Egerváry Research Group
on Combinatorial Optimization

Technical reportS

TR-2014-05. Published by the Egerváry Research Group, Pázmány P. sétány 1/C, H–1117, Budapest, Hungary. Web site: www.cs.elte.hu/egres . ISSN 1587–4451.

Nonseparating cycles in planar and Eulerian graphs

Attila Bernáth and Marcin Kamiński

July 2014

Nonseparating cycles in planar and Eulerian graphs

Attila Bernáth* and Marcin Kamiński**

Abstract

A cycle $C \subseteq E$ in a graph $G = (V, E)$ is *nonseparating* if $G - C$ is connected (note that we only delete the edges of the cycle). We study the algorithmic problem of deciding whether a graph contains a nonseparating cycle. We show how to solve it in polynomial-time in planar graphs and that it is NP-complete in Eulerian graphs. The former was an open problem raised in [1]; the latter is a natural question in the context of greedy improvements of feasible solutions to the graphic TSP Problem.

1 Introduction

In this note we consider undirected graphs with possible parallel edges and loops. We use standard terminology and refer the reader to [3] for what is not defined here. For some subset of edges $C \subseteq E$ of a graph $G = (V, E)$ let $V(C) = \{v \in V : \exists e \in C$ with $v \in e\}$. A *cycle* in a graph $G = (V, E)$ is a subset of edges $C \subseteq E$ such that $(V(C), C)$ is connected and every node in $V(C)$ is incident with two edges of C. A cycle $C \subseteq E$ in a graph $G = (V, E)$ is *nonseparating* if $G - C$ is connected (that is, we only remove the edges of the cycle). There are two similar notions that should not be confused with ours. The first is a **non-separating cycle in a graph embedded on a surface** which is a cycle whose removal from the embedded graph does not disconnect the surface. The second is the notion of a **peripheral cycle** which is an induced cycle – as the set of vertices (and not edges) – whose removal does not disconnect the graph.

The study of nonseparating cycles from the algorithmic perspective was initiated in [1]. The algorithmic problem of deciding whether a graph has a nonseparating cycle was considered there and shown NP-complete.

Theorem 1 (Theorem 2.6 in [1])**.** *It is NP-complete to decide whether a graph has a nonseparating cycle.*

*Hungarian Academy of Sciences, Institute for Computer Science and Control (MTA-SZTAKI), Budapest, Hungary. Research supported by the ERC StG project PAAl no. 259515, the "Momentum - Big Data" grant of the Hungarian Academy of Sciences and OTKA NK 105645. E-mail: `bernath@cs.elte.hu`

**University of Warsaw, Institute of Computer Science. Email: `mjk@mimuw.edu.pl`

Many other related problems were considered in [1]. For example, the following open question was posed there: what is the computational complexity of deciding whether a planar graph has a nonseparating cycle? We give an answer to this question and show how to find a nonseparating cycle in planar graphs in polynomial time. We also study the problem of deciding whether an Eulerian graph has a nonseparating cycle and we prove that it is NP-complete. Here is our motivation to look at this problem.

The *Graphic TSP* problem is the problem of finding in a connected (unweighted) graph a shortest tour that visits every vertex at least once. Given a graph $G = (V, E)$, let $2G = (V, 2E)$ be the graph that we obtain from G by duplicating every edge. The Graphic TSP problem can then be formulated as follows (see [6]).

Problem 1. *Given a connected graph $G = (V, E)$, find a connected Eulerian subgraph of $2G$ (spanning the whole set V) that has the smallest possible number of edges.*

The graphic TSP problem is a special case of the Metric TSP problem for which the best known approximation algorithm is the well-known 1.5-approximation algorithm due to Christofides [2]. The best known approximation algorithm for the Graphic TSP has approximation factor 1.4 [6]. We can observe the following connection between the Graphic TSP problem and the nonseparating cycle problem. Given a feasible solution to the graphic TSP problem H (which is an Eulerian subgraph of $2G$), a greedy way of improving the quality of H would be to delete edges from it, while maintaining feasibility. Obviously, there is such a greedy improvement if and only if the multigraph H has a nonseparating cycle. We show that it is NP-complete to decide whether such a greedy improvement exists or not.

