Algorithms for Shortest Paths and d-cycle Problems

SERGEI BESPAMYATNIKH, *Department of Computer Science, University of British Columbia, Vancouver, B.C. Canada V6T 1Z4, E-mail: besp@cs.ubc.ca*

ANDREI KELAREV, *Faculty of Science and Engineering, University of Tasmania, GPO Box 252-37, Hobart, Tasmania 7001, Australia, E-mail: kelarev@hilbert.maths.utas.edu.au*

ABSTRACT: Let G be a weighted graph with n vertices and m edges. We address the d-cycle problem, i.e., the problem of finding a subgraph of minimum weight with given cyclomatic number d. Hartvigsen [5] presented an algorithm with running time $O(n^2m)$ and $O(n^{2d-1}m^2)$ for the cyclomatic numbers $d = 1$ and $d \geq 2$, respectively. Using a $(d+1)$-shortest-paths algorithm, we develop a new more efficient algorithm for the d-cycle problem with running time $O(n^{2d-1} + n^2m + n^3 \log n)$.

Keywords: shortest path, cyclomatic number, d-cycle problem

1 Introduction

Let $G = (V, E)$ be a weighted undirected graph with n vertices and m edges, where all edges have nonnegative weights. As usual, we denote by $GF(2)$ the field of order two. The *incidence vector* of a cycle is the vector in the linear space $GF(2)^{|E|}$ with ones in all components corresponding to edges which occur in the cycle, and with zeros in all other components. The *cycle space* of G is the vector space over $GF(2)$ generated by all incidence vectors of cycles of G. Dimension of the cycle space of the graph G is denoted by $d(G)$ and is called the *cyclomatic number of* G. There is a well-known formula expressing $d(G)$ in terms of other parameters of the graph (see Proposition 2.1 below). The problem of finding a minimum weight subgraph of G with a given cyclomatic number d is called the *d-cycle problem*. The cyclomatic number d is considered as a constant.

The d-cycle problem is related to the problem of finding a *minimum cycle basis*, that is a minimum weight basis of the cycle space of G. Horton [6] obtained an $O(nm^3)$ algorithm for finding a minimum cycle basis of a graph. Hartvigsen [5] extended Horton's technique to minimum shortest paths (of course, a cycle can be considered as a closed path) and to more general linear spaces. For any $U \subseteq V \times V$, he defined the *U-space*, that is the vector space over $GF(2)$ generated by the incidence vectors of all cycles and paths from u to v, where $(u, v) \in U$. Hartvigsen found an algorithm

 © Hermes Science Publications

which gives a minimum U-basis in $O(n^2m^3)$ time. He used this algorithm to solve the d-cycle problem in $O(n^2m)$ and $O(n^{2d-1}m^2)$ time for cyclomatic numbers $d = 1$ and $d \geq 2$, respectively.

We prove that there exists a relation between the d-cycle problem and the k-shortest-paths problem, that is the problem of finding a list of k vertex-disjoint paths between all pairs of vertices in a graph whose weights form lexicographically smallest sequence (if there are several paths with equal sequence of weights, then we assume that the sequence of vertices is lexicographically smallest with respect to a given ordering of vertices). We show that the k-shortest paths for all pairs can be found in time $O(n^2m + n^3 \log n)$.

We develop a new more efficient d-cycle algorithm with running time

$$O(n^{2d-1} + n^2m + n^3 \log n).$$

It improves the algorithm of Hartvigsen by the factor $O(m^2)$ if $d \leq 4$ (for $d = 2$ and $m = \Omega(\sqrt{\log n})$ the factor is $\min(nm, m^2/\log n)$; for $d = 3$ the factor is $\min(m^2, n^3)$). This improvement has been achieved by reducing the d-cycle problem to the k-shortest-paths problem for $k = d + 1$.

2 Preliminaries

Throughout we assume that $G = (V, E)$ is an undirected graph without isolated vertices, loops or multiple edges. Denote the number of components of the graph G by $c(G)$, and the weight of an edge e by $w(e)$. We need the following simple formula for dimension of the cycle space of a graph or multigraph (see [7, Chap. 4]).

