A NOTE ON THE COMPLEXITY OF COMPUTING CYCLICITY

RICHARD HAMMACK

ABSTRACT. The *cyclicity* of a graph is the largest integer n for which the graph is contractible to the cycle on n vertices. We prove that, for n greater than three, the problem of determining whether an arbitrary graph has cyclicity n is NP-hard. We conjecture that the case $n = 3$ is decidable in polynomial time.

1. INTRODUCTION

A simple graph $G = (V(G), E(G))$ is *contractible* to the graph $H = (V(H), E(H))$ if there is a partition $\{V_y | y \in V(H)\}$ of $V(G)$ for which the subgraph of G induced by V_y is connected for each $y \in V(H)$, and some edge of G joins V_y to V_z if and only if $yz \in E(H)$.

Recent publications have focused on graph cyclicity. The *cyclicity* of a graph is the largest integer n for which the graph is contractible to C_n, the cycle on n vertices. Cyclicity was introduced in [3] as an aid in the study of a related invariant called *circularity* [1, 2, 6]. In [5], formulas are given for cyclicity in several classes of graphs, and a polynomial-time algorithm for computing cyclicity of planar graphs is described. Such results lead one to ask if there is an efficient algorithm for computing the cyclicity of an arbitrary graph. This article casts doubt on the existence of such an algorithm, by showing that the problem of deciding if a graph can be contracted to C_r is NP-complete for $r \geq 5$. Since a graph has cyclicity n if and only if it is contractible to C_n and *not* contractible to C_{n+1}, it follows that the question as to whether a graph has cyclicity n is NP-hard for $n \geq 4$. More precisely, it follows that the question is in the class P^{NP} when $n \geq 5$, and is at least co-NP when $n = 4$. The case $n = 3$ remains open, and may well be tractable.

In what follows, the subgraph of G induced by a set $V \subseteq V(G)$ is denoted by $G[V]$. A contraction of G to C_r will typically be described as a partition

$\{V_0, V_1, V_2, \cdots, V_{r-1}\}$ of $V(G)$, indexed over the cyclic group $\mathbb{Z}_r = \mathbb{Z}/r\mathbb{Z}$, with each $G[V_k]$ connected, and with an edge joining V_k to V_l exactly when $k = l \pm 1$. For the rest of this article, r is a fixed interger no smaller than 5.

2. Transformation

We now begin our proof that deciding if a graph is contractible to C_r is NP-complete when $r \geq 5$. The result is obtained by producing a polynomial transformation of the known NP-complete problem 3SAT ([4], Theorem 3.1) to the problem of contracting a graph to C_r.

Recall that an instance of 3SAT consists of a pair (U, C), where $U = \{U_1, U_2, \cdots, U_m\}$ is a set of variables, and $C = \{C_1, C_2, \cdots, C_n\}$ is a set of clauses. Each clause is a set of three literals: $C_j = \{l_{j1}, l_{j2}, l_{j3}\}$. Each literal is either a variable U_i or its negation $\overline{U_i}$. The variables can take on either of the two values 1 ("true") or 0 ("false"), where $\overline{1} = 0$ and $\overline{0} = 1$. A clause is *satisfied* if at least one of its three literals has a value of 1. The problem 3SAT consists of determining if (U, C) is *satisfiable*, that is, if there exists a truth assignment to the variables that simultaneously satisfies all clauses. For convenience of indexing, we regard the values of 0 and 1 that the variables U_i can take on as being $0, 1 \in \mathbb{Z}_r$.

This section describes how to transform an instance (U, C) of 3SAT to a graph G, and the following section will prove G is contractible to C_r if and only if (U, C) is satisfiable. We build G in three stages. The first stage is construction of a graph G_T whose contractions to C_r are in bijective correspondence with the truth assignments of the variables in $U = \{U_1, U_2, \cdots, U_m\}$. In the lexicon of [4], G_T is a "truth-setting component." The graph $G_T = (V_T, E_T)$ is described as follows.

$$\begin{aligned} V_T = \; & \{v_k \mid k \in \mathbb{Z}_r\} \cup \{u_i(k) \mid 1 \leq i \leq m,\ k \in \mathbb{Z}_r\} \\ E_T = \; & \{v_k v_{k+1} \mid k \in \mathbb{Z}_r\} \cup \\ & \{u_i(k)u_i(k+1),\ v_k u_i(k),\ v_k u_i(k+1) \mid 1 \leq i \leq m,\ k \in \mathbb{Z}_r\} \end{aligned}$$

The bold lines in Figure 1 show G_T for a case $U = \{U_1, U_2, U_3\}$, and $r = 5$.

