PARAMETERIZED COMPLEXITY OF ARC-WEIGHTED DIRECTED STEINER PROBLEMS*

JIONG GUO[†], ROLF NIEDERMEIER[‡], AND ONDŘEJ SUCHÝ[§]

Abstract. We start a systematic parameterized computational complexity study of three NP-hard network design problems on arc-weighted directed graphs: directed Steiner tree, strongly connected Steiner subgraph, and directed Steiner network. We investigate their parameterized complexities with respect to the three parameterizations: "number of terminals," "an upper bound on the size of the connecting network," and the combination of these two. We achieve several parameterized hardness results as well as some fixed-parameter tractability results, in this way extending previous results of Feldman and Ruhl [*SIAM J. Comput.*, 36 (2006), pp. 543–561].

Key words. computational intractability, W[1]-hardness, W[2]-hardness, fixed-parameter tractability, problem kernel, graph algorithm, network design

AMS subject classifications. 05C85, 68R10, 68W99

DOI. 10.1137/100794560

1. Introduction. Steiner-type problems lie at the heart of network design and connectivity problems [26] (see [30] for a broad account on Steiner tree problems). Roughly speaking, the task in these problems is to find in a given weighted graph a low-cost subgraph that satisfies prescribed connectivity requirements. Most of the corresponding optimization problems are NP-hard. Thus, there are numerous results on polynomial-time approximability [26]. By way of contrast, the study of the parameterized complexity of these problems is much less developed (refer to [3], [12], [21] for fixed-parameter tractability and to [11] for parameterized hardness results concerning the undirected case). Our work contributes new algorithmic and computational hardness results concerning the parameterized complexity of three fundamental NP-hard Steiner problems in arc-weighted directed graphs.

Problem descriptions. For the directed Steiner tree (DST) problem, the task is to connect a distinguished root vertex by directed paths to a set of given terminals. For the strongly connected Steiner subgraph (SCSS) problem, the task is to connect all terminals among each other. Finally, for the directed Steiner network (DSN) problem,

the task is to connect given terminal vertex pairs. Obviously, DST and SCSS are special cases of DSN, whereas they are "incomparable" to each other. Note that, following standard modeling, we always assume the underlying directed graph to be complete; arcs that do not exist are modeled by assigning them the weight ∞.

Formally, let $\mathbb{N}$ be the set of natural numbers, and let $W \subseteq \mathbb{N} \cup \{0, \infty\}$. If V is a set of vertices, $w \colon V \times V \to W$ is a weight function,[1] and $A \subseteq V \times V$ is a set of arcs; then we define $w(A) \coloneqq \sum_{a \in A} w(a)$. We study the following:

Directed Steiner Tree (DST):

Instance: A set of vertices V, a weight function $w \colon V \times V \to W$, a set $T \subseteq V$ of terminals ($l \coloneqq |T|$), a root $s \in V$, and a weight bound $p \in \mathbb{N}$.

Question: Is there a set of arcs $A \subseteq V \times V$ of weight $w(A) \leq p$ such that in the digraph $D \coloneqq (V, A)$ for every $t \in T$ there is a directed path from s to t?

Strongly Connected Steiner Subgraph (SCSS):

Instance: A set of vertices V, a weight function $w \colon V \times V \to W$, a set $S \subseteq V$ of terminals ($l \coloneqq |S|$), and a weight bound $p \in \mathbb{N}$.

Question: Is there a set of arcs $A \subseteq V \times V$ of weight $w(A) \leq p$ such that in the digraph $D \coloneqq (V, A)$ for every $s, t \in S$ there is a directed path from s to t?

Directed Steiner Network (DSN):

Instance: A set of vertices V, a weight function $w \colon V \times V \to W$, l pairs of vertices $(s_1, t_1), (s_2, t_2), \ldots, (s_l, t_l)$, and a weight bound $p \in \mathbb{N}$.

Question: Is there a set of arcs $A \subseteq V \times V$ of weight $w(A) \leq p$ such that in the digraph $D \coloneqq (V, A)$ for every $1 \leq i \leq l$ there is a directed path from s_i to t_i?

We set $\min_W \coloneqq \min(W \setminus \{0\})$ and $\max_W \coloneqq \max(W \setminus \{\infty\})$.

To achieve full modeling flexibility (including the cases where one wants to augment an already existing digraph), we sometimes also use arcs of weight 0 to represent already existing connection structure that comes for free. Allowing only arcs of weights 0 and 1 is known in the literature as the *augmentation problem* [13], and allowing only arcs of weights 1 and ∞ models the case that one searches for a minimum-size subgraph, for example, including the classical unweighted DST problem [12]. We distinguish between 0-DSN and DSN, indicating whether 0-weights are allowed or not (analogously, 0-SCSS, SCSS, 0-DST, DST). Moreover, inspired by Papadimitriou and Yannakakis [29], we consider the *ratio* r of arc weights to be the quotient of the maximum occurring arc weight and the minimum occurring arc weight, excluding 0-weights from consideration. If there are ∞-weight arcs, then we call this the *unbounded ratio*. Clearly, a bounded ratio means that in principle every arc is a candidate for being part of the connecting minimum-cost subgraph. Observe that a higher ratio makes the problem more general as well as allowing arcs of weight 0 does. In the spirit of a multivariate algorithmics approach to computational intractability [17], [28], some meaningful parameterizations of the considered Steiner problems are

- the parameter l denoting the number of terminals to be connected;
- the weight p of the solution divided by the minimum arc weight $\min_W$ (again excluding 0), giving the parameter $p / \min_W$;[2] and
- the combined parameter $(l, p / \min_W)$.

Note that a parameterized hardness result with respect to the combined parameter clearly means hardness results for each single parameter and, on the contrary, a

---

[1] Observe that in this way we implicitly deal with complete digraphs in the sense that only arc weights are specified.

[2] This parameter naturally reflects the number of arcs in the spanning subgraph by providing an upper bound on the number of (nonzero) arcs.

fixed-parameter tractability result for a single parameter trivially extends to the combined parameter.

Previous results. We first describe some results on the polynomial-time approximability of Steiner problems in directed graphs, omitting a lot of the rich literature in this direction. Herein, n denotes the number of vertices and m the number of arcs of finite weight. In general terms, one may say that the considered problems are hard to approximate. For instance, it is known that DSN cannot be approximated to within a factor of $O(2^{\log^{1-\epsilon} n})$ for any fixed $\epsilon > 0$, unless NP $\subseteq$ TIME($2^{\text{poly} \log(n)}$) [9]. The best known approximation factor is $O(l^{1/2+\epsilon})$ for any fixed $\epsilon > 0$ [7]. Moreover, 0-DST cannot be approximated to within a factor of $(1 - \epsilon) \ln l$ for any fixed $\epsilon > 0$, unless NP $\subseteq$ DTIME($n^{O(\log \log n)}$) [15]. The best known approximation factor for DST is $O(l^{\epsilon})$ for any fixed $\epsilon > 0$ [6]. For more information, we refer to surveys [14], [25], [26] on the numerous polynomial-time approximation results for Steiner-type problems.

Much less is known about the parameterized complexity [11], [19], [27] of directed Steiner problems. The basic Steiner tree problem (find a subtree of an undirected graph connecting the terminals of minimum size) is known to be W[2]-complete with respect to the parameter "number of nonterminals in the solution" [11]. On the contrary, it is fixed-parameter tractable (FPT) with respect to the parameter "number of terminals" [3], [12], [21]. Both results also transfer to the directed case. In particular, the FPT algorithm can also be used to solve 0-DST, yielding its fixed-parameter tractability with respect to the parameter "number of terminals." Moreover, since the set cover problem is W[2]-complete [11] and it can also be formulated as a special case of both 0-DST and 0-SCSS [20], it follows that 0-DST, 0-SCSS, and 0-DSN are W[2]-hard with respect to the parameter $p/\min_W$. Finally, Feldman and Ruhl [16] showed that 0-DSN can be solved in $O(mn^{4l-2} + n^{4l-1} \log n)$ time using their $O(mn^{2l-3} + n^{2l-2} \log n)$-time algorithm for 0-SCSS as a subprocedure. These algorithmic results directly lead to the question whether there are polynomial-time algorithms whose polynomial degree is independent of l. In other words, Feldman and Ruhl asked for the fixed-parameter tractability of (0-)DSN and (0-)SCSS with respect to the parameter l.

