Math. Program., Ser. A
https://doi.org/10.1007/s10107-018-1270-8

FULL LENGTH PAPER

Beating the 2-approximation factor for global bicut

Kristóf Bérczi[1] · Karthekeyan Chandrasekaran[2] · Tamás Király[1] · Euiwoong Lee[3] · Chao Xu[2]

Received: 29 September 2017 / Accepted: 24 March 2018

Abstract In the fixed-terminal bicut problem, the input is a directed graph with two specified nodes s and t and the goal is to find a smallest subset of edges whose removal ensures that s cannot reach t *and* t cannot reach s. In the global bicut problem, the input is a directed graph and the goal is to find a smallest subset of edges whose removal

A preliminary version of this work appeared in the 20th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX 2017).
Kristóf Bérczi and Tamás Király are supported by the Hungarian National Research, Development and Innovation Office – NKFIH Grants K109240 and K120254 and by the ÚNKP-17-4 New National Excellence Program of the Ministry of Human Capacities.
Chao Xu is supported in part by NSF Grant CCF-1526799.

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s10107-018-1270-8) contains supplementary material, which is available to authorized users.

✉ Karthekeyan Chandrasekaran
karthe@illinois.edu

Kristóf Bérczi
berkri@cs.elte.hu

Tamás Király
tkiraly@cs.elte.hu

Euiwoong Lee
euiwoonl@cs.cmu.edu

Chao Xu
chaoxu3@illinois.edu

1 MTA-ELTE Egerváry Research Group, Department of Operations Research, Eötvös Loránd University, Budapest, Hungary

2 University of Illinois, Urbana-Champaign, USA

3 Carnegie Mellon University, Pittsburgh, USA

Published online: 30 March 2018

ensures that *there exist* two nodes s and t such that s cannot reach t *and* t cannot reach s. Fixed-terminal bicut and global bicut are natural extensions of $\{s, t\}$-min cut and global min-cut respectively, from undirected graphs to directed graphs. Fixed-terminal bicut is NP-hard, admits a simple 2-approximation, and does not admit a $(2-\epsilon)$-approximation for any constant $\epsilon > 0$ assuming the unique games conjecture. In this work, we show that global bicut admits a $(2 - 1/448)$-approximation, thus improving on the approximability of the global variant in comparison to the fixed-terminal variant.

Keywords Digraphs · Bicut · Linear cut · k-cut

Mathematics Subject Classification 05C85

1 Introduction

The global minimum cut problem in undirected graphs is a classic interdiction problem that admits efficient algorithms. In this work, we study the following generalization of this problem from undirected graphs to directed graphs:

BiCut: Given a directed graph, find a smallest subset of edges whose removal ensures that *there exist* two nodes s and t such that s cannot reach t and t cannot reach s.

A natural approach to solving BiCut is by iterating over all pairs of distinct nodes s and t in the input graph and solving the following fixed-terminal bicut problem:

$\{s, t\}$-**BiCut**: Given a directed graph with two specified terminal nodes s, t, find a smallest subset of edges whose removal ensures that s cannot reach t and t cannot reach s.

Clearly, $\{s, t\}$-BiCut is equivalent to 2-terminal multiway-cut in directed graphs (the goal in k-terminal multiway cut is to remove a smallest subset of edges to ensure that s cannot reach t and t cannot reach s for every pair $\{s, t\}$ of the given k terminals). A classic result by Garg, Vazirani and Yannakakis shows that $\{s, t\}$-BiCut is NP-hard [7]. A simple 2-approximation algorithm is to return the union of a minimum $s \to t$ cut and a minimum $t \to s$ cut in the input directed graph. The approximability of $\{s, t\}$-BiCut has seen renewed interest in the last few months culminating in inapproximability results matching the best-known approximability factor [2,13]: $\{s, t\}$-BiCut has no efficient $(2 - \epsilon)$-approximation for any constant $\epsilon > 0$ assuming the Unique Games Conjecture [12]. These results suggest that we have a very good understanding of the complexity and the approximability of the fixed-terminal variant, i.e., $\{s, t\}$-BiCut. In contrast, even the complexity of the global variant, i.e., BiCut, is still an open problem.

The motivations for studying BiCut are multifold. In several network defense/attack applications, global cuts and connectivity are more important than connectivity between fixed pairs of terminals. On the one hand, BiCut is a fundamental global cut problem with interdiction applications involving directed graphs. On the other hand, there is no known complexity theoretic result for BiCut. The fundamental nature of the problem coupled with the lack of basic tractability results are compelling reasons to investigate this problem.

Furthermore, BiCut is an ideal candidate problem to study towards understanding whether cut problems exhibit a dichotomous behaviour between global and fixed-terminal variants in directed graphs. For concreteness, we recall the 3- Cut problem and the 3-way- Cut problem in undirected graphs. In 3- Cut, the input is an undirected graph and the goal is to find a smallest subset of edges whose removal ensures that there exist 3 nodes that cannot reach each other. In 3-way- Cut, the input is an undirected graph with 3 specified nodes and the goal is to find a smallest subset of edges whose removal ensures that the 3 specified nodes cannot reach each other. While the global variant, namely 3- Cut, admits an efficient algorithm [8, 11], the fixed-terminal variant, namely 3-way- Cut, is NP-hard [5]. Such a dichotomy in complexity/approximability between global and fixed-terminal variants is hardly understood in directed graphs. In this work, we exhibit such a dichotomy for directed graphs by focusing on BiCut.

1.1 Results

In this work, we exhibit a dichotomy in the approximability of BiCut and $\{s, t\}$-BiCut. While $\{s, t\}$-BiCut is inapproximable to a constant factor better than 2 assuming UGC, we show that BiCut is approximable to a constant factor that is strictly better than 2. The following is our main result:

Theorem 1 *There exists a polynomial-time* $(2 - 1/448)$*-approximation algorithm for* BiCut.

We emphasize that the complexity of BiCut is still an open problem.

Additional Results on Sub-problems. As a sub-problem in the algorithm for Theorem 1, we consider the following problem:

$(s, *, t)$-**Lin**-3-**Cut**: (abbreviating linear 3-cut): Given a directed graph $D = (V, E)$ and two specified nodes $s, t \in V$, find a smallest subset of edges to remove so that there exists a node r with the property that s cannot reach r and t, and r cannot reach t in the resulting graph.

$(s, *, t)$-Lin- 3- Cut is a global variant of (s, r, t)-Lin- 3- Cut, introduced in [6], where the input specifies three terminals s, r, t and the goal is to find a smallest subset of edges whose removal achieves the property above. A simple reduction from 3-way- Cut shows that (s, r, t)-Lin- 3- Cut is NP-hard. The approximability of (s, r, t)-Lin- 3- Cut was studied by Chekuri and Madan [2]. They showed that the inapproximability factor coincides with the flow-cut gap of an associated *path-blocking linear program* assuming the Unique Games Conjecture. However, the exact approximability factor is still unknown. On the positive side, there exists a simple combinatorial 2-approximation algorithm for (s, r, t)-Lin- 3- Cut.

A 2-approximation for $(s, *, t)$-Lin- 3- Cut can be obtained by iterating over all choices for the terminal r and using the above-mentioned 2-approximation for (s, r, t)-Lin- 3- Cut. However, for the purposes of getting a strictly better than 2-approximation for BiCut, we need a strictly better than 2-approximation for $(s, *, t)$-Lin- 3- Cut. We obtain the following improved approximation factor:

Theorem 2 *There exists a polynomial-time* $3/2$*-approximation algorithm for* $(s, *, t)$-LIN- 3- CUT.

We emphasize that, similar to BICUT, we do not know if $(s, *, t)$-LIN- 3- CUT is NP-hard. Upon encountering cut problems in directed graphs whose complexity is difficult to determine, it is often insightful to consider the complexity of the analogous problem in undirected graphs. Our next result shows that the undirected counterpart of $(s, *, t)$-LIN- 3- CUT is in fact solvable in polynomial time. We observe that reachability in undirected graphs is a symmetric property: if a node s can reach another node t, then the node t can also reach the node s. Hence, the analogous problem in undirected graphs is the following: given an undirected graph with two specified nodes s, t, remove a smallest subset of edges so that the resulting graph has at least 3 connected components with s and t being in different components. More generally, we consider the following:

$\{s, t\}$**-Sep-**k**-Cut**: Given an *undirected graph* $G = (V, E)$ with two specified nodes $s, t \in V$, find a smallest subset of edges to remove so that the resulting graph has at least k connected components with s and t being in different components.
The complexity of $\{s, t\}$-SEP-k-CUT for constant k was posed as an open problem by Queyranne [15]. In this work, we resolve this open problem by showing that $\{s, t\}$-SEP-k-CUT is solvable in polynomial-time for every constant k.

Theorem 3 *For every constant k, there is a polynomial-time algorithm to solve* $\{s, t\}$-SEP-k-CUT.

Organization We set the notation and discuss another cut problem which is useful as a subproblem in our algorithm in Sect. 1.3. We prove Theorems 2 and 3 in Sect. 2 and Theorem 1 in Sect. 3.

1.2 Related work

In spite of an extensive literature on cut problems, we are unaware of any work on BICUT. We mention some work related to the other two problems mentioned in the previous section. (s, r, t)-LIN- 3- CUT was introduced by Erbacher et al. in [6]. They showed that the problem is fixed-parameter tractable when parameterized by the size of the solution.

k-CUT is a well-known partitioning problem in undirected graphs with a rich history. In k-CUT, the input is an undirected graph and the goal is to find a smallest subset of edges to remove so that the resulting graph has at least k connected components. When k is part of the input, this is NP-hard [8] and admits a 2-approximation [16]. When k is a constant, this is solvable in polynomial time [8,11,18].

The fixed-terminal variant of k-CUT is known as k-WAY-CUT. In k-WAY-CUT, the input is an undirected graph with k specified terminals $s_1, \dots, s_k$ and the goal is to find a smallest subset of edges to remove so that no two terminals can reach each other in the resulting graph. It is well-known that k-WAY-CUT is NP-hard [5]. For $k = 3$, a $12/11$-approximation is known [3,9], while for constant k, the current-best approximation factor is 1.2975 due to Sharma and Vondrák [17]. These results are based on an LP-relaxation proposed by Călinescu, Karloff and Rabani [4], known as the CKR relaxation. Manokaran, Naor, Raghavendra and Shwartz [14] showed that the inapproximability factor coincides with the integrality gap of the CKR relaxation. Recently, Angelidakis, Makarychev and Manurangsi [1] exhibited instances with integrality gap at least $6/(5+1/(k-1))-\epsilon$ for every $k \geq 3$ and every $\epsilon > 0$ for the CKR relaxation.

1.3 Preliminaries

We define the notations that we will be using throughout this work. Let $D = (V, E)$ be a directed graph. For two disjoint sets $X, Y \subseteq V$, we denote $\delta_D(X, Y)$ to be the set of edges (u, v) with $u \in X$ and $v \in Y$ and $d_D(X, Y)$ to be the cut value $|\delta_D(X, Y)|$. We use $\delta_D^{in}(X) := \delta_D(V \backslash X, X)$, $\delta_D^{out}(X) := \delta(X, V \backslash X)$, $d_D^{in}(X) := |\delta_D^{in}(X)|$ and $d_D^{out}(X) := |\delta^{out}(X)|$. We define the *cut value* of a set $X \subseteq V$ to be $d_D^{in}(X)$. We drop the subscripts when the graph D is clear from context. We use a similar notation for undirected graphs by dropping the superscripts *in* and *out*. For a subset S of nodes, we define $E[S]$ to be the set of edges in E both of whose end-vertices are in S. For two nodes $s, t \in V$, a subset $X \subset V$ is an $\overline{s}t$*-set* if $t \in X \subseteq V - s$. For two sets $A, B \subseteq V$, let

$$\beta(A, B) := |\delta^{in}(A) \cup \delta^{in}(B)|, \text{ and}$$
$$\sigma(A, B) := |\delta^{in}(A)| + |\delta^{in}(B)|.$$

2 Lin3Cut problems

In this section, we prove Theorems 2 and 3. Theorem 2 gives a 3/2-approximation for $(s, *, t)$-Lin- 3- Cut and is a necessary component of our proof of Theorem 1. Theorem 3 is an investigation of $(s, *, t)$-Lin- 3- Cut in undirected graphs and answers an open problem posed by Queyranne [15].

2.1 A 3/2-approximation for $(s, *, t)$-Lin-3-Cut

One of our main tools used in the approximation algorithm for BiCut is a 3/2-approximation algorithm for $(s, *, t)$-Lin- 3- Cut. We present this algorithm now. We recall the problem $(s, *, t)$-Lin- 3- Cut: Given a directed graph with specified nodes s, t, find a smallest subset of edges whose removal ensures that the graph contains a node r with the property that s cannot reach r and t, and r cannot reach t.

