Twinless Strongly Connected Components

S. Raghavan

The Robert H. Smith School of Business & the Institute for Systems Research
Van Munching Hall
University of Maryland
College Park, MD 20742
raghavan@umd.edu

Summary. Tarjan [9], describes how depth first search can be used to identify Strongly Connected Components (SCC) of a directed graph in linear time. It is standard to study Tarjan's SCC algorithm in most senior undergraduate or introductory graduate computer science algorithms courses. In this paper we introduce the concept of a *twinless strongly connected component* (TSCC) of a directed graph. Loosely stated, a TSCC of a directed graph is (i) strongly connected and (ii) remains strongly connected even if we require the deletion of arcs from the component so that it does not contain a pair of *twin arcs* (twin arcs are a pair of bidirected arcs (i, j) and (j, i) where the tail of one arc is the head of the other and vice versa). This structure has diverse applications, from the design of telecommunication networks [7] to structural stability of buildings [8]. In this paper, we illustrate the relationship between 2-edge connected components of an undirected graph—obtained from the strongly connected components of a directed graph—and twinless strongly connected components. We use this relationship to develop a linear time algorithm to identify all the twinless strongly connected components of a directed graph. We then consider the augmentation problem, and based on the structural properties developed earlier, derive a linear time algorithm for the augmentation problem.

Key words: Digraph augmentation; strong connectivity; linear time algorithm.

1 Introduction

Let $D = (N, A)$ be a directed graph (digraph) with node set N and arc set A. A pair of nodes x and y are *twinless reachable* if there exists a directed path from node x to node y, and a directed path from node y to node x, such that for every arc (i, j) contained in the path from node x to node y, the path from node y to node x does not contain arc (j, i). The *Twinless Strongly Connected Components* (TSCCs) of a digraph are the equivalence classes of nodes under the "twinless reachable" condition (we will show later that the twinless reachable condition defines an equivalence relationship). We

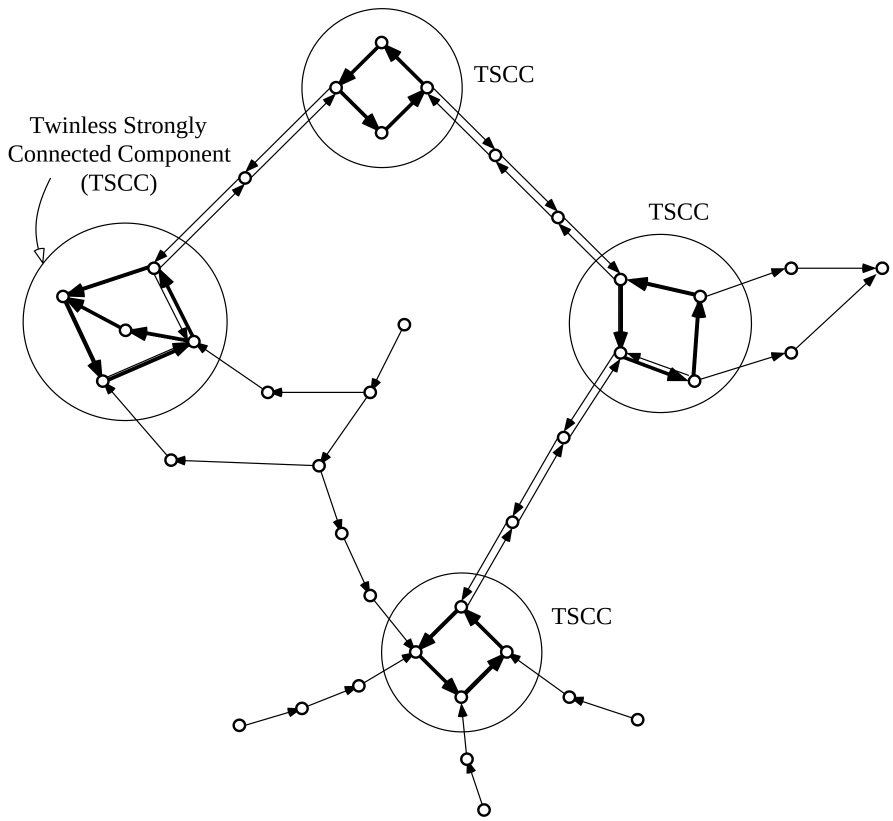

Fig. 1. Twinless Strongly Connected Components of a digraph. Bold arcs show twinless arcs that form a strongly connected component.

say a digraph is *Twinless Strongly Connected* if every pair of nodes is twinless reachable.

We now provide a slightly different, but equivalent, definition of twinless strongly connectedness. We say that a pair of bidirected arcs (i, j) and (j, i) are *twins*. Recall that a digraph is strongly connected if it contains a directed path between every pair of its nodes. Our alternate definition then is as follows. A digraph $D = (N, A)$ is *Twinless Strongly Connected* if for some subset A' of A, the digraph (N, A') is strongly connected and A' does not contain an arc together with its twin. A *Twinless Strongly Connected Component* (TSCC) of a digraph is the node set of a maximal twinless strongly connected subdigraph of D. Figure 1 gives an example of four TSCCs, that contain 3 or more nodes, in a digraph.

It should be apparent that every pair of nodes in a TSCC, as defined by the second definition, are twinless reachable. What may not readily apparent is the converse. That is, if N^i is a TSCC under the first definition of a TSCC. Then, the subdigraph $D^i = (N^i, A^i)$, where $A^i = \{(x, y)|(x, y) \in A, x \in N^i, y \in N^i\}$, is twinless strongly connected as per the second definition. We will work with the second definition until we show, in the next section, that both definitions are indeed equivalent.

Additionally, when considering digraphs, it is clear that reachability is a transitive property. That is, if there is a directed path from node x to node y, and a directed path from node y to node z, then there is a directed path from node x to node z. It turns out that the twinless reachable property is also transitive, but this is not so obvious. Transitivity of the twinless reachable property means that, if a pair of nodes x and y are twinless reachable, and a pair of nodes y and z are twinless reachable, then the pair of nodes x and z are twinless reachable. Transitivity is necessary to define an equivalence relationship and we will show this property in the next section.

In this paper, we consider the following questions (analogous to those for SCCs) in connection with TSCCs. How do we recognize TSCCs of a digraph? Is it possible to recognize TSCCs of a digraph in linear time? We also consider the (unweighted) augmentation problem. That is, given a digraph $D = (N, A)$ find the minimum cardinality set of arcs A' to add to the digraph so that $D = (N, A \cup A')$ is twinless strongly connected (In a seminal paper Eswaran and Tarjan [4] introduced and solved the augmentation problem for strong connectivity). Our answer to these questions is affirmative. Specifically, we develop linear time algorithms to recognize all TSCCs of a digraph and to solve the augmentation problem.