2 Planar graphs

In this section we consider the problem of finding a nonseparating cycle in a planar graph. We give an algorithm that solves this problem in polynomial time. The base of the algorithm is the the following lemma.

Lemma 1. *Let $G = (V, E)$ be a 2-connected plane graph. Let $F \subseteq E$ be a face of G. If $G - F$ is disconnected and C is a nonseparating cycle in G, then $V(C)$ intersects at most one connected component of $G - F$.*

Proof. In 2-connected planar graphs every face is bounded by a cycle. Since G is 2-connected, F is a cycle. Let us suppose that $G - F$ is disconnected and let (V_i, E_i) be the connected components of $G - F$ (for $i = 1, \dots, k$, for some integer $k \geq 2$). Thus, $V_1, \dots, V_k$ is a partition of V and $F, E_1, \dots, E_k$ is a partition of E. Also suppose that there exists a cycle $C \subseteq E$ such that $G - C$ is connected. We will show that C intersects only one of V_i's.

Let us assume that C contains vertices of at least two V_i's. Clearly, there must exist an edge $e = x_1x_2 \in F$ with $x_1, x_2 \in V(C)$ such that x_1 and x_2 are in two different sets, say $x_1 \in V_1$ and $x_2 \in V_2$. We will say that the vertices and the edges in V_1 and E_1 are blue and those in V_2 and E_2 are red. Furthermore, the edges in F are called black. (See Figure 1 for an illustration: note that our figures are in colour.) Without loss of generality, x_2 is the clockwise neighbour of x_1 on the face F and let us number all the other vertices of the face F in the clockwise order as well, $x_1, x_2, \dots, x_n$. Let $x_t \in V_2 \cap V(F)$ be the farthest red vertex from x_2 on the face F in clockwise direction and $f = x_t x_{t+1} \in F$ be the edge incident to x_t in the clockwise direction (we allow $x_t = x_2$ or $x_t = x_n$; in the latter case $x_{t+1} = x_1$). We claim that in $G - \{e, f\}$ (that is, we only remove the edges, but leave their endpoints) there is no path between x_1 and x_2. This implies that there are at most two edge-disjoint paths between x_1 and x_2 in G. (In fact, precisely two, by the two paths using the face F). However, the properties of C imply that there are at least three edge-disjoint paths between x_1 and x_2 in G; a contradiction.

The claim is clearly true if $t = 2$ (i.e. there is only one red vertex on the face F), so from now on we assume that $t > 2$. Let us take a red path $P \subseteq E_2$ between x_2 and x_t. Such a path exists since (V_2, E_2) is connected. Consider the closed plane curve obtained from P by connecting its endpoints (informally, add a "chord" of the circle F between x_2 and x_t, that is not an edge of G). Since F is a face of the plane graph G, this will be simple curve (that is, homeomorphic to the circle). This curve separates x_1 from all the edges of $F - \{e, f\}$ incident to the red vertices. But if there exists a path $P' \subseteq E - \{e, f\}$ between x_1 and x_2, then take the edge $xy \in P'$ so that y is the first red vertex on the path P' when we start from x_1 (and x precedes it). Clearly, this edge must be black (otherwise x was also red); a contradiction. □

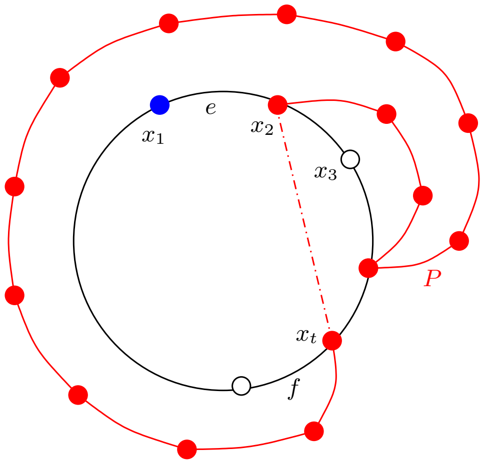

Figure 1: Illustration for the proof of Lemma 1.