Notations Let V be the node set of a graph. A family $\mathcal{C}$ of subsets of V is a *chain* if for every pair of sets $A, B \in \mathcal{C}$, we have $A \subset B$ or $B \subset A$. We observe that a chain family can have at most $|V|$ non-empty sets. Two sets A and B are *uncomparable* if $A \backslash B$ and $B \backslash A$ are non-empty, and *comparable* otherwise. A set A is *compatible* with a chain $\mathcal{C}$ if $\mathcal{C} \cup \{A\}$ is a chain, and it is *incompatible* otherwise.

We first rephrase the problem in a convenient way.

Lemma 1 $(s, *, t)$-LIN- 3- CUT *in a directed graph* $D = (V, E)$ *is equivalent to*

$$\min\{\beta(A, B) : t \in A \subsetneq B \subseteq V - \{s\}\}.$$

Proof Let $F \subseteq E$ be an optimal solution for $(s, *, t)$-LIN- 3- CUT in D and let

$$(A, B) := \operatorname{argmin}\{\beta(A, B) : t \in A \subsetneq B \subseteq V - s\}.$$

Let us fix an arbitrary node $r \in B - A$. Since the deletion of $\delta^{in}(A) \cup \delta^{in}(B)$ results in a graph with no directed path from s to r, from r to t and from s to t, the edge set $\delta^{in}(A) \cup \delta^{in}(B)$ is a feasible solution to (s, r, t)-LIN- 3- CUT in D, thus implying that $|F| \leq \beta(A, B)$.

On the other hand, F is a feasible solution for (s, r', t)-LIN-3-CUT in D for some $r' \in V - \{s, t\}$. Let A' be the set of nodes that can reach t in $D - F$, and R' be the set of nodes that can reach r' in $D - F$. Then, $F \supseteq \delta^{in}(A')$. Moreover, $F \supseteq \delta^{in}(R' \cup A')$ since $R' \cup A'$ has in-degree 0 in $D - F$, and s is not in $R' \cup A'$ because it cannot reach r' and t in $D - F$. Therefore, taking $B' = R' \cup A'$ we get $F \supseteq \delta^{in}(A') \cup \delta^{in}(B')$. □

The above reformulation shows that the optimal solution is given by a chain consisting of two $\overline{s}t$-sets. The following lemma shows that we can obtain a 3/2-approximation to the required chain.

Lemma 2 *There exists a polynomial-time algorithm that given a directed graph* $D = (V, E)$ *with nodes* $s, t \in V$ *returns a pair of* $\overline{s}t$*-sets* $A \subsetneq B \subseteq V$ *such that*

$$\beta(A, B) \leq \frac{3}{2}\min\{\beta(A, B) : t \in A \subsetneq B \subseteq V - \{s\}\}.$$

Proof The objective is to find a chain of two $\overline{s}t$-sets A, B with minimum $\beta(A, B)$. We can assume $|V| \geq 4$, otherwise we check all possibilities. To obtain an approximation, we build a chain $\mathcal{C}$ of $\overline{s}t$-sets with the property that, for some value $k \in \mathbb{Z}_+$,

(i) every set $C \in \mathcal{C}$ is an $\overline{s}t$-set with $d^{in}(C) \leq k$, and
(ii) every $\overline{s}t$-set T with $d^{in}(T)$ strictly less than k is in $\mathcal{C}$.

We use the following procedure to obtain such a chain: We initialize with k being the minimum $\overline{s}t$-cut value and $\mathcal{C}$ consisting of the sink-side of a single minimum $\overline{s}t$-cut. In a general step, we find two $\overline{s}t$-sets: an $\overline{s}t$-set Y compatible with the current chain $\mathcal{C}$, i.e. $\mathcal{C} \cup \{Y\}$ forming a chain, with minimum $d^{in}(Y)$ and an $\overline{s}t$-set Z *not* compatible with the current chain $\mathcal{C}$, i.e. crossing at least one member of $\mathcal{C}$, with minimum $d^{in}(Z)$. Note that it is possible that Y or Z does not exist; the former happens when the chain is maximal, while the latter happens when $\mathcal{C} = \{\{t\}\}$ or $\mathcal{C} = \{V - \{s\}\}$ or $\mathcal{C} = \{\{t\}, V - \{s\}\}$. Since $|V| \geq 4$, at least one of Y and Z exist.

The required sets Y and Z can be found in polynomial-time as follows: let $t \in C_1 \subset \ldots, \subset C_q \subseteq V - s$ denote the members of $\mathcal{C}$, and let $C_0 = \{t\}$, $C_{q+1} = V - s$. (i) Find a minimum cut $Y_i \notin \mathcal{C}$ with $C_i \subseteq Y_i \subseteq C_{i+1}$ for $i = 0, \ldots, q$, and choose Y to be a set with minimum cut value among these cuts. (ii) For each $i \in \{1, \ldots, q\}$ and for each pair x, y of nodes with $y \in C_i \subseteq V - x$, find a minimum cut Z^i_{xy} with

$\{t, x\} \subseteq Z_{xy}^i \subseteq V - \{s, y\}$, and choose Z to be a set with minimum cut value among these cuts. Since $\mathcal{C}$ is a chain, we have that $q \leq |V|$ and hence both sets Y and Z can be found in polynomial-time. If Z does not exist or $d^{in}(Y) \leq d^{in}(Z)$, then we add Y to $\mathcal{C}$, and set k to $d^{in}(Y)$; otherwise we set k to $d^{in}(Z)$ and stop.

Proposition 1 *Let $\mathcal{C}$ denote the chain before any general step of the above-mentioned procedure. Then, for every $C \in \mathcal{C}$ and for every $\overline{s}t$-set A that is not in $\mathcal{C}$, we have*

$$d^{in}(C) \leq d^{in}(A).$$

Proof Let A be an $\overline{s}t$-set that is not in $\mathcal{C}$. Suppose for the sake of contradiction that $d^{in}(A) < d^{in}(C)$ for some $C \in \mathcal{C}$. Let $\mathcal{C}'$ denote the chain consisting of those members of $\mathcal{C}$ that were added before C. Since $A \notin \mathcal{C}$ and C is a set of minimum cut value compatible with $\mathcal{C}'$, we have that A should cross at least one member of $\mathcal{C}'$. Hence, by $d^{in}(A) < d^{in}(C)$, the procedure stops before adding C to the chain $\mathcal{C}'$, a contradiction to C being in $\mathcal{C}$. □

Proposition 2 *The chain $\mathcal{C}$ and the value k obtained at the end of the above-mentioned procedure satisfy* (i) *and* (ii).

Proof The construction immediately guarantees that every set $C \in \mathcal{C}$ is an $\overline{s}t$-set. By Proposition 1 and by construction of $\mathcal{C}$ and k, we have that $d^{in}(C) \leq k$ for every $C \in \mathcal{C}$ and hence, we have (i).

By construction, $\mathcal{C}$ contains all $\overline{s}t$-sets T with $d^{in}(T) < k$ that are compatible with $\mathcal{C}$. Suppose for the sake of contradiction, we have an $\overline{s}t$-set T with $d^{in}(T) < k$ that is not in $\mathcal{C}$. Then, the set T should be incompatible with $\mathcal{C}$. We note that the procedure terminates by setting k to be the minimum cut value $d^{in}(Z)$ among Z that are incompatible with $\mathcal{C}$. Hence, the procedure should have set k to be a value that is at most $d^{in}(T)$ and terminated. This is a contradiction to $d^{in}(T) < k$. Therefore, there does not exist an $\overline{s}t$-set T with $d^{in}(T) < k$ that is not in $\mathcal{C}$ and hence, we have (ii). □

By the above, the procedure stops with a chain $\mathcal{C}$ containing all $\overline{s}t$-sets of cut value less than k, and an $\overline{s}t$-set Z of cut value exactly k which crosses some member X of $\mathcal{C}$. If the optimum value of our problem is less than k, then both members of the optimal pair (A, B) belong to the chain $\mathcal{C}$, and we can find them by taking the minimum of $\beta(A', B')$ where $A' \subset B'$ with $A', B' \in \mathcal{C}$.

We can thus assume that the optimum is at least k. Since $d^{in}(Z) = k$ and $d^{in}(X) \leq k$, the submodularity of the in-degree function implies

$$d^{in}(X \cap Z) + d^{in}(X \cup Z) \leq d^{in}(Z) + d^{in}(X) \leq 2k.$$

Hence either $d^{in}(X \cap Z) \leq k$ or $d^{in}(X \cup Z) \leq k$. Since

$$d(X \backslash Z, X \cap Z) + d(Z \backslash X, X \cap Z) \leq d^{in}(X \cap Z) \text{ and}$$
$$d(V \backslash (X \cup Z), X \backslash Z) + d(V \setminus (X \cup Z), Z \backslash X) \leq d^{in}(X \cup Z),$$

at least one of the following four possibilities holds:

Approximation Algorithm for $(s,*,t)$-Lin-3-Cut

Input: Directed graph $D=(V,E)$ with $s,t\in V$. We assume $|V|\geq 4$.

1. Let S denote the sink-side of a minimum $s\rightarrow t$ cut and α denote its value. Initialize $\mathcal{C}\leftarrow\{S\}$ and $k\leftarrow\alpha$.
2. Repeat:
   (a) $Y\leftarrow\arg\min\{d^{in}(Y): Y$ is a $\overline{s}t$-set compatible with $\mathcal{C}\}$
   (b) $Z\leftarrow\arg\min\{d^{in}(Z): Z$ is a $\overline{s}t$-set incompatible with $\mathcal{C}\}$
   (c) If such a Z does not exist or if $d^{in}(Y)\leq d^{in}(Z)$, then update $\mathcal{C}\leftarrow\mathcal{C}\cup\{Y\}$ and $k\leftarrow d^{in}(Y)$.
   (d) Else, update $k\leftarrow d^{in}(Z)$, set X to be a set in $\mathcal{C}$ that crosses Z and go to Step 3.
3. Let $(A,B)\leftarrow\arg\min\{\beta(A,B): A,B\in\mathcal{C}, A\neq B\}$.
4. Let $(S,T)\leftarrow\arg\min\{\beta(X\cap Z,X),\beta(X\cap Z,Z),\beta(Z,X\cup Z),\beta(X,X\cup Z)\}$
5. Return $\arg\min\{\beta(A,B),\beta(S,T)\}$.

Fig. 1 Approximation Algorithm for $(s,*,t)$-LIN- 3- CUT

1. $d^{in}(X\cap Z)\leq k$ and $d(X\backslash Z, X\cap Z)\leq\frac{1}{2}k$. Choose $A=X\cap Z$, $B=X$. Then $\beta(A,B)=d(X\backslash Z,X\cap Z)+d^{in}(X)\leq\frac{1}{2}k+k=\frac{3}{2}k$.
2. $d^{in}(X\cap Z)\leq k$ and $d(Z\backslash X, X\cap Z)\leq\frac{1}{2}k$. Choose $A=X\cap Z$, $B=Z$. Then $\beta(A,B)=d(Z\backslash X,X\cap Z)+d^{in}(Z)\leq\frac{1}{2}k+k=\frac{3}{2}k$.
3. $d^{in}(X\cup Z)\leq k$ and $d(V\backslash(X\cup Z), X\backslash Z)\leq\frac{1}{2}k$. Choose $A=Z$, $B=X\cup Z$. Then $\beta(A,B)=d^{in}(Z)+d(V\backslash(X\cup Z),X\backslash Z)\leq k+\frac{1}{2}k=\frac{3}{2}k$.
4. $d^{in}(X\cup Z)\leq k$ and $d(V\backslash(X\cup Z), Z\backslash X)\leq\frac{1}{2}k$. Choose $A=X$, $B=X\cup Z$. Then $\beta(A,B)=d^{in}(X)+d(V\backslash(X\cup Z),Z\backslash X)\leq k+\frac{1}{2}k=\frac{3}{2}k$.

Thus a pair (A,B) can be obtained by taking the minimum among the four possibilities above and $\beta(A',B')$ where $A'\subset B'$ with $A',B'\in\mathcal{C}$, concluding the proof of the approximation factor. The algorithm is summarized in Fig. 1. It remains to ensure that the algorithm can be implemented to run in polynomial-time. We have already seen that Steps 2(a) and 2(b) can be implemented to run in polynomial-time above. Furthermore, the size of the chain $\mathcal{C}$ is at most $|V|$ and hence, Step 3 can also be implemented to run in polynomial time. □

Theorem 2 is a consequence of Lemmas 1 and 2. The approximation algorithm is summarized in Fig. 1.

2.2 An exact algorithm for $\{s,t\}$-SEP-k-CUT

In this section, we show that $\{s,t\}$-SEP-k-CUT is solvable in polynomial time if k is a fixed constant. We recall the problem $\{s,t\}$-SEP-k-CUT: Given an *undirected* graph with specified nodes s,t, find a smallest subset of edges whose removal ensures that the resulting graph has at least k connected components with s and t being in different components.