The remainder of this paper is organized as follows. In Section 2 we first derive some structural properties of TSCCs. Specifically we show a correspondence between TSCCs in a strongly connected digraph and 2-edge connected components of an associated undirected graph. Using these structural properties, in Section 3 we describe a linear time algorithm for identifying the TSCCs of a digraph. In Section 4 we consider the augmentation problem and show how to solve the unweighted version of the augmentation problem in linear time. In Section 5 we describe some applications of this structure—one in telecommunications, and one in determining the structural rigidity of buildings. Finally, in Section 6 we discuss a connection between the notion of a twinless strongly connected digraph and strongly connected orientations of mixed graphs.

2 TSCCs of a Strongly Connected Digraph and 2-Edge Connected Components

We now derive some structural properties of TSCCs in a strongly connected digraph D. For ease of exposition, we introduce some additional notation. The *TSCC induced digraph of D* is the digraph $D^{\text{TSCC}} = (N^{\text{TSCC}}, A^{\text{TSCC}})$ obtained by contracting each TSCC in D to a single node. We replace parallel arcs that the contraction creates by a single arc. Every node in the TSCC

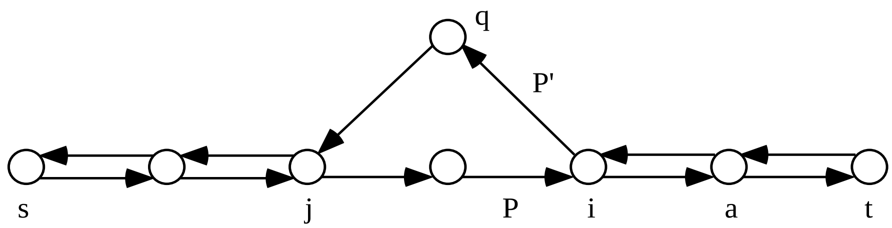

Fig. 2. Proof that the paths P and P' are twin paths.

induced digraph D^{TSCC} corresponds to a TSCC in the original digraph D. Consequently, for any node $i \in N^{\mathrm{TSCC}}$ we refer to the TSCC node i corresponds to in the original digraph, including the arcs and nodes in the TSCC, as TSCC(i).

For any digraph $D = (N, A)$, the *associated undirected graph* $G(D) = (N, E)$ is a graph with edges $E = \{\{i, j\} : (i, j) \in A$ and/or $(j, i) \in A\}$. If (i, j) belongs to A, we refer to the edge $\{i, j\}$ in E as an *image* of this arc. We say that two paths P and P' are *twin paths* if P is a path from node i to node j, and P' is a path from node j to node i that uses exactly the reversal (i.e., the twin) of each arc on the path P.

We first prove a useful property concerning the structure of directed paths between TSCCs in a strongly connected digraph.

Theorem 1 (Twin-arc). *Let $D = (N, A)$ be any strongly connected digraph and let D^{TSCC} be its TSCC induced digraph. The associated undirected graph of D^{TSCC} is a tree. Moreover, every edge in the associated tree is the image of a pair of twin arcs (and no other arcs) of D.*

Proof:
First, consider the TSCC induced subdigraph D^{TSCC} (note that since D is strongly connected, so is D^{TSCC}). We show that D^{TSCC} contains a twin path and no other path between any two nodes. As a result, the associated undirected graph of D^{TSCC} is a tree. We will establish this result by contradiction. Assume the digraph D^{TSCC} contains a path P from a node s to a node t and a path P' from node t to node s that are not twin paths. Let arc (i, q) be the first arc on P' that does not have a twin arc on the path P and let j be the first node following node i on the path P' that lies on the path P (see Figure 2). Then all nodes on P' between nodes i and j and all nodes on P between nodes j and i are twinless strongly connected and thus in the same TSCC. In other words, nodes i and j do not correspond to maximal twinless strongly connected subdigraph's of D (i.e., TSCCs). But, D^{TSCC} is obtained by contracting TSCCs in D and thus each node in D^{TSCC} is a TSCC. We now have a contradiction.

We now show that every pair of twin arcs in D^{TSCC} corresponds to a pair of twin arcs and no other arcs of D. As a result, every edge in the associated tree (i.e., $G(D^{\text{TSCC}})$) is the image of a pair of twin arcs and no other arcs of D. Consider any two adjacent nodes in D^{TSCC}, say nodes a and t. Node t and node a correspond to TSCCs (possibly single nodes) in the expanded digraph (i.e., D). If the original (expanded) digraph contains two non-twin arcs (i,j) with $i \in \text{TSCC}(a)$ and $j \in \text{TSCC}(t)$ and (k,l) with $k \in \text{TSCC}(t)$ and $l \in \text{TSCC}(a)$, then the digraph obtained by the union of TSCC(a), TSCC(t) and the arcs (i,j) and (k,l) is twinless strongly connected, and we have a contradiction. Therefore, only a single arc (a,t) connects TSCC(a) to TSCC(t) and only a single arc, the twin of arc (a,t), joins TSCC(t) to TSCC(a). □

Since D^{TSCC} has the structure of a bidirected tree (that is, a tree with twin arcs in place of each edge [see Figure 3]) when D is strongly connected; we refer to D^{TSCC} as the *TSCC tree*.

Theorem 1 implies the following result concerning the relationship between a strongly connected digraph and its associated undirected graph.

Theorem 2. *The associated undirected graph $G(D)$ of a strongly connected digraph D is 2-edge connected if and only if D is a twinless strongly connected digraph.*

Proof:
If D is a twinless strongly connected digraph, then its associated undirected graph $G(D)$ must be 2-edge connected. Otherwise, if $G(D)$ is not 2-edge connected, deleting some edge $\{i,j\}$ from $G(D)$ disconnects the graph. In D, this edge corresponds to arc (i,j) or arc (j,i) or both arcs (i,j) and its twin (j,i). Eliminating these arcs destroys any directed path between nodes i and j. Consequently D is not twinless strongly connected; a contradiction.

To complete the proof, we show that if the associated undirected graph $G(D)$ is 2-edge connected, then D is a twinless strongly connected digraph. Suppose this is not true. Then $G(D)$ is 2-edge connected while D is not a twinless strongly connected digraph. Consider the TSCC tree of D. If D is not twinless strongly connected then its TSCC tree contains at least two nodes. If the TSCC tree contains 2 or more nodes, then its associated undirected graph (a tree) has at least one edge. Deleting an edge on this graph disconnects it. Since an edge on the associated undirected graph of the TSCC tree is an image of twin arcs and no other arcs in D, deleting the same edge in $G(D)$ disconnects $G(D)$. But then $G(D)$ is not 2-edge connected, resulting in a contradiction. Consequently, the TSCC tree is a single node and D is a twinless strongly connected digraph. □

Theorem 1 and Theorem 2 imply the following characterization of TSCCs in a strongly connected digraph.