The second stage in constructing of G involves adding to G_T "satisfaction-testing" components, which are encodings of the clauses. This consists of adding the following new vertices and edges.

$$V_S = \{c_j \mid 1 \leq j \leq n\}, \quad E_S = \{c_j v_0,\ c_j v_2 \mid 1 \leq j \leq n\}$$

The the dashed lines in Figure 1 show satisfaction-testing edges E_S.

The third stage involves addition of "communication edges" E_C (shown dotted in Figure 1) relating the satisfaction-testing and truth-setting components.

$$E_C = \{c_j u_i(1) \mid 1 \le j \le n,\ l_{jp} = U_i,\ 1 \le p \le 3\}$$
$$\cup \{c_j u_i(2) \mid 1 \le j \le n,\ l_{jp} = \overline{U_i},\ 1 \le p \le 3\}$$

Thus, G is defined as $V(G) = V_T \cup V_S$ and $E(G) = E_T \cup E_S \cup E_C$. Figure 1 is an illustration of G.

3. Result

The previous section described how to transform and instance (U, C) of 3SAT to a graph G. Now this transformation is used to prove our result. The following lemmas prove that (U, C) is satisfiable if and only if G is contractible to C_r.

Lemma 1: *If an instance (U, C) of 3SAT is satisfiable, then the corresponding graph G is contractible to C_r.*

Proof. Suppose each variable in $U = \{U_1, \cdots, U_m\}$ can be given an assignment of 0 or 1, so that the clauses $C = \{C_1, \cdots, C_n\}$ are simultaneously satisfied. We describe a contraction of G to C_r. Consider the following partition of $V(G)$, with indices from $\mathbb{Z}_r$:

$$\begin{aligned}
V_0 &= \{v_0\} \cup \{u_i(0 + \overline{U_i}) \mid 1 \le i \le m\},\\
V_1 &= \{v_1\} \cup \{u_i(1 + \overline{U_i}) \mid 1 \le i \le m\} \cup \{c_j \mid 1 \le j \le n\},\\
V_2 &= \{v_2\} \cup \{u_i(2 + \overline{U_i}) \mid 1 \le i \le m\},\\
V_3 &= \{v_3\} \cup \{u_i(3 + \overline{U_i}) \mid 1 \le i \le m\},\\
&\vdots\\
V_{r-1} &= \{v_{r-1}\} \cup \{u_i(r - 1 + \overline{U_i}) \mid 1 \le i \le m\}.
\end{aligned}$$

The idea is that, for each i and k, $u_i(k) \in V_{k-1}$ if $U_i = 0$, but the condition $U_i = 1$ bumps $u_i(k)$ into V_k. We claim that this partition defines a contraction of G to C_r. First, note that each $G[V_k]$ is connected: This is immediate for $k \ne 1$, since v_k is adjacent to each $u_i(k + \overline{U_i})$ (by an edge in E_T) regardless of whether U_i has value 0 or 1. It is only slightly more trouble to show that $G[V_1]$ is connected. The subgraph of $G[V_1]$ induced by $\{v_1\} \cup \{u_i(1+\overline{U_i}) \mid 1 \le i \le m\}$ is connected for the same reason that $G[V_k]$ is connected for $k \ne 1$. To prove $G[V_1]$ is connected, it suffices to show that each c_j is adjacent to some $u_i(1 + \overline{U_i})$. This is where the communication edges E_C come in. Since C_j is satisfied, one of its literals l_{jp} is true. If

