Egerváry Research Group
on Combinatorial Optimization

Technical reportS

TR-2017-10. Published by the Egerváry Research Group, Pázmány P. sétány 1/C, H–1117, Budapest, Hungary. Web site: www.cs.elte.hu/egres . ISSN 1587–4451.

A tight $\sqrt{2}$-approximation for Linear 3-Cut

Kristóf Bérczi, Karthekeyan Chandrasekaran,
Tamás Király, and Vivek Madan

2017

A tight $\sqrt{2}$-approximation for Linear 3-Cut

Kristóf Bérczi⋆, Karthekeyan Chandrasekaran⋆⋆, Tamás Király⋆, and Vivek Madan⋆⋆

Abstract

We investigate the approximability of the linear 3-cut problem in directed graphs, which is the simplest unsolved case of the linear k-cut problem. The input here is a directed graph $D = (V, E)$ with node weights and three specified terminal nodes $s, r, t \in V$, and the goal is to find a minimum weight subset of non-terminal nodes whose removal ensures that s cannot reach r and t, and r cannot reach t. The problem is approximation-equivalent to the problem of blocking rooted in- and out-arborescences, and it also has applications in network coding and security.

The approximability of linear 3-cut has been wide open until now: the best known lower bound under the Unique Games Conjecture (UGC) was 4/3, while the best known upper bound was 2 using a trivial algorithm. In this work we completely close this gap: we present a $\sqrt{2}$-approximation algorithm and show that this factor is tight assuming UGC. Our contributions are twofold: (1) we analyze a natural two-step deterministic rounding scheme through the lens of a single-step randomized rounding scheme with *non-trivial* distributions, and (2) we construct integrality gap instances that meet the upper bound of $\sqrt{2}$. Our gap instances can be viewed as a weighted graph sequence converging to a "graph limit structure".

1 Introduction

We investigate the complexity of the linear 3-cut problem in directed graphs. In the node weighted variant, abbreviated (s, r, t)-Node-Lin-3-Cut, the input is a directed graph $D = (V, E)$ with specified nodes $s, r, t \in V$ and node weights $w \in \mathbb{R}_+^{V \setminus \{r,s,t\}}$, and the goal is to find a minimum weight node set $U \subseteq V \setminus \{r, s, t\}$ such that $D[V - U]$ has no path from s to t, from s to r and from r to t. The edge-weighted variant, (s, r, t)-Edge-Lin-3-Cut, has edge weights $w \in \mathbb{R}_+^E$, and the goal is to find a minimum

⋆MTA-ELTE Egerváry Research Group, Eötvös Loránd University, Budapest, email: {berkri, tkiraly}@cs.elte.hu. 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.

⋆⋆University of Illinois, Urbana-Champaign, email: {karthe,vmadan2}@illinois.edu. Vivek is supported by the NSF grant CCF-1319376.

weight edge set $F \subseteq E$ such that $D - F$ has no path from s to t, from s to r and from r to t. These two variants are equivalent by standard transformations.

We emphasize that the unreachability requirements are determined by an ordering of the terminal nodes s, r, and t, and this is the origin for the terminology *linear* 3-*cut* [6]. While it might appear to be a fabricated/secondary cut problem, there are fundamental motivations to study linear 3-cut, which are described below. Before explaining the motivations, we note that (s, r, t)-Node-Lin-3-Cut is NP-hard and has no $(4/3 - \epsilon)$-approximation assuming UGC (by an approximation-preserving reduction from node 3-way cut[1] in undirected graphs), and admits a combinatorial 2-approximation[2]. It is a special case of directed multicut[3], which, for constant k, admits a trivial k-approximation and does not admit a $(k - \epsilon)$-approximation assuming UGC [4].

1.1 Motivations

Blocking arborescences. We recall that an *out-r-arborescence* (similarly, an *in-r-arborescence*) in a directed graph is a minimal subset of arcs such that every node has a unique path from r (to r) in the subgraph induced by the arcs. The smallest number of edges/nodes whose removal ensures that the graph has no arborescence holds the key to understanding reliability in networks. Computing this number is also a special case of the interdiction problem of covering bases of two *matroids* [2]. We recall that the problem of finding a minimum weight subset of edges/nodes whose deletion ensures that the remaining graph has no out-r-arborescence for a specified node r can be solved efficiently (by reducing to min $u \to v$ cut in directed graphs). The main motivation behind this work arose from the following closely related problem, abbreviated r-InOut-Node-Blocker: the input is a node-weighted directed graph with a specified terminal node r and the goal is to find a minimum weight set of non-terminal nodes whose removal ensures that the resulting graph has no out-r-arborescence *and* no in-r-arborescence. In this work, we show an approximation-preserving equivalence between r-InOut-Node-Blocker and (s, r, t)-Node-Lin-3-Cut. This equivalence, in turn, motivates the need to investigate the latter.

Global Bicut. In the $\{s, t\}$-Edge-BiCut problem the input is a directed graph and two specified nodes s and t, and the goal is to find a smallest subset of edges whose deletion ensures that s and t cannot reach each other in the resulting graph.

[1] In the node k-way cut problem, the input is a node-weighted undirected graph with k terminal nodes $\{t_1, \ldots, t_k\}$ and the goal is to find a minimum weight set of non-terminal nodes whose removal ensures that the terminals cannot reach each other. It has no $(2 - 2/k - \epsilon)$-approximation assuming UGC [5]. Reduction from node 3-way cut to (s, r, t)-Node-Lin-3-Cut: Bidirect all edges and add new nodes s, r, t with edges $s \to t_1, t_2 \to r \to t_2, t_3 \to t$.

[2] Find a minimum $s \to t$ cut, delete it. In the resulting graph (i) if r can reach t, then find a minimum $r \to t$ cut and delete it, (ii) else if s can reach r, then find a minimum $s \to r$ cut and delete it.

[3] The input to Dir-Multicut is an edge-weighted directed graph $G = (V, E)$ with k source-sink pairs of nodes $(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)$. The goal is to find a minimum weight subset of edges $E' \subseteq E$ such that there is no path from s_i to t_i in $G - E'$ for every $i \in \{1, \ldots, k\}$.

In the global variant of this problem, abbreviated EDGE-BICUT, the input is a directed graph, and the goal is to find a smallest subset of edges whose deletion ensures that the resulting graph has two distinct nodes s and t that cannot reach each other. We note that $\{s,t\}$-EDGE-BICUT and EDGE-BICUT are extensions of min $\{s,t\}$-cut and global min cut in undirected graphs to directed graphs respectively. While $\{s,t\}$-EDGE-BICUT does not admit an efficient $(2-\epsilon)$-approximation assuming UGC [4, 9], EDGE-BICUT admits an efficient $(2-1/448)$-approximation [1], thus exhibiting a dichotomy in the approximability between fixed-terminal and global variants. Intriguingly, determining whether EDGE-BICUT is NP-complete is still an open problem.

The algorithm achieving the $(2-1/448)$-approximation for EDGE-BICUT given in [1] uses a 3/2-approximation for a global version of (s,r,t)-EDGE-LIN-3-CUT as a subroutine. Since this global version can be reduced to (s,r,t)-EDGE-LIN-3-CUT, improving the approximability of the latter beyond 3/2 would improve the approximability of EDGE-BICUT itself. This suggests that the exact approximability of (s,r,t)-EDGE-LIN-3-CUT merits careful investigation.

Network Security. Interdiction problems have long served as a way to understand network reliability and to secure networks. The linear 3-cut problem also arose from one such application. Muthukumaran et al. [10] and Talele et al. [11] formulated the problem of placing security mediators in a distributed system as a cut problem. They modeled a distributed system as a directed graph with arcs indicating the direction of possible communication. The nodes are classified into various levels of integrity by monitoring how much they are compromised. Security is achieved by blocking information traveling from low integrity nodes to high integrity nodes. However, blocking information flow also alters the task that the system is trying to accomplish. Hence, minimum blocking is needed. This is naturally modeled as a cut problem involving ordered terminals, a special case of which is the linear k-cut problem. In the linear k-cut problem (EDGE-LIN-k-CUT), the input is a directed graph and k ordered terminal nodes and the goal is to find a smallest subset of edges whose removal ensures unreachability from any lower terminal node to any higher terminal node. Erbacher et al. [6] showed that EDGE-LIN-k-CUT admits a fixed parameter algorithm when parameterized by the size of the optimal solution.