Corollary 1 *The 2-edge-connected components of the associated undirected graph $G(D)$ of a strongly connected digraph D correspond in a one to one fashion with the TSCCs of D.*

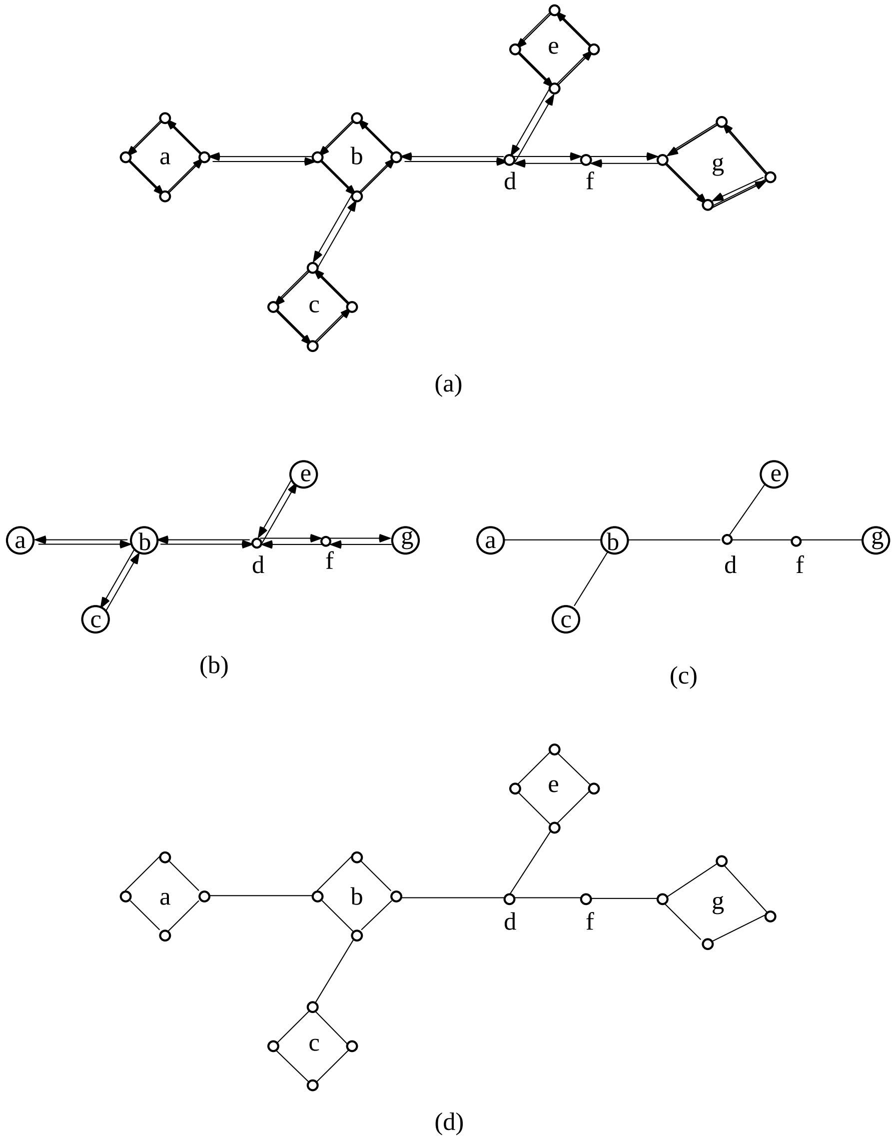

Fig. 3. Illustration of Theorem 1, Theorem 2, and Corollary 1. (a) Strongly connected digraph D, (b) D^{TSCC} the TSCC induced subdigraph of D (the TSCC tree), (c) Associated undirected graph of D^{TSCC}, (d) Associated undirected graph of D.

Notice that Corollary 1 assures us that the TSCCs of a digraph are uniquely defined. Also, from Theorem 1 it follows that the twinless reachable property is transitive.

Lemma 1 *Twinless reachability is a transitive property.*

Proof:
Suppose nodes a and b in a digraph are twinless reachable, and nodes b and c in the same digraph are twinless reachable. It immediately follows that nodes a, b, and c must all be in the same strongly connected component of the digraph. Consider the strongly connected component that contains nodes a, b, and c. From Theorem 1 it follows that nodes a and b must be in the same TSCC, and nodes b and c must be in the same TSCC. But that means nodes a and c are in the same TSCC. From the second definition of twinless strongly connectedness it follows that nodes a and c must be twinless reachable. □

Lemma 1 also shows that the twinless reachable condition defines an equivalence relationship. A binary relationship defines an equivalence relationship if it satisfies reflexivity, symmetry and transitivity. By definition twinless reachability satisfies reflexivity and symmetry, while Lemma 1 shows transitivity proving that it defines an equivalence relationship.

The proof of Lemma 1 also shows the equivalence of the two definitions.

Lemma 2 *The two definitions of a TSCC are equivalent.*

Proof:
It is readily apparent that all node pairs in a TSCC under the second definition are twinless reachable. The proof of Lemma 1 shows any pair of nodes that are twinless reachable must be in the same TSCC (as defined by the second definition of a TSCC). □

The previous lemmas also allow us to show that nodes on any directed path between two nodes in a TSCC are also in the TSCC. This will be useful to us when we consider augmentation problems.

Lemma 3 *Let D be any twinless strongly connected digraph, and P_{ij} be any directed path from node i to j with $i, j \in D$. Then $D_P = D \cup P_{ij}$ is a twinless strongly connected digraph.*

Proof:
Clearly D_P is strongly connected. Consider $G(D_P)$. From Theorem 2 $G(D)$ is 2-edge connected. Thus $G(D) \cup G(P_{ij})$ is also 2-edge connected. But $G(D) \cup G(P_{ij}) = G(D_P)$, showing $G(D_P)$ is 2-edge connected. Thus, by Theorem 2 D_P is also twinless strongly connected. □

3 Identifying Twinless Strongly Connected Components in Linear Time

With the characterization of the relationship between TSCCs in a strongly connected digraph and 2-edge connected components of the associated undirected graph it is now easy to develop a linear time algorithm (based on depth first search) to identify all TSCCs. The first step consists of finding all strongly connected components of the directed graph. As noted in the outset of the paper this is easily done in linear time using depth first search [9]. A singleton node constituting a SCC of the digraph is also a TSCC of the digraph. If a SCC has cardinality 2, i.e., it consists of 2 nodes, then each node in the SCC is a TSCC. For each of the remaining SCCs (i.e., ones with cardinality greater than or equal to 3) we construct the strongly connected digraph (defined by the arcs between nodes of the SCC) and identify the TSCCs on the SCC.

Corollary 1 states that to identify the TSCCs of a strongly connected digraph, it is sufficient to identify all 2-edge-connected components of its associated undirected graph. Let D_S denote a strongly connected component of D. Consequently, we can convert D_S to its associated undirected graph G_S in $\mathcal{O}(|N| + |A|)$ time, and use the well-known method for identifying all 2-edge-connected components that is also based on depth first search (see exercise 23.2 in [3]).

4 The Augmentation Problem

In this section we consider the problem of augmenting a digraph so that it is twinless strongly connected. As mentioned in the introduction to this paper, Eswaran and Tarjan [4] introduced the augmentation problem. They showed how to minimally augment a digraph in linear time so that it is strongly connected. They also showed how to minimally augment an undirected graph, in linear time, so that it is 2-edge connected.