Our results. We extend the above results by initiating a systematic study of the parameterized complexity of the Steiner problems discussed above (also see Table 1.1). First, we show W[1]-hardness results even with respect to the combined parameter $(l, p/\min_W)$ for all four types of problems (0-DSN, DSN, 0-SCSS, SCSS), answering Feldman and Ruhl's question. Interestingly, for arc-weight ratio r for SCSS (0-SCSS) we obtain W[1]-hardness with respect to the combined parameter only if $r \geq 9$ ($r \geq 4$), whereas we obtain fixed-parameter tractability for SCSS when $r \leq 2$ and mixed results for 0-SCSS for $r < 4$ (see Table 1.1 for details). Notably, also with respect to the combined parameter, 0-SCSS turns out to be FPT for $r = 1$, whereas 0-DSN is W[1]-hard for $r = 1$. As a further intractability result, we show that DST and 0-DST parameterized by the combined parameter $(l, p/\min_W)$ have no polynomial-size problem kernels unless NP $\subseteq$ coNP/poly. This means that, under this complexity-theoretic assumption, there is no polynomial-time preprocessing algorithm for the problems transforming an input instance into an equivalent instance of the same problem whose size is bounded by a polynomial of the parameter. Notably, small problem kernel sizes, that is, polynomial-size problem kernels, are desirable for applications in practice. Fixed-parameter tractability only implies exponential-size problem kernels, and it is open in general whether or not polynomial-size problem kernels can be achieved. As indicated in Table 1.1, our work leaves several challenges for future research, particularly concerning

TABLE 1.1
Parameterized complexity results for (0-)DST, (0-)SCSS, and (0-)DSN. Herein, r denotes the ratio between the maximum and the minimum positive arc weights. For $r = 1$ the problems DST, SCSS, and DSN can be solved in polynomial time.

	Parameter:		
Probl.:	l	$p/\min_W$	combined
DST	$r \geq 1$: FPT [8], [12] $r = \infty$: no poly. kernel (Thm. 3.4)	$r \geq 1$: FPT [8], [12] $r = \infty$: no poly. kernel (Thm. 3.4)	$r \geq 1$: FPT [8], [12] $r = \infty$: no poly. kernel (Thm. 3.4)
0-DST	$r \geq 1$: FPT [8], [12] $r = \infty$: no poly. kernel (Thm. 3.4)	$r \geq 1$: W[2]-h. [11]	$r \geq 1$: FPT [8], [12] $r = \infty$: no poly. kernel (Thm. 3.4)
SCSS	$r \geq 9$: W[1]-h. (Thm. 3.1) $2 < r < 9$: open $r \leq 2$: FPT (Thm. 4.1)	$r \geq 9$: W[1]-h. (Thm. 3.1) $2 < r < 9$: open $r \leq 2$: FPT (Thm. 4.1)	$r \geq 9$: W[1]-h. (Thm. 3.1) $2 < r < 9$: open $r \leq 2$: FPT (Thm. 4.1)
0-SCSS	$r \geq 4$: W[1]-h. (Thm. 3.2) $1 < r < 4$: open $r = 1$: FPT (Thm. 4.2)	$r \geq 1$: W[2]-h. [11], [20]	$r \geq 4$: W[1]-h. (Thm. 3.2) $1 < r < 4$: open $r = 1$: FPT (Thm. 4.2)
DSN	$r \geq 9$: W[1]-h. (Thm. 3.1) $1 < r < 9$: open	$r \geq 9$: W[1]-h. (Thm. 3.1) $1 < r < 9$: open	$r \geq 9$: W[1]-h. (Thm. 3.1) $1 < r < 9$: open
0-DSN	$r \geq 1$: W[1]-h. (Thm. 3.3)	$r \geq 1$: W[2]-h. [11], [20]	$r \geq 1$: W[1]-h. (Thm. 3.3)

the parameterized complexity for small arc-weight ratios.

2. Preliminaries. We use $\{u, v\}$ to denote the undirected edge between vertices u and v and use (u, v) to denote the arc directed from u to v. Moreover, in a directed graph $D = (V, A)$, the *in-degree* (*out-degree*) of a vertex u denotes the number of vertices which have arcs directed to u (from u). A graph is *strongly connected* if between each pair of vertices u and v there is a path from u to v and a path from v to u.

A *parameterized problem* is a set $L \subseteq \Sigma^* \times \mathbb{N}$, where Σ is some finite alphabet. An instance $(x, k) \in \Sigma^* \times \mathbb{N}$ of a parameterized problem consists of main part x and parameter k. A given parameterized problem L is FPT with respect to a parameter k if there is an algorithm deciding whether the instance $(x, k) \in \Sigma^* \times \mathbb{N}$ belongs to L in $f(k) \cdot |x|^{O(1)}$ time for some computable function f. It is important to realize that the degree of the polynomial has to be independent of the parameter. A core tool in the development of fixed-parameter algorithms is polynomial-time preprocessing by *data reduction rules*, often yielding a *problem kernel* [4], [22]. Herein, the goal is, given any problem instance G with parameter k, to transform it in polynomial time into a new instance G' with parameter k' such that the size of G' is bounded from above by some function only depending on k, $k' \leq k$, and (G, k) is a yes-instance if and only if (G', k') is a yes-instance. There is a hierarchy for parameterized complexity, called the *W-hierarchy* [11]. At the 0th level of this hierarchy lies the class FPT of fixed-parameter tractable problems. The class of all problems at the ith level of the W-hierarchy ($i > 0$) is denoted by W[i], W[1] being the lowest class of presumably fixed-parameter intractable problems. A parameterized problem Q is *FPT-reducible* to a parameterized problem Q' if there exists an algorithm of running time $f(k) \cdot |x|^{O(1)}$ that on an instance (x, k) of Q produces an instance $(x', g(k))$ of Q' such that (x, k) is a yes-instance of Q if and only if $(x', g(k))$ is a yes-instance of Q', where the functions f and g depend only on k. A parameterized problem Q is W[i]-hard if every problem in W[i] is FPT-reducible to Q. A

parameterized problem shown to be W[1]-hard gives a strong indication for fixed-parameter intractability [11], [19], [27].

In the context of lower bounds for problem kernel sizes we need the concept of composition algorithms [5]. A *composition algorithm* for a parameterized problem $L \subseteq \Sigma^* \times \mathbb{N}$ is an algorithm that receives as input a sequence $((x_1, k), \ldots, (x_t, k))$ with $(x_i, k) \in \Sigma^* \times \mathbb{N}$ for each $1 \leq i \leq t$, uses time polynomial in $\sum_{i=1}^{t} |x_i| + k$, and outputs $(y, k') \in \Sigma^* \times \mathbb{N}$ with

1. $(y, k') \in L \Leftrightarrow (x_i, k) \in L$ for some $1 \leq i \leq t$, and
2. k' being polynomial in k.

3. Parameterized hardness results. In this section, we present new parameterized hardness results for DST, SCSS, and DSN. We start with SCSS. We provide an FPT-reduction from the W[1]-complete multicolored clique (MCC) problem [18]. In MCC we are given an undirected graph that is properly colored[3] by k colors, and the question is whether there is a k-vertex clique in it taking exactly one vertex from each color class. The parameter is k.

THEOREM 3.1. *SCSS with arc-weight ratio at least 9 is W[1]-hard with respect to the combined parameter $(l, p/\min_W)$.*

Proof. Let an undirected graph $G = (V, E)$, an integer $k \geq 1$, and a proper coloring $c: V \rightarrow \{1, \ldots, k\}$ form an instance of MCC. In what follows, we construct an instance (V', w, S, p) of SCSS that corresponds to the given instance of MCC in the sense that it is a yes-instance of SCSS if and only if (G, c, k) is a yes-instance of MCC. Since the instance will be constructible in polynomial time and its parameters l and $p/\min_W$ will be bounded by a function of the original parameter k, the construction provides a parameterized reduction between the problems, showing the W[1]-hardness of SCSS with respect to the combined parameter. The three parts of the high-level idea of the construction of the corresponding SCSS instance are as follows.

First, for every fixed arc-weight ratio $r \geq 9$, we will use only two weights for the arcs between the vertices in V', the cheap arcs having weight $\min_W$ and the expensive arcs having weight $\max_W$ (∞ if the ratio is unbounded) with $r = \max_W / \min_W$. It will be shown that there is a path consisting of at most 9 cheap arcs between any pair of vertices, and hence there is always a solution using only cheap arcs. Thus, we consider in the following only such solutions.

Second, for each color i there will be one terminal b_i that has cheap arcs only to and from vertex gadgets (which will be described later) representing the vertices in the MCC graph G of this color. Thus, the paths between b_i and other terminals, which consist of cheap arcs, have to pass through some arcs in some of these vertex gadgets. This corresponds to taking some vertex from this color class into the solution for the MCC instance. A similar gadget will also be used for every pair of distinct colors, representing the choice of the edge connecting the vertices of the appropriate colors.

Third, the vertex gadget for a vertex v of G will consist of two vertices c_v and c'_v and a cheap arc (c_v, c'_v). This arc will be the only cheap one leaving c_v and will also be the only cheap arc entering c'_v. Taking this uniquely defined arc in solutions for the SCSS instance represents the selection of the corresponding vertex into the solution for the MCC instance. The edges of G will be encoded in a similar way. Note, however, that every edge will be encoded twice. Finally, the vertex and edge gadgets will be connected by cheap arcs according to the incidence so that the selected edges are between the selected vertices.

