MATHEMATICAL ENGINEERING TECHNICAL REPORTS

Finding Coherent Cyclic Orders in Strong Digraphs

Satoru IWATA and Takuro MATSUDA

METR 2006–21 March 2006

DEPARTMENT OF MATHEMATICAL INFORMATICS
GRADUATE SCHOOL OF INFORMATION SCIENCE AND TECHNOLOGY
THE UNIVERSITY OF TOKYO
BUNKYO-KU, TOKYO 113-8656, JAPAN

WWW page: http://www.i.u-tokyo.ac.jp/mi/mi-e.htm

Finding Coherent Cyclic Orders in Strong Digraphs

Satoru Iwata* Takuro Matsuda*

Abstract

A cyclic order in the vertex set of a digraph is said to be coherent if any arc is contained in a directed cycle whose winding number is one. This notion plays a key role in the proof by Bessy and Thomassé (2004) of a conjecture of Gallai (1964) on covering the vertex set by directed cycles. This paper presents an efficient algorithm for finding a coherent cyclic order in a strongly connected digraph, based on a theorem of Knuth (1974). With the aid of ear decomposition, the algorithm runs in $\mathrm{O}(nm)$ time, where n is the number of vertices and m is the number of arcs. This is as fast as testing if a given cyclic order is coherent.

1 Introduction

Let $G = (V, A)$ be a strogly connected digraph with vertex set V of cardinality n and arc set A of cardinality m. Assume that G has no loops. A linear order σ on V is a bijection from V to $\{1, \ldots, n\}$. A cyclic order arises from a linear order σ with the additional relation that the last vertex $\sigma^{-1}(n)$ is followed by the first vertex $\sigma^{-1}(1)$. With respect to a linear order σ, an arc $a = (u, v)$ is called a forward arc if $\sigma(u) < \sigma(v)$. Otherwise, it is called a backward arc. For a directed cycle C in G, the winding number $\mathrm{ind}(C)$ is the number of backward arcs in C. A directed cycle C is called consistent if $\mathrm{ind}(C) = 1$. A cyclic order is said to be coherent if each arc is contained in a consistent cycle.

The notion of coherent cyclic order was introduced by Bessy and Thomassé [1]. They showed that every strongly connected digraph has a coherent cyclic order. Then they gave a min-max theorem for directed graphs with coherent cyclic orders. As a consequence, they obtained an elegant proof of a long-standing conjecture of Gallai that any strongly connected digraph with stability number α can be spanned by α directed cycles. Subsequently, Sebő [5] provided a number of related results including a polynomial-time algorithm for finding a coherent cyclic order in a strongly connected digraph. This algorithm consists of $\mathrm{O}(nm)$ iterations. The main task in each iteration is to test if a given cyclic order is coherent, which can be done in $\mathrm{O}(nm)$ time. Hence the total running time bound is $\mathrm{O}(n^2m^2)$.

Sebő [5] also provided an extension of the min-max theorem to its weighted version. This is shown by reduction to a certain class of minimum cost flow problems, which can be solved in

*Graduate School of Information Science and Technology, University of Tokyo, Tokyo 113-8656, Japan (iwata@mist.i.u-tokyo.ac.jp).

O($nm \log n$) time by the algorithm of Orlin [3]. These results suggest the following two phase method to obtain a collection of at most α directed cycles covering the vertex set. Find a coherent cyclic order, and then solve the minimum cost flow problem. Since the bottleneck part is the first phase, we are interested in a faster algorithm for finding a coherent cyclic order.

Three decades earlier, Knuth [2] had already proved a related theorem on strongly connected graphs, which leads us to a simple alternative proof of the existence of a coherent cyclic order. A straightforward implementation of the original proof results in an O(m^2) algorithm for finding a coherent cyclic order. In this paper, we present an improved algorithm using ear decomposition. The resulting algorithm runs in O(mn) time, which is as fast as testing if a given cyclic order is coherent.

The outline of this paper is as follows. In Section 2, we provide an alternative proof of the theorem of Knuth using ear decomposion. An efficient implementation of this proof is described in Section 3. Finally, in Section 4, we discuss the running time bound of the algorithm.