Our procedure to augment a digraph so that it is twinless strongly connected is roughly as follows. We first apply Eswaran and Tarjan's augmentation procedure to strongly connect the digraph. From Theorem 2, it follows that this strongly connected digraph is twinless strongly connected if and only if its associated undirected graph is 2-edge connected. Consequently, we can apply Eswaran and Tarjan's augmentation procedure (implicitly) to the associated undirected graph to determine the edges to add to make it 2-edge connected. In the corresponding digraph, we add an arc corresponding to each edge added, arbitrarily choosing a direction for the arc in the digraph. Theorem 2 assures us that this procedure gives a twinless strongly connected digraph. We will show that our procedure in fact works (i.e., adds the fewest number of arcs) if the digraph D is *carefully* modified by deleting certain carefully chosen arcs. As a result we present a linear time algorithm to solve the augmentation problem for twinless strong connectivity.

Since our procedure is based on Eswaran and Tarjan's augmentation algorithms we briefly review their procedures.

4.1 Augmenting for Strong Connectivity

Let $D = (N, A)$ be a directed graph, and define $D^{\text{SCC}} = (N^{\text{SCC}}, A^{\text{SCC}})$ to be the SCC induced digraph of D that is obtained by contracting each SCC in D to a single node. We replace parallel arcs that the contraction creates by a single arc. It is well-known (and straightforward) that D^{SCC} is acyclic.

Eswaran and Tarjan show that it is sufficient to focus attention to the augmentation problem on the SCC induced digraph. To be specific let β be a mapping from N^{SCC} to N defined as follows. If $x \in N^{\text{SCC}}$ then $\beta(x)$ defines any node in the strongly connected component of D corresponding to node x. They show that if A^{ASC} is a minimal set of arcs whose addition strongly connects D^{SCC}, then $\beta(A^{\text{ASC}}) = \{(\beta(x), \beta(y)) | (x, y) \in A^{\text{ASC}}\}$ is a minimal set of arcs whose addition that strongly connects D.

In the acyclic digraph D^{SCC}, a source is defined to be a node with outgoing but no incoming arcs, a sink is defined to be a node with incoming but no outgoing arcs, and an isolated node is defined to be a node with no incoming and no outgoing arcs. Let S, T and Q denote the sets of source nodes, sink nodes, and isolated nodes respectively in D^{SCC}, and assume without loss of generality $|S| \le |T|$.

Eswaran and Tarjan's procedure finds an index r and an ordering $s(1), \ldots$ $\ldots, s(|S|)$ of the sources of D^{SCC} and $t(1), \ldots, t(|T|)$ of the sinks of D^{SCC} such that

1. there is a path from $s(i)$ to $t(i)$ for $1 \le i \le r$;
2. for each source $s(i)$, $r + 1 \le i \le |S|$ there is a path from $s(i)$ to some $t(j)$, $1 \le j \le r$; and
3. for each sink $t(j)$, $r+1 \le j \le |T|$, there is a path from some $s(i)$, $1 \le i \le r$, to $t(j)$.

They show that a minimal augmentation of D^{SCC} is obtained from the arc set

$$A^{\text{ASC}} = \{(t(i), s(i+1)) | 1 \le i < r\} \cup \{(t(i), s(i)) | r + 1 \le i \le |S|\}$$
$$\bigcup \begin{cases} (t(r), s(1)) & \text{if } |Q| = 0 \text{ and } |S| = |T| \\ (t(r), t(|S|+1)) \cup \{(t(i), t(i+1)) \,||S| + 1 \le i < |T|\} & \\ \cup (t(|T|), s(1)) & \text{if } |Q| = 0 \text{ and } |S| < |T| \\ (t(r), t(|S|+1)) \cup \{(t(i), t(i+1)) \,||S| + 1 \le i < |T|\} & \\ \cup (t(|T|), q(1)) \cup \{(q(i), q(i+1)) | 1 \le i < |Q|\} & \\ \cup (q(|Q|), s(1)) & \text{otherwise} \end{cases}$$

Notice that the augmenting set contains $|T| + |Q|$ arcs. Since there are $|S| + |Q|$ nodes with no incoming arcs, at least $|S| + |Q|$ arcs are needed to augment D^{SCC} so that it is strongly connected. Similarly, as there are $|T| + |Q|$ nodes with no outgoing arcs, at least $|T| + |Q|$ arcs are needed to augment D^{SCC} so that it is strongly connected. Thus $\max(|S|, |T|) + |Q|$ arcs is a lower bound on the number of arcs needed to augment D^{SCC} so that it is strongly connected.

We now show that the addition of the arcs in A^{ASC} strongly connects the digraph D^{SCC}. Actually we show a stronger result that the addition of these arcs makes D^{SCC} twinless strongly connected. Note however this does not mean adding $\beta(A^{\text{ASC}})$ to D makes it twinless strongly connected.

Lemma 4 *When $|N^{SCC}| > 2$ the augmented digraph $D^{ASC} = (N^{SCC}, A^{SCC} \cup A^{ASC})$ is twinless strongly connected.*

Proof:
Observe that in D^{SCC} any path from a source to sink does not contain a source, sink, or isolated node as an intermediate node. Further D^{SCC} is acyclic and so does not contain twin arcs. Consider the pair $(t(i), s(i+1))$ for any $1 \leq i \leq r$. By design there is a path from $t(i)$ to $s(i)+1$ (i.e., the arc $(t(i), s(i+1))$). Additionally, the addition of arcs has created a directed path from $s(i+1)$ to $t(i)$ that does not use $(s(i+1), t(i))$. The path is defined by following the path from $s(i+1) \rightsquigarrow t(i+1) \rightarrow s(i+2) \rightsquigarrow t(i+2) \rightarrow \ldots \rightsquigarrow t(r) \rightarrow t(|S|+1) \rightarrow t(|S|+2) \rightarrow \ldots \rightarrow t|T| \rightarrow q(1) \rightarrow q(|Q| \rightarrow s(1) \rightsquigarrow t(1) \rightarrow s(2) \rightsquigarrow \ldots \rightarrow s(i) \rightsquigarrow t(i))$.[1] Consequently $s(i+1)$, $t(i)$, and all other nodes on the path from $s(i+1)$ to $t(i)$ are twinless reachable and thus in the same TSCC. Consequently, $s(1), \ldots, s(r), t(1), \ldots, t(r), t(|S|+1), \ldots, t(|T|), q(1), \ldots, q(|Q|)$ are in the same TSCC.

Now consider $s(i)$ and $t(i)$ for any $r+1 \leq i \leq |S|$. Augmentation has created a directed path from $t(i)$ to $s(i)$. Observe, there is a directed path from $s(i)$ to some $t(j), 1 \leq j \leq r$, and a path from some $s(k), 1 \leq k \leq r$, to $t(i)$. The augmentation also creates a directed path from $t(j)$ to $s(k)$ (by following the path described in the first part of the proof). Thus $s(i) \rightsquigarrow t(j) \rightsquigarrow s(k) \rightsquigarrow t(i)$ is a directed path from $s(i)$ to $t(i)$ that does not use arc $(s(i), t(i))$. Consequently $s(i)$ and $t(i)$, and all other nodes on the path from $s(i)$ to $t(i)$ are twinless reachable. This means that $s(1), \ldots, s(|S|), t(1), \ldots, t(|T|), q(1), \ldots, q(|Q|)$ are in the same TSCC.