PROPOSITION 2.1
The cyclomatic number of a graph or multigraph $G = (V, E)$ is equal to $|E| - |V| + c(G)$.

The idea of using vertices of degree at least 3 occurs in [1] in the context of solving the planar k-cut problem and in [5] in the search for a minimum weight subgraph with a specified cyclomatic number.

PROPOSITION 2.2 ([5])
Let G be a graph with at least one edge and no vertices of degree one. Then G has at most $2d(G) - 2$ vertices of degree at least 3.

We need the following lower bound on the number of vertices of degree at least 3. Note that this lower bound meets the upper bound of Proposition 2.2 for graphs with $d(G) = 2$.

PROPOSITION 2.3
Every connected graph G with $d(G) \geq 2$ has at least one vertex of degree ≥ 3.

PROOF. Let G be a connected graph with $d(G) \geq 2$. Suppose to the contrary that G has no vertices of degree at least three. The deletion of a vertex of degree one preserves connectivity and the cyclomatic number of a graph. Hence we may assume that the graph G does not contain vertices of degree one. Then G is a cycle, and so $d(G) = 1$. This contradiction completes the proof. ∎

Figure 1 illustrates that, for each $d \geq 2$, there is a graph with cyclomatic number d and exactly one vertex of degree at least 3.

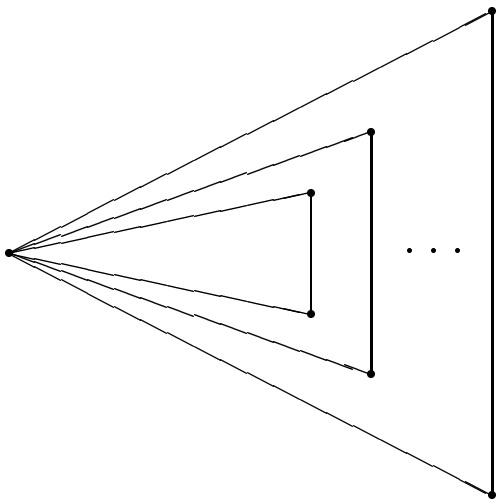

Fig. 1: A graph G with $d(G) \geq 2$ and one vertex v of degree ≥ 3.

3 Shortest paths

In this section we reduce the problem of finding the minimum weight subgraph with a specified cyclomatic number to the k-shortest-paths problem. The main idea behind this reduction is that every subgraph of this sort can be represented as a collection of shortest paths.

Two paths $u_1, \ldots, u_a$ and $v_1, \ldots, v_b$ are said to be *vertex-disjoint* if their internal vertices are disjoint, i.e. $u_i \neq v_j$ for all $i = 2, \ldots, a-1$ and $j = 2, \ldots, b-1$. Otherwise we say that the paths are *crossing*.

For any pair of vertices u, v of V, we are going to define certain vertex-disjoint paths from u to v. Take any positive integer k and consider the family $\mathcal{F}$ of all sets of pairwise vertex-disjoint paths from u to v such that each set in $\mathcal{F}$ contains precisely k paths. If $\mathcal{F}$ is empty then we take all sets of vertex-disjoint paths from u to v, which contain the maximum possible number $\ell < k$ of paths, and include all these sets in $\mathcal{F}$. In both the cases choose the set S in $\mathcal{F}$ such that the sequence of weights the paths of S forms lexicographically smallest word.

This is equivalent to saying that we choose paths in the set $\mathcal{F}$ in the following greedy fashion. First, we choose the shortest path $p_{1k}(u, v)$ among all paths in the sets in $\mathcal{F}$. Second, we find a shortest path $p_{2k}(u, v)$ with the property that both $p_{1k}(u, v)$ and $p_{2k}(u, v)$ belong to some sets in $\mathcal{F}$. Next, we find a shortest path $p_{3k}(u, v)$ among all paths such that $\{p_{1k}(u, v), p_{2k}(u, v), p_{3k}(u, v)\}$ is contained in some sets in $\mathcal{F}$. If the path $p_{3k}(u, v)\}$ does not exist, then we say that it is undefined. We continue this process until we have found k paths.