---

[3]That is, each vertex is assigned one color such that no two neighboring vertices receive the same color.
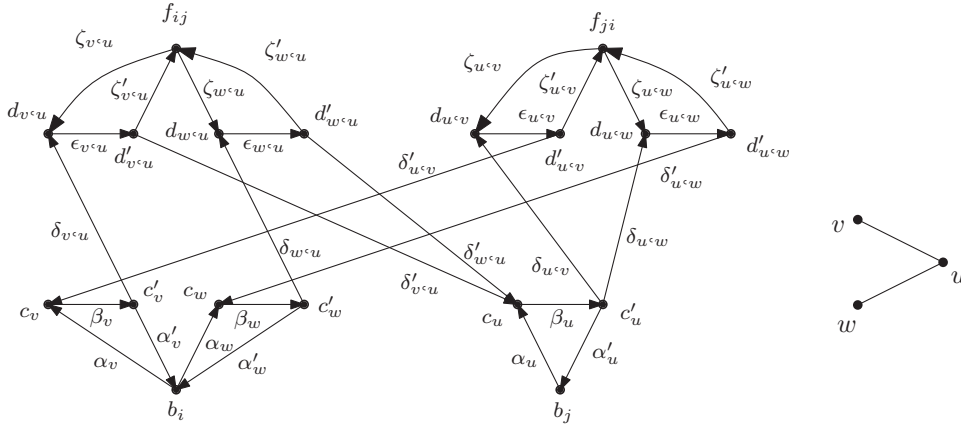
FIG. 3.1. *Part of the construction from Theorem 3.1 with three vertices—v and w of color i and u of color j—and two edges $\{u, v\}$ and $\{u, w\}$. The original graph is drawn on the right. Only the arcs of $\mathcal{Y} \setminus \Gamma$ are drawn for simplicity. The gadget representing the choice of a vertex of color i is in the bottom left corner; the one for a vertex of color j is in the bottom right. The edge selection is represented as a selection of an arc from color-i vertices to color-j vertices in the top left and as a selection of an arc from color-j vertices to color-i vertices in the top right. The gadgets are interconnected according to the incidence of the vertices and the edges.*

After this informal high-level description of the parameterized reduction from MCC to SCSS, we now come to the mathematical details. We construct our instance (V', w, S, p) of SCSS as follows. The set of vertices V' consists of the following six vertex subsets (also see Figure 3.1 for an illustrative example):

$$B := \{b_i | 1 \le i \le k\}, \quad D := \{d_{u,v}, d_{v,u} | \{u, v\} \in E\},$$
$$C := \{c_v | v \in V\}, \quad D' := \{d'_{u,v}, d'_{v,u} | \{u, v\} \in E\},$$
$$C' := \{c'_v | v \in V\}, \quad F := \{f_{ij} | 1 \le i, j \le k, i \ne j\}.$$

The following arcs are given the weight $\min_W$; that is, they are cheap arcs (see Figure 3.1):

$$\mathcal{A} := \{\alpha_v := (b_{c(v)}, c_v) | v \in V\},$$
$$\mathcal{A}' := \{\alpha'_v := (c'_v, b_{c(v)}) | v \in V\},$$
$$\mathcal{B} := \{\beta_v := (c_v, c'_v) | v \in V\},$$
$$\Gamma := \{\gamma_{u,v} := (c'_u, c_v) | u, v \in V\},$$
$$\mathcal{D} := \{\delta_{u,v} := (c'_u, d_{u,v}), \delta_{v,u} := (c'_v, d_{v,u}) | \{u, v\} \in E\},$$
$$\mathcal{D}' := \{\delta'_{u,v} := (d'_{u,v}, c_v), \delta'_{v,u} := (d'_{v,u}, c_u) | \{u, v\} \in E\},$$
$$\mathcal{H} := \{\epsilon_{u,v} := (d_{u,v}, d'_{u,v}), \epsilon_{v,u} := (d_{v,u}, d'_{v,u}) | \{u, v\} \in E\},$$
$$\mathcal{Z} := \{\zeta_{u,v} := (f_{c(u),c(v)}, d_{u,v}), \zeta_{v,u} := (f_{c(v),c(u)}, d_{v,u}) | \{u, v\} \in E\},$$
$$\mathcal{Z}' := \{\zeta'_{u,v} := (d'_{u,v}, f_{c(u),c(v)}), \zeta'_{v,u} := (d'_{v,u}, f_{c(v),c(u)}) | \{u, v\} \in E\},$$
$$\mathcal{Y} := \mathcal{A} \cup \mathcal{A}' \cup \mathcal{B} \cup \Gamma \cup \mathcal{D} \cup \mathcal{D}' \cup \mathcal{H} \cup \mathcal{Z} \cup \mathcal{Z}'.$$

All remaining arcs are set to be expensive arcs; that is, for $(x, y) \in ((V \times V) \setminus \mathcal{Y})$, $w((x, y)) := \max_W$ (∞ if the ratio is unbounded). The terminal set S is $B \cup F$ and hence $l = |S| = |B| + |F| = k + k(k-1) = k^2$. Finally, set $p := (3k + 5k(k-1)) \cdot \min_W$. It is clear that the instance is constructible in polynomial time and that both l and $p / \min_W$

only depend on the parameter k. Next, we show that every k-vertex clique of the MCC instance one-to-one corresponds to a weight-p solution of the SCSS instance.

"⇒": If K is a multicolored clique in G, then one obtains a set A of arcs that form a solution to the SCSS instance as follows:

$$A := \{\alpha_v, \alpha'_v, \beta_v | v \in K\} \cup \{\delta_{u,v}, \delta'_{u,v}, \epsilon_{u,v}, \zeta_{u,v}, \zeta'_{u,v} | u,\ v \in K,\ u \neq v\}.$$

To show that the vertices of S are mutually connected in the digraph (V', A), assume that $v_h \in K$ is the vertex of color h. Now, for every two terminals b_i and b_j, the arcs α_{v_i}, $\beta_{v_i}, \delta_{v_i,v_j}, \epsilon_{v_i,v_j}, \delta'_{v_i,v_j}, \beta_{v_j}, \alpha'_{v_j}$ form a path from b_i to b_j. The arcs $\alpha_{v_i}, \beta_{v_i}, \delta_{v_i,v_j}, \epsilon_{v_i,v_j}, \zeta'_{v_i,v_j}$ form a path from b_i to $f_{i,j}$, and the arcs $\zeta_{v_i,v_j}, \epsilon_{v_i,v_j}, \delta'_{v_i,v_j}, \beta_{v_j}, \alpha'_{v_j}$ form a path from $f_{i,j}$ to b_j. Hence, the set S is strongly connected, and with an easy calculation one shows $w(A) = p$.

"⇐": To show that a weight-p solution of the SCSS instance implies a k-vertex clique of the MCC instance, we need the following claim.

Claim. For every set of arcs $A \subseteq V' \times V'$, there is a set of arcs $A' \subseteq \mathcal{Y}$ with $w(A') \leq w(A)$ such that, for every two vertices $x, y \in V'$, if there is a path from x to y in (V', A), then there is a length-9 path from x to y in (V', A').

Proof of Claim. We construct a set A' by replacing each arc from $A \setminus \mathcal{Y}$ by a set of arcs in $\mathcal{Y}$ and show that, for every $(x, y) \in A$, there is a path from x to y in (V, A'). Since the weight of the replaced arc is $\max_W$ whereas the weight of each replacement arc is $\min_W$, if we introduce at most r replacement arcs for each replaced arc, then $w(A') \leq w(A)$ and the claim follows.

To define the set A'_e of the replacement arcs for a replaced arc $e = (x, y) \in A \setminus \mathcal{Y}$, we distinguish several cases depending on whether the endpoints represent a color, a pair of colors, a vertex, or an edge. Moreover, in order to decrease the number of cases in the analysis, we build A'_e in three steps: first the replacement arcs forming a path from x to some vertex $x' \in C'$, then the replacement arcs forming a path from some vertex $y' \in C$ to y, and, finally, the replacement arc connecting x' to y'. In the first step, we consider the following cases for x:

- If $x = b_i$ for some $1 \leq i \leq k$, then consider an arbitrary vertex v of color $c(v) = i$, add the arcs α_v and β_v to A'_e, and set $x' := c'_v$.
- If $x = c_v$ for some $v \in V$, then add the arc β_v to A'_e, and set $x' := c'_v$.
- If $x = c'_v$ for some $v \in V$, then add no arc to A'_e, and set $x' := x$.
- If $x = d_{u,v}$ for some $u, v \in V$, then add the arcs $\epsilon_{u,v}, \delta'_{u,v}, \beta_v$ to A'_e, and set $x' := c'_v$.
- If $x = d'_{u,v}$ for some $u, v \in V$, then add the arcs $\delta'_{u,v}, \beta_v$ to A'_e, and set $x' := c'_v$.
- If $x = f_{i,j}$ for some $1 \leq i, j \leq k$ then consider an arbitrary edge $\{u, v\} \in E$ such that $c(u) = i$ and $c(v) = j$, add arcs $\zeta_{u,v}, \epsilon_{u,v}, \delta'_{u,v}, \beta_v$ to A'_e, and set $x' := c'_v$.