Linear k-cut and Network coding. The information capacity in networks with delay constraints is closely related to a variant of multicut, namely Skew-Multicut [3]. In Skew-Multicut, the input consists of a directed graph with two ordered sets of terminals $(s_1,\ldots,s_{k-1})$, $(t_1,\ldots,t_{k-1})$ and the goal is to find a smallest subset of edges whose deletion ensures that s_i cannot reach t_j for every $i \le j$. Skew-Multicut is equivalent to EDGE-LIN-k-CUT[4]. Chekuri et al. [3] showed that the upper bound on

[4]Reduction from Skew-$(k-1)$-Multicut to EDGE-LIN-k-CUT: Add new nodes $s'_1,\ldots,s'_k$ and infinite weight edges $s'_i \to s_i$ for $i \in [1,k-1]$, $t_{i-1} \to s'_i$ for $i \in [2,k]$ and solve the EDGE-LIN-k-CUT instance with terminals $(s'_1,\ldots,s'_k)$. Reduction from EDGE-LIN-k-CUT to Skew-$(k-1)$-Skew-Multicut: Given a directed graph with terminals $(s_1,\ldots,s_k)$, add new nodes $s'_1,\ldots,s'_{k-1},t'_1,\ldots,t'_{k-1}$ and infinite weight edges $s'_i \to s_i$ for $i \in [1,k-1]$, $s_i \to t'_{i-1}$ for $i \in [2,k]$ and solve the Skew-Multicut problem w.r.t. terminal sets $(s'_1,\ldots,s'_{k-1})$, $(t'_1,\ldots,t'_{k-1})$

the integrality gap of a natural LP relaxation (Distance LP) for Skew-Multicut gives an upper bound on the gap between routing and optimal network coding in a delay constrained graph. Thus, obtaining tight bounds on the integrality gap of the Distance LP for Skew-Multicut/EDGE-LIN-k-CUT is of special significance to network coding. In particular, it is an intriguing open question to determine whether the integrality gap of the Distance LP for EDGE-LIN-k-CUT is constant for arbitrary k.

There is a straightforward rounding scheme showing an upper bound of $\lceil \log_2(k) \rceil$ on the integrality gap by recursively partitioning the terminal set and cutting all paths from terminals on the left to terminals on the right based on the LP solution. A simple reduction from node k-way cut to EDGE-LIN-k-CUT, shows a lower bound of $2(1-1/k)$ on the integrality gap. Chekuri-Madan [4] proved that the hardness of approximation for EDGE-LIN-k-CUT matches the integrality gap of the Distance LP. However, they do not improve the upper bound of $\lceil \log_2 k \rceil$ or the lower bound of $2(1-1/k)$ on the integrality gap. In this work, we improve both these bounds for $k=3$.

1.2 Results

The following is our main result.

Theorem 1.1. *There is an efficient $\sqrt{2}$-approximation for (s,r,t)-NODE-LIN-3-CUT. Assuming UGC, the problem has no efficient $(\sqrt{2}-\epsilon)$-approximation for any $\epsilon > 0$.*

Both the algorithm and the hardness result are based on a natural distance-based LP relaxation of the problem. We briefly remark on some of the salient features of our results.

Approximation. Our main contribution for the upper bound is an analysis exhibiting the tight approximation factor for a natural rounding scheme. A natural rounding scheme is to take the best of the following two alternatives: (i) first ensure that s and r cannot reach t by suitably rounding the LP-solution to obtain a node set K_1 to be removed, and then find a minimum $s \to r$ directed cut K_2 in the graph obtained after deleting K_1, and return $K_1 \cup K_2$; (ii) first ensure that s cannot reach r and t by suitably rounding the LP-solution to obtain a node set K_1 to be removed, and then find a minimum $r \to t$ directed cut K_2 in the graph obtained after deleting K_1, and return $K_1 \cup K_2$. We note that in both alternatives, the first step can be implemented by standard *deterministic ball-cut rounding schemes*[5] while the second step can be solved exactly in polynomial time. The main technical challenge lies in analyzing the approximation factor of such a *best of alternatives rounding scheme* where the second step in each alternative depends on the first. We overcome this challenge by showing that a weaker, single-step randomized ball-cut rounding scheme already achieves the desired expected value. The distribution underlying our single-step scheme turns out

[5] Pick $\theta \in (0,1)$ and set K_1 to be the set of nodes which have incoming (outgoing) arcs to nodes which are within a distance θ from the terminal(s) of interest. Since there are only finitely many θ values of interest, the best solution can be obtained in polynomial time.

to be extremely non-trivial in nature. In the proofs, we derive the distribution with the goal of obtaining the best approximation factor instead of stating the distribution upfront and bounding the approximation factor.

Inapproximability. It is known that the inapproximability factor under UGC for (s,r,t)-Node-Lin-3-Cut is identical to the integrality gap of a natural distance-based LP [4]. We construct a sequence of instances such that the sequence of integrality gaps of the distance-based LP converges to $\sqrt{2}$. Our gap instances are also non-trivial and can be viewed as a weighted graph sequence converging to a kind of "graph limit structure" having irrational weights. While irrational gap instances for semi-definite programming relaxations of natural combinatorial optimization problems are known to exist (e.g., the max-cut problem [7, 8]), the authors are unaware of irrational gap instances for natural LP-relaxations of natural combinatorial optimization problems besides the one studied in this work.

We next turn towards the applications that motivated our study of (s,r,t)-Node-Lin-3-Cut. We show that the approximability factors of r-InOut-Node-Blocker and (s,r,t)-Node-Lin-3-Cut coincide by exhibiting a combinatorial reduction between the two problems.

Theorem 1.2. *There exists an efficient α-approximation algorithm for r-*InOut-Node-Blocker *if and only if there exists an efficient α-approximation for* (s,r,t)-Node-Lin-3-Cut.

We finally mention that our upper bound on the approximability of (s,r,t)-Node--Lin-3-Cut in Theorem 1.1 in turn improves the approximability of Edge-BiCut. The new approximation factor is $(2-(\sqrt{2}-1)/(72+58\sqrt{2})) \approx 1.9973$ thus improving upon the previously known $(2-1/448) \approx 1.9977$ [1]. We refrain from including a proof of this result since it is identical to the one presented in [1] and the improved factor is obtained by directly plugging in the improved approximation factor for (s,r,t)-Node-Lin-3-Cut from Theorem 1.1.

Organization. We present the upper bound of Theorem 1.1 in Section 2 and the integrality gap instances leading to the lower bound of Theorem 1.1 in Section 3. We show the approximation-preserving equivalence between r-InOut-Node-Blocker and (s,r,t)-Node-Lin-3-Cut (Theorem 1.2) in Section 4.

2 A $\sqrt{2}$-approximation algorithm for (s,r,t)-Node-Lin-3-Cut

Let $D=(V,E)$ be an input digraph with specified nodes $s,r,t \in V$, and node weights $w \in \mathbb{R}_+^{V\setminus\{s,r,t\}}$. The (s,r,t)-Node-Lin-3-Cut problem asks for a minimum weight node set $U \subseteq V \setminus \{s,r,t\}$ such that $D[V-U]$ has no path from s to t, from s to r, and from r to t. The collection of feasible solutions remains the same if we add the arcs $t \to r$ and $r \to s$ to the digraph. In the rest of this section, we assume that these arcs are present in D.

For a subset $U \subseteq V$, let us denote $w(U) := \sum_{u \in U} w_u$. For nodes $u, v \in V$, let $\mathcal{P}_{uv}$ denote the set of all directed paths from u to v in D. For $x \in \mathbb{R}_+^V$ and a path P in D, we define $x(P) := \sum_{v \in V(P)} x_v$. For $u, v \in V$, let $\mathsf{dist}_x(u,v) := \min\{x(P) : P \in \mathcal{P}_{uv}\}$. A natural LP relaxation of (s,r,t)-NODE-LIN-3-CUT is the following, denoted DISTANCE-LP:

$$\begin{aligned} \min w^T x \\ x \in \mathbb{R}_+^V \\ \mathsf{dist}_x(s,t),\ \mathsf{dist}_x(s,r),\ \mathsf{dist}_x(r,t) \geq 1 \\ x_s = x_r = x_t = 0. \end{aligned} \tag{DISTANCE-LP}$$

This LP is solvable in polynomial time, since separation amounts to finding shortest paths. If x is a feasible solution to DISTANCE-LP, then there is a feasible solution x' to DISTANCE-LP such that $x'_v \leq x_v$ for every $v \in V$ and moreover:

$$\text{if } x'_v > 0, \text{ then } \mathsf{dist}_{x'}(r,v) + \mathsf{dist}_{x'}(v,r) \leq 1 + x'_v. \tag{1}$$

To achieve this property, we observe that if $x_v > 0$ and $\mathsf{dist}_x(r,v) + \mathsf{dist}_x(v,r) > 1 + x_v$, then $x(P) > 1$ for all $P \in \mathcal{P}_{st} \cup \mathcal{P}_{sr} \cup \mathcal{P}_{rt}$ that contains v. Indeed, for any such path P, there is a subset F of arcs from the set of arcs $\{t \to r, r \to s\}$ such that $F \cup P$ is the concatenation of $P_1 \in \mathcal{P}_{rv}$ and $P_2 \in \mathcal{P}_{vr}$; therefore, $x(P) = x(P_1) + x(P_2) - x_v > 1$. This means that we can decrease x_v until the property is satisfied.

