman problem," in *Polyhedral Combinatorics, Proceedings DIMACS Workshop, Morristown, New Jersey, 1989*, vol. 1 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pp. 189–202, American Mathematical Society, Providence, Rhode, Island, 1990.
- [5] F. Hadlock, "Finding a maximum cut of a planar graph in polynomial time," *SIAM Journal on Computing*, vol. 4, no. 3, pp. 221–225, 1975.
- [6] G. I. Orlova and Y. G. Dorfman, "Finding the maximal cut in a graph," *Engineering Cybernetics*, pp. 502–506, 1972.
- [7] V. Guruswami, "Maximum cut on line and total graphs," *Discrete Applied Mathematics*, vol. 92, no. 2–3, pp. 217–221, 1999.
- [8] H. L. Bodlaender and K. Jansen, "On the complexity of the maximum cut problem," *Nordic Journal of Computing*, vol. 7, no. 1, pp. 14–31, 2000.
- [9] A. P. Terebenkov, "NP-completeness of maximum-cut and cycle-covering problems for a planar graph," *Cybernetics and Systems Analysis*, vol. 27, no. 1, pp. 16–20, 1991.
- [10] F. Barahona, M. Gröetschel, M. Juenger, and G. Reinelt, "Application of combinatorial optimization to statistical physics and circuit layout design," *Operations Research*, vol. 36, no. 3, pp. 493–513, 1988.
- [11] S. Poljak and Z. Tuza, "Maximum cuts and large bipartite subgraphs," in *American Mathematical Society*, vol. 20 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pp. 181–244, American Mathematical Society, 1995.
- [12] W. Ben-Ameur, A. R. Mahjoub, and J. Neto, "The maximum cut problem in : paradigms of combinatorial optimization," in *Problems and New Approaches*, V. T. Paschos, Ed., J. Wiley and Sons, USA, 2nd edition, 2014.
- [13] W.-K. Shih, S. Wu, and Y. S. Kuo, "Unifying maximum cut and minimum cut of a planar graph," *IEEE Transactions on Computers*, vol. 39, no. 5, pp. 694–697, 1990.
- [14] T. Williams, *The Circuit Designer's Companion*, Butterworth-Heinemann, 2005.
- [15] M. Baiou and J. R. Correa, "The node-edge weighted 2-edge connected subgraph problem: linear relaxation, facets and separation," *Discrete Optimization*, vol. 3, no. 2, pp. 123–135, 2006.
- [16] B. Chaourar, "An $O(K n \log[K n])$ algorithm for the K th best spanning tree in series parallel graphs," *Arabian Journal for Science and Engineering. AJSE. Mathematics*, vol. 35, no. 1D, pp. 29–35, 2010.
- [17] G. Chen and G. Xue, "A PTAS for weight constrained Steiner trees in series-parallel graphs," *Theoretical Computer Science*, vol. 304, no. 1–3, pp. 237–247, 2003.
- [18] I. Demginsky, H. Noltemeier, and H.-C. Wirth, "Optimizing cost flows by edge cost and capacity upgrade," *Journal of Discrete Algorithms*, vol. 2, no. 4, pp. 407–423, 2004.
- [19] P. Detti, C. Meloni, and M. Pranzo, "Minimum dominating trail set for two-terminal series parallel graphs," *Electronic Notes in Discrete Mathematics*, vol. 17, pp. 117–122, 2004.
- [20] M. Didi Biha and A. R. Mahjoub, " k -edge connected polyhedra on series parallel graphs," *Operations Research Letters*, vol. 19, no. 2, pp. 71–78, 1996.
- [21] L. Finta, Z. Liu, I. Milis, and E. Bampis, "Scheduling UET-UCT series-parallel graphs on two processors," *Theoretical Computer Science*, vol. 162, no. 2, pp. 323–340, 1996.
- [22] J. Fonlupt and D. Naddef, "The traveling salesman problem in graphs with some excluded minors," *Mathematical Programming*, vol. 53, no. 2, Ser. A, pp. 147–172, 1992.
- [23] H. Kervin and A. R. Mahjoub, "On survivable network polyhedra," *Discrete Mathematics*, vol. 290, no. 2–3, pp. 183–210, 2005.
- [24] M. Laurent, "The real positive semidefinite completion problem for series-parallel graphs," *Linear Algebra and its Applications*, vol. 252, pp. 347–366, 1997.
- [25] T. Nishizeki, J. Vygen, and X. Zhou, "The edge-disjoint paths problem is NP-complete for series-parallel graphs," *Discrete Applied Mathematics*, vol. 115, no. 1–3, pp. 177–186, 2001.
- [26] J. Rader and G. J. Woeginger, "The quadratic 0-1 knapsack problem with series-parallel support," *Operations Research Letters*, vol. 30, no. 3, pp. 159–166, 2002.
- [27] P. D. Seymour, "On odd cuts and plane multicommodity flows," *Proceedings of the London Mathematical Society*, vol. 3–42, no. 1, pp. 178–192, 1981.
- [28] Y. L. Wang, R. S. Chang, and C. C. Hsu, "A note on the tour problems in two-terminal series-parallel graphs," *Information Sciences*, vol. 72, no. 1–2, pp. 179–189, 1993.
- [29] S. Xu, "The line index and minimum cut of weighted graphs," *European Journal of Operational Research*, vol. 109, no. 3, pp. 672–685, 1998.
- [30] C.-C. Yen and R. C. T. Lee, "A linear time algorithm to solve the weighted perfect domination problem in series-parallel graphs," *European Journal of Operational Research*, vol. 73, no. 1, pp. 192–198, 1994.
- [31] X. Zhou, Y. Matsuo, and T. Nishizeki, "List total colorings of series-parallel graphs," *Journal of Discrete Algorithms*, vol. 3, no. 1, pp. 47–60, 2005.