Notations. Let $G=(V,E)$ be an undirected graph. Let the minimum size of an $\{s,t\}$-cut in G be denoted by $\lambda_G(s,t)$. For two subsets of nodes X,Y, we recall that $d(X,Y)$ denotes the number of edges between X and Y and that $d(X)=d(X,V\backslash X)$. The *cut* *value* of a partition $\{V_1, \ldots, V_q\}$ of V is defined to be the total number of crossing edges, that is, $(1/2)\sum_{i=1}^{q} d(V_i)$, and is denoted by $\gamma(V_1, \ldots, V_q)$. Let $\gamma^q(G)$ denote the value of an optimum q- Cut in G, i.e.,

$$\begin{aligned}\gamma^q(G) := \min\{&\gamma(V_1, \ldots, V_q) : V_i \neq \emptyset\ \forall\, i \in [q],\\ &V_i \cap V_j = \emptyset\ \forall\, i, j \in [q], \cup_{i=1}^{q} V_i = V\}.\end{aligned}$$

Proof of Theorem 3 Let γ^* denote the optimum value of $\{s,t\}$-Sep-k-Cut in $G = (V, E)$, and let H denote the graph obtained from G by adding an edge of infinite capacity between s and t. The algorithm is based on the following observation (we recommend the reader to consider $k = 3$ for ease of understanding):

Proposition 3 *Let $\{V_1, \ldots, V_k\}$ be a partition of V corresponding to an optimal solution of $\{s,t\}$-Sep-k-Cut, where s is in V_{k-1} and t is in V_k. Then $\gamma(V_1, \ldots, V_{k-2}, V_{k-1} \cup V_k) \le 2\gamma^{k-1}(H)$.*

Proof Let $W_1, \ldots, W_{k-1}$ be a minimum $(k-1)$-cut in H. Clearly, s and t are in the same part, so we may assume that they are in W_{k-1}. Let U_1, U_2 be a minimum $\{s,t\}$-cut in $G[W_{k-1}]$. Then $\{W_1, \ldots, W_{k-2}, U_1, U_2\}$ gives an $\{s,t\}$-separating k-cut, showing that

$$\gamma^* \le \gamma(W_1, \ldots, W_{k-2}, U_1, U_2) = \gamma^{k-1}(H) + \lambda_{G[W_{k-1}]}(s,t). \tag{1}$$

By Menger's theorem, we have $\lambda_G(s,t)$ pairwise edge-disjoint paths $P_1, \ldots, P_{\lambda_G(s,t)}$ between s and t in G. Consider one of these paths, say P_i. If all nodes of P_i are from $V_{k-1} \cup V_k$, then P_i has to use at least one edge from $\delta(V_{k-1}, V_k)$. Otherwise, P_i uses at least two edges from $\delta(V_1 \cup \cdots \cup V_{k-2})$. Hence the maximum number of pairwise edge-disjoint paths between s and t is

$$\lambda_G(s,t) \le d(V_{k-1}, V_k) + \frac{1}{2} d(V_1 \cup \cdots \cup V_{k-2}).$$

Thus, we have

$$\begin{aligned}
\gamma^* &= d(V_{k-1}, V_k) + d(V_1 \cup \cdots \cup V_{k-2}) + \sum_{i<j\le k-2} d(V_i, V_j)\\
&\ge \lambda_G(s,t) + \frac{1}{2}\left(d(V_1 \cup \cdots \cup V_{k-2}) + 2\sum_{i<j\le k-2} d(V_i, V_j)\right)\\
&\ge \lambda_G(s,t) + \frac{1}{2}\left(d(V_1 \cup \cdots \cup V_{k-2}) + \sum_{i<j\le k-2} d(V_i, V_j)\right)\\
&= \lambda_G(s,t) + \frac{1}{2}\gamma(V_1, \ldots, V_{k-2}, V_{k-1} \cup V_k)\\
&\ge \lambda_{G[W_{k-1}]}(s,t) + \frac{1}{2}\gamma(V_1, \ldots, V_{k-2}, V_{k-1} \cup V_k)
\end{aligned}$$

Algorithm for $\{s,t\}$-Sep-k-Cut

Input: Undirected graph $G=(V,E)$ with $s,t\in V$

1. Let H be the graph obtained from G by adding an edge of infinite capacity between s and t. In H, enumerate all feasible solutions to $(k-1)$-Cut—namely the vertex partitions $\{W_1,\ldots,W_{k-1}\}$—whose cut value $\gamma_H(W_1,\ldots,W_{k-1})$ is at most $2\gamma^{k-1}(H)$. Without loss of generality, assume $s,t\in W_{k-1}$.
2. For each feasible solution to $(k-1)$-Cut in H listed in Step 1, find a minimum $\{s,t\}$-cut in $G[W_{k-1}]$, say U_1,U_2.
3. Among all feasible solutions $\{W_1,\ldots,W_{k-1}\}$ to $(k-1)$-Cut listed in Step 1 and the corresponding U_1,U_2 found in Step 2, return the k-cut $\{W_1,\ldots,W_{k-2},U_1,U_2\}$ with minimum $\gamma(W_1,\ldots,W_{k-2},U_1,U_2)$.

Fig. 2 Algorithm for $\{s,t\}$-Sep-k-Cut

that is,

$$\gamma^* \geq \lambda_{G[W_{k-1}]}(s,t)+\frac{1}{2}\gamma(V_1,\ldots,V_{k-2},V_{k-1}\cup V_k). \tag{2}$$

By combining (1) and (2), we get $\gamma(V_1,\ldots,V_{k-2},V_{k-1}\cup V_k)\leq 2\gamma^{k-1}(H)$, proving the proposition. □

Karger and Stein [11] showed that the number of feasible solutions to k-Cut in an undirected graph G with value at most $2\gamma^k(G)$ is $O(n^{4k})$. All these solutions can be enumerated in polynomial-time for fixed k [10,11,19]. This observation together with Proposition 3 gives the algorithm for finding an optimal solution to $\{s,t\}$-Sep-k-Cut. The algorithm is summarized in Fig. 2.

The correctness of the algorithm follows from Proposition 3: one of the choices enumerated in Step 1 will correspond to the partition $(V_1,\ldots,V_{k-2},V_{k-1}\cup V_k)$, where $(V_1,\ldots,V_k)$ is the partition corresponding to the optimal solution. □

3 BiCut

In this section, we present our approximation algorithm (Theorem 1) for BiCut. We begin with the high-level ideas of the approximation algorithm in Sect. 3.1. The full algorithm and the proof of its approximation ratio are presented in Sect. 3.2.

We recall the problem BiCut: Given a directed graph, find a smallest number of edges in it whose removal ensures that there exist two distinct nodes s and t such that s cannot reach t and t cannot reach s. We begin with a reformulation of BiCut that is helpful for the purposes of designing an algorithm. We recall that two sets A and B are *uncomparable* if $A\backslash B\neq\emptyset$ and $B\backslash A\neq\emptyset$. We also recall that for two sets of nodes A and B, the quantity $\beta(A,B)=|\delta^{in}(A)\cup\delta^{in}(B)|$.

Definition 1 For a directed graph $D=(V,E)$, let

$$\beta:=\min\{\beta(A,B):\ A \text{ and } B \text{ are uncomparable}\}.$$

The following lemma shows that bicut is equivalent to finding an uncomparable pair of subsets of nodes A, B with minimum $\beta(A,B)$.

Lemma 3 *Let $D = (V, E)$ be a directed graph. The minimum number of edges in D whose removal ensures that there exist two distinct nodes s and t such that s cannot reach t and t cannot reach s is exactly equal to β.*

Proof We show the inequality in both directions. Suppose β is attained by two sets $A, B \subseteq V$ such that A and B are uncomparable. Let $s \in A \setminus B$ and $t \in B \setminus A$. For $F := \delta^{in}(A) \cup \delta^{in}(B)$, we consider the graph $D' := D - F$. Since there are no edges entering the set B in D', the node s cannot reach the node t in D'. Since there are no edges entering the set A in D', the node t cannot reach the node s in D'. Thus, the minimum number of edges whose removal ensures that there exist two distinct nodes s and t such that s cannot reach t and t cannot reach s is at most $|F| = \beta$.

Suppose F is a smallest set of edges of D such that the graph $D' := D - F$ has two nodes s and t such that s cannot reach t and t cannot reach s. Let A be the set of nodes that can reach s in D' and B be the set of nodes that can reach t in D'. The sets A and B are uncomparable since $s \in A \setminus B$ and $t \in B \setminus A$. Moreover, $|\delta^{in}_{D'}(A) \cup \delta^{in}_{D'}(B)| = 0$. Thus, we have $\beta \leq \beta(A, B) \leq |F|$. □

Using the above formulation, and by recalling that $\sigma(A, B) = |\delta^{in}(A)| + |\delta^{in}(B)|$, we have the following natural relaxation of bicut:

Definition 2 For a directed graph $D = (V, E)$, let

$$\sigma := \min\{\sigma(A, B) : \ A \text{ and } B \text{ are uncomparable}\}.$$

A pair where the latter value is attained is called a *minimum uncomparable cut-pair*.

3.1 Overview of the approximation algorithm

In this section, we sketch the argument for a $(2 - \epsilon)$-approximation for some small enough ϵ. We observe that for every pair of subsets of nodes (A, B), we have

$$\beta(A, B) = \sigma(A, B) - d(V \setminus (A \cup B), A \cap B). \tag{3}$$

Therefore, $\beta(A, B) \leq \sigma(A, B) \leq 2\beta(A, B)$ for every pair of subsets of nodes (A, B) and hence $\beta \leq \sigma \leq 2\beta$. Furthermore, σ can be computed in polynomial-time (see Lemma 4), and the optimal solution is a $(2-\epsilon)$-approximation for BiCut if $\sigma \leq (2-\epsilon)\beta$. On the other hand, if $\sigma > (2-\epsilon)\beta$, then $d(V \setminus (A \cup B), A \cap B) > (1-\epsilon)\beta$ for every minimizer (A, B) of $\beta(A, B)$, thus providing a structural handle on optimal solutions. Our algorithm proceeds by making several further attempts at finding pairs (A', B') that could give a $(2-\epsilon)$-approximation. Each attempt that is unsuccessful at giving a $(2-\epsilon)$-approximation implies some structural property of the optimal solution. These structural properties are together exploited by the last attempt to succeed.

Let us fix an uncomparable minimizer (A, B) for $\beta(A, B)$. From (3), we note that if $A \cap B$ or $V \setminus (A \cup B)$ is empty, then $\sigma(A, B) = \beta(A, B)$ and hence, computing σ would have found the optimum bicut value already. So, we may assume that $A \cap B$ and $V \setminus (A \cup B)$ are non-empty. In the subsequent attempts, we guess nodes $x \in A \setminus B$,

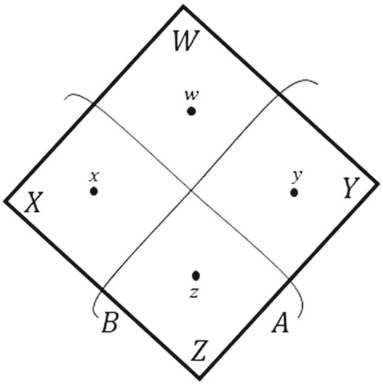

Fig. 3 The partitioning of the node set in the graph D. Here, (A, B) denotes the optimum bicut that is fixed

$y \in B \backslash A$, $w \in V \backslash (A \cup B)$, and $z \in A \cap B$. We use the notation $X := A \backslash B$, $Y := B \backslash A$, $W := V \backslash (A \cup B)$, and $Z := A \cap B$ (see Fig. 3).

We now observe that A is the sink-side of a $\{w, y\} \to \{x, z\}$-cut while B is the sink-side of a $\{w, x\} \to \{y, z\}$-cut. Our next attempt in the algorithm is to find (X', Y'), where X' is the sink-side of the unique inclusionwise minimal minimum $\{w, y\} \to \{x, z\}$-cut, and Y' is the sink-side of the unique inclusionwise minimal minimum $\{w, x\} \to \{y, z\}$-cut. The hope behind this attempt is that X' could recover A and Y' could recover B as these are feasible solutions to the respective problems and thus, they would together help us recover the optimal solution. Unfortunately, this favorable best-case scenario may not happen. Yet, owing to the feasibility of A and B for the respective problems, we may conclude that $\sigma(X', Y') \le \sigma(A, B) \le 2\beta(A, B) = 2\beta$.