In the second step, we consider y:

- If $y = b_i$ for some $1 \leq i \leq k$, then consider an arbitrary vertex v with $c(v) = i$, add the arcs β_v and α'_v to A'_e, and set $y' := c_v$.
- If $y = c_v$ for some $v \in V$, then add no arc to A'_e, and set $y' := y$.
- If $y = c'_v$ for some $v \in V$, then add the arc β_v to A'_e, and set $y' := c_v$.
- If $y = d_{u,v}$ for some $u, v \in V$, then add the arcs β_u and $\delta_{u,v}$ to A'_e, and set $y' := c_u$.
- If $y = d'_{u,v}$ for some $u, v \in V$, then add arcs $\beta_u, \delta_{u,v}, \epsilon_{u,v}$, to A'_e, and set $y' := c_u$.
- If $y = f_{i,j}$ for some $1 \leq i, j \leq k$, then consider an arbitrary edge $\{u, v\} \in E$ such that $c(u) = i$ and $c(v) = j$, add arcs $\beta_u, \delta_{u,v}, \epsilon_{u,v}, \zeta'_{u,v}$ to A'_e, and set $y' := c_u$.

In the third step, we add the arc (x', y') to A_e'. This arc exists since $x' \in C'$, $y' \in C$, and Γ has an arc from u to v for every pair of vertices u and v with $u \in C'$ and $v \in C$. Altogether, we add at most four arcs in the first step, at most four in the second step, and the arc (x', y') in the third step to A_e'. That is, we add at most nine replacement arcs for each arc from $A \setminus \mathcal{Y}$. Moreover, it is not hard to check that the replacement arcs form a length-9 path from x to y. This finishes the proof of the claim. □

Given a solution A to the SCSS instance, we can assume, due to the above claim, that $A \subseteq \mathcal{Y}$. We take an arbitrary vertex $f_{i,j} \in F$. All the paths to and from $f_{i,j}$ in $\mathcal{Y}$ pass through some vertex in $D \cup D'$. In order to connect $f_{i,j}$ to other terminals, A must contain at least one arc from each of the sets $\mathcal{Z}, \mathcal{Z}', \mathcal{H}, \mathcal{D}, \mathcal{D}'$, each of them having indices u and v such that $c(u) = i$ and $c(v) = j$ (if more than five arcs are used, then the indices u and v do not have to be the same for all the arcs). This means that A contains a disjoint union of sets of at least five arcs, each of the sets one-to-one corresponding to an $f_{i,j}$. Now take an arbitrary vertex $b_i \in B$. Clearly, A must contain two arcs α_u and α_u' with $c(u) = i$ which connect b_i to all other terminals (again possibly the indices can be different). Moreover, since the vertices in C are sinks and the vertices in C' are sources in the digraph induced by $\mathcal{Y} \setminus B$, there is at least one β_u with $c(u) = i$ in A. This means that A also contains a disjoint union of sets of at least three arcs, each of these sets one-to-one corresponding to a b_i. Since the above mentioned arcs together already give weight $(5 \cdot |F| + 3 \cdot |B|) \cdot \min_W = p$, there is no other arc in A. Let $K := \{v \in V \mid \beta_v \in A\}$. We show that K is a multicolored clique in G.

For each color i there is exactly one vertex of color i in K, since there is exactly one β_u in A among those with $c(u) = i$, as we have shown above. It remains to show that, for two arbitrary vertices u_0 and v_0 in K, there is an edge between u_0 and v_0 in E. As we have shown, there are exactly one $\delta_{u,v}$, exactly one $\delta_{u,v}'$, exactly one $\epsilon_{u,v}$, exactly one $\zeta_{u,v}$, and exactly one $\zeta_{u,v}'$ in A with $c(u) = c(u_0)$, $c(v) = c(v_0)$, and $\{u, v\} \in E$. It is not hard to see that the indices u, v must be the same for these five arcs. Now, if $u \neq u_0$, then β_u is not in A and c_u' is a source in (V', A). Thus, there is a path to $f_{c(u_0),c(v_0)}$ only from c_u', $d_{u,v}$, and $d_{u,v}'$ but from no vertex in S. This is a contradiction, since A is a solution. Hence, $u = u_0$. Similarly, $v = v_0$, and thus $\{u_0, v_0\}$ is an edge of G. □

A similar reduction as above also works for 0-SCSS; however, we can prove hardness already for a smaller arc-weight ratio.

THEOREM 3.2. *0-SCSS with ratio at least 4 is $W[1]$-hard with respect to the combined parameter $(l, p/\min_W)$.*

Proof. Giving a parameterized reduction from MCC to 0-SCSS, we use the same construction as for the proof of Theorem 3.1 and only replace the weight function w by a function w' that is defined as follows. For any $x, y \in V'$, we set $w'((x, y)) := \min_W$ if $(x, y) \in \mathcal{B} \cup \mathcal{H}$, $w'((x, y)) := 0$ if $(x, y) \in \mathcal{Y} \setminus (\mathcal{B} \cup \mathcal{H})$, and $w'((x, y)) := \max_W$ otherwise. We set $p := k^2 \cdot \min_W$. Again, $S := B \cup F$ and $l = k^2$.

It is not hard to check that if K is a multicolored clique in G, then

$$A := \{\alpha_v, \alpha_v', \beta_v \mid v \in K\} \cup \{\delta_{u,v}, \delta_{u,v}', \epsilon_{u,v}, \zeta_{u,v}, \zeta_{u,v}' \mid u, v \in K, u \neq v\}$$

is again a solution to the instance (V', w', S, p) of 0-SCSS. It is also not hard to check that an analogue of the claim shown in the proof of Theorem 3.1 holds in this case. It suffices to take the same replacement and then check that the replacement arcs have weight at most $4 \cdot \min_W$, whereas the replaced arc has weight $\max_W$ which is at least $4 \cdot \min_W$ since the ratio is at least 4.

It remains to show that if there is a solution A such that $A \subseteq \mathcal{Y}$, then there is a multicolored clique in G. To this end, first observe that in $(V', \mathcal{Y} \setminus (\mathcal{B} \cup \mathcal{H}))$ the vertices

in C and D are sinks, while the vertices in C' and D' are sources. Thus, to connect the vertex b_i to the other terminals, there must be at least one β_v in A with $c(v) = i$. Similarly, to connect $f_{i,j}$ to the other terminals it requires at least one $\epsilon_{u,v}$ in A with $c(u) = i$ and $c(v) = j$. Summing these up, we know that there is exactly one such arc in each of these subsets of $\mathcal{Y}$ and, by similar reasons as in the proof of Theorem 3.1, the set $K := \{v \mid \beta_v \in A\}$ forms a multicolored clique in G. $\square$

Theorem 3.2 clearly extends to the more general 0-DSN problem. In what follows, however, we strengthen the W[1]-hardness result by showing that, in case of 0-DSN, it already holds for arc-weight ratio $r = 1$. The reduction for 0-DSN with ratio 1 again is from MCC. However, with now only one value of nonzero weight allowed, a completely different construction is used.

THEOREM 3.3. *0-DSN with arc-weight ratio at least 1 is W[1]-hard with respect to the combined parameter $(l, p / \min_W)$.*

Proof. We reduce from MCC. For each edge and each vertex color there is a pair of vertices that can be connected either directly or by one of several paths formed by two arcs each, exactly one of them of weight zero. The middle vertex of the chosen path represents the choice of the appropriate edge/vertex. Formally, let $G = (V, E)$, k, and $c\colon V \to \{1, \ldots, k\}$ be an instance of MCC. We construct our instance of 0-DSN as follows. The set of vertices V' consists of the following six vertex subsets (see Figure 3.2):

$$B := \{b_i | 1 \le i \le k\},$$
$$B' := \{b'_i | 1 \le i \le k\},$$
$$F := \{f_{ij} | 1 \le i < j \le k\},$$
$$F' := \{f'_{ij} | 1 \le i < j \le k\},$$
$$C := \cup_{i=1}^k C_i, \quad \text{where}$$
$$C_i := \{c_v | v \in V, c(v) = i\},$$
$$D := \cup_{1 \le i < j \le k} D_{ij}, \quad \text{where}$$
$$D_{ij} := \{d_{u,v} | \{u, v\} \in E, c(u) = i, c(v) = j\}.$$