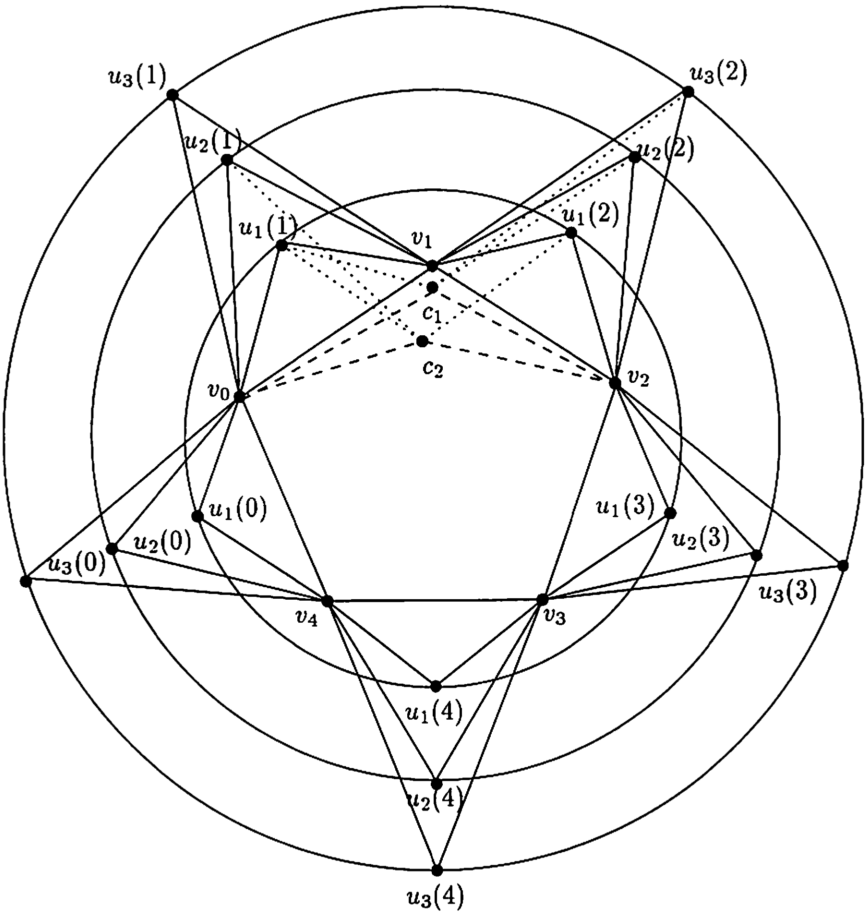

FIGURE 1. An example of G arising from the instance $U = \{U_1, U_2, U_3\}$, $C_1 = \{U_1, \overline{U_2}, \overline{U_3}\}$, $C_2 = \{U_1, \overline{U_1}, U_2,\}$ of 3SAT, and with $r = 5$.

$l_{jp} = U_i$, then $U_i = 1$, and $c_j u_i(1) = c_j u_i(1 + \overline{U_i})$ is a communication edge joining c_j to $u_i(1 + \overline{U_i})$. On the other hand, if $l_{jp} = \overline{U_i}$, then $U_i = 0$, and $c_j u_i(2) = c_j u_i(1 + \overline{U_i})$ is a communication edge joining c_j to $u_i(1 + \overline{U_i})$. Thus is $G[V_1]$ connected.

Next, we confirm that some edge of G joins V_k to V_l if and only if $k = l \pm 1$. If $k = l \pm 1$, then $v_k v_l \in E_T$ joins V_k to V_l. Conversely, suppose

$e \in E(G) = E_T \cup E_S \cup E_C$ joins V_k to V_l $(k \neq l)$. The cases $e \in E_T$, $e \in E_S$, and $e \in E_C$ are considered separately. If $e \in E_T$, then, by definition of E_T, either $e = v_q v_{q+1}$, or $e = u_i(q)u_i(q+1)$, or $e = v_q u_i(q)$, or $e = v_q u_i(q+1)$, for some i, q. In the first case, $v_q \in V_q$ and $v_{q+1} \in V_{q+1}$, so $k = l \pm 1$. In the second case, $u_i(q) \in V_{q-\overline{U_i}}$ and $u_i(q+1) \in V_{q+1-\overline{U_i}}$, so $k = l \pm 1$. In the last two cases, the edge starts at $v_q \in V_q$ and ends at either $u_i(q) \in V_{q-1} \cup V_q$ or $u_i(q+1) \in V_q \cup V_{q+1}$. Either way, $k = l \pm 1$. Next, suppose $e \in E_S$, so $e = c_j v_0$ or $e = c_j v_2$. Now, $c_j \in V_1$, $v_0 \in V_0$, and $v_2 \in V_2$. Thus e either joins V_0 to V_1 or V_1 to V_2, and either way, $k = l \pm 1$. Finally, suppose $e \in E_C$, so $e = c_j u_i(1)$ or $e = c_j u_i(2)$. Since $c_j \in V_1$, $u_i(1) \in V_0 \cup V_1$, and $u_i(2) \in V_1 \cup V_2$, it follows that $k = l \pm 1$. The proof is complete. ∎