Our subsequent attempts are more complex and proceed by modifying X' and Y'. We observe that Z is the sink-side of a $\{w, x, y\} \to \{z\}$-cut. So, our next attempt in the algorithm would be to find Z' as the sink-side of a minimum $\{w, x, y\} \to \{z\}$-cut and expand X' and Y' to include Z' thereby obtaining an uncomparable pair $(A' = X' \cup Z', B' = Y' \cup Z')$. Our hope is to find a Z' so that the resulting $\beta(A', B')$ is small. While finding Z', we prefer not to have many edges of $E[X'] \cup E[Y']$ in the new bicut (A', B'). This is because such edges enter only one among the two sets A' and B'. (We recall that if we have an uncomparable pair (A', B') with lot of edges from $V \backslash (A' \cup B')$ to $A' \cap B'$, then the value of $\beta(A', B')$ is going to be much less than $\sigma(A', B')$—e.g., see (3)—thus leading to a $(2 - \epsilon)$-approximation.) So, in order to avoid the edges of $E[X'] \cup E[Y']$ in the new bicut (A', B'), we make such edges more expensive by duplicating them before finding Z'. Let D_1 be the digraph obtained by duplicating the edges in $E[X'] \cup E[Y']$, and let Z' be the sink-side of the minimum $\{w, x, y\} \to \{z\}$-cut in D_1. We then show that the pair $(X' \cup Z', Y' \cup Z')$ is a $(2 - \epsilon)$-approximation unless $|\delta^{in}_{D_1}(Z)| > (2 - 3\epsilon)\beta$, thus giving us more structural handle on the optimum solution.

We next make an analogous attempt by shrinking X' and Y' instead of expanding. Let D_2 be the digraph obtained by duplicating the edges in $E[V \backslash X'] \cup E[V \backslash Y']$, and let W' be the source-side of the minimum $\{w\} \to \{x, y, z\}$-cut in D_2. We obtain that the pair $(X' \backslash W', Y' \backslash W')$ is a $(2 - \epsilon)$-approximation unless $|\delta^{out}_{D_2}(W)| > (2 - 3\epsilon)\beta$.

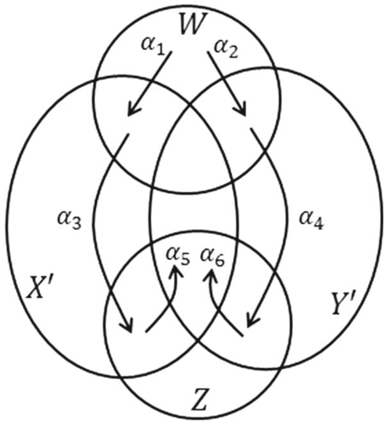

Fig. 4 The quantities $\alpha_1, \dots, \alpha_6$

Let $\alpha_1, \dots, \alpha_6$ be the number of edges in each position indicated in Fig. 4. If the attempts so far are unsuccessful, then we use the structural properties derived so far to arrive at the following:

1. All but $O(\epsilon\beta)$ edges in $\delta^{in}(X') \cup \delta^{in}(Y') \cup \delta^{out}(W) \cup \delta^{in}(Z)$ are as positioned in Fig. 4.
2. The quantities $\alpha_1, \alpha_3, \alpha_5$ are within $O(\epsilon\beta)$ of each other (see (34), (35), (36)) and so are $\alpha_2, \alpha_4, \alpha_6$.
3. Furthermore, $(1 - O(\epsilon))\beta = \alpha_3 + \alpha_4 \leq \beta$ (see Proposition 7).

Without loss of generality, we may assume that $\alpha_3 \geq \alpha_4$. Hence, by conclusion (3) from above, we have that $\alpha_3 \geq \beta/2 - O(\epsilon)\beta$.

Our final attempt in the algorithm to obtain a $(2 - \epsilon)$-approximate bicut is to expand Y' by including some nodes from $X' \backslash Y'$ and to shrink X' by excluding some nodes from $X' \backslash Y'$. We now explain the motivation behind this choice of expanding and shrinking. Consider $S := Y' \cup (X' \cap Z)$, which is obtained by expanding Y' by including some nodes from $X' \backslash Y'$ and $T := X' \backslash (X' \cap (W \backslash Y'))$, which is obtained by shrinking X' by excluding some nodes from $X' \backslash Y'$ (see Fig. 5). By definition, (S, T) is an uncomparable pair. We will now see that the bicut value of (S, T) is much smaller than 2β. Using conclusions (1) and (2) from above, we obtain that

$$\begin{aligned}
\beta(S, T) &= |\delta^{in}(Y' \cup (X' \cap Z)) \cup \delta^{in}(X' \backslash (X' \cap (W \backslash Y')))| \\
&= |\delta^{in}(Y')| - \alpha_5 + \alpha_3 + |\delta^{in}(X')| - \alpha_1 + O(\epsilon)\beta && (4)\\
&= \sigma(X', Y') - \alpha_1 - \alpha_5 + \alpha_3 + O(\epsilon)\beta && (5)\\
&\leq 2\beta - \alpha_3 + O(\epsilon)\beta && (6)\\
&\leq \frac{3}{2}\beta + O(\epsilon)\beta. && (7)
\end{aligned}$$

In the above, Eq. (4) is by using conclusion (1), Eq. (5) is by definition of σ, inequality (6) is by using conclusion (2) and $\sigma(X', Y') \leq \sigma(A, B) \leq 2\beta$, and inequality (7) is because $\alpha_3 \geq \beta/2 - O(\epsilon)\beta$.

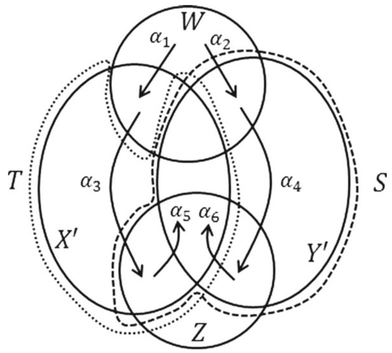

Fig. 5 The motivation behind the last attempt

Although (S, T) is a good approximation to the optimal bicut, we cannot obtain the sets S and T without the knowledge of W and Z (which, in turn, depend on the optimal bicut (A, B)). Instead, our algorithmic attempt is to expand Y' by including some nodes from $X' \backslash Y'$ and to shrink X' by excluding some nodes from $X' \backslash Y'$. In other words, our candidate is a pair $(B', Y' \cup A')$ for some $X' \cap Y' \subseteq A' \subsetneq B' \subseteq X'$ (we need the condition $A' \subsetneq B'$ because B' and $Y' \cup A'$ should be uncomparable) with minimum $\beta(B', Y' \cup A')$ value. When choosing A' and B', we ignore the edges whose contribution to the objective do not depend on A' and B'. Let H be the digraph obtained by removing the edges in $E[Y' \cup (V \backslash X')]$. Our aim is to minimize $|\delta_H^{in}(B') \cup \delta_H^{in}(Y' \cup A')|$. However, using conclusion (1), we note that this quantity differs from $|\delta_H^{in}(A') \cup \delta_H^{in}(B')|$ by $O(\epsilon\beta)$, so we may instead aim to minimize the latter.

The crucial observation now is that this latter minimization problem is an instance of $(s, *, t)$-Lin- 3- Cut. While we do not know how to solve $(s, *, t)$-Lin- 3- Cut optimally, we can obtain a 3/2-approximation in polynomial-time by Theorem 2. By the reformulation of $(s, *, t)$-Lin- 3- Cut in Lemma 1, we get a pair of subsets (A', B') for which $X' \cap Y' \subseteq A' \subsetneq B' \subseteq X'$ and which is a 3/2-approximation. In particular, $|\delta_H^{in}(A') \cup \delta_H^{in}(B')| \leq (3/2)|\delta_H^{in}((X' \cap (Z \cup Y')) \cup \delta_H^{in}(X' \backslash (W \setminus Y'))| \leq 3(\alpha_3 + O(\epsilon)\beta)/2$. Using this and proceeding similar to the calculations shown above to obtain the bound on $\beta(S, T)$ (i.e., 4, 5, 6, and 7), we derive that $\beta(B', Y' \cup A') \leq (7/4 + O(\epsilon))\beta$, concluding the proof.

3.2 Approximation algorithm and analysis

In this section we prove Theorem 1 by giving a polynomial-time $(2-\epsilon)$-approximation algorithm for BiCut for a constant $\epsilon > 0$. We will describe the algorithm, analyze its approximation factor to show that it is $(2 - \epsilon)$ for some constant $\epsilon > 0$ and compute the value of ϵ at the end of the analysis.

We begin by showing that a natural relaxation of β, namely σ, can be solved.

Lemma 4 *For a directed graph $D = (V, E)$, there exists a polynomial-time algorithm to find a minimum uncomparable cut-pair.*

Proof For fixed vertices a and b, there is an efficient algorithm to find A and B such that $a \in A \backslash B$ and $b \in B \backslash A$ and $\sigma(A, B)$ is minimized. Indeed, this is precisely

Approximation Algorithm for BiCut

Input: Directed graph $D = (V, E)$

1. Compute $(S, T) \leftarrow \arg\min\{\sigma(S, T) : S \text{ and } T \text{ are uncomparable}\}$ using Lemma 4 and set $\mu_1 \leftarrow \beta(S, T)$
2. Initialize $\mu_2 \leftarrow \infty$
3. For each ordered tuple of nodes (x, y, z, w)
   - (i) $X' \leftarrow$ sink-side of the unique inclusionwise minimal minimum $\{w, y\} \rightarrow \{x, z\}$-cut
   - (ii) $Y' \leftarrow$ sink-side of the unique inclusionwise minimal minimum $\{w, x\} \rightarrow \{y, z\}$-cut
   - (iii) $E_1 \leftarrow E[X'] \cup E[Y']$
   - (iv) $E_2 \leftarrow E[V \setminus X'] \cup E[V \setminus Y']$
   - (v) $D_1 \leftarrow D$ with the arcs in E_1 duplicated
   - (vi) $D_2 \leftarrow D$ with the arcs in E_2 duplicated
   - (vii) $Z' \leftarrow$ sink-side of minimum $\{w, x, y\} \rightarrow \{z\}$-cut in D_1
   - (viii) $W' \leftarrow$ source-side of minimum $\{w\} \rightarrow \{x, y, z\}$-cut in D_2
   - (ix) $H \leftarrow$ contract $X' \cap Y'$ to z', contract $V \setminus X'$ to w', remove all $w'z'$ arcs
   - (x) In H, find $\overline{w'}z'$-sets $A' \subsetneq B'$ such that $\beta(A', B')$ is at most $(3/2)\min\{\beta(A, B) : z' \in A \subsetneq B \subset V - \{w'\}\}$ using Lemma 2
   - (xi) $A_1 \leftarrow (A' \setminus \{z'\}) \cup (X' \cap Y')$ and $B_1 \leftarrow (B' \setminus \{z'\}) \cup (X' \cap Y')$
   - (xii) $\mu_2' \leftarrow \min\{\beta(X', Y'), \beta(X' \cup Z', Y' \cup Z'), \beta(X' \setminus W', Y' \setminus W'), \beta(B_1, Y' \cup A_1)\}$.
   - (xiii) If $\mu_2' < \mu_2$, update $\mu_2 \leftarrow \mu_2'$
4. Return $\mu \leftarrow \min\{\mu_1, \mu_2\}$.

Fig. 6 Approximation Algorithm for BiCut

finding the sink side of a min $a \rightarrow b$ cut and that of a min $b \rightarrow a$ cut. Trying all distinct pairs of nodes a and b and taking the minimum gives the desired result. □

We need the following definition.

Definition 3 If c is a capacity function on a directed graph D, then $d_c^{in}(U) = \sum_{e \in \delta^{in}(U)} c(e)$ is the sum of the capacities of incoming edges of U. Similarly, $d_c^{out}(U) = \sum_{e \in \delta^{out}(U)} c(e)$.

The rest of the section is devoted to presenting the approximation algorithm and its analysis (i.e., proving Theorem 1).

Proof of Theorem 1 The algorithm is summarized in Fig. 6. We first note that the algorithm indeed returns the bicut value of an uncomparable pair. The run-time of the algorithm being polynomial follows from Lemmas 2 and 4. In the rest of the proof, we analyze the approximation factor. We will show that the algorithm achieves a $(2 - \epsilon)$-approximation factor and compute ϵ at the end.

We note that both values μ_1, μ_2 computed by the algorithm are bicut values of uncomparable pairs of sets. Indeed, by definition, it is clear that μ_1 is the bicut value of an uncomparable pair of sets. The value μ_2' computed in Step 3(xii) is also the bicut value of an uncomparable pair of sets: A pair $(P, Q) \in \{(X', Y'), (X' \cup Z', Y' \cup Z'), (X' \backslash W', Y' \backslash W')\}$ is uncomparable since the vertex $x \in P \backslash Q$ while the vertex $y \in Q \backslash P$. The pair $(B_1, Y' \cup A_1)$ is uncomparable since the vertex $y \in (Y' \cup A_1) \backslash B_1$ while the set $B_1 \backslash (Y' \cup A_1)$ is non-empty since $A' \subsetneq B'$.

To analyze the approximation factor, let us fix a minimizer (A, B) for BiCut in the input graph $D = (V, E)$, i.e. fix an uncomparable pair (A, B) such that $\beta(A, B) = \beta$.