The following arc sets are important for the construction (see Figure 3.2):

$$\mathcal{A} := \{\alpha_v := (b_{c(v)}, c_v) | v \in V\},$$
$$\mathcal{A}' := \{\alpha'_v := (c_v, b'_{c(v)}) | v \in V\},$$
$$\mathcal{B} := \{\beta_{u,v} := (c_u, d_{u,v}), \beta_{v,u} := (c_v, d_{u,v}) | \{u, v\} \in E, c(u) < c(v)\},$$
$$\mathcal{C} := \{\gamma_{u,v} := (f_{c(u),c(v)}, d_{u,v}) | \{u, v\} \in E, c(u) < c(v)\},$$
$$\mathcal{C}' := \{\gamma'_{u,v} := (d_{u,v}, f'_{c(u),c(v)}) | \{u, v\} \in E, c(u) < c(v)\}.$$

Now let $w((x, y)) := 0$ if $(x, y) \in \mathcal{A}' \cup \mathcal{B} \cup \mathcal{C}$, and $w((x, y)) := \max_W = \min_W$ otherwise. The solution is required to connect for every $1 \le i \le k$ the vertex b_i to vertex b'_i and for every $1 \le i < j \le k$ the vertices b_i, b_j and $f_{i,j}$ to $f'_{i,j}$. Hence $l = k + 3k(k-1)/2$. Let the bound on the weight of arcs which are in the solution be $p = (k + k(k-1)/2) \cdot \min_W$.
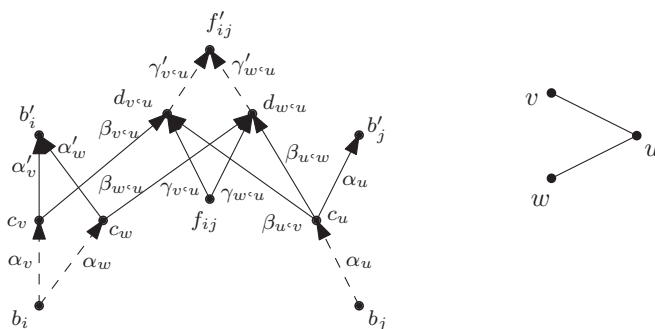
FIG. 3.2. *Part of the construction from Theorem* 3.3 *with three vertices—v and w of color i and u of color j—and two edges $\{u, v\}$ and $\{u, w\}$. The original graph is drawn on the right. The 0-weight arcs of $\mathcal{A}' \cup \mathcal{B} \cup \mathcal{C}$ are drawn as solid, the arcs of $\mathcal{A} \cup \mathcal{C}'$ with weight $\max_W$ as dashed lines.*

It is clear that the instance is constructible in polynomial time and that both l and $p/\min_W$ only depend on the parameter k. It remains to show that there is a k-vertex MCC in G if and only if there is a weight-p solution to the constructed 0-DSN instance.

If K is a multicolored clique in G, then we get a set A of arcs that form a solution to the constructed 0-DSN instance by setting

$$A := \{\alpha_v, \alpha_v' | v \in K\} \cup \{\beta_{u,v}, \beta_{v,u}, \gamma_{u,v}, \gamma_{u,v}' | u, \ v \in K, \ c(u) < c(v)\}.$$

Indeed, if we assume that $v_i \in K$ is the vertex of color i, then for each $i < j$ the arcs α_{v_i} and α_{v_i}' form a directed path from b_i to b_i', whereas the arcs $\alpha_{v_i}, \beta_{v_i,v_j}, \gamma_{v_i,v_j}'$ the arcs $\alpha_{v_j}, \beta_{v_j,v_i}, \gamma_{v_i,v_j}'$ and the arcs $\gamma_{v_i,v_j}, \gamma_{v_i,v_j}'$ form the directed paths to $f_{i,j}'$ from b_i, b_j and $f_{i,j}$, respectively.

For the reverse direction, we make use of the following claim.

Claim. Every solution $A \subseteq V' \times V'$ to the constructed 0-DSN instance uses only arcs in $\mathcal{A} \cup \mathcal{A}' \cup \mathcal{B} \cup \mathcal{C} \cup \mathcal{C}'$. Moreover, for each vertex in $B \cup F'$, there is exactly one arc in A incident with it.

Proof of Claim. We concentrate on the arcs in $\mathcal{X} := (V' \times V') \setminus (\mathcal{A}' \cup \mathcal{B} \cup \mathcal{C})$, that is, the arcs of nonzero weight. The solution A contains at most $p/\min_W$ of them. Each vertex in B has out-degree at least 1 with respect to $A \cap \mathcal{X}$ since it is connected to a vertex from B' in (V', A). For each $1 \le i < j \le k$, at least one of the vertices in $D_{ij} \cup \{f_{ij}\}$ has out-degree at least 1 in $A \cap \mathcal{X}$ since the vertex f_{ij} is connected to f_{ij}'. But this already gives $k + (1/2)k(k-1) = p/\min_W$ arcs. Hence, no vertex in $C \cup B' \cup F'$ has nonzero out-degree in $A \cap \mathcal{X}$, the vertices in B have out-degree exactly one, and for each $1 \le i < j \le k$ exactly one vertex from $D_{ij} \cup \{f_{ij}\}$ has out-degree exactly one. By similar reasons the vertices in F' have in-degree exactly one, for each $1 \le i \le k$ there is exactly one vertex in $C_i \cup \{b_i'\}$ with in-degree one, and all the other vertices have in-degree zero.

Now suppose that for some i the solution contains an arc (b_i, x), where $x \notin C_i \cup \{b_i'\}$. Then, however, either $x \in F' \cup B' \setminus \{b_i'\}$, which is not possible since this would mean that x has out-degree 0 in A and b_i is not connected to b_i', or $x \in C$. In this case there must be an arc from some vertex in D to some vertex in C_i or to b_i' in A in order to connect b_i to b_i'. Note that, by what we have already proved, the number of nonzero arcs of the solution that start in B equals the number of arcs that end in $C \cup B'$ and similarly for $F \cup D$ and F'. Hence, if a solution contains an arc from a vertex of D to a vertex of $C \cup B'$, it must also contain an arc from some vertex in B to some vertex in

F', which is impossible as we have already shown. Similarly, we can show that the arc ending in f'_{ij} starts in some vertex of $D_{ij} \cup \{f_{ij}\}$.

Finally, if for some i the arc (b_i, b'_i) was present in the solution, then there would be no chance to get from b_i to f_{ij} for any j, as we have proved that b'_i is a sink in the solution and there is only one arc starting in b_i in any solution. Similarly for the arcs (f_{ij}, f'_{ij}). This proves the claim. □

Given a solution A to the DSN instance, we can assume, due to the above claim, that $A \subseteq \mathcal{A} \cup \mathcal{A}' \cup \mathcal{B} \cup \mathcal{C} \cup \mathcal{C}'$. For each i denote with v_i the uniquely determined vertex such that the arc (b_i, c_{v_i}) is in A. We claim that $K := \{v_i \mid 1 \le i \le k\}$ is a multicolored clique in G. It is clearly multicolored since the vertex v_i must have color i. Now, due to the claim, for each $1 \le i < j \le k$ there is exactly one edge $\{v'_i, v'_j\}$ in E such that $c(v'_i) = i$, $c(v'_j) = j$, and $\gamma'_{v'_i, v'_j} \in A$. Hence, if $v'_i \ne v_i$, then there is no path in A from b_i to f_{ij} since there is no arc from c_{v_i} to $d_{v'_i, v'_j}$ in $\mathcal{B}$. Thus, $v'_i = v_i$. By the same reasoning $v'_j = v_j$ and $\{v_i, v_j\}$ is an edge. This implies that K is a clique in G, completing the proof. □

So far, we neglected studying (0-)DST. It is already known from the literature that DST is FPT for both single parameterizations by l and $p/\min_W$ [8], [12], while 0-DST is FPT with respect to l [8], [12] and it is W[2]-hard with respect to $p/\min_W$ [11]. We can add to these results a proof for the presumable nonexistence of a polynomial-size problem kernel for DST parameterized by the combined parameter $(l, p/\min_W)$. Recall that this implies the nonexistence of polynomial-size problem kernels for the more general 0-DST case as well as for the single parameter cases. To this end, we use a technique introduced by Bodlaender et al. [5]. A similar result for the undirected unweighted Steiner tree problem was proved by Dom, Lokshtanov, and Saurabh [10] as a byproduct of a more complicated proof of presumable nonexistence of a polynomial-size problem kernel for a different problem.

THEOREM 3.4. *There is no polynomial-size problem kernel for DST with unbounded ratio with respect to the combined parameter* $(l, p/\min_W)$ *unless* NP $\subseteq$ coNP$/$poly.

Proof. We show the claim by applying the lower bound technique of Bodlaender et al. [5] to DST. More specifically, we show that there exists a composition algorithm for DST, which implies, according to Lemmas 1 and 2 of [5], that a polynomial problem kernel for DST with respect to the combined parameter would lead to NP $\subseteq$ coNP$/$poly.