Finally observe that every node in D^{SCC} that is not a source, sink, or an isolated node is on a path from a source to a sink. Thus, using Lemma 3, D^{ASC} is a twinless strongly connected digraph. □

4.2 Augmenting a Strongly Connected Digraph so that it is Twinless Strongly Connected

We now consider the following problem. Given a strongly connected digraph how do we minimally augment it so that it is twinless strongly connected. Before we begin we first make the following observation that immediately follows from the transitivity of the twinless reachable condition.

[1] We denote a path from node i to j by $i \rightsquigarrow j$, and an arc from node i to node j by $i \rightarrow j$.

Property 1 *Let γ be a mapping from N^{TSCC} to N defined as follows. If $x \in N^{TSCC}$ then $\gamma(x)$ defines any node in the twinless strongly connected component of D corresponding to node x. If A^{ATSC} is a set of arcs whose addition twinless strongly connects D^{TSCC}, then $\gamma(A^{ATSC}) = \{(\gamma(x), \gamma(y))|(x, y) \in A^{ATSC}\}$ is a set of arcs whose addition twinless strongly connects D.*

Observe however that the converse is not true. Consequently, this property does not immediately show that it suffices to focus on D^{TSCC}.

Suppose D is a strongly connected digraph. Recall that D^{TSCC}, the TSCC induced digraph has the structure of a bidirected tree. Consider the set of leaf nodes L (also referred to as leaf TSCCs) of this TSCC tree and observe a leaf node on the TSCC tree has a pair of twin arcs directed into and out of it (referred to as twin leaf arcs). Consequently to make D twinless strongly connected we need to either add an arc directed into the leaf TSCC or out of the leaf TSCC that is not the twin of the twin leaf arcs. Since there are $|L|$ leaf TSCCs, we need at least $\lceil \frac{|L|}{2} \rceil$ arcs to make D twinless strongly connected.

The procedure to make D twinless strongly connected is as follows. Consider the associated undirected graph of D^{TSCC}. Recall since D is strongly connected, $G(D^{\mathrm{TSCC}})$ is a tree. Select one of the leaf nodes and perform DFS from this node. Number the leaf nodes of the TSCC tree in the order they are visited in the DFS procedure, and let $l(1), \ldots, l(|L|)$ denote this ordering. Augment D^{TSCC} with the arc set $A^{\mathrm{ATSC}} = \{(l(i), l(i + \lfloor |L|/2 \rfloor))|1 \leq i \leq \lceil |L|/2 \rceil\}$.

Lemma 5 *D^{ATSC} is Twinless Strongly Connected.*

Proof:
Observe that D^{ATSC} is strongly connected and thus by Theorem 2 to prove it is twinless strongly connected it suffices to show that $G(D^{\mathrm{ATSC}})$ is 2-edge connected. Consider $G(D^{\mathrm{TSCC}})$. The procedure described above is exactly Eswaran and Tarjan's procedure to augment a graph so that it is 2-edge connected. In other words Eswaran and Tarjan's procedure adds the edges[2] A^{ATSC} to $G(D^{\mathrm{TSCC}})$ to obtain a 2-edge connected graph $G(D^{\mathrm{ATSC}})$. □

Observe that the procedure adds $\lceil \frac{|L|}{2} \rceil$ arcs. Thus $\gamma(A^{\mathrm{ATSC}})$ minimally augments D so that it is twinless strongly connected.

4.3 Augmenting an Arbitrary Digraph so that it is Twinless Strongly Connected

We now describe how to put together the two procedures *carefully* so that the number of arcs added by the augmentation procedure is minimal. In particular, we will modify the digraph D by deleting certain arcs[3] so that when

[2] With a slight abuse of notation we use A^{ATSC} to denote the edges corresponding to the arcs in the set.

[3] We note that if $\hat{D}$ is a digraph obtained by deleting arcs in D, then an arc set that augments $\hat{D}$ so that it is twinless strongly connected also augments D to be twinless strongly connected.

the two procedures are applied in sequence, the number of arcs added in the augmentation procedure is minimal.

First, we define some notation that we need to describe our procedure. Let $v \in N^{\text{TSCC}}$. Then $\theta(v)$ is the node in D^{SCC} corresponding to the strong component in D^{TSCC} that contains v. For every $w \in N^{\text{SCC}}$, $\psi(w)$ is the set of nodes in the strong component of D^{TSCC} corresponding to node w. We will call a TSCC in D with exactly one incoming arc and one outgoing arc that are twin arcs a leaf TSCC, and refer to the pair of twin arcs directed into and out of a leaf TSCC as twin leaf arcs. It is fairly straightforward to see that in D^{TSCC}, a leaf TSCC is a node (i.e., TSCC) with exactly one incoming arc and one outgoing arc that are twin arcs.

Let L denote the set of leaf TSCCs in D^{TSCC}. As before, let S, T, and Q, denote the set of source nodes, sink nodes, and isolated nodes in D^{SCC}. We further classify the sources in D^{SCC}, based on whether or not they contain a leaf TSCC when expanded (to its constituent TSCCs) in D^{TSCC}. We denote the set of sources in D^{SCC} that do not contain a leaf TSCC as S^o. Similarly, we denote the set of sinks in D^{SCC} that do not contain a leaf TSCC as T^o. With respect to the set of isolated nodes Q in D^{SCC}, observe that an isolated node in D^{SCC} either corresponds to an isolated node in D^{TSCC}, or it corresponds to several TSCCs in D^{TSCC} that are strongly connected with each other. In the former case the isolated node does not contain any leaf TSCCs. In the latter case, from the fact that this strongly connected component is isolated (i.e., does not have any arcs directed into or out of it from nodes that are not in the strongly connected component) and the fact that the twinless strong component graph has the structure of a bidrected tree when the underlying digraph is strongly connected, it follows that there must be at least two leaf TSCCs contained in it. Let Q^o denote the set of isolated nodes in D^{SCC} that do not contain leaf TSCCs.

We now show that

$$\max\left\{|S|+|Q|, |T|+|Q|, \left\lceil\frac{|L|+|S^o|+|T^o|+2|Q^o|}{2}\right\rceil\right\}, \tag{1}$$

is a lower bound for the number of arcs needed to augment D so that it is strongly connected.

Claim 1 *Equation 1 describes a lower bound on the number of arcs needed to augment D so that it is strongly connected.*

Proof:
Clearly the number of arcs needed to make D strongly connected is a lower bound on the number of arcs needed to make D twinless strongly connected. Thus, $|S|+|Q|$ and $|T|+|Q|$ are lower bounds.