Let x be a feasible solution to DISTANCE-LP that satisfies (1). We present an algorithm that, given x as input, constructs in polynomial time a feasible solution U to (s,r,t)-NODE-LIN-3-CUT that satisfies $w(U) \leq \sqrt{2} w^T x$. The algorithm itself is a simple and natural deterministic ball-cut scheme, described below. The main novelty is the proof of the approximation ratio, which is obtained by considering a weaker, randomized ball-cut algorithm.

For a node $u \in V$ and $0 < \theta \leq 1$, let

$$\begin{aligned} B^{out}(u,\theta) &:= \{v \in V : \mathsf{dist}_x(u,v) < \theta\}, \\ S^{out}(u,\theta) &:= \{v \in V : \mathsf{dist}_x(u,v) - x_v < \theta \leq \mathsf{dist}_x(u,v)\}, \\ B^{in}(u,\theta) &:= \{v \in V : \mathsf{dist}_x(v,u) < \theta\}, \\ S^{in}(u,\theta) &:= \{v \in V : \mathsf{dist}_x(v,u) - x_v < \theta \leq \mathsf{dist}_x(v,u)\}. \end{aligned}$$

One can think of $B^{out}(u,\theta)$ as the open ball of radius θ around u with respect to distances from u, and $S^{out}(u,\theta)$ can be thought of as the boundary of $B^{out}(u,\theta)$. The sets $B^{in}(u,\theta)$ and $S^{in}(u,\theta)$ are analogous, but with respect to distances *to* u. We note that $S^{out}(u,\theta)$ and $S^{in}(u,\theta)$ cannot contain nodes v with $x_v = 0$.

Claim 2.1. *For any $\theta \in (0,1]$, there exists $\theta' \in (0,1]$ such that $S^{out}(r,\theta') = S^{out}(r,\theta)$, and $\theta' = \mathsf{dist}_x(r,v)$ or $\theta' = \mathsf{dist}_x(r,v) - x_v$ for some $v \in V$. A similar statement holds for $S^{in}(r,\theta)$.*

Proof. Let $\theta' = \min\{\gamma : \gamma \geq \theta,\ \gamma = \mathsf{dist}_x(r,v) \text{ or } \gamma = \mathsf{dist}_x(r,v) - x_v \text{ for some } v \in V\}$. The minimum is chosen from a non-empty set because $\mathsf{dist}_x(r,t) = 1$. Now it is easy to verify that $S^{out}(r,\theta) = S^{out}(r,\theta')$. ☐

As a consequence, there are at most $2n$ distinct sets of the form $S^{out}(r,\theta)$ (where $\theta \in (0,1]$). The deterministic ball-cut scheme is based on enumerating these.

Deterministic Ball-Cut Algorithm for (s,r,t)-Node-Lin-3-Cut

Input: feasible solution x to DISTANCE-LP that satisfies (1)
Compute distances $\mathsf{dist}_x(r,v)$ and $\mathsf{dist}_x(v,r)$ for every $v \in V$
$U := V \setminus \{r,s,t\}$
for every $v \in V$ **do**
  for every $\theta \in (0,1] \cap \{\mathsf{dist}_x(r,v), \mathsf{dist}_x(r,v) - x_v\}$ **do**
    $K_1 := S^{out}(r,\theta)$
    Find minimum weight $s \to r$ cut K_2 in $D[V \setminus K_1]$
    if $w(K_1 \cup K_2) < w(U)$ **then** $U := K_1 \cup K_2$
  for every $\theta \in (0,1] \cap \{\mathsf{dist}_x(v,r), \mathsf{dist}_x(v,r) - x_v\}$ **do**
    $K_1 := S^{in}(r,\theta)$
    Find minimum weight $r \to t$ cut in $D[V \setminus K_1]$
    if $w(K_1 \cup K_2) < w(U)$ **then** $U := K_1 \cup K_2$
return U

The algorithm has the running time of $O(|V|)$ max flow computations. The following claim implies that the output is a feasible solution to (s,r,t)-NODE-LIN-3-CUT.

Claim 2.2. *If $\theta \in (0,1]$ and K is an $s \to r$ cut in $D[V \setminus S^{out}(r,\theta)]$, then $S^{out}(r,\theta) \cup K$ is a feasible solution to (s,r,t)-NODE-LIN-3-CUT. Similarly, if K is an $r \to t$ cut in $D[V \setminus S^{in}(r,\theta)]$, then $S^{in}(r,\theta) \cup K$ is a feasible solution.*

Proof. We prove the first part of the claim, the second part being similar. We observe that for every $u,v \in V$, every $P \in \mathcal{P}_{uv}$, and every two consecutive nodes w and w' in the direction of P, we have $\mathsf{dist}_x(u,w) \geq \mathsf{dist}_x(u,w') - x_{w'}$ and $\mathsf{dist}_x(w',v) \geq \mathsf{dist}_x(w,v) - x_w$.

We now show that every path $P \in \mathcal{P}_{rt}$ contains a node in $S^{out}(r,\theta)$. Let $P \in \mathcal{P}_{rt}$ with the nodes $w_0 := r, w_1, w_2, \ldots, w_k, w_{k+1} := t$ appearing in order. If $\mathsf{dist}_x(r,w_i) < \theta$ for every $i \in [k]$, then $\mathsf{dist}_x(r,w_k) < \theta \leq 1$ and hence, $\mathsf{dist}_x(r,t) < 1$, a contradiction. Hence, there exists a node w_i such that $\mathsf{dist}_x(r,w_i) \geq \theta$. Pick the node w_i with the smallest index i such that $\mathsf{dist}_x(r,w_i) \geq \theta$. By the observation from the previous paragraph, we have $\mathsf{dist}_x(r,w_i) - x_{w_i} \leq \mathsf{dist}_x(r,w_{i-1}) < \theta$, where the second inequality is by the choice of the index i. Thus, $w_i \in S^{out}(r,\theta)$ and hence, the path P contains a node in $S^{out}(r,\theta)$.

Due to the presence of the edge $r \to s$ in the graph and the fact that $s,r \notin S^{out}(r,\theta)$, we also have that every path $P \in \mathcal{P}_{st}$ contains a node in $S^{out}(r,\theta)$. Now, let us consider a path $P \in \mathcal{P}_{sr}$ without any nodes in $S^{out}(r,\theta)$. Since K is an $s \to r$ cut in $D[V \setminus S^{out}(r,\theta)]$, P contains a node in K. This means that $S^{out}(r,\theta) \cup K$ is a feasible solution to (s,r,t)-NODE-LIN-3-CUT. □

The difficulty of analyzing the approximation factor is due to the way the choice of K_2 depends on the choice of K_1. We overcome this difficulty by abandoning the minimum weight cuts K_2 in favor of random ball cuts that are easier to analyze. To do this, we need to define two types of feasible solutions to (s,r,t)-NODE-LIN-3-CUT.

For $0 < \theta_1 \le 1$ and $0 < \theta_2 \le 1$, the **vertical T-shaped cut** $V(\theta_1,\theta_2)$ is defined as

$$V(\theta_1,\theta_2) := S^{out}(r,\theta_1) \cup (B^{out}(r,\theta_1) \cap S^{in}(r,\theta_2)),$$

while the **horizontal T-shaped cut** $H(\theta_1,\theta_2)$ is defined as

$$H(\theta_1,\theta_2) := S^{in}(r,\theta_2) \cup (B^{in}(r,\theta_2) \cap S^{out}(r,\theta_1)).$$

The name "T-shaped cut" comes from the observation that if each node v is represented in the plane by the square $(\mathsf{dist}_x(r,v)-x_v, \mathsf{dist}_x(r,v)] \times (\mathsf{dist}_x(v,r)-x_v, \mathsf{dist}_x(v,r)]$, then the cut consists of nodes whose square is intersected by two segments forming a rotated "T" shape (see Figure 1).