Let $(I_1, l, d), \ldots, (I_r, l, d)$ be a set of DST instances where, for $1 \le i \le r$, I_i consists of a vertex set V_i, a weight function $w_i : V_i \times V_i \to W_i$, a set $T_i \subseteq V_i$ of terminals, a root $s_i \in V_i$, and a weight bound $p_i \in \mathbb{N}$. Moreover, $l = |T_1| = \cdots = |T_r|$ and $d = p_1/\min\{W_1 \setminus \{0\}\} = \cdots = p_r/\min\{W_r \setminus \{0\}\}$. By rescaling, we can assume that $\min\{W_1 \setminus \{0\}\} = \cdots = \min\{W_r \setminus \{0\}\}$ and, thus $p_1 = \cdots = p_r$. Note that this can be done without loss of generality, as the minimum obtained this way is upper-bounded by a product of the original minima, and thus the length of its encoding is upper-bounded by the sum of lengths of encodings of the original minima. Also the length of encoding of each other number involved in the instances is this way increased only by an additive factor upper-bounded by the length of encoding of the new minimum. Hence, this operation can be clearly done within polynomial time and does not change the value of the parameter.

Now we show how a composition algorithm constructs a DST instance $(I, l, 2d + 2l)$ which is a yes-instance if and only if there is one i such that (I_i, l, d) is a yes-instance. First, the algorithm adds a set U of $l + 1$ new vertices to $V := \cup_{1 \le i \le r} V_i$. Herein, one new vertex s is the new root, and the remaining l new vertices form the new terminal set $T := \{t_1, \ldots, t_l\}$. The set W of weights of the new instance contains all possible weights

of the old instances. We set $\min_W := \min\{W_1 \setminus \{0\}\}$ and $p := 2_{p_1} + 2l \cdot \min_W$. Let $T_i := \{t_1^i, \ldots, t_l^i\}$ denote the respective terminal sets of the given instances. The new weight function w is defined as follows:

- $w((u, v)) := w_i((u, v))$ if both u and v are from V_i;
- $w((u, v)) := \infty$ if u and v are from different instances;
- $w((u, s)) := \infty$;
- $w((s, u)) := p_i + l \cdot \min_W$ if $u = s_i$ for some $1 \le i \le r$; otherwise, $w((s, u)) := \infty$;
- $w((t_i, u)) := \infty$ for all $1 \le i \le l$;
- $w((u, t_i)) := \min_W$ if $u = t_i^j$ for some $1 \le j \le r$; otherwise, $w((u, t_i)) := \infty$.

Clearly, the above algorithm runs in polynomial time. To show that $(I, l, 2d + 2l)$ is a yes-instance if and only if one of (I_i, l, d) is a yes-instance, observe that from the new root s we can only use the arcs between s and the old roots s_i. By the weight upper bound p, we know that we cannot afford to use more than one of such arcs. Therefore, connecting s to T can be reduced to connecting one of the old roots to its corresponding terminals. This completes the proof. □

4. Algorithmic results. In this section, we present two fixed-parameter algorithms for two variants of SCSS and 0-SCSS that restrict the allowed arc-weight ratio. Recall that, for the combined parameters, in Theorems 3.1 and 3.2 we have shown W[1]-hardness of SCSS for arc-weight ratio $r \ge 9$ and of 0-SCSS for $r \ge 4$. Now, we complement this with fixed-parameter tractability results for $r \le 2$ and $r = 1$, respectively, leaving a small gap of unsettled cases. In addition, we show that these algorithms directly imply a significant running time improvement of the algorithm by Feldman and Ruhl [16] for these relevant cases.

THEOREM 4.1. *SCSS with arc-weight ratio at most* 2 *is solvable in* $O(2^l \cdot l^2 + n^2)$ *and* $O(2^{(p/\min_W)} \cdot (p/\min_W)^2 + n^2)$ *time.*

Proof. We consider only the case that $p/(2\min_w) < l \le p/\min_W$ since a Hamiltonian cycle over the terminals gives a total weight at least $l \cdot \min_W$ and at most $2l \cdot \min_W$. This means that to strongly connect the terminals in S we need an arc set with a minimum weight at least $l \cdot \min_w$ and a maximum weight at most $l \cdot \max_W \le 2l \cdot \min_W$. Thus, $p \ge 2l \cdot \min_W$ always gives yes-instances, while $p < l \cdot \min_W$ gives no-instances. Therefore, the parameters l and $p/\min_W$ are linearly related to each other. In particular, it suffices to show that the problem is solvable in $O(2^l \cdot l^2 + n^2)$ time. To this end, we claim that there is always a Hamiltonian cycle on S having the minimum total weight among all arc sets strongly connecting S. Since a minimum-weight Hamiltonian cycle on S can be found in $O(2^l \cdot l^2)$ time [24], the theorem follows. We allow $O(n^2)$ time to read the entire input.

Claim. Among all arc sets strongly connecting the terminals in S, there is always one arc set A with a minimum total weight such that A induces a Hamiltonian cycle on S.

Proof. Let A denote an arc set strongly connecting S with a minimum total weight. If A induces a Hamiltonian cycle on S, then we are done; otherwise, we construct a new arc set A' from A such that A' induces a Hamiltonian cycle on S and $w(A') \le w(A)$.

Consider a maximal arc set $P \subseteq A \cap (S \times S)$ that forms vertex-disjoint paths on S. Let n_p be the number of such paths. There have to be n_p arcs in A that enter the start-vertices of paths in P and n_p arcs that leave the end-vertices. If there is only one arc a leaving the end-vertex of some path and this arc enters the start-vertex of this path, then the path together with the arc a form a cycle. As this cycle does not contain all terminals (due to our assumptions), there is an arc (x, y) in A leaving this cycle. We modify P by

removing the arc of P leaving x and adding a. If there is an arc from x to a starting point of some other path, we add it also. Further, we add an arc leaving the end-vertex of some other path and entering the start-vertex of this path, as long as this does not create a cycle. Then P is again a maximal set of arcs that forms vertex-disjoint paths on S. We apply the above strategy until every path has an arc leaving its end-vertex and not entering its start-vertex. This is possible as each application reduces the number of paths that do not satisfy the condition. We let n_p still be the number of paths of P. Now there are n_p arcs that leave the end-vertices of the paths and do not enter the start-vertex of the same path. Note that these arcs are distinct from the n_p arcs entering the start-vertices of the paths: if an arc leaving a path p_1 was equal to an arc entering a path p_2, for some paths $p_1 \neq p_2$ of P, then we could add this arc to P and still have a set of paths, contradicting the fact that P is maximal.

We connect the paths of P directly using arcs between their endpoints to construct a Hamiltonian cycle over the terminals. Let A' be the arc set of this Hamiltonian cycle. To connect the paths into a Hamiltonian cycle, we need at most n_p arcs of total weight at most $2n_p \cdot \min_W$, while the arcs of $A \setminus P$ mentioned in the previous step have weight at least $2n_p \cdot \min_W$. Thus, $w(A') \leq w(A)$. ☐ ☐

Next, we consider the "augmentation case" of 0-SCSS, namely, the case that there are only two weights and one of them is zero. In contrast to the W[1]-hardness of the augmentation case of 0-DSN (cf. Theorem 3.3), for 0-SCSS we achieve fixed-parameter tractability with respect to the number of terminals l.

THEOREM 4.2. *0-SCSS with arc-weight ratio 1 is solvable in $O(4^{l^2-l} + n^3)$ time.*

Proof. First note that in this case we have only two weights 0 and $\min_W = \max_W$. Suppose that we are given an input instance of 0-SCSS consisting of V, w, S (where $|S| = l$), and p. If $p/\min_W \geq l$, then the answer is always yes, since we can connect all terminals to a cycle that costs at most $l \cdot \min_W$. So, for the rest of the proof we will assume that $p/\min_W < l$.

We provide four polynomial-time executable data reduction rules that lead to a problem kernel with at most $2 \cdot 2^l + l$ vertices. Let $A_0 := \{a \in V \times V \mid w(a) = 0\}$. To simplify the presentation, the rules are described as modifications of the digraph $H := (V, A_0)$. The vertices of $V \setminus S$ are called *nonterminals*. In the following, we use $N_A^+(v)$ and $N_A^-(v)$ to denote the sets of vertices which have arcs in a set A directed from and to v, respectively. If clear from the context, then we omit the index A. The rules are ordered, and the next rule is always applied after the previous one cannot be applied any more. Later rules never produce a situation where an earlier rule could again be applied. The correctness of the rules follows from the proven fact that an instance produced by a rule is a yes-instance if and only if the original instance is a yes-instance.

Rule 1. Contract strongly connected components of H into a single vertex.

Since the arcs in A_0 can be added to any solution, Rule 1 is clearly correct. Moreover it can be exhaustively applied in $O(n^2)$ time.

