2-edge-twinless blocks

Raed Jaberi

Abstract
Let $G = (V, E)$ be a directed graph. A 2-edge-twinless block in $G$ is a maximal vertex set $C^t \subseteq V$ with $|C^t| > 1$ such that for any distinct vertices $v, w \in C^t$, and for every edge $e \in E$, the vertices $v, w$ are in the same twinless strongly connected component of $G \setminus \{e\}$. In this paper we study this concept and describe algorithms for computing 2-edge-twinless blocks.

Keywords: Directed graphs, Strong bridges, Graph algorithms, 2-blocks, Twinless strongly connected graphs

1. Introduction
Let $G = (V, E)$ be a directed graph. A 2-edge block in $G$ is a maximal vertex set $C^e \subseteq V$ with $|C^e| \geq 2$ such that for each pair of distinct vertices $v, w \in C^e$, there are two edge-disjoint paths from $v$ to $w$ and two edge-disjoint paths from $w$ to $v$ in $G$. By Menger’s Theorem for edge connectivity [14], there exist two edge-disjoint paths from $v$ to $w$ and two edge-disjoint paths from $w$ to $v$ in $G$ if and only if the vertices $v, w$ belong to the same strongly connected component of $G \setminus \{e\}$ for any edge $e \in E$. In this paper we introduce and study a new concept the 2-edge-twinless blocks. A 2-edge-twinless block in $G$ is a maximal vertex set $C^t \subseteq V$ of size at least 2 such that for any distinct vertices $v, w \in C^t$, and for every edge $e \in E$, the vertices $v, w$ lie in the same twinless strongly connected component of $G \setminus \{e\}$. As Figure 1 demonstrates, the vertices of a 2-edge-twinless block do not necessarily lie in the same 2-edge block.

The concept of twinless strongly connected components was first introduced by Raghavan [15] in 2006. Raghavan [15] proved that twinless strongly connected components of a directed graph can be identified in linear time. The first linear time algorithm for testing 2-vertex connectivity of directed graphs was described by Georgiadis [2] in 2010. Italiano et al. [7, 5] gave linear time algorithms for determining all the strong articulation points and strong bridges of a directed graph. In 2014, Jaberi [9] presented algorithms for computing the 2-vertex-connected components of directed graphs in $O(nm)$ time (published in [8]). An experimental study (2015) [13] showed that our
Figure 1: A twinless strongly connected graph $G$. This graph contains two 2-edges blocks $C_1 = \{12, 18\}, C_2 = \{2, 7\}$ and one 2-edge-twinless block $C = \{12, 18\}$. Since the vertices 2, 7 are not in the same twinless strongly connected component of $G \setminus \{(3, 8)\}$, the set $C_2$ is not a 2-edge-twinless block.

algorithm performs well in practice. Henzinger et al. [6] introduced algorithms for calculating the 2-vertex-connected components and 2-edge-connected components of a directed graph in $O(n^2)$ time. Jaberi [10] presented algorithms for computing the 2-directed blocks, 2-strong blocks, and 2-edge blocks of a directed graph. Georgiadis et al. [3] gave linear time algorithms for finding 2-edge blocks. The same authors [4] gave linear time algorithms for calculating 2-directed blocks and 2-strong blocks. In 2019, Jaberi [11] presented an algorithm for computing 2-twinless-connected components. Jaberi [12] also presented algorithms for computing 2-twinless blocks.

The paper is organized as follows. In section 2 we present an algorithm for computing 2-edge-twinless blocks in $O((b_t n + m)n)$ time, where $b_t$ is the number of twinless bridges. We then study in section 3 the relation between 2-edge-twinless blocks and 2-edge block and we show that 2-edge-twinless
blocks can be computed in $O((b_t - b_s + n)m)$, where $b_t$ and $b_s$ are the number of twinless bridges and strong bridges, respectively. Finally, in section \[4\] we pose some open problems.

2. Computing 2-edge-twinless blocks

In this section we present an algorithm for computing the 2-edge-twinless blocks of directed graphs. The strongly connected components of a directed graph are disjoint and they can be found in linear time using Tarjan’s algorithm [16]. Moreover, the twinless strongly connected components of a strongly connected graph can be identified in linear time using Raghavan’s algorithm [15]. The following lemma shows that we need only consider twinless strongly connected graphs.

**Lemma 2.1.** Let $G = (V, E)$ be a strongly connected graph. Let $C^t$ be a 2-edge-block of $G$. Then all the vertices of $C^t$ lie in the same twinless strongly connected component of $G$.

**Proof.** Assume for the sake of contradiction that $C^t$ contains two vertices $v, w$ such that $v, w$ are in distinct twinless strongly connected components $C_v, C_w$, respectively. If we contract every twinless strongly connected component of $G$ into a single vertex, we obtain a graph $G^{tss}$. By [15, Theorem], the underlying graph of $G^{tss}$ is a tree and each edge corresponds to antiparallel edges of $G$. Therefore, there is an edge $e$ such that $G \setminus \{e\}$ is not strongly connected, a contradiction. □

The 2-edge-twinless blocks of a strongly connected graph are the union of the 2-edge-twinless blocks of its twinless strongly connected components.