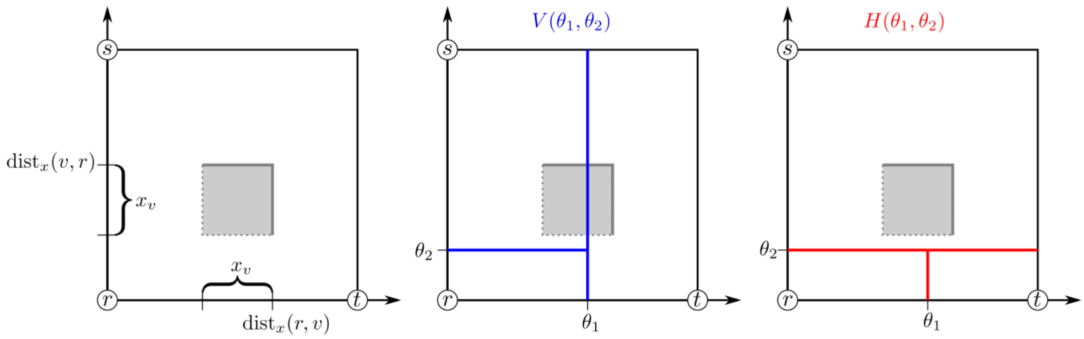

Figure 1: Representation of T-shaped cuts. Left: the square corresponding to node v. Center: v is in $V(\theta_1,\theta_2)$ because one of the blue lines intersects the square. Right: v is not in $H(\theta_1,\theta_2)$ because the red lines do not intersect the square.

Lemma 2.1. *The set* $B^{out}(r,\theta_1) \cap S^{in}(r,\theta_2)$ *is an* $s \to r$ *cut in* $D[V \setminus S^{out}(r,\theta_1)]$*, and* $B^{in}(r,\theta_2) \cap S^{out}(r,\theta_1)$ *is an* $r \to t$ *cut in* $D[V \setminus S^{in}(r,\theta_2)]$.

Proof. We prove only the first part of the claim, the proof of the second part being similar. Let us consider a path $P \in \mathcal{P}_{sr}$ without any nodes in $S^{out}(r,\theta_1)$. Let the nodes in P be $w_0 := s, w_1, w_2, \dots, w_k, w_{k+1} := r$ appearing in order. We will show that $w_i \in B^{out}(r,\theta_1)$ for every $i \in \{1,\dots,k\}$ by induction on i. For the base case, owing to the presence of the edge $r \to s$ in the graph, we have that $\mathsf{dist}_x(r,w_0) - x_{w_0} = 0 < \theta_1$ and since $w_0 \notin S^{out}(r,\theta_1)$, it follows that $\mathsf{dist}_x(r,w_0) < \theta_1$. For the induction step, we have that $\mathsf{dist}_x(r,w_{i+1}) - x_{w_{i+1}} \le \mathsf{dist}_x(r,w_i) < \theta_1$, where the second inequality follows by induction hypothesis. Now, since $w_{i+1} \notin S^{out}(r,\theta_1)$, it follows that $\mathsf{dist}_x(r,w_{i+1}) < \theta_1$. Hence, all nodes of P are in $B^{out}(r,\theta_1)$.

We now show that at least one of the nodes in P should be in $S^{in}(r,\theta_2)$. If $\mathsf{dist}_x(w_i,r) < \theta_2$ for every $i \in [k]$, then $\mathsf{dist}_x(w_1,r) < \theta_2 \le 1$ and hence, $\mathsf{dist}_x(s,r) < 1$, a contradiction. Hence, there exists a node w_i such that $\mathsf{dist}_x(w_i,r) \ge \theta_2$. Pick the node w_i with the largest index i such that $\mathsf{dist}_x(w_i, r) \geq \theta_2$. We have $\mathsf{dist}_x(w_i, r) - x_{w_i} \leq \mathsf{dist}_x(w_{i+1}, r) < \theta_2$, where the second inequality is by the choice of the index i. Thus, $w_i \in S^{in}(r, \theta_2)$ and hence, the path P contains a node in $S^{in}(r, \theta_2)$. Consequently, P contains a node in $B^{out}(r, \theta_1) \cap S^{in}(r, \theta_2)$. □

Corollary 2.1. *Every T-shaped cut is a feasible solution to (s,r,t)-*NODE-LIN-3-CUT*, and the weight of the cut found by the Deterministic Ball-Cut Algorithm is at most the minimum weight of a T-shaped cut.*

Proof. Feasibility follows directly from Lemma 2.1 and Claim 2.2. For the second statement, consider $V(\theta_1, \theta_2)$ for some $\theta_1, \theta_2 \in (0, 1]$. By Claim 2.1, there exists $\theta' \in \{\mathsf{dist}_x(r, v), \mathsf{dist}_x(r, v) - x_v\}$ for some $v \in V$ such that $S^{out}(r, \theta') = S^{out}(r, \theta_1)$. When the algorithm considers v and θ', it finds a minimum weight $s \to r$ cut K_2 in $D[V \setminus S^{out}(r, \theta)]$. As $B^{out}(r, \theta_1) \cap S^{in}(r, \theta_2)$ is also an $s \to r$ cut in $D[V \setminus S^{out}(r, \theta)]$ by Lemma 2.1, $w(S^{out}(r, \theta') \cup K_2) \leq w(S^{out}(r, \theta_1) \cup (B^{out}(r, \theta_1) \cap S^{in}(r, \theta_2))$. □

We can bound the approximation factor of Deterministic Ball-Cut Algorithm by estimating the minimum weight of a T-shaped cut. We show that this gives the desired factor of $\sqrt{2}$.

Theorem 2.1. *There exists a T-shaped cut U such that $w(U) \leq \sqrt{2} w^T x$.*

To prove Theorem 2.1, we will follow a probabilistic argument. We will exhibit a distribution over T-shaped cuts for which the expected weight satisfies the bound mentioned in Theorem 2.1. This distribution turns out to be non-trivial in nature. Instead of stating this distribution upfront and analyzing its approximation factor, we will derive the optimal distribution as a natural consequence of the following lemma, which provides a sufficient condition for achieving a certain approximation factor.

Lemma 2.2. *Let $\xi : [0,1]^2 \to \mathbb{R}_+$ be a function satisfying*

$$\int_0^1 (\xi(a,z) + \xi(b,z))\,\mathrm{d}z + \int_a^1 \xi(z,b)\,\mathrm{d}z + \int_b^1 \xi(z,a)\,\mathrm{d}z = 1 \quad \forall a, b \in \mathbb{R}_+,\ a + b \leq 1. \tag{2}$$

*Let $\alpha := 2\int_0^1 \int_0^1 \xi(z_1, z_2)\,\mathrm{d}z_1\,\mathrm{d}z_2$. Then, for any instance of (s,r,t)-*NODE-LIN-3-CUT*, there exists a T-shaped cut U such that*

$$w(U) \leq \left(\frac{1}{\alpha}\right) w^T x.$$

Proof. We define a probability distribution on the set of T-shaped cuts by giving a weighing function $f : \{\mathrm{Ver}, \mathrm{Hor}\} \times [0,1]^2 \to \mathbb{R}_+$. For $(\theta_1, \theta_2) \in [0,1]^2$, let $f(\mathrm{Ver}, \theta_1, \theta_2) := \xi(\theta_1, \theta_2)/\alpha$ and $f(\mathrm{Hor}, \theta_1, \theta_2) := \xi(\theta_2, \theta_1)/\alpha$. For a T-shaped cut U, let

$$\mathsf{Pr}(U) := \int_{(\theta_1,\theta_2): V(\theta_1,\theta_2) = U} f(\mathrm{Ver}, \theta_1, \theta_2)\,\mathrm{d}\theta_1\,\mathrm{d}\theta_2 + \int_{(\theta_1,\theta_2): H(\theta_1,\theta_2) = U} f(\mathrm{Hor}, \theta_1, \theta_2)\,\mathrm{d}\theta_1\,\mathrm{d}\theta_2.$$

We mention that a node set U could be both a horizontal and a vertical T-shaped cut in which case, the probability mass for U comes from both integrals in the above sum.

Furthermore $\mathsf{Pr}(\cdot)$ is a probability distribution supported over the set of T-shaped cuts because of the definition of α. Let U be a T-shaped cut chosen according to this distribution.