Figure 2 illustrates this preliminary definition with a graph G whose edges have weight 1, and shows that the greedy selection alone is not sufficient to define the k-shortest paths uniquely, and it is essential to add the condition that we choose lexicographically smallest sequence of weights of the paths we obtain in this greedy way.[1] In this example, the 2-shortest paths from a to f with respect to the ordinary ordering of vertices are the paths $abcf$ and $adef$. Note that the shortest path $abef$ cannot be supplemented with another vertex-disjoint path from a to f.

Thus, to avoid ambiguity, if there exist several sets S with equal sequences of weights, we fix an ordering of vertices of V, and choose the set S with the sequence of vertices which is lexicographically smallest with respect to this ordering. In this way the k-shortest-paths are defined uniquely. We denote these paths by $p_{1k}(u,v), p_{2k}(u,v), \ldots, p_{kk}(u,v)$. If the number k considered has been specified, then we omit the second index in this notation, and denote the k-shortest paths by $p_1(u,v), p_2(u,v), \ldots, p_k(u,v)$. If k is the largest positive integer such that there exist k vertex-disjoint paths from u to v, then for every $\ell > k$, the ℓ-shortest paths coincide with the k-shortest paths, and the paths $p_{k+1}(u,v), \ldots, p_\ell(u,v)$ are undefined.

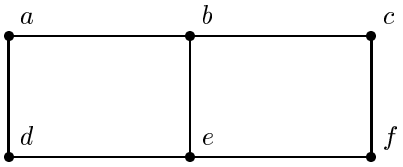

Fig. 2: 1-shortest path $(abef)$ and 2-shortest paths $(abcf, adef)$.

We say that a subgraph $G' = (V', E')$ of the graph G is *ℓ-based* if there is a set V'' of ℓ vertices of G' such that edges in E' can be partitioned into vertex-disjoint paths P with terminal vertices in V'' and the paths with the same terminal vertices u, v coincide with the k-shortest paths $p_1(u,v), p_2(u,v), \ldots$ in the whole graph G where k is the number of (u,v) paths in P. The set V'' will be called the *base* of G'.

LEMMA 3.1
Let $G' = (V', E')$ be a minimum weight subgraph of G with cyclomatic number d. Let u and v be vertices of G', and let P be a set of k vertex-disjoint paths in G' from u to v such that the graph $G' \setminus P = (V', E' \setminus P)$ has no paths from u to v. Then $k \leq d+1$ and if we modify the graph G' by replacing all paths of P by the k-shortest paths from u to v in G, then the weight of G' does not increase and the cyclomatic number of G' does not decrease.

PROOF. Since G' has k vertex-disjoint paths from u to v, it follows that all k-shortest paths in G' are defined. Denote these paths by $p_1, \ldots, p_k$. Evidently, the incidence vectors of the cycles (p_1, p_2), (p_1, p_3), …, (p_1, p_k) of G' are linearly independent in the cycle space. Hence the cyclomatic number $d(G')$ is not less than $k-1$. Since $d(G') \leq d$, we get $k \leq d+1$.

[1] The authors are grateful to a referee for this example and clarification.

Next, we find the shortest path $p(u, v)$ among all paths of P with the property that $p(u, v)$ does not belong to the set of k-shortest path from u to v in G.

If all edges of the k-shortest paths from u to v belong to E', then the assertion is obvious, because the graph G' remains unchanged when we replace P by the k-shortest paths. Suppose that one of these edges does not belong to E'.

Let $i \leq d + 1$ be the smallest index such that the i-th shortest path has an edge e that is not in E'. The weight of the path $p(u, v)$ is not less than the weight of the path $p_i(u, v)$ by the definition of $p_i(u, v)$ in view of its greedy nature. Replacing the path $p(u, v)$ in G' by $p_i(u, v)$ does not increase the weight of G'. It remains to show that the cyclomatic number does not decrease, either.