Let $X := A\backslash B$, $Y := B\backslash A$, $Z := A \cap B$, and $W := V\backslash(A \cup B)$ (see Fig. 3). With this notation, we have

$$\beta = d(W \cup Y, X) + d(W \cup X, Y) + d^{in}(Z) = d(Y, X \cup Z) + d(X, Y \cup Z) + d^{out}(W). \tag{8}$$

We may assume that both Z and W are non-empty, otherwise $\beta(A, B) = \sigma(A, B)$ and consequently, the algorithm finds the optimum since it returns a value $\mu \leq \mu_1 \leq \sigma(A, B) = \beta(A, B)$. Let $\epsilon > 0$ be a constant whose value will be determined later.

Lemma 5 *If one of the following is true, then $\sigma \leq (2 - \epsilon)\beta$:*

(i) $d(W, Z) \leq (1 - \epsilon)\beta$.
(ii) *For every $z \in Z$, there exists a subset U of nodes containing z but not all nodes of Z with $d^{in}(U) < (1 - \epsilon)\beta$.*
(iii) *For every $w \in W$, there exists a subset U of nodes not containing w but intersecting W with $d^{in}(U) < (1 - \epsilon)\beta$.*

Proof (i) If $d(W, Z) \leq (1 - \epsilon)\beta$, then $\sigma(A, B) = \beta(A, B) + d(W, Z) \leq (2 - \epsilon)\beta$. The pair (A, B) is uncomparable, and hence $\sigma \leq \sigma(A, B) \leq (2 - \epsilon)\beta$.
(ii) Suppose condition (ii) holds. This in particular, implies that $|Z| \geq 2$. Since condition (ii) holds, there exist sets M with in-degree less than $(1 - \epsilon)\beta$ such that $Z\backslash M \neq \emptyset$. Among all such sets, consider a set M with inclusionwise maximal intersection with Z. Let $z \in Z\backslash M$. There exists a set U containing z but not Z with $d^{in}(U) < (1 - \epsilon)\beta$. Because of the maximal intersection of M with Z, we have that $M \not\subseteq U$. Hence M and U are uncomparable and therefore $\sigma \leq \sigma(M, U) \leq (2 - 2\epsilon)\beta$.
(iii) An argument similar to the proof of (ii) shows that $\sigma \leq (2 - 2\epsilon)\beta$ if condition (iii) holds.

□

Our aim is to show that the algorithm in Fig. 6 achieves a $(2 - \epsilon)$-approximation. Therefore, we may assume for the rest of the proof that

$$\sigma > (2 - \epsilon)\beta \tag{9}$$

since otherwise, the algorithm returns $\mu \leq \mu_1 = \sigma \leq (2 - \epsilon)\beta$. By Lemma 5, we have

$$d(W, Z) \geq (1 - \epsilon)\beta. \tag{10}$$

We also have vertices $z \in Z$ and $w \in W$ violating conditions (ii) and (iii) of Lemma 5 respectively. Let us fix such vertices, i.e.,

(a) if $|Z| = 1$, then fix $z \in Z$, else if $|Z| \geq 2$, then fix $z \in Z$ such that $d^{in}(U) \geq (1 - \epsilon)\beta$ for all subsets U of nodes containing z but not all nodes of Z, and
(b) if $|W| = 1$, then fix $w \in W$, else if $|W| \geq 2$, then fix $w \in W$ such that $d^{in}(U) \geq (1 - \epsilon)\beta$ for all subsets U of nodes not containing w but intersecting W.

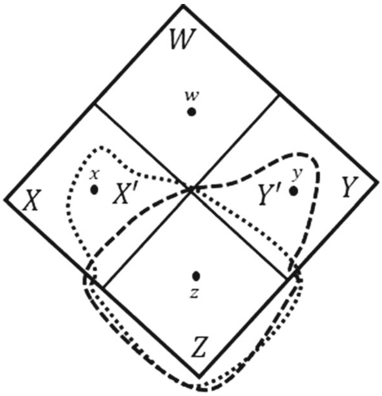

Fig. 7 The case where $W \cap (X' \cup Y') = \emptyset$ and $Z \subseteq X' \cap Y'$

Also let us fix an arbitrary choice of $x \in X$, $y \in Y$ (since A and B are uncomparable, we have that X and Y are non-empty and hence such an x and y can be chosen). Henceforth, we will consider the iteration of Step 3 in the algorithm for this choice of x, y, z, w.

We note that (X', Y') form an uncomparable pair since $x \in X' \backslash Y'$ and $y \in Y' \backslash X'$. If $\beta(X', Y') \leq (2-\epsilon)\beta$, then the algorithm returns $\mu \leq \mu_2 \leq (2-\epsilon)\beta$. Therefore, we may assume that

$$\beta(X', Y') \geq (2-\epsilon)\beta. \tag{11}$$

Also, we have $d^{in}(X') \leq d^{in}(X \cup Z)$ because X' is the sink-side of a min $\{w, y\} \to \{x, z\}$ cut. Since $d^{in}(X \cup Z) = d^{in}(A) \leq \beta$, we have that

$$d^{in}(X') \leq \beta. \tag{12}$$

Similarly,

$$d^{in}(Y') \leq d^{in}(Y \cup Z) \leq \beta. \tag{13}$$

Consequently,

$$\sigma(X', Y') \leq d^{in}(X') + d^{in}(Y') \leq 2\beta. \tag{14}$$

We consider four cases depending on the relations between W and $X' \cup Y'$, and between Z and $X' \cap Y'$.

Case 0. Suppose $W \cap (X' \cup Y') = \emptyset$ and $Z \subseteq X' \cap Y'$ (see Fig. 7). In this case $\delta^{in}(X')$ and $\delta^{in}(Y')$ both contain all edges counted in $d(W, Z)$. Hence $\beta(X', Y') \leq \sigma(X', Y') - d(W, Z) \leq (1+\epsilon)\beta$. The second inequality here is because $\sigma(X', Y') \leq 2\beta$ by (14) and $d(W, Z) \geq (1-\epsilon)\beta$ by (10). This shows that (X', Y') is a $(1+\epsilon)$-approximation. This completes the proof for this case.

For the remaining three cases, we will use the next lemma. Let c be the capacity function obtained by increasing the capacity of each edge in E_1 to 2, and let $\bar{c}$ be the capacity function obtained by increasing the capacity of each edge in E_2 to 2.

Recall that Z' is the sink-side of a minimum $\{w, x, y\} \to \{z\}$-cut in D_1, and W' is the source-side of minimum $\{w\} \to \{x, y, z\}$-cut in D_2.

Lemma 6 *If $d^{in}(X' \cap Z') \geq (1-\epsilon)\beta$ and $d^{in}(Y' \cap Z') \geq (1-\epsilon)\beta$, then $\beta(X' \cup Z', Y' \cup Z') \leq 2\epsilon\beta + d_c^{in}(Z)$. If $d^{out}(W' \backslash X') \geq (1-\epsilon)\beta$ and $d^{out}(W' \backslash Y') \geq (1-\epsilon)\beta$, then $\beta(X' \backslash W', Y' \setminus W') \leq 2\epsilon\beta + d_c^{out}(W')$.*

Proof If $d^{in}(X' \cap Z') \geq (1-\epsilon)\beta$, then $d^{in}(X') - d^{in}(X' \cap Z') \leq \epsilon\beta$. So

$$\begin{aligned} d^{in}(X' \cup Z') &= d^{in}(Z') + d^{in}(X') - d^{in}(X' \cap Z') \\ &\quad - d(X' \backslash Z', Z' \backslash X') - d(Z' \backslash X', X' \backslash Z') \\ &\leq d^{in}(Z') + \epsilon\beta - d(X' \backslash Z', Z' \backslash X') - d(Z' \backslash X', X' \backslash Z'). \end{aligned} \tag{15}$$

Hence, we have

$$d^{in}(X' \cup Z') \leq d^{in}(Z') + \epsilon\beta - d(X' \backslash Z', Z' \backslash X'). \tag{16}$$

Similarly,

$$d^{in}(Y' \cup Z') \leq d^{in}(Z') + \epsilon\beta - d(Y' \backslash Z', Z' \backslash Y'). \tag{17}$$

We need the following proposition.

Proposition 4

$$\begin{aligned} \beta(X' \cup Z', Y' \cup Z') \leq \sigma(X' \cup Z', Y' \cup Z') + d_c^{in}(Z') - 2d^{in}(Z') \\ + d(X' \backslash Z', Z' \backslash X') + d(Y' \backslash Z', Z' \backslash Y'). \end{aligned} \tag{18}$$

Proof By counting the edges entering Z', we have

1. $d_c^{in}(Z') = d^{in}(Z') + |\delta^{in}(Z') \cap E_1|$.
2. $d^{in}(Z') = d(V \backslash (X' \cup Y' \cup Z'), Z') + |\delta^{in}(Z') \cap E_1| + d(X' \backslash Z', Z' \backslash X') + d(Y' \backslash Z', Z' \backslash Y') - d((X' \cap Y') \backslash Z', Z' \backslash (X' \cup Y'))$.

The first equation can be rewritten as

$$d_c^{in}(Z') - 2d^{in}(Z') = -d^{in}(Z') + |\delta^{in}(Z') \cap E_1|.$$

Using this and the second equation, we get

$$\begin{aligned} & d_c^{in}(Z') - 2d^{in}(Z') + d(X' \backslash Z', Z' \backslash X') + d(Y' \backslash Z', Z' \backslash Y') \\ & \quad = -d(V \backslash (X' \cup Y' \cup Z'), Z') + d((X' \cap Y') \backslash Z', Z' \backslash (X' \cup Y')). \end{aligned}$$

Thus, the desired inequality (18) simplifies to

$$\begin{aligned} \beta(X' \cup Z', Y' \cup Z') \leq \sigma(X' \cup Z', Y' \cup Z') - d(V \backslash (X' \cup Y' \cup Z'), Z') \\ + d((X' \cap Y') \backslash Z', Z' \backslash (X' \cup Y')). \end{aligned}$$

To prove this inequality, we observe that the edges counted by $d(V \backslash (X' \cup Y' \cup Z'), Z')$ are counted twice in $\sigma(X' \cup Z', Y' \cup Z')$. Hence we have the desired relation (18). □

Using (18), (17) and (16) we get

$$\begin{aligned}\beta(X'\cup Z', Y'\cup Z') &\le d^{in}(X'\cup Z') + d^{in}(Y'\cup Z') + d_c^{in}(Z') - 2d^{in}(Z')\\ &\quad + d(X'\backslash Z', Z'\backslash X') + d(Y'\backslash Z', Z'\backslash Y')\\ &\le d^{in}(Z') + \epsilon\beta + d^{in}(Z') + \epsilon\beta + d_c^{in}(Z') - 2d^{in}(Z')\\ &= 2\epsilon\beta + d_c^{in}(Z')\\ &\le 2\epsilon\beta + d_c^{in}(Z).\end{aligned}$$

The last inequality above is because Z is a feasible solution for the minimization problem that obtains Z' and hence $d_c^{in}(Z') \le d_c^{in}(Z)$. This completes the proof of the first part of the lemma. The second part follows by a symmetric argument. $\square$

We are now ready to prove the three remaining cases.

Case 1. Suppose $W \cap (X' \cup Y') = \emptyset$ and $Z \not\subseteq X' \cap Y'$. Without loss of generality, let $Z \not\subseteq X'$. This implies that $|Z| \ge 2$. The set $X' \cap Z'$ contains z but not the whole Z, hence $d^{in}(X' \cap Z') \ge (1-\epsilon)\beta$ by (a).

We first consider the subcase where $d^{in}(Y' \cap Z') < (1-\epsilon)\beta$. By the choice of the vertex z and (a), this means that $Z \subseteq Y' \cap Z'$. In this case $Y' \cap Z'$ crosses X', because X' does not contain all vertices in Z, and $Y' \cap Z'$ does not contain x. Thus $(X', Y' \cap Z')$ is an uncomparable pair. Now we observe that $\sigma(X', Y' \cap Z') = d^{in}(X') + d^{in}(Y' \cap Z') \le (2-\epsilon)\beta$. Thus, $\sigma \le (2-\epsilon)\beta$, a contradiction to (9).

Next we consider the other subcase where $d^{in}(Y' \cap Z') \ge (1-\epsilon)\beta$. Then, by Lemma 6, we get

$$\beta(X'\cup Z', Y'\cup Z') \le 2\epsilon\beta + d_c^{in}(Z).$$

We are in the case where $(X'\cup Y')\cap W = \emptyset$, so $d_c^{in}(Z) \le d^{in}(Z) + d(X, Z) + d(Y, Z)$. We now note that $d^{in}(Z) + d(X, Z) + d(Y, Z) = 2d^{in}(Z) - d(W, Z) \le 2\beta - (1-\epsilon)\beta = (1+\epsilon)\beta$ since $d(W, Z) \ge (1-\epsilon)\beta$ and $d^{in}(Z) \le \beta$ which follows from (8). Hence we have $\beta(X'\cup Z', Y'\cup Z') \le (1+3\epsilon)\beta$. Since $(X'\cup Z', Y'\cup Z')$ is an uncomparable pair, we have that $\mu_2 \le (1+3\epsilon)\beta$.