Lemma 2: *If G is contractible to C_r, then the instance (U, C) of 3SAT is satisfiable.*

Proof. Let $\{V_k \mid k \in \mathbb{Z}_r\}$ be a partition of $V(G)$ giving a contraction of G to C_r, so each $G[V_k]$ is connected, and an edge joins V_k to V_l if and only if $k = l \pm 1$. In order to describe the truth assignments, it is necessary to first make several observations concerning this partition.

First note that, for each $k \in \mathbb{Z}_r$, V_k contains exactly one vertex of the r-cycle $Z = v_0 v_1 v_2 v_3 \cdots v_{r-1} v_0$. To see this, observe that, for any k, one of the sets V_k or V_{k+1} must intersect $V(Z)$. The reason is that each edge of G is either incident with Z, or lies on a triangle that shares a vertex with Z. Thus, any edge joining V_k to V_{k+1} either touches Z, or lies on a triangle that does. The vertices of such a triangle necessarily lie in $V_k \cup V_{k+1}$, so at least one of V_k or V_{k+1} contains vertices of Z. Now suppose there were a $k \in \mathbb{Z}_r$ for which $V_k \cap V(Z) = \emptyset$; we show this leads to a contradiction. Were there such a k, then Z would intersect V_{k+1} and V_{k-1} (by what was said above), and, since Z is connected, it would have vertices in each of the $r-1$ sets V_{k+1}, V_{k+2}, $V_{k+3} \cdots V_{k-1}$. The cycle Z could be viewed as starting in V_{k+1}, passing through V_{k+2}, V_{k+3}, etc., eventually going out as far as V_{k-1}, then returning to V_{k+1}. Then Z would have a positive even number of edges joining V_{k+s} to V_{k+s+1}, $1 \leq s \leq r-2$, and therefore it would have at least $2(r-2)$ edges. This is a contradiction since Z has exactly r edges, and $r < 2(r-2)$ because $r \geq 5$. (This is the only place the condition $r \geq 5$ is used.) We conclude that each V_k contains one (and hence, only one) vertex of Z.

By relabeling the indices of the sets V_k if necessary, we can (and do) assume $v_k \in V_k$ for each $k \in \mathbb{Z}_r$. Next, observe that, for any $1 \leq i \leq m$, each

V_k contains exactly one vertex of the r-cycle $u_i(0)u_i(1)\cdots u_i(r-1)u_i(0)$. The reason is that, since each $u_i(k)$ is ajdacent to both $v_{k-1} \in V_{k-1}$ and $v_k \in V_k$, it must be in either V_{k-1} or V_k. From this, it readily follows that if $u_i(1) \in V_0$ then $u_i(k) \in V_{k-1}$ for each k, while $u_i(1) \in V_1$ implies $u_i(k) \in V_k$ for each k.

Now the truth assignments can be made. Notice that, for a fixed i, $1 \le i \le m$, the vertex $u_i(1)$ is adjacent to both $v_0 \in V_0$ and $v_1 \in V_1$, so it must be in V_0 or V_1. Give each variable U_i the following truth assignment.

$$U_i = \begin{cases} 1 \text{ if } u_i(1) \in V_1 \\ 0 \text{ if } u_i(1) \in V_0 \end{cases}$$

