COMBINATORICA
Bolyai Society – Springer-Verlag

COMBINATORICA **28** (1) (2008) 83–88
DOI: 10.1007/s00493-008-2305-1

FINDING COHERENT CYCLIC ORDERS IN STRONG DIGRAPHS

SATORU IWATA, TAKURO MATSUDA

Received May 9, 2006

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 strongly 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 a round if $\mathrm{ind}(C)=1$. A cyclic order is said to be coherent if each arc is contained in a round.

A linear order τ is called a cyclic shift of σ if there exists a number ℓ such that $\tau(u)=\sigma(u)+\ell$ holds modulo n for every vertex $u\in V$. We interpret that all the cyclic shifts represent the same cyclic order. Although taking a

Mathematics Subject Classification (2000): 05C20, 90C27 cyclic shift changes the sets of forward and backward arcs, it does not change the winding number of each directed cycle. This invariance of the winding number suggests that the coherence is in fact a property of a cyclic order.

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 digraphs 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 $\mathrm{O}(nm \log n)$ time by the algorithm of Orlin [3]. Based on these results, Sebő suggested 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. This requires $\mathrm{O}(n^2m^2)$ time in total. 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 digraphs, 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 $\mathrm{O}(m^2)$ algorithm for finding a coherent cyclic order. We improve this approach by making use of ear decompositions. The resulting algorithm runs in $\mathrm{O}(nm)$ time, which is as fast as testing if a given cyclic order is coherent. Thus we reduce the complexity of the algorithmic proof of Gallai's conjecture from $\mathrm{O}(n^2m^2)$ to $\mathrm{O}(nm \log n)$.

The outline of this paper is as follows. In Section 2, we show that the theorem of Knuth implies the existence of a coherent cyclic order. An efficient algorithm for finding a coherent cyclic order is described in Section 3. Finally, in Section 4, we discuss the running time bound of the algorithm.

2. Coherent Cyclic Orders via Ear Decompositions

In this section, we provide an alternative proof of the existence of a coherent cyclic order.

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. We denote by $V(P)$ and $A(P)$ the vertex set and the arc set of P, respectively. 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 $G_0, G_1, \dots, G_k$ 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 $i = 1, \dots, k$, 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, which can be constructed by a recursive algorithm in $\mathrm{O}(m^2)$ time. 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 partition 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 $G_0, G_1, \dots, G_k$ 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. Because of Lemma 3 below, it suffices to show that the statement holds for some specific vertex v. Let v be the initial vertex of P_j. Then by the inductive assumption the arc set A_j of G_j can be partitioned into F and B so that the three conditions are satisfied for v. 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 for v. ∎

Lemma 3 (Knuth [2]). *Given a partition of the arc set A of a strongly connected digraph $G = (V, A)$ into B and F such that the conditions (i) and (ii) hold, one can modify the partition so that (iii) also holds for a specified vertex $v \in V$.*

To prove Lemma 3, Knuth described the following procedure to update the partition. 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 coincides with V, then (iii) holds. Otherwise, all the arcs entering X must belong to B, and therefore it follows from (ii) that 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, which completes the proof of Lemma 3. We refer to this procedure as $\mathsf{Partition}(G,v)$. We will see in Section 4 that this procedure requires $\mathrm{O}(m)$ time.

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, which is defined to be a linear order σ on V such that $\sigma(u)<\sigma(v)$ for every arc $a=(u,v)\in F$. It follows from (ii) 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.

3. Algorithm

In this section, we present an efficient algorithm for finding a coherent cyclic order.

The above proof of Theorem 2 suggests an algorithm for finding a coherent cyclic order via ear decomposition. Starting with an arbitrary vertex $u \in V$, the algorithm grows a subgraph $\hat{G}$ with a partition of its arc set satisfying the three conditions for some vertex. The algorithm performs $\mathrm{O}(n)$ iterations. In each iteration, the algorithm finds a vertex v in $\hat{G}$ such that there exists an arc $a=(v,w)$ leaving $\hat{G}$ and applies $\mathsf{Partition}(G,v)$ so that v become reachable from any vertex in $\hat{G}$ through the arcs in F. Then it repeatedly adds ears starting from v to $\hat{G}$. The first arc of each ear is added to B, while the rest is added to F. The resulting subgraph $\hat{G}$ has a partition of its arc set satisfying the three conditions for v. The algorithm terminates when $\hat{G}$ coincides with G.

We now provide a formal description of this 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 $D(v) := D(v) \setminus \{a\}$ and $B := B \cup \{a\}$.
- If $w \notin U$, then find a directed path P from w to U, and 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 initial 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 starting with the set of vertices that become reachable in the previous iteration. Then the procedure identifies the set of arcs between X and $V \setminus X$ by examining the arcs incident to the set of new vertices that become reachable in the current iteration. All these operations scan each arc at most constant times 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. If the depth-first search visits an arc a, it finds an ear that uses a or it detects that there is no ear from v to U through a. In either case, the algorithm does not have to visit a again during the execution of Step 4. Thus one execution of Step 4 takes $\mathrm{O}(m)$ time.

Since each iteration adds at least one vertex to $\hat{G}$, the algorithm performs $\mathrm{O}(n)$ iterations. Both of Steps 3 and 4 in each iteration requires $O(m)$ time. Therefore, the entire algorithm runs in $\mathrm{O}(nm)$ time. Thus we obtain the following result.

Theorem 4. *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. 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 4 implies that Algorithm EDO is as fast as the above algorithm for testing coherence of a given cyclic order.

Acknowledgements

The authors are grateful to Yuji Matsuoka for discussions and to András Sebő for very helpful comments on the manuscript.

References

[1] S. Bessy and S. Thomassé: Three min-max theorems concerning cyclic orders of strong digraphs, in *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), 42–46.

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

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

[5] A. Sebő: Minmax relations for cyclically ordered digraphs, *J. Combinatorial Theory, Ser. B* **97(4)** (2007), 518–552.

Satoru Iwata

Research Institute for Mathematical Sciences
Kyoto University
Kyoto 606-8502
Japan
iwata@kurims.kyoto-u.ac.jp

Takuro Matsuda

Fujitsu
Shinkamata
Ohta-ku, Tokyo 144-8588
Japan