Denote the updated graph by F. Choose a basis in the cycle space of G' and denote by $A_1, \ldots, A_d$ the cycles of this basis. Suppose that the path $p(u, v)$ occurs in exactly m first cycles only. This means that each cycle A_j, where $j \leq m$, is the union of the path $p(u, v)$ and another path, say $q_j(u, v)$, from u to v. Take any $j \leq m$ and consider the cycle $A'_j = p_i(u, v) \cup q_j(u, v)$ of the graph F corresponding to the cycle A_j. If m is odd, then the edge e (chosen in the preceeding paragraph) occurs an odd number of times in the cycles $A'_1, \ldots, A'_m, A_{m+1}, \ldots, A_d$ and they are independent. If m is even, then the sets of cycles $A_1, \ldots, A_d$ and $A'_1, \ldots, A'_m, A_{m+1}, \ldots, A_d$ are both independent. Thus $c(F) \geq d$, which completes the proof. ∎

THEOREM 3.2
Let C be the minimum weight cycle of the graph G. There exists a $(2d - 2)$-based subgraph G'' of G such that either G'' or $C \cup G''$ is the minimum weight subgraph of G with cyclomatic number d.

PROOF. Let $G' = (V', E')$ be a minimum weight subgraph of G with cyclomatic number d. Let D_1 be the set of components of G' with cyclomatic number 1. Obviously, these components are cycles, because G' is minimal and we can delete all redundant edges and keep only the single cycle in each component in D_1. Denote by V_1 the set of all vertices of G' with degree at least 3. We see that all cycles in D_1 do not have any vertices in V_1. Let D_2 be the set of cycles in G' that have exactly one vertex in V_1. Put $d_1 = |D_1|$ and $d_2 = |D_2|$.

First, consider the case where $d = d_1 + d_2$. If two distinct cycles of G' have a common edge, then they have at least two common vertices of degree ≥ 3. This is impossible with cycles in $V_1 \cup V_2$. Therefore distinct cycles of $V_1 \cup V_2$ have no edges in common. We can keep the cycles of V_1 and V_2 in G', and remove all the edges not involved in these cycles without decreasing the cyclomatic number o G', because these cycles have linearly independent incidence vectors, and therefore ensure that the cyclmatic number of G' is at least $d_1 + d_2$. In view of the minimality og G' we see that G' coincides with the union (in another terminology, the sum) of all cycles in V_1 and V_2. Let C' be the minimum weight cycle of G'. If we replace the cycle C' by C in G', then the weight of G' does not increase and the cyclomatic number cannot become less than d, because there are d cycles in the new set $(\{C\} \cup V_1 \cup V_2) \setminus \{C'\}$, and obviously their incidence vectors are linearly independent. If, after that replacment, the new numbers d_1 and d_2 decrease and the inequality $d > d_1 + d_2$ holds, then we proceed as in the second case below. Thus in the first case we may assume that C is

in G' and $d = d_1 + d_2$.

Then we define the set V'' in the following way. First, we include the whole set V_1 into V''. Then, for each cycle of D_1, pick any two vertices and put them into V'' (evidently, these vertices don't belong to V_1, and so we have to add them to V''). Finally, for each cycle of D_2, pick any of its vertices which is not in V_1, and put it into V'', too.

Clearly, the total number of vertices in V'' is at most $2d - 2$. The edges $E' \setminus C$ can be partitioned into vertex-disjoint paths with endpoints in V''. Lemma 3.1 shows that they can be replaced by the corresponding k-shortest paths, and this will produce the $(2d - 2)$-based subgraph G'' such that $G' = G'' \cup C$.

Second, consider the case where $d > d_1 + d_2$. It is easily seen that V_1 has at most $2(d - d_1) - d_2 - 2$ vertices. Indeed, if we delete all edges of a cycle which is in D_1, then the set of vertices of degree at least 3 remains unchanged. Similarly, the deletion of edges of a cycle in D_2 either does not change the set of vertices of degree at least 3 at all, or decreases their number by one.