Observe that to make D twinless strongly connected, for each leaf TSCC, we will need to add an arc directed into or out of the leaf TSCC that is distinct from the twin leaf arcs. Notice further any source node in S^o corresponds to a strongly connected component with no leaf TSCC in D. This strongly connected component has no incoming arc and so requires at least one arc directed into it to make D strongly (or twinless strongly) connected. Similarly, any sink node in T^o corresponds to a strongly connected component with no leaf TSCC in D. This strongly connected component has no outgoing arc and so requires at least one arc directed out of it to make D strongly (or twinless strongly) connected. Finally, observe that an isolated node in Q^o corresponds to a strongly connected component with no leaf TSCCs in D. This strongly connected component has no outgoing arcs or incoming arcs, and so requires at least one arc directed into it, and one arc directed out of it to make D strongly (or twinless) connected. Putting all of this together, we obtain that at least $\left\lceil \frac{|L|+|S^o|+|T^o|+2|Q^o|}{2} \right\rceil$ are required to make D twinless strongly connected. $\square$

We now describe how to modify D so that when the two procedures described in Sections 4.1 and 4.2 are applied in sequence a minimal augmentation is obtained. To motivate the modifications needed consider the first step of the augmentation procedure (i.e., the application of Eswaran and Tarjan's augmentation procedure that we described in Section 4.1). Consider D^{SCC} and observe that the augmentation procedure adds exactly one incoming arc to the sources $s(1), \ldots, s(|S|)$ in N^{SCC}, adds exactly one outgoing arc from the sinks $t(1), \ldots, t(|S|)$, and adds one arc directed into and one arc directed out of the isolated nodes $q(1), \ldots, q(|Q|)$. If any $\psi(s(i))$ contains a leaf TSCC, then ensuring that the arc directed into $\psi(s(i))$ is directed into the leaf TSCC takes care of the leaf TSCCs requirement while simultaneously ensuring that an arc is added that is directed into the source $s(i)$. Similarly, if any $\psi(t(i))$ $(1 \leq i \leq |S|)$ contains a leaf TSCC, then ensuring that the arc directed out of $\psi(t(i))$ is directed out of the leaf TSCC takes care of the leaf TSCCs requirement while simultaneously ensuring that an arc is added that is directed out of the sink $t(i)$. For the isolated nodes, as noted earlier, $\psi(q(i))$ either contains two or more leaf TSCCs or is a singleton set. In the former case we can direct the arc into $\psi(q(i))$ into one of the leaf TSCCs in $\psi(q(i))$, and the arc out of $\psi(q(i))$ out of a different leaf TSCC in $\psi(q(i))$. In the latter case there is no choice in selecting the node in $\psi(q(i))$.

Finally consider the sink nodes $t(|S|+1), \ldots, t(|T|)$. For each of these sinks the augmentation procedure adds both arcs directed into the sink and out of the sink. If $\psi(t(i))$ contains two or more leaf TSCCs then we may proceed as the isolated nodes, selecting one leaf TSCC in $\psi(t(i))$ for the incoming arc, and another leaf TSCC in $\psi(t(i))$ for the outgoing arc. However, if $\psi(t(i))$ contains none or one TSCC then the augmentation procedure, if applied without adaptation, may add more arcs than necessary (as it adds an arc directed into this sink as well as directed out of the sink). Therein lies the problem (i.e., if $|S| = |T|$ we would not have had this problem).

To get around this problem we now describe how to modify the augmentation procedure. The modification we propose will delete arcs from the digraph D, to obtain a new digraph $\hat{D}$, so that the number of sources is increased.

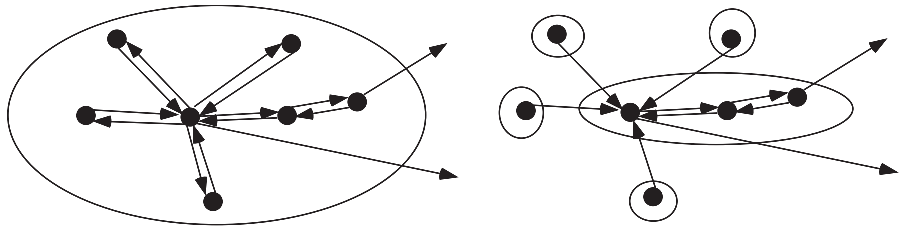

Fig. 4. Leaf TSCCs in a source may be converted into sources by deleting their incoming arcs.

Specifically, we will increase the number of sources by taking a leaf TSCC and deleting its incoming arc. We will do this until the number of sources is equal to the number of sinks, or there are no more leaf TSCCs available for this purpose. We will show that when the two augmentation procedures are applied in sequence to the digraph $\hat{D}$ the number of arcs added is equal to the lower bound in Equation 1. We now elaborate on how this may be done.

Consider a source $s(i)$ with $\psi(s(i))$ containing x leaf TSCCs. Then the number of sources in D^{SCC} may be increased by $y \leq x - 1$ by taking $y + 1$ leaf TSCCs in $\psi(s(i))$ and deleting their incoming arcs (see Figure 4 for an example). For a sink $t(i)$ with $\psi(t(i))$ containing x leaf TSCCs, the number of sources in D^{SCC} may be increased by $y \leq x-1$ by taking one of the leaf TSCCs and deleting its outgoing arc (creating a sink), and taking y of the remaining TSCCs and deleting their incoming arcs (creating sources). For an isolated node $q(i)$ with $\psi(q(i))$ containing x (≥ 2) leaf TSCCs, we may increase the number of sources by $y \leq x - 1$ and sinks by 1 by taking 1 leaf TSCC and deleting its outgoing arc (creating a sink) and taking y of the remaining leaf TSCCs and deleting their incoming arc. We refer to nodes that are neither, source nodes, sink nodes, or isolated nodes in D^{SCC} as *intermediate* nodes. Consider an intermediate node i in D^{SCC}. If $\psi(i)$ contains x leaf TSCCs then the number of sources may be increased by $y \leq x$ by deleting the incoming arc into y of the leaf TSCCs.

We are now ready to explain how to modify D^{TSCC} and apply the augmentation procedure. The algorithm TSCAUG is summarized in Figure 5. The first step is to identify the strongly connected components of D^{TSCC}, and each leaf TSCC in the strongly connected components. These may be done in linear time following the procedure described in Section 3. The next step is to classify each strongly connected component of D^{TSCC} as a source, sink, isolated, or intermediate strongly connected component. This may also be done in linear time (in fact since the procedure to find TSCCs requires identifying SCCs first it may be done as part of the procedure to identify TSCCs). Next, we consider the strongly connected component of D^{TSCC} one by one, while keeping track of the difference between the number of sinks and sources, to identify the arcs that are to be deleted to create $\hat{D}$. When considering a strongly connected component that is a source the procedure deletes an incoming arc from one