Case 2. Suppose $W \cap (X' \cup Y') \ne \emptyset$ and $Z \subseteq X' \cap Y'$. This is similar to Case 1 by symmetry. The uncomparable pair of sets that are of interest in this case are $(X'\backslash W', Y'\backslash W')$.

Case 3. Suppose $W \cap (X' \cup Y') \ne \emptyset$ and $Z \not\subseteq X' \cap Y'$. Consequently, we have that $|Z|, |W| \ge 2$ and hence by (a) and (b), we have that $d^{in}(U) \ge (1-\epsilon)\beta$ for all subsets U of nodes containing z but not all nodes of Z and for all subsets U of nodes not containing w but intersecting W. For the rest of the proof, we may also assume that

$$\mu_2 > (2-\epsilon)\beta \tag{19}$$

for otherwise, the algorithm returns $\mu \le \mu_2 \le (2-\epsilon)\beta$ and we are done. With this, we have the following proposition.

Proposition 5

$$d_c^{in}(Z) \geq (2-3\epsilon)\beta, \text{ and} \tag{20}$$
$$d_{\overline{c}}^{out}(W) \geq (2-3\epsilon)\beta. \tag{21}$$

Proof We know that $Z \nsubseteq X' \cap Y'$. Without loss of generality, suppose $Z \nsubseteq X'$. The set $X' \cap Z'$ contains z but not the whole Z, hence $d^{in}(X' \cap Z') \geq (1-\epsilon)\beta$. By the same argument as in the first subcase of Case 1 (first paragraph), we may assume that $d^{in}(Y' \cap Z') \geq (1-\epsilon)\beta$ (otherwise, $\sigma \leq (2-\epsilon)\beta$, a contradiction to (9)). The inequality $\beta(X' \cup Z', Y' \cup Z') \leq 2\epsilon\beta + d_c^{in}(Z)$ holds using Lemma 6. If $d_c^{in}(Z) \leq (2-3\epsilon)\beta$, then these imply $\beta(X' \cup Z', Y' \cup Z') \leq (2-\epsilon)\beta$. Since $(X' \cup Z', Y' \cup Z')$ is an uncomparable pair, we would thus have $\mu_2 \leq (2-\epsilon)\beta$, a contradiction to (19). Similarly, if $d_{\overline{c}}^{out}(W) \leq (2-3\epsilon)\beta$, then we obtain $\mu_2 \leq \beta(X' \setminus W', Y' \setminus W') \leq (2-\epsilon)\beta$, a contradiction to (19). Thus, we have the conclusion. □

Let us define the following quantities (see Fig. 4):

1. $\alpha_1 := d(W \setminus (X' \cup Y'), W \cap (X' \setminus Y'))$,
2. $\alpha_2 := d(W \setminus (X' \cup Y'), W \cap (Y' \setminus X'))$,
3. $\alpha_3 := d(W \cap (X' \setminus Y'), Z \cap (X' \setminus Y'))$,
4. $\alpha_4 := d(W \cap (Y' \setminus X'), Z \cap (Y' \setminus X'))$,
5. $\alpha_5 := d(Z \cap (X' \setminus Y'), X' \cap Y' \cap Z)$, and
6. $\alpha_6 := d(Z \cap (Y' \setminus X'), X' \cap Y' \cap Z)$.

In Propositions 6, 7, 8, 9, 10 and 11, we show a sequence of inequalities involving these quantities.

Proposition 6 *Each of the values $d^{in}(X' \cap Y')$, $d^{in}(X' \cup Y')$, $d^{in}(X' \cap Z)$, $d^{in}(X' \cup Z)$, $d^{in}(Y' \cap Z)$, $d^{in}(Y' \cup Z)$ is at least $(1-\epsilon)\beta$ and is at most $(1+\epsilon)\beta$.*

Proof By submodularity,

$$d^{in}(X' \cap Y') + d^{in}(X' \cup Y') \leq d^{in}(X') + d^{in}(Y') \leq 2\beta.$$

We note that $d^{in}(X' \cap Y') \geq (1-\epsilon)\beta$ by the choice of the vertex z. This shows $d^{in}(X' \cup Y') \leq (1+\epsilon)\beta$. Similarly, $d^{in}(X' \cup Y') \geq (1-\epsilon)\beta$ by the choice of the vertex w, and hence $d^{in}(X' \cap Y') \leq (1+\epsilon)\beta$.

We argue the bounds for $d^{in}(X' \cup Z)$ and $d^{in}(X' \cap Z)$. The bounds for $d^{in}(Y' \cup Z)$ and $d^{in}(Y' \cap Z)$ follow using a similar proof strategy. By the assumption of Case 3, we have $Z \nsubseteq X' \cap Y'$. We will argue that $d^{in}(X' \cap Z) \geq (1-\epsilon)\beta$ by considering two sub-cases. Sub-case (i): Suppose $Z \nsubseteq X'$. Hence $X' \cap Z$ contains z but not all of Z. By the choice of the vertex z, we have $d^{in}(X' \cap Z) \geq (1-\epsilon)\beta$. Sub-case (ii): Suppose $Z \subsetneq X'$. Then, $d^{in}(X' \cap Z) = d^{in}(Z)$. We have $d^{in}(Z) \geq d(W, Z) \geq (1-\epsilon)\beta$ using (10) and hence, $d^{in}(X' \cap Z) \geq (1-\epsilon)\beta$.

By submodularity,

$$d^{in}(X' \cup Z) \leq d^{in}(X') + d^{in}(Z) - d^{in}(X' \cap Z) \leq 2\beta - (1-\epsilon)\beta = (1+\epsilon)\beta.$$

Next, we notice that $X' \cup Z$ and Y' are uncomparable, so $\sigma(X' \cup Z, Y') \geq (2-\epsilon)\beta$ by (9). However, we have

$$\sigma(X' \cup Z, Y') = d^{in}(X' \cup Z) + d^{in}(Y') \leq d^{in}(X' \cup Z) + \beta.$$

Hence, $d^{in}(X' \cup Z) \geq (1-\epsilon)\beta$. Using submodularity, we obtain $d^{in}(X' \cap Z) \leq (1+\epsilon)\beta$. □

Proposition 7 $(1-6\epsilon)\beta \leq \alpha_3 + \alpha_4 \leq \beta$.

Proof We have $\alpha_3+\alpha_4 \leq d(W, Z)$ by definition of α_3 and α_4. Moreover, $d(W, Z) \leq \beta$ by (8). Hence, we have the upper bound that $\alpha_3 + \alpha_4 \leq \beta$. We next show the lower bound. From (20), we recall that $(2-3\epsilon)\beta \leq d_c^{in}(Z) = d^{in}(Z) + |\delta^{in}(Z) \cap E_1|$ and from (21), we recall that $(2-3\epsilon)\beta \leq d_{\bar{c}}^{out}(W) = d^{out}(W) + |\delta^{out}(W) \cap E_2|$. Moreover, we have $d^{in}(Z) \leq \beta$ and $d^{out}(W) \leq \beta$ by (8).

Let C be the set of edges from W to Z, i.e. those counted by $d(W, Z)$. We next argue that $\alpha_3 + \alpha_4 = |C \cap E_1 \cap E_2|$. In order to show this equality, we show the inequality in both directions. For the first direction, we observe that every edge e that is counted by $\alpha_3 + \alpha_4$ is in C as well as E_1 as well as E_2, and hence $\alpha_3 + \alpha_4 \leq |C \cap E_1 \cap E_2|$. For the other direction, consider $e \in C \cap E_1 \cap E_2$. Then, e is counted either in α_3 or α_4 but not both. Hence, $|C \cap E_1 \cap E_2| \leq \alpha_3 + \alpha_4$.

Let $a := |\delta^{in}(Z) \backslash C|$ and $b := |\delta^{out}(W) \backslash C|$. Using (8), we have

$$\begin{aligned}
\beta &= d(W \cup Y, X) + d(W \cup X, Y) + d^{in}(Z) \\
&\geq d(W, X) + d(W, Y) + d^{in}(Z) \\
&= b + d^{in}(Z) \\
&= b + |C| + |\delta^{in}(Z) \backslash C| \\
&= b + |C| + a.
\end{aligned}$$

Thus, we have $|C| + a + b \leq \beta$. Furthermore, we have $|C \cap E_1| \geq |\delta^{in}(Z) \cap E_1| - a$ and $|C \cap E_2| \geq |\delta^{out}(W) \cap E_2| - b$. From all the above, we get the following sequence of inequalities that shows the lower bound:

$$\begin{aligned}
|C \cap E_1 \cap E_2| &\geq |C| - |C \backslash E_1| - |C \backslash E_2| \\
&= |C| - (|C| - |C \cap E_1|) - (|C| - |C \cap E_2|) \\
&= |C \cap E_1| + |C \cap E_2| - |C| \\
&\geq |\delta^{in}(Z) \cap E_1| - a + |\delta^{out}(W) \cap E_2| - b - |C| \\
&\geq (2-3\epsilon)\beta - d^{in}(Z) + (2-3\epsilon)\beta - d^{out}(W) - (a + b + |C|) \\
&\geq (4-6\epsilon)\beta - 3\beta \\
&= (1-6\epsilon)\beta.
\end{aligned}$$

□

Proposition 8 $(1-8\epsilon)\beta \leq \alpha_1+\alpha_2 \leq (1+\epsilon)\beta$ *and* $(1-8\epsilon)\beta \leq \alpha_5+\alpha_6 \leq (1+\epsilon)\beta$.

Proof We first show the upper bounds. We have $\alpha_1+\alpha_2 \le d^{in}(X' \cup Y')$ which is at most $(1+\epsilon)\beta$ by Proposition 6. Similarly, we have $\alpha_5+\alpha_6 \le d^{in}(X' \cap Y') \le (1+\epsilon)\beta$. We next show the lower bounds.

We first note that

$$\alpha_5+\alpha_6 \ge d^{in}(X' \cap Y' \cap Z) - |\delta^{in}(Z) \cap \delta^{in}(X' \cap Y' \cap Z)| \\ - d(V \backslash (X' \cup Y'), X' \cap Y' \cap Z). \tag{22}$$

We bound each of the terms in the RHS now. We observe that $X' \cap Y' \cap Z$ contains z but not all nodes in Z, hence

$$d^{in}(X' \cap Y' \cap Z) \ge (1-\epsilon)\beta. \tag{23}$$

Moreover, we have

$$\begin{aligned} |\delta^{in}(X') \cap \delta^{in}(Y')| &= d^{in}(X') + d^{in}(Y') - |\delta^{in}(X') \cup \delta^{in}(Y')| \\ &= \sigma(X', Y') - \beta(X', Y') \\ &\le 2\beta - (2-\epsilon)\beta \qquad \text{(Using (14) and (11))} \\ &= \epsilon\beta. \end{aligned}$$

Here, $|\delta^{in}(X') \cap \delta^{in}(Y')| \le \epsilon\beta$ implies that we have at most $\epsilon\beta$ edges entering $X' \cap Y' \cap Z$ from $V \backslash (X' \cup Y')$. Thus, we have

$$d(V \backslash (X' \cup Y'), X' \cap Y' \cap Z) \le \epsilon\beta. \tag{24}$$

We further have $|\delta^{in}(Z) \cap \delta^{in}(X' \cap Y' \cap Z)| \le d^{in}(Z) - \alpha_3 - \alpha_4$. Using Proposition 7, we obtain that

$$|\delta^{in}(Z) \cap \delta^{in}(X' \cap Y' \cap Z)| \le d^{in}(Z) - \alpha_3 - \alpha_4 \le 6\epsilon\beta. \tag{25}$$

Substituting the bounds from (23), (24), and (25) in (22), we obtain that $\alpha_5+\alpha_6 \ge (1-\epsilon)\beta - 6\epsilon\beta - \epsilon\beta = (1-8\epsilon)\beta$.