Lemma 1 implies a divide-and-conquer polynomial-time algorithm for the problem of finding a nonseparating cycle. We sketch the algorithm below (see Algorithm `FindNonSeparatingCycle(`G`)`). The proof of the following theorem is straightforward from Lemma 1 and the algorithm.

Theorem 2. *There exists a polynomial-time algorithm to decide whether a planar graph has a nonseparating cycle.*

Algorithm FindNonSeparatingCycle(G)

begin

INPUT: A planar graph G (given with a planar representation)

OUTPUT: YES, if G has a nonseparating cycle, NO otherwise

1.1. If G is not 2-connected, then iterate through the blocks of G.

1.2. If G is 2-connected, take an arbitrary face $F \subseteq E$ of G.

1.3. If $G - F$ is connected, then return YES.

1.4. Otherwise, let (V_i, E_i) $(i = 1, \ldots, k)$ be the connected components of $G - F$. For $i = 1, \ldots, k$ do

1.5. If FindNonSeparatingCycle($(V_i, E_i \cup F_i)$), then return YES (where $F_i = \{uv \in F : u, v \in V_i\}$. Note that we need these edges, see Figure 2).

1.6. Return NO.

end

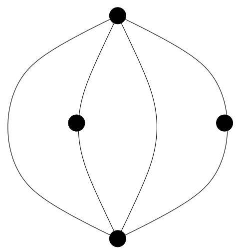

Figure 2: Illustration for the algorithm.

We mention that planar duality gives that the Problem of finding a nonseparating cycle in a planar graph is eqivalent to finding a cut in a planar graph that contains no circuit. By Theorem 2, this problem is polynomially solvable for planar graphs. However, van den Heuvel [5] has shown that this problem is NP-complete in general (i.e. not necessarily planar) graphs.

We can clearly modify the algorithm above to solve the following problem.

Problem 2. *Given a graph $G = (V, E)$ embedded in the plane and an edge $e \in E$, find a cycle $C \subseteq E$ containing e such that $G - C$ is connected.*

A problem related to the problem of finding a nonseparating cycle in a graph is the following.

Problem 3. *Given a graph $G = (V, E)$ and two vertices $s, t \in V$, find an $s - t$ path $P \subseteq E$ so that $G - P$ is connected.*

This problem was shown to be NP-complete in [1], but it was stated as an open problem for a planar graph G. If s and t fall in the same face of G then we can solve the problem as follows: connect s and t with a new edge e and solve Problem 2 in the obtained planar graph G' and this edge e. However, we don't know the status of Problem 3 for planar graphs in general.

3 Nonseparating cycles in Eulerian graphs

In this section we consider the problem of finding a nonseparating cycle in an Eulerian graph. This problem is motivated by the graphic TSP problem, as mentioned in the Introduction.

Theorem 3. *It is NP-complete to decide whether an Eulerian graph H has a nonseparating cycle.*