algorithm TSCAUG:
1: Construct D^{TSCC}. Identify each strongly connected component in D^{TSCC} and identify the leaf TSCCs in each strongly connected component of D^{TSCC}.
2. Classify each strongly connected component of D^{TSCC} as a source, sink, isolated, or intermediate strongly connected component.
3. Set k to be the difference between the number of sinks and sources. Consider each strongly connected component of D^{TSCC}.
    If it is a source containing leaf TSCCs
        Delete an incoming arc from one leaf TSCC.
        If $k > 0$, create upto k sources from the remaining
            leaf TSCCs by deleting their incoming arcs. Update k.
    If it is a sink containing leaf TSCCs
        Delete an outgoing arc from one leaf TSCC.
        If $k > 0$, create upto k sources from the remaining
            leaf TSCCs by deleting their incoming arcs. Update k.
    If it is isolated containing leaf TSCCs
        Delete an incoming arc from one leaf TSCC and
            delete an outgoing arc from another leaf TSCC.
        If $k > 0$, create upto k sources from the remaining
            leaf TSCCs by deleting their incoming arcs. Update k.
    If it is intermediate containing leaf TSCCs
        If $k > 0$, create upto k sources from the
            leaf TSCCs by deleting their incoming arcs. Update k.
4. Let A^{DEL} denote the arcs deleted from D^{TSCC} in Step 3.
Set $\hat{D} = (N, A \backslash \gamma(A^{\text{DEL}}))$. Or $\hat{D}^{\text{TSCC}} = (N^{\text{TSCC}}, A^{\text{TSCC}} \backslash A^{\text{DEL}})$.
5. Apply Eswaran and Tarjan's strong connectivity augmentation algorithm to $\hat{D}$ to obtain the set of arcs $\beta(\hat{A}^{\text{ASC}})$.
Set $\hat{D} = (N, \hat{A} \cup \beta(\hat{A}^{\text{ASC}}))$.
6. Apply the algorithm described in Section 4.2 to $\hat{D}$ to obtain $\gamma(\hat{A}^{\text{ATSC}})$.
Set $\hat{D} = (N, \hat{A} \cup \gamma(\hat{A}^{\text{ATSC}}))$.
7. Output the arcs $\beta(\hat{A}^{\text{ASC}}) \cup \gamma(\hat{A}^{\text{ATSC}})$.

Fig. 5. Algorithm to solve the twinless strong connectivity augmentation problem.

leaf TSCC in the strongly connected component. If the number of sinks is greater than the number of sources, it also increases the number of sources by converting leaf TSCCs into sources so that the number of sources is equal to the number of sinks, or no more leaf TSCCs remain in the strongly connected component. When considering a strongly connected component that is a sink it deletes an outgoing arc from a leaf TSCC in the strongly connected component, and creates additional sources using leaf TSCCs until the number of sources is equal to the number of sinks, or the leaf TSCCs in the strongly connected component are exhausted. When considering a strongly connected component that is isolated it creates an additional source and an additional sink, and also converts leaf TSCCs to sources as needed. Similarly when considering a strongly connected component that is intermediate it converts leaf TSCCs into sources if needed. Observe that this procedure considers each leaf TSCC in the digraph once and thus is an $\mathcal{O}(|N|)$ procedure. After obtaining $\hat{D}$ we first apply Eswaran and Tarjan's augmentation procedure to strongly connect $\hat{D}$. Recall that this procedure takes $\mathcal{O}(|A|)$ time. Next we apply the procedure described in Section 4.2 which also takes $\mathcal{O}(|A|)$ time. Since each of the steps in the algorithm take linear time, the procedure is an $\mathcal{O}(|N| + |A|)$ procedure.

We now show that the number of arcs added by the procedure is equal to the lower bound shown in Equation 1.

Claim 2 *The number of arcs added by algorithm TSCAUG is equal to the bound in Equation 1.*

Proof:
When the first step of the procedure is applied the procedure adds $|\hat{T}| + \hat{Q}|$ arcs. Observe that in $\hat{D}^{\text{SCC}}$ no source, sink, or isolated node contains leaf TSCCs. Thus no arc added in the first step is directed into or out of a leaf TSCC in $\hat{D}$. Consequently, the second step adds $\left\lceil \frac{|\hat{L}|}{2} \right\rceil$ arcs. Thus a total of $|\hat{T}| + |\hat{Q}| + \left\lceil \frac{|\hat{L}|}{2} \right\rceil$ arcs are added. Notice that,
$|\hat{T}| = |T| + (|Q| - |Q^o|)$,
$|\hat{Q}| = |Q^o|$,
$|\hat{L}| = \max\{0, |L| - (|S| - |S^o| + |T| - |T^o| + 2(|Q| - |Q^o|) + |T| - |S|)\}$.

So the procedure adds

$$(|T| + |Q| - |Q^o|) + |Q^o| +$$
$$\left\lceil \frac{\max\{0, |L| - (|S| - |S^o| + |T| - |T^o| + 2(|Q| - |Q^o|) + |T| - |S|)\}}{2} \right\rceil$$

arcs. Or
$$\max\left\{ |T| + |Q|, \left\lceil \frac{|L| + |S^o| + |T^o| + 2|Q^o|}{2} \right\rceil \right\}$$
arcs, which is the bound in Equation 1. □

5 Applications and Extensions

We now discuss two applications of the TSCC structure. The first application is in the design of survivable telecommunications networks. Magnanti and Raghavan [7] describe a dual-ascent (primal-dual) algorithm for the network design problem with low connectivity requirements (NDLC). In this problem, we are given an undirected graph $G = (N, E)$, with node set N and edge set E, a cost vector $\mathbf{c} \in \mathcal{R}_+^{|E|}$ on the edges E, and a connectivity requirement $r_i \in (0, 1, 2)$ for each node, and wish to design a minimum cost network that contains $r_{st} = \min(r_s, r_t)$ edge-disjoint paths between nodes s and t. To develop a strong mixed-integer programming formulation for the problem they direct the problem by replacing every edge $\{i, j\}$ by a pair of arcs (i, j) and (j, i), each with the same cost as edge $\{i, j\}$. They show that in the equivalent directed problem, we wish to design a minimum cost directed network that contains a directed path from every node with $r_i = 2$ to every node with $r_j = 1$ or 2. Further, any two nodes i and j with $r_i = r_j = 2$ must be twinless reachable. In their dual-ascent algorithm, it is necessary to identify twinless strongly connected components of a so-called auxiliary digraph (to identify ascent directions). The linear time algorithm described in Section 3 serves this purpose.

A second application, described in Baglivo and Graver [1], arises in determining whether the bracing of an architectural structure is rigid (i.e., will not deform). In this application, we are given an $n \times m$ square grid and wish to brace this structure by placing *tension bracings*. A tension bracing consists of a tension wire that can be compressed but not stretched. Baglivo and Graver show that the question of whether a square grid with tension bracings is rigid may be answered by looking at its associated *bracing digraph*. The bracing digraph is constructed as follows. Create a node for each row and each column of the $n \times m$ square grid (i.e., $n + m$ nodes). If a tension wire is attached to the lower left and upper right hand corners of the cell in the ith row and jth column of the square grid, then create a directed arc from the node representing column j to the node representing row i. On the other hand, if the tension wire is attached to the lower right and upper left hand corners of the cell in the ith row and jth column, then create a directed arc from the node representing row i to the node representing column j. Figure 6 gives an example of this construction. Baglivo and Graver show that a tension bracing of an $n \times m$ square grid is rigid if and only if the bracing digraph is strongly connected.