Claim 2.3. *For $v \in V \setminus \{r, s, t\}$, probability that v is in the chosen T-shaped cut U is at most x_v/α.*

Proof. We may assume that $x_v \neq 0$ since every vertex in a T-shaped cut necessarily has this property. Let $a := \mathsf{dist}_x(r, v)$ and $b := \mathsf{dist}_x(v, r)$. We recall that a vertical T-shaped cut $V(\theta_1, \theta_2)$ is defined as $S^{out}(r, \theta_1) \cup (B^{out}(r, \theta_1) \cap S^{in}(r, \theta_2))$. Thus, $V(\theta_1, \theta_2)$ contains the node v if and only if either (1) $a - x_v < \theta_1 \leq a$, or (2) $a < \theta_1$ and $b - x_v < \theta_2 \leq b$. Similarly, a horizontal T-shaped cut $H(\theta_1, \theta_2)$ contains the node v if and only if either (3) $b - x_v < \theta_2 \leq b$, or (4) $b < \theta_2$ and $a - x_v < \theta_1 \leq a$. Therefore the probability of v being in a random T-shaped cut is at most

$$\begin{aligned} P := \frac{1}{\alpha} \Bigg(& \int_{z_2=0}^{1} \int_{z_1=a-x_v}^{a} \xi(z_1, z_2) \, \mathrm{d}z_1 \, \mathrm{d}z_2 + \int_{z_2=b-x_v}^{b} \int_{z_1=a}^{1} \xi(z_1, z_2) \, \mathrm{d}z_1 \, \mathrm{d}z_2 \\ & + \int_{z_2=b-x_v}^{b} \int_{z_1=0}^{1} \xi(z_2, z_1) \, \mathrm{d}z_1 \, \mathrm{d}z_2 + \int_{z_2=b}^{1} \int_{z_1=a-x_v}^{a} \xi(z_2, z_1) \, \mathrm{d}z_1 \, \mathrm{d}z_2 \Bigg) \end{aligned}$$

By change of variables, we have that

$$\begin{aligned} P = \frac{1}{\alpha} \int_{y=0}^{x_v} \Bigg(& \int_{z=0}^{1} \xi(a - y, z) \, \mathrm{d}z + \int_{z=a}^{1} \xi(z, b - x_v + y) \, \mathrm{d}z \\ & + \int_{z=0}^{1} \xi(b - x_v + y, z) \, \mathrm{d}z + \int_{z=b}^{1} \xi(z, a - y) \, \mathrm{d}z \Bigg) \, \mathrm{d}y. \end{aligned}$$

For $0 \leq y \leq x_v$, we have $a - y \leq a$ and $b - x_v + y \leq b$. By assumption, $\xi(z_1, z_2)$ is non-negative in the domain. Therefore, we have

$$\int_{y=0}^{x_v} \int_{z=a}^{1} \xi(z, b - x_v + y) \, \mathrm{d}z \, \mathrm{d}y \leq \int_{y=0}^{x_v} \int_{z=a-y}^{1} \xi(z, b - x_v + y) \, \mathrm{d}z \, \mathrm{d}y, \text{ and}$$
$$\int_{y=0}^{x_v} \int_{z=b}^{1} \xi(z, a - y) \, \mathrm{d}z \, \mathrm{d}y \leq \int_{y=0}^{x_v} \int_{z=b-x_v+y}^{1} \xi(z, a - y) \, \mathrm{d}z \, \mathrm{d}y.$$

Hence,

$$\begin{aligned} P \leq \frac{1}{\alpha} \int_{y=0}^{x_v} \Bigg(& \int_{z=0}^{1} \xi(a - y, z) \, \mathrm{d}z + \int_{z=a-y}^{1} \xi(z, b - x_v + y) \, \mathrm{d}z \\ & + \int_{z=0}^{1} \xi(b - x_v + y, z) \, \mathrm{d}z + \int_{z=b-x_v+y}^{1} \xi(z, a - y) \, \mathrm{d}z \Bigg) \, \mathrm{d}y \\ = \frac{1}{\alpha} \int_{y=0}^{x_v} 1 \, \mathrm{d}y = \frac{x_v}{\alpha}, & \end{aligned}$$

where the equality at the beginning of the last row follows from (2), since for $0 \leq y \leq x_v$, we have $(a - y) + (b - x_v + y) = a + b - x_v = \mathsf{dist}_x(r, v) + \mathsf{dist}_x(v, r) - x_v \leq 1$ by (1), and moreover $a - y \geq a - x_v = \mathsf{dist}_x(r, v) - x_v \geq 0$ and $b - x_v + y \geq b - x_v = \mathsf{dist}_x(v, r) - x_v \geq 0$ by the definition of $\mathsf{dist}_x(\cdot, \cdot)$. ∎

Since every node v is in the random T-shaped cut with probability at most $\frac{x_v}{\alpha}$, expected weight of a random T-shaped cut is at most $\frac{w^T x}{\alpha}$. Therefore, there is a T-shaped cut U with $w(U) \leq \frac{w^T x}{\alpha}$. □

To prove the Theorem 2.1, it is enough by Lemma 2.2 to show the existence of a function $\xi : [0,1]^2 \to \mathbb{R}_+$ satisfying (2) for which

$$\int_0^1 \int_0^1 \xi(z_1, z_2)\, \mathrm{d}z_1\, \mathrm{d}z_2 = \frac{1}{2\sqrt{2}}. \tag{3}$$

It turns out that such a function exists, but its structure is surprisingly complex. We define two regions where the function ξ will have positive values, see Figure 2.

$$\mathcal{R}_1 := \left\{ (z_1, z_2) : \frac{1}{\sqrt{2}} < z_1, z_2 \leq 1 \right\},$$
$$\mathcal{R}_2 := \left\{ (z_1, z_2) : \frac{\sqrt{2}-1}{\sqrt{2}} \leq z_1 \leq \frac{1}{\sqrt{2}},\ z_1 + z_2 \leq 1 \right\}.$$

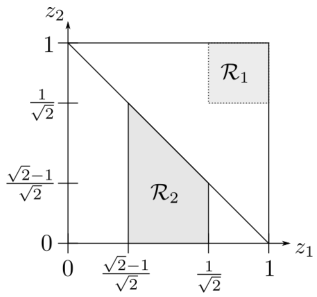

Figure 2: The regions $\mathcal{R}_1$ and $\mathcal{R}_2$.

Remark. The reason for this restriction on the support of ξ will become apparent in Section 3, where we present an infinite sequence of node-weighted graphs for which the integrality gap of DISTANCE-LP converges to $\sqrt{2}$. It can be seen that $\mathcal{R}_1 \cup \mathcal{R}_2$ consists of the pairs (z_1, z_2) for which the weight of the vertical T-shaped cut $V(z_1, z_2)$ (based on the optimal LP solution x) converges to 1 in the graph sequence. Informally, the region $\mathcal{R}_1 \cup \mathcal{R}_2$ is the region where the complementary slackness conditions allow positive density, if we consider the "limit" of the weighted graph sequence defined in Section 3. However, this is not the usual notion of graph limit, so we do not formalize this statement as it is not necessary for the proof.

Proof of Theorem 2.1. To prove the theorem, we define a function ξ with the above properties. The value of ξ is defined to be 0 for $(z_1, z_2) \in [0,1]^2 \setminus (\mathcal{R}_1 \cup \mathcal{R}_2)$. For every $(z_1, z_2) \in \mathcal{R}_1$, we set $\xi(z_1, z_2) := (\sqrt{2}+1)/\sqrt{2}$. In the region $\mathcal{R}_2$, the value of $\xi(z_1, z_2)$ will depend only on z_1. In particular, we will define a function $\zeta : [\frac{\sqrt{2}-1}{\sqrt{2}}, \frac{1}{\sqrt{2}}] \to \mathbb{R}_+$, and define $\xi(z_1, z_2)$ in the region $\mathcal{R}_2$ as

$$\xi(z_1, z_2) := \zeta(z_1).$$

Let us examine the properties that must be satisfied by ζ in order for ξ to satisfy (2).