We proceed by a similar argument now to show the lower bound for $\alpha_1+\alpha_2$. We note that

$$\alpha_1+\alpha_2 \ge d^{out}(W \backslash (X' \cup Y')) - |\delta^{out}(W) \cap \delta^{out}(W \backslash (X' \cup Y'))| \\ - d(W \backslash (X' \cup Y'), X' \cap Y'). \tag{26}$$

We bound each of the terms in the RHS now. We observe that $d^{out}(W \backslash (X' \cup Y')) = d^{in}(V \backslash (W \setminus (X' \cup Y')))$. The set $V \backslash (W \setminus (X' \cup Y'))$ does not contain w but intersects W, hence

$$d^{out}(W \backslash (X' \cup Y')) = d^{in}(V \backslash (W \setminus (X' \cup Y'))) \ge (1-\epsilon)\beta. \tag{27}$$

Moreover, by $|\delta^{in}(X') \cap \delta^{in}(Y')| \leq \epsilon\beta$ derived as above, we have at most $\epsilon\beta$ edges entering $X' \cap Y'$ from $W \backslash (X' \cup Y')$. Thus, we have

$$d(W \backslash (X' \cup Y'), X' \cap Y') \leq \epsilon\beta. \tag{28}$$

We further have $|\delta^{out}(W) \cap \delta^{out}(W \backslash (X' \cup Y'))| \leq d^{out}(W) - \alpha_3 - \alpha_4$. Using Proposition 7, we obtain that

$$|\delta^{out}(W) \cap \delta^{out}(W \backslash (X' \cup Y'))| \leq d^{out}(W) - \alpha_3 - \alpha_4 \leq 6\epsilon\beta. \tag{29}$$

Substituting the bounds from (27), (28), and (29) in (26), we obtain that $\alpha_1 + \alpha_2 \geq (1-\epsilon)\beta - 6\epsilon\beta - \epsilon\beta = (1-8\epsilon)\beta$. □

Proposition 9 $(1-16\epsilon)\beta \leq \alpha_1 + \alpha_6 \leq \beta$ *and* $(1-16\epsilon)\beta \leq \alpha_2 + \alpha_5 \leq \beta$.

Proof The upper bounds follow by $\alpha_1 + \alpha_6 \leq d^{in}(X') \leq \beta$ and $\alpha_2 + \alpha_5 \leq d^{in}(Y') \leq \beta$. On the other hand, combining the two inequalities in Proposition 8 gives $(2 - 16\epsilon)\beta \leq \alpha_1 + \alpha_2 + \alpha_5 + \alpha_6$. Now using the upper bound $\alpha_2 + \alpha_5 \leq \beta$ gives $(1-16\epsilon)\beta \leq \alpha_1 + \alpha_6$. Similarly, we obtain $(1-16\epsilon)\beta \leq \alpha_2 + \alpha_5$. □

Proposition 10 $(1-23\epsilon)\beta \leq \alpha_3 + \alpha_6 \leq (1+\epsilon)\beta$.

Proof Consider the set $M := X' \cap Z$. We note that $\alpha_3 + \alpha_6 \leq d^{in}(X' \cap Z)$. By Proposition 6, we have $d^{in}(M) \leq (1+\epsilon)\beta$, which gives the upper bound. We now show the lower bound.

By Proposition 6, we have

$$(1-\epsilon)\beta \leq d^{in}(M). \tag{30}$$

Next we have

$$d^{in}(M) = \alpha_6 + d((Z \backslash X') \cap Y', M \backslash Y') + d(Z \setminus (X' \cup Y'), M) + d(V \backslash Z, M). \tag{31}$$

Also,

$$\alpha_1 + \alpha_6 + d((Z \backslash X') \cap Y', M \backslash Y') + d(Z \setminus (X' \cup Y'), M) \leq d^{in}(X') \leq \beta.$$

Using Proposition 9, we thus obtain

$$d((Z \backslash X') \cap Y', M \backslash Y') + d(Z \setminus (X' \cup Y'), M) \leq 16\epsilon\beta. \tag{32}$$

We next note that $\alpha_4 + d(V \backslash Z, M) \leq d^{in}(Z) \leq \beta$. Using Proposition 7, we have $(1-6\epsilon)\beta - \alpha_3 \leq \alpha_4$. We thus obtain

$$d(V \backslash Z, M) \leq 6\epsilon\beta + \alpha_3. \tag{33}$$

Using (30), (31), (32), and (33), we obtain

$$\begin{aligned}(1-\epsilon)\beta &\le d^{in}(M)\\ &= \alpha_6 + d((Z\backslash X')\cap Y', M\backslash Y')\\ &\quad + d(Z\backslash(X'\cup Y'), M) + d(V\backslash Z, M)\\ &\le \alpha_6 + 16\epsilon\beta + \alpha_3 + 6\epsilon\beta\\ &= \alpha_3 + \alpha_6 + 22\epsilon\beta.\end{aligned}$$

Rewriting the final inequality gives $(1-23\epsilon)\beta \le \alpha_3 + \alpha_6$. □

Proposition 11 $\alpha_1 + \alpha_5 \ge 2\alpha_3 - 51\epsilon\beta$.

Proof The above propositions give us a chain of relations:

$$\begin{aligned}(1-16\epsilon)\beta - \alpha_6 \le \alpha_1 &\le \beta - \alpha_6,\\ (1-8\epsilon)\beta - \alpha_1 \le \alpha_2 &\le (1+\epsilon)\beta - \alpha_1,\\ (1-16\epsilon)\beta - \alpha_2 \le \alpha_5 &\le \beta - \alpha_2,\\ (1-23\epsilon)\beta - \alpha_3 \le \alpha_6 &\le (1+\epsilon)\beta - \alpha_3.\end{aligned}$$

By substitution, we get

$$\alpha_3 - 17\epsilon\beta \le \alpha_1 \le \alpha_3 + 23\epsilon\beta, \tag{34}$$

$$\alpha_1 - 17\epsilon\beta \le \alpha_5 \le \alpha_1 + 8\epsilon\beta. \tag{35}$$

By substituting again, we get

$$\alpha_3 - 34\epsilon\beta \le \alpha_5 \le \alpha_3 + 31\epsilon\beta. \tag{36}$$

Using (34) and (36), we obtain $\alpha_1 + \alpha_5 \ge 2\alpha_3 - 51\epsilon\beta$. □

Without loss of generality, we may assume that $\alpha_3 \ge (\alpha_3 + \alpha_4)/2$, since if not, there is another iteration of the algorithm where x and y are switched and thus, the unique choices of X' and Y' also get switched. Therefore, by Proposition 7, we have

$$\alpha_3 \ge (1/2 - 3\epsilon)\beta. \tag{37}$$

Let H be the directed graph obtained in Step 3(ix) of the algorithm, i.e., by contracting $X' \cap Y'$ to a node z', contracting $V\backslash X'$ to a node w', and removing all $w'z'$ arcs. Let

$$\begin{aligned}A_0 &:= ((X'\cap Z)\backslash Y') \cup \{z'\} \text{ and}\\ B_0 &:= (X'\backslash(W\cup Y')) \cup \{z'\}.\end{aligned}$$

We note that (A_0, B_0) is a feasible solution for Step 3(x) of the algorithm: both sets contain z' and do not contain w' by definition and moreover $A_0 \subsetneq B_0$ since the vertex x is in $B_0 \backslash A_0$. The following proposition shows an upper bound on the value of $\beta(A_0, B_0)$ in H:

Proposition 12

$$|\delta_H^{in}(A_0) \cup \delta_H^{in}(B_0)| \leq \alpha_3 + 39\epsilon\beta. \tag{38}$$

Proof For notational convenience, we will use $d(P, Q)$ to denote $d_D(P, Q)$ for two subsets $P, Q \subseteq V$. We have that

$$\begin{aligned} |\delta_H^{in}(A_0)| &= |\delta_H(V(H)\backslash A_0, (X' \cap Z)\backslash Y')| + |\delta_H(X'\backslash(Y' \cup Z), z')| \\ &= d(V\backslash((X' \cap Z) \cup (X' \cap Y')), (X' \cap Z)\backslash Y') \\ &\quad + d(X'\backslash(Y' \cup Z), X' \cap Y'), \end{aligned} \tag{39}$$

and

$$\begin{aligned} |\delta_H^{in}(B_0)\backslash\delta_H^{in}(A_0)| &= d(V\backslash X', X'\backslash(Y' \cup W \cup Z)) \\ &\quad + d((X' \cap W)\backslash Y', X'\backslash(Y' \cup W \cup Z)). \end{aligned} \tag{40}$$

We would like to bound the sum of the above four terms. We further decompose the first term as follows:

$$\begin{aligned} &d(V\backslash((X' \cap Z) \cup (X' \cap Y')), (X' \cap Z)\backslash Y') \\ &\quad = d(Z\backslash X', (X' \cap Z)\backslash Y') \\ &\quad\quad + d(V\backslash((X' \cap Z) \cup (X' \cap Y') \cup Z), (X' \cap Z)\backslash Y'). \end{aligned} \tag{41}$$

We note that $d(V\backslash((X' \cap Z) \cup (X' \cap Y') \cup Z), (X' \cap Z)\backslash Y')$ counts a subset of the edges entering Z. Since we have $d(W, Z) \geq (1 - \epsilon)\beta$, while $d^{in}(Z) \leq \beta$, it follows that all but $\epsilon\beta$ edges entering Z are from W. Hence,

$$\begin{aligned} &d(V\backslash((X' \cap Z) \cup (X' \cap Y') \cup Z), (X' \cap Z)\backslash Y') \\ &\quad \leq d((V\backslash((X' \cap Z) \cup (X' \cap Y') \cup Z)) \cap W, (X' \cap Z)\backslash Y') + \epsilon\beta \\ &\quad = d(W\backslash(X' \cap Y'), (X' \cap Z)\backslash Y') + \epsilon\beta. \end{aligned} \tag{42}$$

The last equation above is because the set $(V\backslash((X' \cap Z) \cup (X' \cap Y') \cup Z)) \cap W$ is precisely $W\backslash(X' \cap Y')$. Hence, using (39), (40), (41), and (42), we have that $|\delta_H^{in}(A_0) \cup \delta_H^{in}(B_0)| - \epsilon\beta$ is at most

$$\begin{aligned} &d(Z\backslash X', (X' \cap Z)\backslash Y') + d(W \setminus (X' \cap Y'), (X' \cap Z)\backslash Y') \\ &\quad + d(X'\backslash(Y' \cup Z), X' \cap Y') \\ &\quad + d(V\backslash X', X'\backslash(Y' \cup W \cup Z)) \\ &\quad + d((X' \cap W)\backslash Y', X'\backslash(Y' \cup W \cup Z)). \end{aligned} \tag{43}$$

We now bound this sum by suitably grouping the terms.

1. The first term $d(Z\backslash X', (X'\cap Z)\backslash Y')$ and the fourth term $d(V\backslash X', X'\backslash(Y'\cup W\cup Z))$ together count a subset of the edges entering X'. We have

$$d(Z\backslash X', (X' \cap Z)\backslash Y') + d(V \setminus X', X'\backslash(Y' \cup W \cup Z)) \\ + \alpha_1 + \alpha_6 \leq d^{in}(X') \leq \beta.$$

   Using $\alpha_1 + \alpha_6 \geq (1 - 16\epsilon)\beta$ from Proposition 9, we obtain

$$d(Z\backslash X', (X' \cap Z)\backslash Y') + d(V \setminus X', X'\backslash(Y' \cup W \cup Z)) \leq 16\epsilon\beta.$$

2. The third term $d(X'\backslash(Y' \cup Z), X' \cap Y')$ counts a subset of the edges entering Y'. We have

$$d(X'\backslash(Y' \cup Z), X' \cap Y') + \alpha_2 + \alpha_5 \leq d^{in}(Y') \leq \beta.$$

   Using $\alpha_2 + \alpha_5 \geq (1 - 16\epsilon)\beta$ from Proposition 9, we obtain

$$d(X'\backslash(Y' \cup Z), X' \cap Y') \leq 16\epsilon\beta.$$

3. The second term $d(W\backslash(X' \cap Y'), (X' \cap Z)\backslash Y')$ and the fifth term $d((X' \cap W)\backslash Y', X'\backslash(Y' \cup W \cup Z))$ together count a subset of the edges leaving W. We have

$$d(W\backslash(X' \cap Y')), (X' \cap Z)\backslash Y') \\ + d((X' \cap W)\backslash Y', X'\backslash(Y' \cup W \cup Z)) + \alpha_4 \leq d^{out}(W) \leq \beta.$$

   Using $\alpha_3 + \alpha_4 \geq (1 - 6\epsilon)\beta$ from Proposition 7, we obtain

$$d(W\backslash(X' \cap Y'), (X' \cap Z)\backslash Y') \\ + d((X' \cap W)\backslash Y', X'\backslash(Y' \cup W \cup Z)) \leq 6\epsilon\beta + \alpha_3.$$

Thus, the total contribution of the five terms is at most $38\epsilon\beta + \alpha_3$, thus proving the proposition. $\square$

Using Proposition 12, Step 3(x) of the algorithm finds $\overline{w'}z'$-sets $A' \subsetneq B'$ such that

$$\begin{aligned} |\delta_H^{in}(A') \cup \delta_H^{in}(B')| &\leq \frac{3}{2}|\delta_H^{in}(A_0) \cup \delta_H^{in}(B_0)| \\ &\leq \frac{3}{2}(\alpha_3 + 39\epsilon\beta) = \frac{3}{2}\alpha_3 + \frac{117}{2}\epsilon\beta. \end{aligned} \tag{44}$$

Let $A_1 := (A'\backslash\{z'\}) \cup (X' \cap Y')$ and $B_1 := (B'\backslash\{z'\}) \cup (X' \cap Y')$ as obtained in Step 3(xi) of the algorithm, i.e., A_1 and B_1 are the corresponding sets in V obtained by replacing z' by $X' \cap Y'$ (see Fig. 8). Now we consider the pair $(B_1, Y' \cup A_1)$. Since $A' \subsetneq B'$, we have that $B_1\backslash(Y' \cup A_1) \neq \emptyset$. Moreover, the vertex $y \in (Y' \cup A_1)\backslash B_1$ and hence $(Y' \cup A_1)\backslash B_1 \neq \emptyset$. Hence, $(B_1, Y' \cup A_1)$ is an uncomparable pair. We next

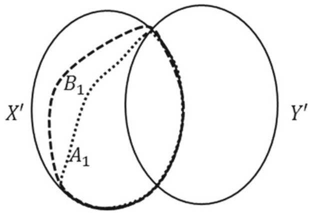

Fig. 8 The sets A_1 and B_1 are completely contained in X'

compute the bicut value $\beta(B_1, Y' \cup A_1)$ of this pair in the original directed graph. The next proposition will help in bounding the bicut value.