For aesthetic reasons an architect may want to ensure that no cell in the square grid has two diagonal tension bracings [8]. For an existing structure, a question that then arises is whether tension bracings can be removed from cells that contain two diagonal tension bracings while keeping the structure rigid. It is easy to see this corresponds to the question of whether some subdigraph of the bracing digraph is strongly connected and does not contain a pair of twin arcs. This is precisely the definition of twinless strong connectedness, and again the algorithm in Section 3 can be used to determine whether the bracing digraph is twinless strongly connected.

If the bracing digraph is not twinless strongly connected, then a natural question that arises is how it may be minimally augmented so that it is twinless strongly connected. This is an augmentation problem with a twist: the augmented digraph also needs to be bipartite (arcs need to go from nodes

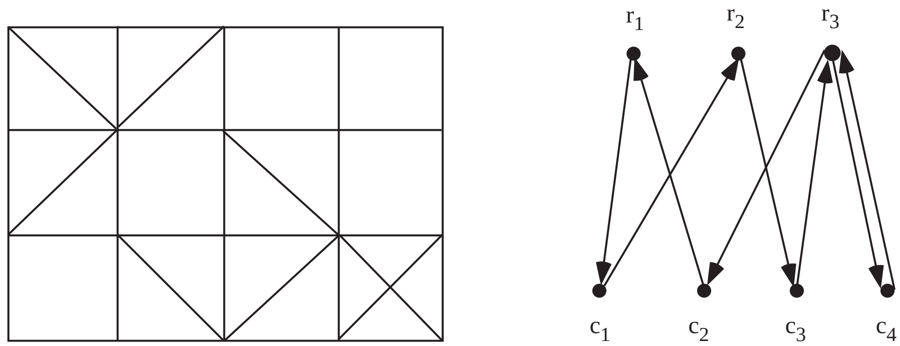

Fig. 6. A 3×4 grid with tension bracings and its corresponding bracing digraph.

representing rows to columns or vice versa); and the solution to this problem remains an open question. Interestingly, the case when we wish to augment the bracing digraph (preserving bipartiteness) so that it is strongly connected was recently solved in [5].

6 On Strongly Connected Orientations of Mixed Graphs and Twinless Strongly Connected Digraphs

A referee pointed out the connection between the notion of a twinless strongly connected digraph and strongly connected orientations of mixed graphs. We discuss this connection here.

A mixed graph $M = (N, A, E)$ is a graph with node set N that contains both directed arcs A and undirected edges E. A natural question that arises in such graphs is whether the undirected edges can be directed (oriented) so that the resulting digraph is strongly connected. Boesch and Tindall [2] show that a mixed graph has a strongly connected orientation if and only if it is connected[4] and the underlying undirected graph[5] of the mixed graph is 2-edge connected.

The question of whether a digraph D is twinless strongly connected can be translated into a question of whether a mixed graph has a strongly connected orientation as follows. Replace each pair of twin arcs (i, j) and (j, i) in the digraph D by an undirected edge $\{i, j\}$ to obtain a mixed graph $M(D)$. It is easy to observe that the digraph D is twinless strongly connected if and only if $M(D)$ has a strongly connected orientation. Based on this connection it is possible to provide an alternate proof of some of the results in Section 2.

[4] In other words, it is possible to go from any node to any other node between using a sequence of undirected and directed arcs (directed arcs must be traversed in the direction they are oriented).

[5] The underlying undirected graph is obtained by replacing all directed arcs by undirected edges.

After learning about the connection between strong orientations of mixed graphs and twinless strongly connected digraphs, we did a web search and found Gusfield [6] considers the following mixed graph augmentation problem. Given a mixed graph $M = (N, A, E)$ find the minimum number of directed arcs to add to the mixed graph so that it has a strongly connected orientation. By transforming the digraph D to the mixed graph $M(D)$, as described in the previous paragraph, our problem may be solved as a mixed graph augmentation problem on $M(D)$. Alternatively, the mixed graph augmentation problem may be transformed to the problem of augmenting a digraph so that it is twinless strongly connected as follows. First, take each pair of twin arcs (i, j) and (j, i) in the mixed graph M and replace (j, i) by adding a new node k_{ij} to the mixed graph and two arcs (j, k_{ij}) and (k_{ij}, i).[6] Next replace each undirected edge $\{i, j\}$ by a pair of twin arcs (i, j) and (j, i) to obtain the digraph $D(M)$. It is easy to observe that the mixed graph M has a strong orientation if and only if $D(M)$ is twinless strongly connected. Thus the solution to the augmentation problem for the mixed graph M may be obtained by solving the twinless strongly connected augmentation problem on $D(M)$.

Gusfield's augmentation algorithm determines an optimal orientation of the undirected edges of the mixed graph (thus transforming it into a directed graph) after which Eswaran and Tarjan's augmentation algorithm for strong connectivity is applied. This procedure and especially the proofs of correctness are complex and quite intricate (as determining the optimal orientation of the undirected edges is quite challenging!). Our procedure in the context of directed graphs is much simpler, and the proofs of correctness are fairly straightforward. Our procedure nicely illustrates how the sequential application of Eswaran and Tarjan's two augmentation solve the problem of optimally augmenting a graph so it is twinless strongly connected, or alternatively shows how to optimally augment a mixed graph so that it admits a strong orientation.

References

1. J. A. Baglivo and J. E. Graver. *Incidence and Symmetry in Design and Architecture.* Cambridge University Press, Cambridge, UK, 1983.
2. F. Boesch and R. Tindall. Robbin's Theorem for mixed multigraphs. *The American Mathematical Monthly*, 87(9):716–719, 1980.

[6] Observe that the mixed graphs considered by Gusfield may contain both arcs (i, j) and (j, i), in which case nodes i and j are strongly connected. However, without this transformation, in our digraph i and j are not twinless strongly connected. Notice, that it suffices to focus on the mixed graph augmentation problem on this transformed mixed graph. Let $\xi(k_{ij}) = i$ or j for the new nodes added and $\xi(i) = i$ otherwise. The set of arcs A^{AMSC} added to the transformed mixed graph are mapped to arcs on the original mixed graphs by $\xi(A^{\text{AMSC}}) = \{(\xi(x), \xi(y)) | (x, y) \in A^{\text{AMSC}}\}$.

3. T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. MIT Press, 1990.
4. K. Eswaran and R. Tarjan. Augmentation problems. *SIAM Journal on Computing*, 5(4):653–665, 1976.
5. H. N. Gabow and T. Jordán. How to make a square grid framework with cables rigid. *SIAM Journal on Computing*, 30:649–680, 2000.
6. D. Gusfield. Optimal mixed graph augmentation. *SIAM Journal on Computing*, 16:599–612, 1987.
7. T. L. Magnanti and S. Raghavan. A dual-ascent algorithm for low-connectivity network design. Technical report, Smith School of Business, University of Maryland, College Park, MD 20742, 2006.
8. A. Recksi. Private Communication at Discrete Optimization 1999, Rutgers University, 1999.
9. R. Tarjan. Depth-first-search and linear graph algorithms. *SIAM Journal on Computing*, 1:146–160, 1972.