Rule 2. For any nonterminal $v \in V \setminus S$ with both $N^-(v) \neq \varnothing$ and $N^+(v) \neq \varnothing$ delete v and connect its neighbors appropriately; that is, continue with the digraph $H' := (V \setminus \{v\}, A_0 \setminus (((\{v\} \times N^+(v)) \cup (N^-(v) \times \{v\})) \cup (N^-(v) \times N^+(v)))$.

After this rule is exhaustively applied, there remain only sources, terminals, and sinks in the digraph, and the connections between them are preserved. Hence, the resulting digraph does not depend on the order in which the vertices are considered. To see the correctness of the rule, it is enough to realize that any arc $a \notin A_0$ of the solution starting in v can be replaced by an arc starting in some sink reachable from v in H. Similarly, any arc $a \notin A_0$ ending in v can be replaced by an arc ending in some source

from which v can be reached in H. This rule can be exhaustively applied in $O(n^3)$ time, since there are n vertices and to apply the rule to one vertex takes $O(n^2)$ time.

Rule 3. Delete all weight-0 arcs between two nonterminals.

The rule can be applied in $O(n^2)$ time. The following claim shows its correctness.

Claim. If neither Rule 1 nor Rule 2 can be applied anymore, then there is an optimal solution that uses no arc of weight 0 between two nonterminals.

Proof of claim. Suppose, on the contrary, that each optimal solution uses some arc in $A_0 \cap ((V \setminus S) \times (V \setminus S))$. Let A be an optimal solution with the minimum number of such arcs, and let $a := (x, y) \in A \cap A_0$ be such an arc; that is, $x, y \in V \setminus S$. Clearly, x is a source and y is a sink in $H = (V, A_0)$; otherwise we could apply Rule 2. There is some arc (of nonzero weight) in A ending in x and some nonzero arc in A starting in y since $A \setminus \{a\}$ is not a solution. We can assume that there is only one arc ending in x in A for the following reason. If $|N_A^-(x)| \geq 2$, then select an arbitrary $x' \in N_A^-(x)$, and replace the arcs ending in x (except (x', x)) by arcs ending in x', and get another optimal solution that satisfies this assumption (each replaced arc was nonzero, as x is a sink in H). Let us call this unique arc (x', x). Similarly, we assume that there is a unique arc starting in y, and we call it (y, y').

Let V_0 denote the minimal set $S \subseteq V_0 \subseteq V$ such that $A \subseteq V_0 \times V_0$, and also assume that A is minimal in the sense that (V_0, A) is a strongly connected digraph. Now distinguish the following sets of vertices (see Figure 4.1):

$$P := \{v \in V_0 \setminus \{y\} | \exists \text{ a path in } (V_0, A \setminus \{a\}) \text{ from } v \text{ to } y\},$$
$$N := \{v \in V_0 \setminus \{x\} | \exists \text{ a path in } (V_0, A \setminus \{a\}) \text{ from } x \text{ to } v\},$$
$$O := V_0 \setminus (P \cup N \cup \{x, y\}).$$

Observe that in $A \setminus \{a\}$ there is no path from any vertex in N to any vertex in P (in particular, $N \cap P = \varnothing$), since otherwise there would be a path from x to y different from (x, y), and thus $A \setminus \{a\}$ would be a solution. There is also no path from O to P and from N to O according to the definition of N, P, and O. If N is empty, then $A' := (A \setminus \{(x', x), a\}) \cup \{(x', y)\}$ is a better solution, since $(V_0 \setminus \{x\}, A')$ is strongly connected, x is a nonterminal, and $w((x', x)) \geq w((x', y))$. Hence, N is nonempty. Similarly, P is nonempty since otherwise $(A \setminus \{a, (y, y')\}) \cup \{(x, y')\}$ would be a better solution.

Since N is nonempty and (V_0, A) is strongly connected, there must be some arc from some vertex in N to some vertex outside N. But, as we have shown, it can end neither in
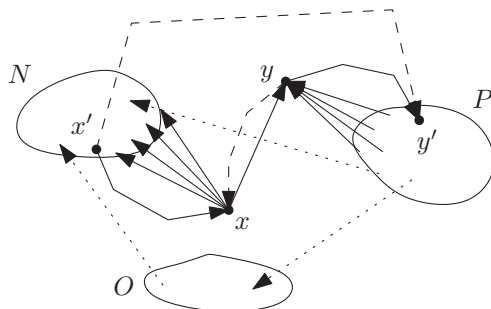

FIG. 4.1. *Illustration to the proof of the claim in Theorem* 4.2. *Solid lines represent the sure connections in* $A \setminus \{(x, y)\}$. *No other connections are possible in* $A \setminus \{(x, y)\}$, *except for those drawn by dotted lines. Dashed lines represent the arcs in* $A' \setminus A$.

O nor in P nor in y. Hence, it ends in x, and thus $x' \in N$. Similarly, $y' \in P$. Now let $A' := (A \setminus \{a, (x', x), (y, y')\}) \cup \{(x', y'), (y, x)\}$. To check that $H' := (V_0, A')$ is again strongly connected, observe that there is a path from x to y in H' formed by a path from x to x' ($x' \in N$), the arc (x', y'), and a path from y' to y ($y' \in P$). The path from x' to x is formed by the arc (x', y'), a path from y' to y, and the arc (y, x), and finally the path from y to y' is formed by the arc (y, x), a path from x to x', and the arc (x', y'). Since $w((x', x)) + w(y, y')) = 2\min_W \geq w((x', y'))$, we have $w(A') \leq w(A)$. Also, due to optimality of A we have $w(A') \leq w(A)$ and $w((x', y')) = w((y, x)) = \min_W$. Thus A' is an optimal solution which uses fewer arcs of weight 0 between two nonterminals—a contradiction.   □

Rule 4. If there are several nonterminals with the same neighborhood in H, then delete all of them except for one.

Rule 4 can be exhaustively applied in $O(n \cdot l \cdot \log n)$ time by first encoding the neighborhood of each vertex as an l-bit string, then sorting these strings in $O(n \cdot l \cdot \log n)$-time and comparing the strings for each of $O(n)$ pairs of vertices consecutive in the order; each comparison takes $O(l)$ time, and if they are the same, deleting one of them can be done in $O(n)$ time. To see the correctness, observe that we can reconnect the solution arcs with nonzero weight incident with deleted vertices to make them incident with the appropriate remaining vertex without affecting any connection between terminals.

Claim. If Rule 4 cannot be applied, then the digraph has at most $2 \cdot 2^l + l$ vertices.

Proof of claim. The neighborhood of a nonterminal is formed only by terminal vertices. Moreover a nonterminal is always either a source or a sink. Hence, there are at most $2 \cdot 2^l$ different neighborhoods and thus by Rule 4 at most $2 \cdot 2^l$ nonterminals. Together with l terminals this gives the claimed bound on the number of the vertices.   □

By the above claim, we have at most $2 \cdot 2^l$ nonterminals in the reduced instance. To solve 0-SCSS on the reduced instance, try all possibilities to connect at most $p/\min_W$ sinks out of 2^l many to at most $p/\min_W$ sources out of 2^l many, and check whether in the resulting digraph the terminals are mutually interconnected. We have $\binom{2^l}{l-1}$ ways to choose the sinks, $\binom{2^l}{l-1}$ ways to choose the sources, and $(l-1)!$ ways to interconnect them. Since at most $3l - 2$ vertices are involved in such a solution, its strong connectivity can be checked in $O(l^2)$ time. This yields time $O(\binom{2^l}{l-1} \cdot \binom{2^l}{l-1} \cdot (l-1)! \cdot l^2) = O(4^{l^2-l})$ for this part of the algorithm. Thus, 0-SCSS with ratio 1 can be solved in $O(4^{l^2-l} + n^3)$ time.   □

The (0-)DSN algorithm developed by Feldman and Ruhl [16] uses an algorithm for (0-)SCSS as a subprocedure. Using the algorithms developed in the proofs of Theorems 4.1 and 4.2 in case of arc-weight ratios 2 and 1, respectively, as the subprocedure, the running time of the overall algorithm of Feldman and Ruhl can be significantly improved by roughly halving the degree of its running time polynomial for the relevant case of these small arc-weight ratios.