Claim 2.4. *Condition* (2) *is satisfied by* ξ *if the following hold for* ζ*:*

$$\zeta\left(\frac{\sqrt{2}-1}{\sqrt{2}}\right) = 0, \tag{4}$$

$$(1-y)\zeta(y) + \int_y^{1-y} \zeta(z)\,\mathrm{d}z = \frac{1}{2} \quad \text{if } \frac{\sqrt{2}-1}{\sqrt{2}} \le y \le \frac{1}{2}, \tag{5}$$

$$(1-y)\zeta(y) - \int_{1-y}^{y} \zeta(z)\,\mathrm{d}z = \frac{1}{2} \quad \text{if } \frac{1}{2} \le y \le \frac{1}{\sqrt{2}}. \tag{6}$$

Proof. We consider several cases based on the values of a and b. Let $\Gamma(a,b)$ denote the LHS of (2). By taking $y = \frac{\sqrt{2}-1}{\sqrt{2}}$ in (5) and substituting $\zeta(\frac{\sqrt{2}-1}{\sqrt{2}}) = 0$, we obtain that

$$\int_{\frac{\sqrt{2}-1}{\sqrt{2}}}^{\frac{1}{\sqrt{2}}} \zeta(z)\,\mathrm{d}z = \frac{1}{2}. \tag{7}$$

Case 1(i): Suppose $a > \frac{1}{\sqrt{2}}$. Then $b \le \frac{\sqrt{2}-1}{\sqrt{2}}$ since $a+b \le 1$. Now,

$$\begin{aligned}\Gamma(a,b) &= \int_0^1 \xi(a,z)\,\mathrm{d}z + \int_0^1 \xi(b,z)\,\mathrm{d}z + \int_a^1 \xi(z,b)\,\mathrm{d}z + \int_b^1 \xi(z,a)\,\mathrm{d}z \\ &= \left(1+\frac{1}{\sqrt{2}}\right)\left(1-\frac{1}{\sqrt{2}}\right) + 0 + 0 + \left(1+\frac{1}{\sqrt{2}}\right)\left(1-\frac{1}{\sqrt{2}}\right) = 1.\end{aligned}$$

Case 1(ii): Suppose $b > \frac{1}{\sqrt{2}}$. Proceeding similar to Case 1(i), we obtain $\Gamma(a,b) = 1$.

Case 2: Suppose $a \le \frac{\sqrt{2}-1}{\sqrt{2}}$ and $b \le \frac{\sqrt{2}-1}{\sqrt{2}}$. In this case, we have

$$\begin{aligned}\Gamma(a,b) = \int_0^1 \xi(a,z)\,\mathrm{d}z + \int_0^1 \xi(b,z)\,\mathrm{d}z + \int_a^1 \xi(z,b)\,\mathrm{d}z + \int_b^1 \xi(z,a)\,\mathrm{d}z \\ = 0 + 0 + 2\int_{\frac{\sqrt{2}-1}{\sqrt{2}}}^{\frac{1}{\sqrt{2}}} \zeta(z)\,\mathrm{d}z = 1.\end{aligned}$$

Case 3(i): Suppose $\frac{\sqrt{2}-1}{\sqrt{2}} < a \le \frac{1}{\sqrt{2}}$ and $b \le \frac{\sqrt{2}-1}{\sqrt{2}}$. Then

$$\begin{aligned}\Gamma(a,b) &= \int_0^1 \xi(a,z)\,\mathrm{d}z + \int_0^1 \xi(b,z)\,\mathrm{d}z + \int_a^1 \xi(z,b)\,\mathrm{d}z + \int_b^1 \xi(z,a)\,\mathrm{d}z \\ &= \int_0^{1-a} \zeta(a)\,\mathrm{d}z + 0 + \int_a^{\frac{1}{\sqrt{2}}} \zeta(z)\,\mathrm{d}z + \int_{\frac{\sqrt{2}-1}{\sqrt{2}}}^{1-a} \zeta(z)\,\mathrm{d}z \\ &= (1-a)\zeta(a) + \int_a^{\frac{1}{\sqrt{2}}} \zeta(z)\,\mathrm{d}z + \int_{\frac{\sqrt{2}-1}{\sqrt{2}}}^{1-a} \zeta(z)\,\mathrm{d}z.\end{aligned} \tag{8}$$

If $a \le \frac{1}{2}$, then the RHS from (8) can be written as

$$\Gamma(a, b) = (1-a)\zeta(a) + \int_{a}^{1-a} \zeta(z)\,\mathrm{d}z + \int_{\frac{\sqrt{2}-1}{\sqrt{2}}}^{\frac{1}{\sqrt{2}}} \zeta(z)\,\mathrm{d}z = 1$$

by (5) and (7). If $a > \frac{1}{2}$, then the RHS from (8) can be written as

$$\Gamma(a, b) = (1-a)\zeta(a) - \int_{1-a}^{a} \zeta(z)\,\mathrm{d}z + \int_{\frac{\sqrt{2}-1}{\sqrt{2}}}^{\frac{1}{\sqrt{2}}} \zeta(z)\,\mathrm{d}z = 1$$

by (6) and (7).

Case 3(ii): Suppose $a \le \frac{\sqrt{2}-1}{\sqrt{2}}$ and $\frac{\sqrt{2}-1}{\sqrt{2}} < b \le \frac{1}{\sqrt{2}}$. Proceeding similar to Case 3(a), we obtain that $\Gamma(a, b) = 1$.

Case 4: Suppose $\frac{\sqrt{2}-1}{\sqrt{2}} < a \le \frac{1}{\sqrt{2}}$ and $\frac{\sqrt{2}-1}{\sqrt{2}} < b \le \frac{1}{\sqrt{2}}$. Moreover, we have $a + b \le 1$.

$$\begin{aligned}\Gamma(a, b) &= \int_0^1 \xi(a, z)\,\mathrm{d}z + \int_0^1 \xi(b, z)\,\mathrm{d}z + \int_a^1 \xi(z, b)\,\mathrm{d}z + \int_b^1 \xi(z, a)\,\mathrm{d}z \\ &= \int_0^{1-a} \zeta(a)\,\mathrm{d}z + \int_0^{1-b} \zeta(b)\,\mathrm{d}z + \int_a^{1-b} \zeta(z)\,\mathrm{d}z + \int_b^{1-a} \zeta(z)\,\mathrm{d}z \\ &= (1-a)\zeta(a) + (1-b)\zeta(b) + \int_a^{1-b} \zeta(z)\,\mathrm{d}z + \int_b^{1-a} \zeta(z)\,\mathrm{d}z.\end{aligned}$$

We will assume that $a \le b$ (the other case is similar). If $b \le \frac{1}{2}$, then

$$\int_a^{1-b} \zeta(z)\,\mathrm{d}z + \int_b^{1-a} \zeta(z)\,\mathrm{d}z = \int_a^{1-a} \zeta(z)\,\mathrm{d}z + \int_b^{1-b} \zeta(z)\,\mathrm{d}z,$$

and hence $\Gamma(a, b) = 1$ follows from (5). If $b > \frac{1}{2}$, then, $a \le 1 - b < \frac{1}{2} \le b$ and hence, we have

$$\int_a^{1-b} \zeta(z)\,\mathrm{d}z + \int_b^{1-a} \zeta(z)\,\mathrm{d}z = \int_a^{1-a} \zeta(z)\,\mathrm{d}z - \int_{1-b}^{b} \zeta(z)\,\mathrm{d}z.$$

Therefore, $\Gamma(a, b) = 1$ follows from (5) and (6). □

We note that conditions (4)-(6) on ζ are actually necessary for (2) to hold. This fact is not needed for the proof of Theorem 2.1, but we prove it in Appendix A to facilitate future investigations on our distribution.

In order to complete the proof of the theorem, we have to find a function $\zeta : [\frac{\sqrt{2}-1}{\sqrt{2}}, \frac{1}{\sqrt{2}}] \to \mathbb{R}_+$ that satisfies properties (4)-(6). By solving the differential equations corresponding to (4)-(6), we get that the function satisfying these properties is

$$\zeta(y) := \frac{2y(2-y) - 1}{4y(1-y)^2}. \tag{9}$$

By substituting the function values, it can be verified that $\int_0^1 \int_0^1 \xi(z_1, z_2)\,\mathrm{d}z_1\,\mathrm{d}z_2 = \frac{1}{2\sqrt{2}}$. We present the calculations needed for verification in Appendix B. □

For clarity, we conclude the section by describing the obtained distribution explicitly. The probability of choosing a given T-shaped cut U is

$$\mathsf{Pr}(U) := \sqrt{2}\left(\int_{(\theta_1,\theta_2):V(\theta_1,\theta_2)=U} \xi(\theta_1,\theta_2)\,\mathrm{d}\theta_1\,\mathrm{d}\theta_2 + \int_{(\theta_1,\theta_2):H(\theta_1,\theta_2)=U} \xi(\theta_2,\theta_1)\,\mathrm{d}\theta_1\,\mathrm{d}\theta_2\right),$$

where

$$\xi(\theta_1,\theta_2) := \begin{cases} \frac{\sqrt{2}+1}{\sqrt{2}} & \text{if } \frac{1}{\sqrt{2}} < \theta_1, \theta_2 \leq 1, \\ \frac{2\theta_1(2-\theta_1)-1}{4\theta_1(1-\theta_1)^2} & \text{if } \frac{\sqrt{2}-1}{\sqrt{2}} \leq \theta_1 \leq \frac{1}{\sqrt{2}} \text{ and } \theta_1+\theta_2 \leq 1, \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

3 Integrality Gap

It is known that the inapproximability factor of (s,r,t)-Node-Lin-3-Cut under UGC coincides with the integrality gap of the Distance-LP [4]. In this section, we present an integrality gap instance for (s,r,t)-Node-Lin-3-Cut which will in turn show the inapproximability result in Theorem 1.1.