Proposition 13

$$\beta(B_1, Y' \cup A_1) + \alpha_5 + \alpha_1 \leq \sigma(X', Y') + |\delta_H^{in}(A') \cup \delta_H^{in}(B')|.$$

Proof The proposition follows by counting the edges on the left hand side. We use a figure to easily visualize the counting argument. We recall that $X' \cap Y' \subseteq A_1 \subseteq B_1 \subseteq X'$.

We use Fig. 9. Each arrow represents that all edges from the set of nodes in the rectangle containing its tail to the set of nodes in the rectangle containing its head are counted in the left hand side of Proposition 13. In particular, edges corresponding to $\delta^{in}(B_1)$ are marked as thin continuous arrows and $\delta^{in}(Y' \cup A_1) \backslash \delta^{in}(B_1)$ are marked as thin dotted arrows. Edges counted by α_1, i.e., corresponding to $\delta(W \backslash (X' \cup Y'), W \cap (X' \backslash Y'))$, are marked as thick $\rightarrow W$ arrows to indicate that the head v of the edges are in $W \cap S$ where S is the set of nodes in the rectangle containing the head. Edges counted by α_5, i.e., corresponding to $\delta(Z \cap (X' \backslash Y'), X' \cap Y' \cap Z)$, are marked as thick dotted $Z \rightarrow Z$ arrows to indicate that the tail u and the head v of the edges are in $Z \cap S_1$ and $Z \cap S_2$ respectively where S_1 and S_2 are the set of nodes in the rectangles containing the tail and head respectively.

We note that the edges that are counted twice in the left hand side are exactly the ones in the following four sets:

1. $\delta(Z \cap (X' \backslash B_1), Z \cap (X' \cap Y'))$ since these edges are also contained in $\delta(X' \backslash B_1, X' \cap Y')$,
2. $\delta(Z \cap (B_1 \backslash A_1), Z \cap (X' \cap Y'))$ since these edges are also contained in $\delta(B_1 \backslash A_1, X' \cap Y')$,
3. $\delta(W \backslash (X' \cup Y'), W \cap (B_1 \backslash A_1))$ since these edges are also contained in $\delta(W \backslash (X' \cup Y'), B_1 \backslash A_1))$, and
4. $\delta(W \backslash (X' \cup Y'), W \cap (A_1 \backslash Y'))$ since these edges are also contained in $\delta(W \backslash (X' \cup Y'), A_1 \backslash Y')$.

In order to prove the proposition, we need to show that every edge in the left hand side that is counted exactly once is counted in the right hand side and moreover, those edges that are counted twice in the left hand side are counted by two different terms in the right hand side. In order to show this, we mark the tail of the arrows as follows: $\square$ indicates that the edge is counted in $\delta^{in}(X')$, $\diamond$ indicates that the edge is counted in

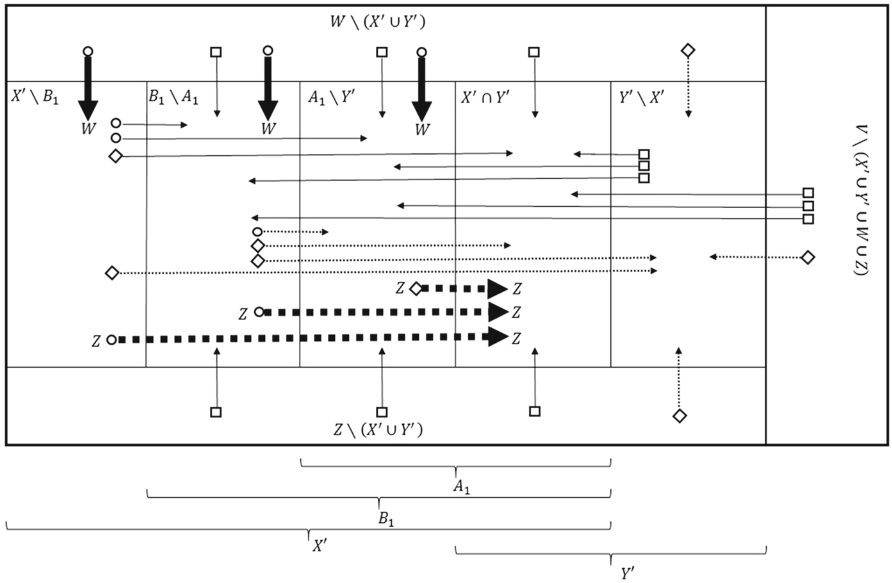

Fig. 9 Proof of Proposition 13

$\delta^{in}(Y')$ and $\circ$ indicates that the edge is counted in $\delta_H^{in}(A')\cup\delta_H^{in}(B')$. We note that the edges that are counted twice have different tail marks as follows:

1. the tail marks of $\delta(Z\cap(X'\backslash B_1), Z\cap(X'\cap Y'))$ and $\delta(X'\backslash B_1, X'\cap Y')$ are different,
2. the tail marks of $\delta(Z\cap(B_1\backslash A_1), Z\cap(X'\cap Y'))$ and $\delta(B_1\backslash A_1, X'\cap Y')$ are different,
3. the tail marks of $\delta(W\backslash(X'\cup Y'), W\cap(B_1\backslash A_1))$ and $\delta(W\backslash(X'\cup Y'), B_1\backslash A_1))$ are different, and
4. the tail marks of $\delta(W\backslash(X'\cup Y'), W\cap(A_1\backslash Y'))$ and $\delta(W\backslash(X'\cup Y'), A_1\backslash Y')$ are different.

Thus, the left hand side is at most the right hand side. $\square$

Using Proposition 13 and inequality (44), we get

$$\beta(B_1, Y'\cup A_1)\le\sigma(X',Y')+|\delta_H^{in}(A')\cup\delta_H^{in}(B')|-\alpha_5-\alpha_1$$
$$\le 2\beta+\frac{3}{2}\alpha_3+\frac{117}{2}\epsilon\beta-\alpha_5-\alpha_1.$$

Next, using Proposition 11, we get

$$\beta(B_1, Y'\cup A_1)\le 2\beta+\frac{3}{2}\alpha_3+\frac{117}{2}\epsilon\beta-(2\alpha_3-51\epsilon\beta)=2\beta-\frac{1}{2}\alpha_3+\frac{219}{2}\epsilon\beta.$$

Finally, we recall that $\alpha_3 \geq (1/2 - 3\epsilon)\beta$ from (37) and hence,

$$\begin{aligned}\beta(B_1, Y' \cup A_1) &\leq \left(2 + \frac{219}{2}\epsilon\right)\beta - \frac{1}{2}\left(\frac{1}{2} - 3\epsilon\right)\beta \\ &= \left(\frac{7}{4} + 111\epsilon\right)\beta.\end{aligned}$$

Based on all the cases analyzed above, the approximation factor is at most

$$\max\left\{1+\epsilon, 1+3\epsilon, 2-\epsilon, \frac{7}{4} + 111\epsilon\right\} = \max\left\{2-\epsilon, \frac{7}{4} + 111\epsilon\right\}.$$

In order to minimize the factor, we set $\epsilon = 1/448$ to get the desired approximation factor, thus concluding the proof of Theorem 1. $\square$

4 Conclusion and open problems

In this work, we considered BiCut which is a natural extension of the global minimum cut problem from undirected graphs to directed graphs. While its fixed-terminal variant is well-understood both in terms of complexity and approximability, BiCut has hardly been investigated in the literature. In this work, we gave a $(2-1/448)$-approximation for BiCut, thus exhibiting a dichotomous behaviour in the approximability between the global and the fixed-terminal variants. Intriguingly, the complexity of BiCut remains elusive and is an open problem that merits thorough investigation.

Our approximation algorithm for BiCut needs to solve $(s, *, t)$-Lin- 3- Cut as an intermediate subproblem. In this work, we gave a 3/2-approximation for $(s, *, t)$-Lin- 3- Cut and used this factor in the analysis of the approximation factor for BiCut. If $(s, *, t)$-Lin- 3- Cut is solvable in polynomial-time or if its approximability is better than 3/2, then the approximability of BiCut would also improve using our techniques. Hence, it would be interesting to resolve the complexity of $(s, *, t)$-Lin- 3- Cut as well.

Acknowledgements The authors would like to thank the anonymous reviewers for their helpful comments in improving the presentation of this work.

References

1. Angelidakis, H., Makarychev, Y., Manurangsi, P.: An improved integrality gap for the Călinescu–Karloff–Rabani relaxation for multiway cut. In: Proceedings of the 19th International Conference on Integer Programming and Combinatorial Optimization, IPCO'17, pp. 40–50 (2017)
2. Chekuri, C., Madan, V.: Approximating multicut and the demand graph. In: Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA'17, pp. 855–874 (2017)
3. Cheung, K., Cunningham, W., Tang, L.: Optimal 3-terminal cuts and linear programming. Math. Program. **106**(1), 1–23 (2006)
4. Călinescu, G., Karloff, H., Rabani, Y.: An improved approximation algorithm for multiway cut. J. Comput. Syst. Sci. **60**(3), 564–574 (2000)

5. Dahlhaus, E., Johnson, D., Papadimitriou, C., Seymour, P., Yannakakis, M.: The complexity of multiterminal cuts. SIAM J. Comput. **23**(4), 864–894 (1994)
6. Erbacher, R., Jaeger, T., Talele, N., Teutsch, J.: Directed multicut with linearly ordered terminals. Preprint: arXiv:1407.7498 (2014)
7. Garg, N., Vazirani, V., Yannakakis, M.: Multiway cuts in directed and node weighted graphs. In: Proceedings of the 20th International Colloquium on Automata, Languages and Programming, ICALP'94, pp. 487–498 (1994)
8. Goldschmidt, O., Hochbaum, D.: A polynomial algorithm for the k-cut problem for fixed k. Math. Oper. Res. **19**(1), 24–37 (1994)
9. Karger, D., Klein, P., Stein, C., Thorup, M., Young, N.: Rounding algorithms for a geometric embedding of minimum multiway cut. Math. Oper. Res. **29**(3), 436–461 (2004)
10. Karger, D., Motwani, R.: Derandomization through approximation. In: Proceedings of the 26th Annual ACM Symposium on Theory of Computing, STOC'94, pp. 497–506 (1994)
11. Karger, D., Stein, C.: A new approach to the minimum cut problem. J. ACM **43**(4), 601–640 (1996)
12. Khot, S.: On the power of unique 2-prover 1-round games. In: Proceedings of the 34th Annual ACM Symposium on Theory of Computing, STOC'02, pp. 767–775 (2002)
13. Lee, E.: Improved hardness for cut, interdiction, and firefighter problems. In: Proceedings of the 44th International Colloquium on Automata, Languages, and Programming, ICALP, pp. 92:1–92:14 (2017)
14. Manokaran, R., Naor, J., Raghavendra, P., Schwartz, R.: SDP gaps and UGC hardness for multiway cut, 0-extension, and metric labeling. In: Proceedings of the 40th Annual ACM Symposium on Theory of Computing, STOC'08, pp. 11–20 (2008)
15. Queyranne, M.: On optimum k-way partitions with submodular costs and minimum part-size constraints. Talk Slides, https://smartech.gatech.edu/bitstream/handle/1853/43309/Queyranne.pdf (2012). Accessed 26 Mar 2018
16. Saran, H., Vazirani, V.: Finding k cuts within twice the optimal. SIAM J. Comput. **24**(1), 101–108 (1995)
17. Sharma, A., Vondrák, J.: Multiway cut, pairwise realizable distributions, and descending thresholds. In: Proceedings of the 46th Annual ACM Symposium on Theory of Computing, STOC'14, pp. 724–733 (2014)
18. Thorup, M.: Minimum k-way cuts via deterministic greedy tree packing. In: Proceedings of the 40th Annual ACM Symposium on Theory of Computing, STOC'08, pp. 159–166 (2008)
19. Vazirani, V., Yannakakis, M.: Suboptimal cuts: their enumeration, weight and number (extended abstract). In: Proceedings of the 19th International Colloquium on Automata, Languages and Programming, ICALP'92, pp. 366–377 (1992)