Consider an arbitrary clause $C_j = \{l_{j1}, l_{j2}, l_{j3}\}$. The vertex c_j is adjacent to both $v_0 \in V_0$ and $v_2 \in V_2$, so it must be in V_1. Since $G[V_1]$ is connected, there must be a path in $G[V_1]$ joining c_j to $v_1 \in V_1$. This path begins with neither edge c_jv_0 nor c_jv_2, for these lead outside of $G[V_1]$. The only other edges incident with c_j are communication edges in E_C, so the path must begin with an edge $c_ju_i(1)$ or $c_ju_i(2)$. In the first case, $c_ju_i(1) \in E_C$ implies $l_{jp} = U_i$ for some $1 \le p \le 3$. Combine this with the fact that $u_i(1) \in V_1$ implies $U_i = 1$, and l_{jp} has a value of 1, so C_j is satisfied. In the second case, $c_ju_i(2) \in E_C$ implies $l_{jp} = \overline{U_i}$ for some $1 \le p \le 3$. Moreover, $u_i(2) \in V_1$ implies $u_i(1) \in V_0$, because, as mentioned above, V_1 can not contain more than one vertex of the cycle $u_i(0)u_i(1)u_i(2)\cdots u_i(r-1)u_i(0)$. Now, $u_i(1) \in V_0$ means $U_i = 0$, so the literal $l_{jp} = \overline{U_i}$ is true, and C_j is satisfied. In this way, every clause is satisfied, and the proof is complete. ■

Theorem 1: *The problem of determining whether a graph is contractible to C_r is NP-complete for $r \ge 5$.*

Proof. This problem is in the class NP, for the vertices of an arbitrary graph can be nondeterministically partitioned into r sets $\{V_k \mid k \in \mathbb{Z}_r\}$. Then, in polynomial time, it can be determined if this partition is a contraction to C_r by checking whether each $G[V_k]$ is connected, and if an edge joins V_k to V_l exactly when $k = l \pm 1 \pmod r$.

Lemmas 1 and 2 imply that we can transform any instance (U, C) of the NP-complete problem 3SAT to a graph G such that (U, C) is satisfiable if and only if G is contractible to C_r. Moreover, as G has $r + r|U| + |C|$ vertices, the number of steps needed for its construction is bounded by a polynomial function of the sizes of U and C. Hence, we have a polynomial transformation from 3SAT to the problem of contracting a graph to C_r. It follows that the problem of determining whether a graph can be contracted to C_r is NP-complete. ∎

4. Conclusion

We have seen that for $r \geq 5$ it is NP-complete to determine if a graph is contractible to C_r. Our method does not adapt to proving that the question of contracting to C_4 is NP-complete. The problem is that, if G is built by indexing over $\mathbb{Z}_4$ rather than $\mathbb{Z}_r$, the proof of Lemma 2 does not go through. There is a possibility that some V_k contains more than one of the vertices $\{v_i \mid i \in \mathbb{Z}_4\}$, and this allows cases where a contraction to C_4 does not yield a truth assignment that solves (U, C). It may be that it can be decided in polynomial time if G can be contracted to C_4. Work is currently underway to find an algorithm that does this.

As mentioned in the introduction, Theorem 1 implies that it is NP-hard to decide if a graph has cyclicity n for $n \geq 4$, and the question is in the class P^{NP} when $n \geq 5$, and is at least co-NP for cyclicity $n = 4$. What about cyclicity 3? It is easy to see that a connected graph is contractible to C_3 exactly when it contains a cycle, and this can be determined in polynomial time. A graph has cyclicity 3 if and only if it is contractible to C_3 and *not* contractible to C_4. This suggests an open problem.

Conjecture 1: *It can be decided in polynomial time if an arbitrary graph has cyclicity 3. Equivalently, it can be decided in polynomial time if an arbitrary graph can be contracted to C_4.*

References

[1] Bell, Brown, Dickman, and Green, "Circularity of graphs and continua: combinatorics," Houston J. Math., **6**: 455–469, 1980.

[2] Bell, Brown, Dickman, and Green, "Circularity of graphs and continua: topology," Fund. Math., **122**: 103–110, 1981.

[3] D. Blum, "Circularity of Graphs," PhD Thesis, Virginia Polytechnic Institute and State University, 1982.

[4] M. Garey and D. Johnson, *Computers and Intractability*, W. H. Freeman, San Francisco, 1979.

[5] R. Hammack, "Cyclicity of graphs," J. Graph Theory, **32(2)**: 160–170, 1999.

[6] R. Hammack, "Circularity of planar graphs," Houston J. Math., **25(2)**: 213–221, 1999.

(Richard Hammack) Department of Mathematics, Randolph-Macon College, P.O. Box 5005, Ashland, VA 23005, USA

E-mail address: rhammack@rmc.edu