2 Coherent Cyclic Orders via Ear Decomposition

In this section, we provide an alternative proof of the exisitence of a coherent cyclic order. For a directed path P in G, we denote by $V(P)$ and $A(P)$ the vertex set and the arc set of P, respectively.

An ear in G is a directed path P such that the in- and out-degrees of the inner vertices of P are all one. Let I denote the set of inner vertices of an ear P. Then we say that G is obtained by adding an ear P to the subgraph $G' = (V \setminus I, A \setminus A(P))$. An ear decomposition of G is a sequence $G_0, G_1, \ldots, G_k$ of subgraphs of G such that G_0 consists of only one vertex, G_i is obtained by adding an ear to G_{i-1} for $i = 1, \ldots, k$, and G_k coincides with G. The following lemma is well known to hold [4, Theorem 6.9].

Lemma 1. *A digraph is strongly connected if and only if it has an ear decomposition.*

The following theorem was shown by Knuth [2] as an application of a structural result which says that every strongly connected digraph has a "wheels-within-wheels" decomposition. We describe here an alternative proof based on ear decomposition.

Theorem 2 (Knuth [2])**.** *A strongly connected digraph $G = (V, A)$ with a specified vertex $v \in V$ has a parition of its arc set A into F and B such that the following three conditions hold.*

(i) *Every directed cycle in G contains an arc in B.*

(ii) *Any arc $a \in A$ is contained in a directed cycle that consists of a number of arcs in F and only one arc in B.*

(iii) *For each vertex $u \in V$, there exists a directed path from u to v that consists of arcs in F.*

Proof. Let $G_0, G_1, \ldots, G_k$ be an arbitrary ear decomposition of G. Suppose that the statement holds for G_j. We now intend to prove that it also holds for G_{j+1} obtained by adding an ear P_j to G_j. Let v be the initial vertex of P_j. Then the arc set A_j of G_j can be partitioned into F and B so that the three conditions are satisfied. Add the first arc of P_j to B and the rest to F. Then F and B form a partition of the arc set A_{j+1} of G_{j+1} satisfying the three conditions. □

The above partition of A into F and B yields a coherent cyclic order as follows. Since the subgraph $\vec{G} = (V, F)$ of G is acyclic by (i), there is a topological order, i.e., a linear order σ on V such that $\sigma(u) < \sigma(v)$ for every arc $a = (u, v) \in F$. It follows from (ii) and (iii) that the topological order σ is in fact a coherent cyclic order. Thus the existence of a coherent cyclic order can be regarded as a corollary of Theorem 2. The proof in Knuth [2] provides an efficient recursive algorithm for finding an appropriate partition in $\mathrm{O}(m^2)$ time.

3 Algorithm

In this section, we present an efficient algorithm for finding a coherent cyclic order. The algorithm makes use of ear decomposition of directed graphs.

The above proof of Theorem 2 has already suggested an approach to finding a coherent cyclic order via ear decomposition. In order to accomplish this approach, however, one needs to develop a procedure to update the partition when the target vertex v has changed.

Suppose that the arc set A of G is partitioned into F and B so that (i) and (ii) are satisfied. We now intend to update this partition so that (iii) also holds for $v \in V$. Let X be the set of vertices from which v is reachable through arcs in F. If X conincides with V, we may assert that (iii) holds. Otherwise, all the arcs entering X must belong to B, and all the arcs leaving X must belong to F. Then we interchange these arcs between F and B. The arc set F continues to be acyclic, and every directed cycle that has only one arc in B continues to enjoy this property. Consequently, the conditions (i) and (ii) continue to hold, whereas new vertices become reachable to v through the arcs in F. Therefore, repeating this process at most $\mathrm{O}(n)$ times, we will obtain a partition of A into F and B such that (i)–(iii) are all satisfied. We refer to this procedure as $\mathsf{Partition}(G, v)$.

We now describe the entire algorithm EDO (ear decomposition ordering) for finding a coherent cyclic order.

Algorithm EDO

Step 1: Select an arbitrary vertex $u \in V$. Initialize $U := \{u\}$, $F := \emptyset$, and $B := \emptyset$. Let $D(v)$ be the set of arcs emanating from v for each $v \in V$.