In this case we define the set V'' as follows. The set V'' contains

- all vertices of V_1,
- any two distinct vertices of each cycle in D_1,
- one vertex v chosen for each cycle in D_2 so that $v \notin V_1$.

The total number of vertices in V'' is at most $2d - 2$. All edges of E' can be partitioned into vertex-disjoint paths with terminal vertices in V''. Lemma 3.1 implies that they can be replaced by the corresponding k-shortest paths producing the $(2d - 2)$-based subgraph G'' that is the minimum weight subgraph of G with cyclomatic number d. This completes our proof. ∎

Therefore the following problem is of interest.

The k-Shortest Paths Problem. Given a graph $G = (V, E)$ and a positive integer k, find the k-shortest paths from u to v, for all pairs $(u, v) \in V \times V$.

4 A k-shortest-paths algorithm

The problem of finding shortest paths in graphs has been studied very well. A standard algorithm given by Dijkstra [2] allows one to compute a tree of shortest paths from any given source vertex to all other vertices of the graph $G = (V, E)$ with n vertices and m edges. Early implementation of Dijkstra's algorithm required time $O(n^2)$ or $O(m \lg n)$. Fredman and Tarjan [4] gave $O(m + n \lg n)$ implementation using Fibonacci heaps.

Eppstein [3] consider the problem of computing k shortest paths from a source in a weighted digraph. His algorithm takes $O(m + n \log n + k)$ time. Our shortest path problem is different from the one of Eppstein in that we look for the vertex-disjoint paths with lexicographically smallest sequence of weights.

Using the algorithm due to Fredman and Tarjan we can solve the $(d + 1)$-shortest-path problem in the following way. Fix any pair of vertices u and v. (Note that u and v may coincide.) First, we denote by $\overline{G}$ an isomorphic copy of the graph, which we are going to modify. The shortest path $p_1(u, v)$ from u to v can be computed in

$O(m + n \log n)$ time. To find the second shortest path we remove all internal vertices and all edges of $p_1(u, v)$ from $\overline{G}$. We proceed in this way until either the $(d + 1)$-th shortest path is found or there is no path from u to v in the graph $\overline{G}$.

Algorithm `ShortestPaths`(G, k)
// Input: A graph $G = (V, E)$ and a positive integer k.
// Output: For all pairs $(u, v) \in V \times V$, nonnegative integers $n(u, v)$, and for each $n(u, v) > 0$ the set of $n(u, v)$-shortest paths from u to v.
1. **for** all $(u, v) \in V \times V$
2. $\quad \overline{G} = G$
3. $\quad$ **for** $i = 1$ **to** k **do**
4. $\quad\quad p_i(u, v) =$ `ShortestPath`$(\overline{G}, u, v)$
5. $\quad\quad$ **if** $p_i(u, v) =$ `NIL` **then break**

In order to simplify the exposition, if an ℓ-th shortest path does not exist for some $\ell \leq k$, then we still write that the algorithm has found it and that this path is undefined. In particular, if u and v are not connected, then our algorithm tells us that all the k-shortest paths are undefined.

LEMMA 4.1
Algorithm `ShortestPaths` finds a solution to the k-shortest paths problem on a graph with n vertices and m edges in in $O(n^2m + n^3 \log n)$ worst case time.

PROOF. It is routine to verify the correcness of Algorithm `ShortestPaths`. Its running time is bounded by the number of vertex pairs G multiplied by the running time of Subroutine `ShortestPath`, that is

$$O(\frac{n(n-1)}{2}(m + n \log n)) = O(n^2m + n^3 \log n).$$

∎

5 The d-cycle problem

We use the reduction of d-cycle problem to the k-shortest paths problem as described in Section 3, and apply the d-cycle algorithm stated below.