Proof. The proof is similar to the proof of Theorem 2.6 in [1]. It is clear that the problem is in NP. The completeness will be shown by a reduction from the well known NP-complete problem 3SAT (Problem LO2 in [4]). Let φ be a 3-CNF formula with variable set $\{x_1, x_2, \ldots, x_n\}$ and clause set $\mathcal{C} = \{C_1, C_2, \ldots, C_m\}$. Assume that literal x_j appears in k_j clauses $C_{a_1^j}, C_{a_2^j}, \ldots, C_{a_{k_j}^j}$, and literal $\overline{x_j}$ occurs in l_j clauses $C_{b_1^j}, C_{b_2^j}, \ldots, C_{b_{l_j}^j}$. Construct the following graph $G_\varphi = (V, E)$. For any clause $C \in \mathcal{C}$ and any literal y occurring in C introduce a vertex $v(y, C)$. Furthermore, for every variable x_j introduce 4 new vertices z_j^1, z_j^2, p_j, q_j. For every variable x_j, let G_φ contain a cycle on the $k_j + l_j + 4$ vertices $z_j^1, v(x_j, C_{a_1^j}), v(x_j, C_{a_2^j}), \ldots, v(x_j, C_{a_{k_j}^j}), p_j, z_j^2, q_j, v(\overline{x_j}, C_{b_{l_j}^j}), v(\overline{x_j}, C_{b_{l_j-1}^j}), \ldots, v(\overline{x_j}, C_{b_1^j})$ in this order. We say that this cycle is *associated to the variable* x_j. The edges of these cycles associated to the variables will be called *black*, all other edges are *red*. Add a cycle of length 2 on the vertices p_j, q_j. Identify the vertex pairs z_j^2, z_{j+1}^1 for every $j = 1, 2, \ldots, n$ (where we mean $z_{n+1}^1 = z_1^1$).

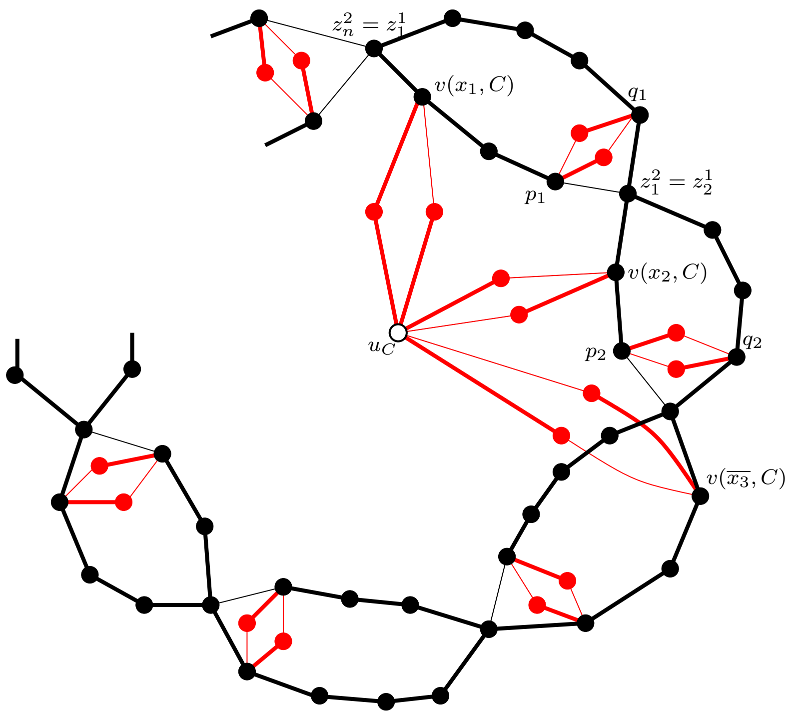

Figure 3: Part of the construction of graph G_φ for clause $C = \{x_1, x_2, \overline{x_3}\}$.

For every clause $C \in \mathcal{C}$ we introduce a new vertex u_C. Introduce a cycle of length 2 on $u_C v(y,C)$for every clause C and every literal y in C (i.e., the vertices u_C will have degree 6 and the ones $v(y,C)$ will have degree 4). Finally, put a subdivision vertex on every red edge. The construction of the graph G_φ is finished. Clearly, G_φ is Eulerian. An illustration can be found in Figure 3.