Theorem 3.1. *The integrality gap of the* Distance-LP *is at least* $\sqrt{2}$.

We will construct a sequence of node-weighted graphs for which the integrality gap converges to $\sqrt{2}$. In most previously known integrality gap instances for distance-based linear programs for directed multicut-like problems, the node weights were uniformly set to be one. In contrast, our gap instance assigns varying weights to the nodes.

3.1 Gap Instance Construction

Let M be a positive integer. We will construct a graph $G = (V, E)$ on $(M+1)^2 + 3$ nodes with weights on the nodes. For convenience, let us define $V' := \{(i,j) : i,j \in \{0,1,\ldots,M\}\}$. Thus, we may view V' as the nodes of a $(M+1)\times(M+1)$-grid whose columns and rows are indexed from 0 to M (we will follow the convention that the first index denotes the column while the second index denotes the row). The node set of G is given by $V := \{s,r,t\} \cup V'$. We now define the weights on the nodes. The construction involves a parameter $\alpha \in (0, 1/2)$ that will be determined later. We denote the weight of node (i,j) to be w_{ij} and define[6]

$$w_{ij} := \begin{cases} 0 & \text{if } i+j > M, \\ \frac{1-\alpha}{M} & \text{if } i+j < M, \\ \frac{1}{2} - \frac{(1-\alpha)i}{M} & \text{if } i+j = M,\ i < M\left(1 - \frac{1}{2(1-\alpha)}\right), \\ \alpha & \text{if } i+j = M,\ M\left(1-\frac{1}{2(1-\alpha)}\right) \leq i \leq M\left(\frac{1}{2(1-\alpha)}\right), \text{ and} \\ \frac{1}{2} - \frac{(1-\alpha)j}{M} & \text{if } i+j = M,\ i > M\left(\frac{1}{2(1-\alpha)}\right). \end{cases}$$

[6]The various boundary conditions in the definition of the node weights will have to use appropriately rounded down and rounded up boundary values. We avoid this technicality in the interests of simplicity.

The edge set E consists of undirected and directed edges. The undirected edges consist of the following: every node (i,j) is adjacent to all nodes in $V' \cap \{(i-1,j-1),(i-1,j),(i-1,j+1),(i,j-1),(i,j+1),(i+1,j-1),(i+1,j),(i+1,j+1)\}$. The directed edges consist of $r \to s, t \to r$, $s \to (i,M)$ and $(i,0) \to r$ for every $i \in \{0,1,\ldots,M\}$, and $(M,j) \to t$ and $r \to (0,j)$ for every $j \in \{0,1\ldots,M\}$. See Figure 3.

We will refer to the subgraph of G induced by the vertex-set V' as a *diagonalized-grid*. The leftmost column, rightmost column, bottommost row, topmost row, and diagonal refer to $\{(0,j) : j \in \{0,1,\ldots,M\}\}$, $\{(M,j) : j \in \{0,1,\ldots,M\}\}$, $\{(i,0) : i \in \{0,1,\ldots,M\}\}$, $\{(i,M) : i \in \{0,1,\ldots,M\}\}$, and $\{(i,j) : i,j \in \{0,1,\ldots,M\}, i+j = M\}$ respectively.

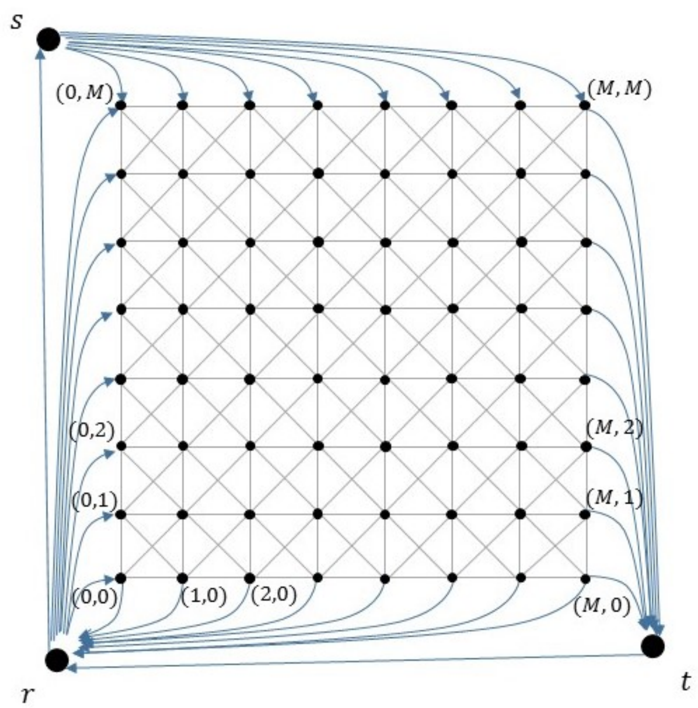

Figure 3: The graph corresponding to the integrality gap instance. The black edges are undirected while the blue edges are directed. The node weights are not shown.

3.2 Proof of Gap

The following lemma bounds the value of an optimal solution to the linear program.

Lemma 3.1. *An optimal solution to the* Distance-LP *for the node-weighted graph constructed above has weight at most*

$$\left(\frac{1}{M}\right)\sum_{i=0}^{M}\sum_{j=0}^{M} w_{ij}.$$

Proof. It is sufficient to exhibit a feasible solution to the linear program whose objective value is as specified in the lemma. We will show that $x_{(i,j)} := 1/M$ for every $i, j \in \{0, 1, \ldots, M\}, i+j \leq M$ and $x_{(i,j)} := 1$ for every $i, j \in \{0, 1, \ldots, M\}, i+j > M$, is a feasible solution to the linear program. We recall that nodes (i, j) with $i+j > M$ have weight $w_{ij} = 0$. Let us consider the graph H obtained from G by removing all nodes (i, j) with $i + j > M$. To show feasibility of x, it suffices to show that every path from s to r, from r to t and from s to t has at least M intermediate nodes in H.

A path in H from a node $(i, j) \in V(H)$ to r has to cross j intermediate rows and hence has at least j internal nodes. Hence, for every node (i, j) with $i, j \in \{0, 1 \ldots, M\}, i + j \leq M$, the number of internal nodes in every path from (i, j) to r in H is at least j. Now every path from s to r in H has to go through $(0, M)$ and hence has at least M internal nodes from $V(H) \cap V'$. Similarly, every path from r to t in H has at least M internal nodes from $V(H) \cap V'$. Finally, the distance from s to t in H is at least the distance from r to t in H owing to the edge $r \to s$ and hence, the number of internal nodes from $V(H) \cap V'$ in any path from s to t in H is also at least M. □

The next lemma shows a lower bound on the objective value of an integral optimum solution.

Lemma 3.2. *An optimal solution to* (s, r, t)-NODE-LIN-3-CUT *in the node-weighted graph constructed above has weight at least* 1.

Proof. Let U^* be an integral optimal solution. We will show the lower bound on the weight of U^* in two steps. We define the *axis-parallel neighbors* of a node (i, j) to be the nodes in $\{(i + 1, j), (i - 1, j), (i, j + 1), (i, j - 1)\}$ and a path to be an *axis-parallel path* if all neighbors of a node (i, j) occurring in the path are its axis-parallel neighbors. In the first step of the proof, we will show that U^* consists of an axis-parallel path from a node in the topmost row to a node in the bottommost row and an axis-parallel path from a node in the leftmost column to a node in rightmost column. In the second step of the proof, we will show a lower bound on the total weight of the union of the nodes in these two paths.

Lemma 3.3. *The optimal solution U^* contains a set of nodes which form an axis-parallel path P_1 from a node in the bottommost row to a node in the topmost row in G.*

Lemma 3.4. *The optimal solution U^* contains a set of nodes which form an axis-parallel path P_2 from a node in the leftmost column to a node in the rightmost column in G.*

We defer the proof of Lemmas 3.3 and 3.4 and proceed with the proof of Lemma 3.2. The next claim follows immediately from the definition of axis-parallel paths.

Claim 3.1. *Every axis-parallel path from node (i_1, j_1) to (i_2, j_2) contains at least $|i_2 - i_1| + |j_2 - j_1| - 1$ internal nodes.* □

We also have the following claim from the definition of the node weights.