Step 2: Select a vertex $v \in U$ with $D(v) \neq \emptyset$.

Step 3: Apply $\mathsf{Partition}(\hat{G}, v)$ to $\hat{G} = (U, F \cup B)$.

Step 4: Repeat the following until $D(v) = \emptyset$.

- Select an arc $a = (v, w) \in D(v)$. Update $D(v) := D(v) \setminus \{v\}$ and $B := B \cup \{a\}$.
- Find a directed path P from w to U.
- Update $U := U \cup V(P)$ and $F := F \cup A(P)$.

Step 5: If $F \cup B \neq A$, then go to Step 2.

Step 6: Find a topological order on the subgraph $\vec{G} = (V, F)$ of G. The resulting order yields a coherent cyclic order in G.

4 Complexity

This section is devoted to complexity analysis of Algorithm EDO.

We first discuss the running time of $\mathsf{Partition}(G, v)$. This procedure performs $\mathrm{O}(n)$ iterations. In each iteration, the arcs that connect X and $V \setminus X$ are moved between B and F. Then all the end-vertices of these arcs become reachable to v through the arcs in F, and they will never be moved again. In order to identify X, the procedure performs the depth-first search in the reverse order, which scans each arc exactly once throughout the procedure. Thus the procedure runs in $\mathrm{O}(m)$ time.

We next consider the running time of Step 4. In order to find a path from w to U, the algorithm performs the depth-first search. Since the digraph $\hat{G}$ is strongly connected, it finds a desired path without doing back track. The arcs that are scanned in the depth-first search must be added to $\hat{G}$, and they will never be scanned again. Therefore, the running time spent for Step 4 is $\mathrm{O}(m)$ throughout the algorithm.

The algorithm consists of $\mathrm{O}(n)$ iterations. Since each iteration performs $\mathsf{Partition}(\hat{G}, v)$, which takes $\mathrm{O}(n)$ time, the entire algorithm runs in $\mathrm{O}(nm)$ time. Thus we obtain the following result.

Theorem 3. *Algorithm EDO finds a coherent cyclic order in* $\mathrm{O}(nm)$ *time.*

To make a comparison with this result, we now briefly describe the running time bound of the algorithm of Sebő [5], which repeatedly tests if a given cyclic order is coherent. The number of iterations is shown to be $\mathrm{O}(nm)$.

A natural way to test the coherence of a given cyclic order σ is as follows. A linear order τ is called a shift of σ if there there is a number ℓ such that $\tau(u) = \sigma(u) + \ell$ holds modulo n for every vertex $u \in V$. Note that a shift of a coherent cyclic order is also coherent. For each vertex $v \in V$, consider a shift τ such that $\tau(v) = 1$, and then check if every vertex $u \in V$ with $(u, v) \in A$ is reachable from v by a directed path that consists of forward arcs with respect to τ. This can be done by the depth-first search in $\mathrm{O}(m)$ time for each vertex v. The answer is positive for every vertex v if and only if σ is coherent. Thus the coherence of a cyclic order can be tested in $\mathrm{O}(nm)$ time. Therefore, the algorithm of Sebő finds a coherent cyclic order in $\mathrm{O}(n^2m^2)$ time.

Theorem 3 implies that Algorithm EDO is as fast as the above algorithm for testing coherence of a given cyclic order.

References

[1] S. Bessy and S. Thomassé: Three min-max theorems concerning cyclic orders of strong digraphs, *Integer Programming and Combinatorial Optimization*, D. Bienstock and G. Nemhauser, eds., LNCS 3064, Springer-Verlag, 2004, pp. 132–138.

[2] D. E. Knuth: Wheels within wheels, *J. Combinatorial Theory*, Ser. B, 16 (1974), pp. 42–46.

[3] J. B. Orlin: A faster strongly polynomial minimum cost flow algorithm, *Oper. Res.*, 41 (1993), pp. 338–350.

[4] A. Schrijver: *Combinatorial Optimization — Polyhedra and Efficiency*, Springer-Verlag, 2003.

[5] A. Sebő: Minmax relations for cyclically ordered digraphs, November, 2004.