The next lemma shows that 2-edge-twinless blocks have no vertices in common.

**Lemma 2.2.** Let $G = (V, E)$ be a twinless strongly connected graph. The 2-edge-twinless blocks of $G$ are disjoint.

**Proof.** Let $C^t_1, C^t_2$ be two distinct 2-edge-twinless blocks of $G$. Suppose for the purpose of contradiction that $C^t_1 \cap C^t_2 \neq \emptyset$. Let $v \in C^t_1 \cap C^t_2$ and let $x \in C^t_1, y \in C^t_2$ with $x, y \notin C^t_1 \cap C^t_2$. Since $x, y$ are not in the same 2-edge-twinless block, there is an edge $e \in E$ such that $x, y$ are not in the same twinless strongly connected component of $G \setminus \{e\}$. The vertices $x, v$ lie in the same twinless strongly connected component of $G \setminus \{e\}$ because $C^t_i$ is a 2-edge-twinless block. Furthermore, $v, y$ belong to the same twinless strongly connected component of $G \setminus \{e\}$. By [15, Lemma 1] $x$ and $y$ are in the same twinless strongly connected component of $G \setminus \{e\}$. □
Lemma 2.3. Let \( G = (V, E) \) be a twinless strongly connected graph and let \( x, y \) be distinct vertices in \( G \). Let \( e \) be an edge in \( G \) such that \( e \) is not a twinless bridge. Then the vertices \( x \) and \( y \) lie in the same twinless strongly connected component of \( G \setminus \{e\} \).

Proof. Immediate from the definition. \( \square \)

Lemma 2.2 and Lemma 2.3 lead to an algorithm (Algorithm 2.3) which might be helpful when the number of twinless bridges is small.

Algorithm 2.4.
Input: A twinless strongly connected graph \( G = (V, E) \).
Output: The 2-edge-twinless blocks of \( G \).
1. If \( G \) is 2-edge-twinless connected then.
2. Output \( V \).
3. else
4. Let \( A \) be an \( n \times n \) matrix.
5. Initialize \( A \) with 1s.
6. Compute the twinless bridge of \( G \).
7. for each twinless bridge \( e \) of \( G \) do
8. for each pair \((v, w) \in V \times V\) do
9. if \( v, w \) in distinct twinless strongly connected components of \( G \setminus \{e\} \) then
10. \( A[v, w] \leftarrow 0 \).
11. \( E' \leftarrow \emptyset \).
12. for each pair \((v, w) \in V \times V\) do
13. if \( A[v, w] = 1 \) and \( A[w, v] = 1 \) then
14. Add the undirected edge \((v, w)\) to \( E' \).
15. Calculate the connected components of size > 1 of \( G' \) and output them.

Theorem 2.5. Algorithm 2.4 runs in \( O((b_t n + m)n) \) time, where \( b_t \) is the number of twinless bridges.

Proof. The twinless strongly connected components of a strongly connected graph can be calculated in linear time using Raghavan’s algorithm \([15]\). Jaberi \([11]\) shows that twinless bridges can be computed in \( O(nm) \) time. Moreover, Lines 7–11 take \( O(b_t n^2) \) time. \( \square \)

3. The relationship between 2-edge-twinless blocks and 2-edge-blocks

In this section we discuss the relationship between 2-edge blocks and 2-edge-twinless blocks. The following lemma shows that each 2-edge-twinless block is a subset of a 2-edge block.
Lemma 3.1. Let $G = (V, E)$ be a twinless strongly connected graph and let $C_t$ be a 2-edge twinless block in $G$. Then all the vertices of $C_t$ are in the same 2-edge block of $G$.

Proof. Let $v$ and $w$ be distinct vertices in $C_t$ and let $e$ be an edge in $G$. By definition, the vertices $v, w$ are in the same twinless strongly connected component of $G \setminus \{e\}$. Let $C$ be the twinless strongly connected component of $G \setminus \{e\}$ that contains $v, w$. Clearly, all the vertices of $C$ lie in the same strongly connected component of $G \setminus \{e\}$. Consequently, $v, w$ are in the same 2-strong block in $G$. □

The next lemma demonstrates which vertices are important when we compute 2-edge-twinless blocks in 2-edge-connected graphs.

Lemma 3.2. Let $G = (V, E)$ be a 2-edge connected graph. Let $x, y$ be two distinct vertices in $V$ such that $x, y$ are not in the same 2-edge-twinless block in $G$. Then $G$ contains a twinless bridge $e$ such that $e$ is not a strong bidge of $G$ and the vertices $x, y$ do not lie in the same twinless strongly connected component of $G \setminus \{e\}$.