COROLLARY 4.3. *DSN with arc-weight ratio* 2 *and* 0-*DSN with arc-weight ratio* 1 *can be solved in* $O(2^{2l} \cdot l^2 \cdot n^{2l})$ *time and* $O(128^{2l^2-l} n^{2l} + n^{2l+3})$ *time, respectively.*

Proof. The (0-)DSN algorithm of Feldman and Ruhl [16] is based on a simulation of a game in which one moves some tokens along the arcs of the input digraph, the price of the move being the total weight of the arcs used. To this end, they construct a so-called "game graph," with vertices representing the possible token positions and arcs representing the legal moves. For (0-)DSN with l terminal pairs a game graph for l tokens is constructed, and the solution is then determined by a single shortest path computation in this graph. One type of legal move is moving some $k \leq l$ tokens along some strongly

connected subgraph to their new positions. The price of such a move can be determined by solving an instance of (0-)SCSS with up to $2l$ terminals corresponding to old and new positions of the tokens. For this computation our algorithms can be used. Since there are at most n^{2l} such moves and the time needed to perform the corresponding computations overshadows all other steps of the algorithm (see section 6.2 of [16] for the details on the time complexity of the original algorithm), the result follows.   □

While DSN with arc-weight ratio 2 only allows for a running time exponentially depending on the number l of terminal pairs, in case of arc-weight ratio 1 a simple argument suffices to achieve polynomial-time solvability independent of the value of l.

PROPOSITION 4.4. *DSN with arc-weight ratio* 1 *is solvable in polynomial time.*

Proof. First observe that we can focus attention to arcs between terminals. If a nonterminal was part of an optimal solution, then we could reroute the arcs incident with it to some terminal without increasing the weight of the solution. Let $(V, (s_1, t_1), \ldots, (s_l, t_l))$ and p form an instance of DSN with ratio 1 (a weight function is not necessary in this case). Consider the digraph $D = (V_t, A_t)$, where $V_t \coloneqq \cup_{i=1}^{l} \{s_i, t_i\}$ is the set of terminals and $A_t \coloneqq \{(s_1, t_1), \ldots, (s_l, t_l)\}$.

It is easy to see that each (weakly) connected component can be treated separately. If such a component C is acyclic (that is, it does not contain a directed cycle), then we can compute a topological order of the vertices of C and then connect them according to this order by a simple path, from the first one to the last one. This can be done in polynomial time by standard techniques, and it is easy to see that in this way we obtain a solution of weight $(|C| - 1) \cdot \min_W$. As any solution must use at least $|C| - 1$ arcs to make this component connected, the solution is optimal.

If the considered component C contains a cycle, then at least $|C|$ arcs are needed, and any cycle over the vertices of C constitutes an optimal solution.   □

5. Conclusion. We extended previous work on directed Steiner problems [16]. In particular, we examined the impact of the ratio of the arc weights on the parameterized complexity of three Steiner problems with respect to the considered parameterizations. Table 1.1 in the introduction summarizes known and new results and indicates open questions. Specifically, the parameterized complexity of (0-)SCSS and DSN is unsettled for some small values of the arc-weight ratio. Given the vast literature on polynomial-time approximation algorithms, one may encounter many more questions to study concerning the parameterized complexity of network design problems in general—for instance, the connectivity augmentation problems with arc reversal and complement operations [1]. Moreover, it would also be interesting to investigate whether some restrictions on the graph structure, such as planarity, could lead to fixed-parameter tractability results of Steiner-type problems (see Bateni, Hajiaghayi, and Marx [2] for some approximation results). Finally, we contributed to parameterized complexity results for NP-hard problems on *directed* graphs, still a comparatively little-developed field within parameterized algorithmic graph theory (cf. [23]).

REFERENCES

[1] E. M. ARKIN, R. HASSIN, AND S. SHAHAR, *Increasing digraph arc-connectivity by arc addition, reversal and complement*, Discrete Appl. Math., 122 (2002), pp. 13–22.

[2] M. Bateni, M. Hajiaghayi, and D. Marx, *Approximation schemes for Steiner forest on planar graphs and graphs of bounded treewidth*, in Proc. 42nd STOC, ACM, New York, 2010, pp. 211–220.

[3] A. Björklund, T. Husfeldt, P. Kaski, and M. Koivisto, *Fourier meets Möbius: Fast subset convolution*, in Proc. 39th STOC, ACM, New York, 2007, pp. 67–74.

[4] H. L. Bodlaender, *Kernelization: New upper and lower bound techniques*, in Proc. 4th IWPEC, Lecture Notes in Comput. Sci. 5917, Springer, New York, 2009, pp. 17–37.

[5] H. L. Bodlaender, R. G. Downey, M. R. Fellows, and D. Hermelin, *On problems without polynomial kernels*, J. Comput. System Sci., 75 (2009), pp. 423–434.

[6] M. Charikar, C. Chekuri, T.-Y. Cheung, Z. Dai, A. Goel, S. Guha, and M. Li, *Approximation algorithms for directed Steiner problems*, J. Algorithms, 33 (1999), pp. 73–91.

[7] C. Chekuri, G. Even, A. Gupta, and D. Segev, *Set connectivity problems in undirected graphs and the directed Steiner network problem*, in Proc. 19th SODA, SIAM, Philadelphia, 2008, pp. 532–541.

[8] B. Ding, J. X. Yu, S. Wang, L. Qin, X. Zhang, and X. Lin, *Finding top-k min-cost connected trees in databases*, in Proc. 23rd ICDE, IEEE, Washington, DC, 2007, pp. 836–845.

[9] Y. Dodis and S. Khanna, *Designing networks with bounded pairwise distance*, in Proc. 31th STOC, ACM, New York, 1999, 750–759.

[10] M. Dom, D. Lokshtanov, and S. Saurabh, *Incompressibility through colors and IDs*, in Proc. 36th ICALP, Lecture Notes in Comput. Sci. 5555, Springer, New York, 2009, pp. 378–389.

[11] R. G. Downey and M. R. Fellows, *Parameterized Complexity*, Springer, New York, 1999.

[12] S. E. Dreyfus and R. A. Wagner, *The Steiner problem in graphs*, Networks, 1 (1972), pp. 195–207.

[13] K. P. Eswaran and R. E. Tarjan, *Augmentation problems*, SIAM J. Comput., 5 (1976), pp. 653–665.

[14] G. Even, *Recursive greedy methods*, in Handbook of Approximation Algorithms and Metaheuristics, T. F. Gonzalez, ed., CRC, Boca Raton, FL, 2007, ch. 5.

[15] U. Feige, *A threshold of* ln n *for approximating set cover*, J. ACM, 45 (1998), pp. 634–652.

[16] J. Feldman and M. Ruhl, *The directed Steiner network problem is tractable for a constant number of terminals*, SIAM J. Comput., 36 (2006), pp. 543–561.

[17] M. R. Fellows, *Towards fully multivariate algorithmics: Some new results and directions in parameter ecology*, in Proc. 20th IWOCA, Lecture Notes in Comput. Sci. 5874, Springer, New York, 2009, pp. 2–10.

[18] M. R. Fellows, D. Hermelin, F. A. Rosamond, and S. Vialette, *On the parameterized complexity of multiple-interval graph problems*, Theoret. Comput. Sci., 410 (2009), pp. 53–61.

[19] J. Flum and M. Grohe, *Parameterized Complexity Theory*, Texts in Theoretical Computer Science. An EATCS Series, Springer, New York, 2006.

[20] A. Frank, *Augmenting graphs to meet edge-connectivity requirements*, SIAM J. Discrete Math., 5 (1992), pp. 25–53.

[21] B. Fuchs, W. Kern, D. Mölle, S. Richter, P. Rossmanith, and X. Wang, *Dynamic programming for minimum Steiner trees*, Theory Comput. Syst., 41 (2007), 493–500.

[22] J. Guo and R. Niedermeier, *Invitation to data reduction and problem kernelization*, ACM SIGACT News, 38 (2007), pp. 31–45.

[23] G. Gutin and A. Yeo, *Some parameterized problems on digraphs*, The Computer Journal, 51 (2008), 363–371.

[24] M. Held and R. M. Karp, *A dynamic programming approach to sequencing problems*, J. SIAM, 10 (1962), 196–210.

[25] S. Khuller, *Approximation algorithms for finding highly connected subgraphs*, in Approximation Algorithms for NP-hard Problems, D. S. Hochbaum, ed., PWS, Boston, 1997.

[26] G. Kortsarz and Z. Nutov, *Approximating minimum cost connectivity problems*, in Handbook of Approximation Algorithms and Metaheuristics, T. F. Gonzalez, ed., CRC, Boca Raton, FL, 2007, ch. 58.

[27] R. Niedermeier, *Invitation to Fixed-Parameter Algorithms*, Oxford University Press, London, 2006.

[28] R. Niedermeier, *Reflections on multivariate algorithmics and problem parameterization*, in Proc. 27th STACS, LIPIcs 5, Schloss Dagstuhl—Leibniz-Zentrum für Informatik, Leibniz, Germany, 2010, pp. 17–32.

[29] C. H. Papadimitriou and M. Yannakakis, *The traveling salesman problem with distances one and two*, Math. Oper. Res., 18 (1993), 1–11.

[30] H. J. Prömel and A. Steger, *The Steiner Tree Problem: A Tour Through Graphs, Algorithms, and Complexity*, Vieweg+Teubner Verlag, Wiesbaden, Germany, 2002.