Claim 1. *The formula φ is satisfiable if and only if G_φ has a nonseparating cycle.*

The rest of the proof is a proof of the claim (in this proof letter K will be used for cycles, since letter C is used for clauses). If τ is a truth assignment to the variables $x_1, x_2, \ldots, x_n$ then we define a cycle K_τ in G_φ as follows: for every $j = 1, 2, \ldots, n$, if x_j is set to TRUE then let K_τ go through the vertices $z_j^1, v(\overline{x_j}, C_{b_1^j}), v(\overline{x_j}, C_{b_2^j}), \ldots, v(\overline{x_j}, C_{b_{l_j}^j}), q_j, z_j^2$, otherwise (i.e., if x_j is set to FALSE) let K_τ go through $z_j^1, v(x_j, C_{a_1^j}), v(x_j, C_{a_2^j}), \ldots, v(x_j, C_{a_{k_j}^j}), p_j, z_j^2$. If the truth assignment τ satisfies φ then the cycle K_τ is a nonseparating cycle: since every clause C contains a true literal in τ we can reach u_C in $G_\varphi - K_\tau$ from z_1^1, therefore we can reach every node of V in $G_\varphi - K_\tau$ from z_1^1.

On the other hand assume that G_φ contains a nonseparating cycle K. Note that K can only use black edges due to the subdivision vertices on the red edges. Furthermore, K cannot be the cycle associated to a variable x_j due to the subdivision vertices on the red edges connecting p_j and q_j. Therefore K naturally defines a truth assignment τ as above: for any $j = 1, 2, \ldots, n$, if K goes through the vertices $z_j^1, v(\overline{x_j}, C_{b_1^j}), v(\overline{x_j}, C_{b_2^j}), \ldots, v(\overline{x_j}, C_{b_{l_j}^j}), q_j, z_j^2$ then let $\tau(x_j)$ be TRUE, otherwise (i.e. if K goes through $z_j^1, v(x_j, C_{a_1^j}), v(x_j, C_{a_2^j}), \ldots, v(x_j, C_{a_{k_j}^j}), p_j, z_j^2)$ then let $\tau(x_j)$ be FALSE. By the nonseparating property of K, the assignment τ satisfies the formula φ, finishing our prof. □

Recall that given a graphic TSP instance with graph $G = (V, E)$, the algorithm of Christofides works as follows. First, it finds an arbitrary spanning tree $F \subseteq E$ in G. Second, it finds a minimum cost T_F-join $J_F \subset E$ to correct the parities of the vertices, where T_F is the set of vertices that have an odd degree in the spanning tree F and the cost is one for every edge. Finally the algorithm outputs $F + J_F$ where this $+$ means that edges in $F \cap J_F$ are doubled in the output (so the output is a subgraph of $2G$). We note that a slight modification of our construction above might be the output of this algorithm: in Figure 3 the bold edges form a spanning tree F for which the optimal T_F-join is the rest of the edges (if we add some more subdivision nodes on the thick red edges). This remark indicates that this simple greedy heuristic to improve the Christofides algorithm is NP-hard to implement.

References

[1] Attila Bernáth and Zoltán Király, *On the tractability of some natural packing, covering and partitioning problems*, Proceedings of the 7th Hungarian-Japanese

Symposium on Discrete Mathematics and Its Applications, May 31 - June 3, 2011, RIMS, Kyoto University, Kyoto, Japan, 2011, pp. 19–28.

[2] Nicos Christofides, *Worst-case analysis of a new heuristic for the travelling salesman problem*, Tech. report, CMU, research Report, 1976.

[3] Reinhard Diestel, *Graph theory*, Springer-Verlag, Electronic Edition, 2005.

[4] Michael R Garey and David S Johnson, *Computers and intractability: A guide to the theory of NP-completeness*, W. H. Freeman & Co., New York, NY, USA, 1979.

[5] Jan van den Heuvel, *Private communication.*

[6] András Sebő and Jens Vygen, *Shorter tours by nicer ears: 7/5-approximation for graphic tsp, 3/2 for the path version, and 4/3 for two-edge-connected subgraphs*, arXiv preprint arXiv:1201.1870 (2012), (to appear in Combinatorica).