Proof. Since the vertices $x, y$ are not in the same 2-edge-twinless block in $G$, there is an edge $(v, w) \in E$ such that $x, y$ are in distinct twinless strongly connected components of $G \setminus \{(v, w)\}$. Moreover, $(v, w)$ is a twinless bridge because the graph $G \setminus \{(v, w)\}$ is not twinless strongly connected. But the edge $(v, w)$ is not a strong bridge because $G$ is 2-edge-connected. □

Algorithm 3.3 describes an algorithm for computing 2-edge-twinless blocks.
Lemma 3.4. Algorithm 3.3 calculates 2-edge-twinless blocks.

Proof. This follows from Lemma 3.1 and Lemma 3.2. □

Theorem 3.5. The running time of Algorithm 3.3 is $O((b_t - b_s + n)m)$, where $b_t$ and $b_s$ are the number of twinless bridges and strong bridges, respectively.

Proof. The 2-edge blocks of a directed graph can be calculated in linear time using the algorithm of Georgiadis et al. [3]. Italiano et al. [7, 1] presented linear time algorithms for calculating all the strong bridges of a directed graph. Jaberi [11] proved that the twinless bridges can be computed in $O(nm)$ time. The twinless strongly connected components of a directed graph can be found in linear time using Raghavan’s algorithm [15]. Line 9 can be implemented in linear time by using a similar idea to [3, Lemma 3.2]). □

Italiano et al. [7] proved that the number of strong bridges is at most $2n - 2$. Let $G$ be a twinless strongly connected graph. Jaberi [11] described how we can obtain a twinless strongly connected subgraph from a strongly connected subgraph in a twinless strongly connected graph without increasing the number of its edges. This means that the number of twinless bridges is at most $2n - 2$.

4. Open Problems

We leave as open problem whether the 2-edge-twinless blocks of a directed graph can be identified in linear time. Another open question is whether the twinless bridges of a directed graph can be calculated in linear time.

Let $G = (V, E)$ be a directed graph. A $k$-edge-twinless block is a maximal vertex set $U \subseteq V$ such that for any distinct vertices $v, w \in U$ and for each edge subset $L \subseteq E$ with $|L| < K$, the vertices $v, w$ belong to the same twinless strongly connected component of $G \setminus L$.

Lemma 4.1. The $k$-edge-twinless blocks of a directed graph are disjoint

Proof. Similar to the proof of Lemma 2.2. □

We leave as open problem whether $k$-edge-twinless blocks can be calculated efficiently.
References

[1] D. Firmani, G.F. Italiano, L. Laura, A. Orlandi, F. Santaroni, Computing strong articulation points and strong bridges in large scale graphs, SEA, LNCS 7276, (2012) 195–207.

[2] L. Georgiadis, Testing 2-vertex connectivity and computing pairs of vertex-disjoint s-t paths in digraphs, In Proc. 37th ICALP, Part I, LNCS 6198 (2010) 738–749.

[3] L. Georgiadis, G.F. Italiano, L. Laura, N. Parotsidis, 2-Edge Connectivity in Directed Graphs, SODA (2015) 1988–2005.

[4] L. Georgiadis, G.F. Italiano, L. Laura, N. Parotsidis, 2-Vertex Connectivity in Directed Graphs, ICALP (1)2015 : 605–616

[5] G. F. Italiano, L. Laura, F. Santaroni, Finding Strong Bridges and Strong Articulation Points in Linear Time. COCOA (1)2010 : 157–169

[6] M. Henzinger, S. Krinninger, V. Loitzenbauer, Finding 2-Edge and 2-Vertex Strongly Connected Components in Quadratic Time. ICALP (1)2015 : 713–724

[7] G.F. Italiano, L. Laura, F. Santaroni, Finding strong bridges and strong articulation points in linear time, Theoretical Computer Science 447 (2012) 74–84.

[8] R. Jaberi, On computing the 2-vertex-connected components of directed graphs. Discrete Applied Mathematics 204: (2016)164–172

[9] R. Jaberi, On Computing the 2-vertex-connected components of directed graphs, (January, 2014) CoRR abs/1401.6000 .

[10] R. Jaberi, Computing the 2-blocks of directed graphs. RAIRO - Theor. Inf. and Applic. 49(2)(2015)93–119

[11] R. Jaberi, Twinless articulation points and some related problems, (2019), arxiv, abs/1912.11799

[12] R. Jaberi, Computing 2-twinless blocks, (2019), arxiv, abs/1912.12790

[13] W. D. Luigi, L. Georgiadis, G. F. Italiano, L. Laura, N. Parotsidis, 2-Connectivity in Directed Graphs, An Experimental Study. ALENEX. (2015) 173–187.
[14] K. Menger. Zur allgemeinen kurventheorie. Fund. Math., 10 : 96–115, (1927).

[15] S. Raghavan, Twinless Strongly Connected Components, Perspectives in Operations Research, (2006) 285–304.

[16] R.E. Tarjan, Depth-first search and linear graph algorithms, SIAM J. Comput. 1(2) (1972) 146–160