Algorithm d–`cycle` (G, d)
// Input: Graph $G = (V, E), d > 0$.
// Output: Minimum subgraph G with cyclomatic number d if any.
1. // Shortest paths
For all pairs (u, v) of vertices of G, find at most $d + 1$ shortest paths $p_1(u, v), \ldots, p_{d+1}(u, v)$ from u to v using the algorithm `ShortestPaths` of Section 4.
2. // Minimum weight cycle
For each pair of vertices of G, compute the total weight of the first and second shortest path between them. The minimum weight cycle C is defined by a pair of paths with minimum weight.
3. // Standard k-based subgraphs
For all subsets $W \subseteq V$ with at most $2d - 2$ vertices construct all standard $2d - 2$ subgraphs G' with base W. Let $W = \{w_1, \ldots, w_{2d-2}\}$. The standard subgraph G' defines the bounding matrix $M = [m_{ij}]$, where the element $m_{ij}, i < j$, is the number of shortest paths from w_i to w_j. The element m_{ij} is in the range $\{0, \ldots, d + 1\}$. For each graph G' check if the paths are vertex-disjoint and G' has cyclomatic number d. Also, make such verification for the graph $G' \cup C$.
4. // Output
Output a graph that minimizes the weight over all graphs G' and $G' \cup C$ in Step 3.

THEOREM 5.1
Let $G = (V, E)$ be a nonnegative edge-weighted graph with n vertices and m edges. Let $T(n, m, k)$ be the running time of an algorithm for finding k-shortest paths between all pairs of vertices of G. Then Algorithm d–`cycle` (G, d) finds the minimum subgraph of G with cyclomatic number d in $O(T(n, m, d + 1) + n^{2d-1})$ time.

PROOF. Correctness follows from Theorem 3.2. Algorithm d–`cycle` (G, d) finds the minimum weight cycle in Step 2 and constructs all $(2d-2)$-based subgraphs of G in Step 3.

It remains to analyse the running time of the algorithm. Finding all shortest paths in Step 1 of this algorithm requires $T(n, m, d + 1)$ time. Step 2 takes $O(n^3)$ time. The number of subsets W which are considered in Step 3 is $O(n^{2d-2})$. Since each set W has size at most $2d - 2$, the number of bounding matrices M is bounded by a constant. The number of graphs G' corresponding to the partition W is also bounded by a constant (although possibly quite a large one). The verification of whether G' is vertex-disjoint can be done in $O(n)$ time by marking the vertices of G. The running time of Steps 3 and 4 is $O(n^{2d-1})$. Hence the total running time is

$$O(T(n, m, d + 1) + n^{2d-1}).$$

∎

Lemma 4.1 yields us the following Corollary.

COROLLARY 5.2
The minimum weight subgraph with cyclomatic number d can be computed in time $O(n^{2d-1} + n^2 m + n^3 \log n)$.

6 Conclusion

Our approach reduces the d-cycle problem to the k-shortest-paths problem. This leads to an efficient algorithm for finding a subgraph with a specified cyclomatic number d. Note that the interesting question of whether the d-cycle problem with the input including d is NP-complete remains open.

The authors are grateful to three referees for several corrections and suggestions improving the text of this paper.

References

[1] Dahlhaus, E., Johnson, D. S., Papadimitriou, C. H., Seymour, P. D., Yannakakis, M.: The complexity of multiterminal cuts. SIAM J. Computing, **23** (1994) 864–894

[2] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[3] Eppstein, D.: Finding the k shortest paths. SIAM J. Computing, **28** (2) (1999) 652–673

[4] M. Fredman and R. E. Tarjan. Fibonacci heaps and their uses in improved network optimization problems. *J. ACM*, 34:596–615, 1987.

[5] Hartvigsen, D.': Minimum path bases. Journal of Algorithms, **15** (1993) 125–142

[6] Horton, J. D.: A polynomial time algorithm to find the shortest cycle basis of a graph. SIAM J. Comput., **16** (2) (1987) 358–366

[7] Seshu, S., Reed, M. B.: Linear graphs and Electical Networks. Addison-Wesley, Reading, MA, (1961)

Received 21 October 1999.