Claim 3.2. *For every node (i,j) for $i,j \in \{0,1,\dots,M\}$ with $i+j=M$, we have*

$$w_{ij} = \max\left\{\frac{1}{2} - \frac{(1-\alpha)i}{M}, \alpha, \frac{1}{2} - \frac{(1-\alpha)j}{M}\right\}. \qquad \square$$

Let P_1 and P_2 be the node sets guaranteed by Lemmas 3.3 and 3.4 respectively. Without loss of generality, let P_1 be the node set in U^* that induces an axis-parallel path from a node $(a,0)$ to a node (b,M). Similarly, let P_2 be the node set in U^* that induces an axis-parallel path from a node $(0,c)$ to a node (M,d) (see Figure 4).

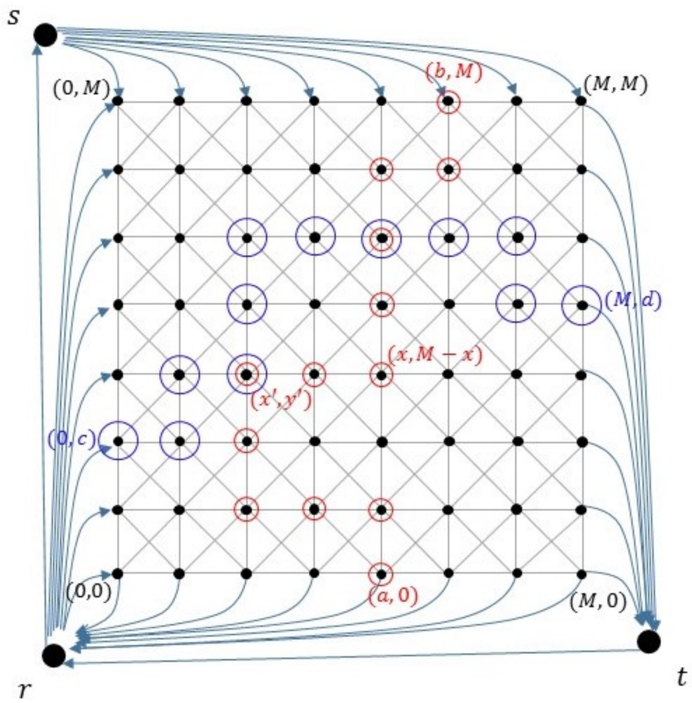

Figure 4: The red circled nodes denote the axis-parallel path P_1 and the blue circled nodes denote the axis-parallel path P_2.

Since P_1 is an axis-parallel path from a node in the bottommost row to a node in the topmost row, there exists a node in P_1 from the diagonal. Let $(x, M-x)$ be the first node along the axis-parallel path P_1 that is in the diagonal. Let P_1' be the restriction of P_1 from $(a,0)$ to $(x, M-x)$. Let (x', y') be the first node along the axis-parallel path P_2 that is either in the diagonal or in P_1'. Let P_2' be the restriction of P_2 from $(0,c)$ to (x',y'). By construction, all nodes (p,q) of $P_1' \cup P_2'$ satisfy $p+q \leq M$. We will show that the total weight of the nodes in $P_1' \cup P_2'$ is at least 1. This suffices since $P_1' \cup P_2' \subseteq U^*$. We distinguish two cases.

1. Suppose P_2' is a path from $(0,c)$ to a node (i,j) of P_1', where $i+j \leq M$. By Claim 3.1, the axis-parallel path P_2' has at least $|i-0|+|j-c|-1 \geq i-1$ internal nodes. Furthermore, the path P_1' is the concatenation of an axis-parallel path Q_1 from $(a,0)$ to (i,j) and an axis-parallel path Q_2 from (i,j) to $(x, M-x)$.

Hence, by Claim 3.1, the axis-parallel path P_1' has at least $|i-a|+|j-0|-1+1+|x-i|+|M-x-j|-1 \geq j+|x-i|+|M-x-j|-1 \geq M-i-1$ internal nodes. We recall that all these nodes have weight $(1-\alpha)/M$. Additionally, the nodes $(a,0)$ and $(0,c)$ have weight $(1-\alpha)/M$ each. The node $(x, M-x)$ on the diagonal has weight at least α by Claim 3.2. Combining these, we get that the total weight of the nodes in $P_1' \cup P_2'$ is at least

$$((i-1)+(M-i-1)+2)\left(\frac{1-\alpha}{M}\right)+\alpha=1.$$

2. Suppose P_2' is a path from $(0,c)$ to a node $(x', M-x')$ on the diagonal. In this case, we will show that the total weight of the nodes in P_1' and P_2' are each lower bounded by 1/2. By Claim 3.1, the axis-parallel path P_1' has at least $|x-a|+|M-x-0|-1 \geq M-x-1$ internal nodes each of which has weight $(1-\alpha)/M$. Additionally, the end-node $(a,0)$ also has weight $(1-\alpha)/M$ and the end-node $(x, M-x)$ has weight at least $1/2-(1-\alpha)((M-x)/M)$ by Claim 3.2. Thus, the total weight of the nodes in P_1' is at least

$$((M-x-1)+1)\left(\frac{1-\alpha}{M}\right)+\frac{1}{2}-\frac{(1-\alpha)(M-x)}{M}=\frac{1}{2}.$$

We proceed by a similar argument for the total weight of the nodes in P_2'. By Claim 3.1, the axis-parallel path P_2' has at least $|x'-0|+|M-x'-c|-1 \geq x'-1$ internal nodes each of which has weight $(1-\alpha)/M$. Additionally, the end-node $(0,c)$ also has weight $(1-\alpha)/M$ and the end-node $(x', M-x')$ has weight at least $1/2-(1-\alpha)(x'/M)$ by Claim 3.2. Thus, the total weight of the nodes in P_2' is at least

$$((x'-1)+1)\left(\frac{1-\alpha}{M}\right)+\frac{1}{2}-\frac{(1-\alpha)x'}{M}=\frac{1}{2}.$$

This completes the proof of Lemma 3.2.

We now prove Lemma 3.3. The proof of Lemma 3.4 is similar.

Proof of Lemma 3.3. Let U_t^* denote an inclusionwise-minimal subset of U^* such that $G-U_t^*$ contains no path from r to t. We will show that U_t^* contains a path P_1 as required. Showing this is equivalent to showing the following combinatorial statement: every subset of nodes that intersects all paths from left to right in a diagonalized-grid (since G contains a diagonalized-grid) has a subset of nodes that induce an axis-parallel path from a node in the topmost row to a node in the bottommost row in G. We proceed to show this now.

We will show the combinatorial statement using a coloring argument. Let

$$R := U_t^* \cup \{(i,-1),(i,M+1) : i \in \{0,1,\ldots,M\}\} \text{ and}$$
$$B := (V' \setminus U_t^*) \cup \{(-1,j),(M+1,j) : j \in \{0,1,\ldots,M\}\}.$$

and call the corresponding nodes as red and blue nodes respectively (we observe that the sets R and B have extra nodes in addition to the nodes in the diagonalized-grid of G, but this is only for the purposes of notational convenience in this proof). We will construct an auxiliary graph for the purposes of the proof—for clarity, we will refer to the vertices of G as *nodes* and the vertices of the auxiliary graph as *vertices*.

We construct an undirected graph H as follows. The vertex set of H is given by $V(H) := \{v_{i,j} : i,j \in \{0,1,\ldots,M+1\}\} \cup \{a,b,c,d\}$. We call a vertex $v_{i,j}$ to be in column i and row j. We define $a' := v_{0,0}, b' := v_{M+1,0}, c' := v_{0,M+1}, d' := v_{M+1,M+1}$ and call them to be the *corner vertices*.

The edge set of H is denoted by $E(H)$: a vertex $v_{i,j}$ is adjacent to all vertices in $V(H) \cap \{v_{i-1,j}, v_{i+1,j}, v_{i,j-1}, v_{i,j+1}\}$ (i.e., the undirected grid edges) and vertices a,b,c,d are adjacent to a',b',c',d' respectively.

We note that H is a plane graph that corresponds to a square grid with four pendant vertices that are adjacent to the four corner vertices of the grid. In the following, it is helpful to consider overlaying H on top of G as shown in Figure 5 with each internal face of H containing exactly one node from V'. For a vertex $v_{i,j} \in V(H) \setminus \{a,a',b,b',c,c',d,d'\}$, we define $D_1^{i,j} := \{(i,j),(i-1,j-1)\}$ and $D_2^{i,j} := \{(i-1,j),(i,j-1)\}$. We emphasize that $D_1^{i,j}$ and $D_2^{i,j}$ consist of nodes from the original graph G. They are the nodes in the two diagonally opposite faces adjacent to $v_{i,j}$ in the overlay (see Figure 5).

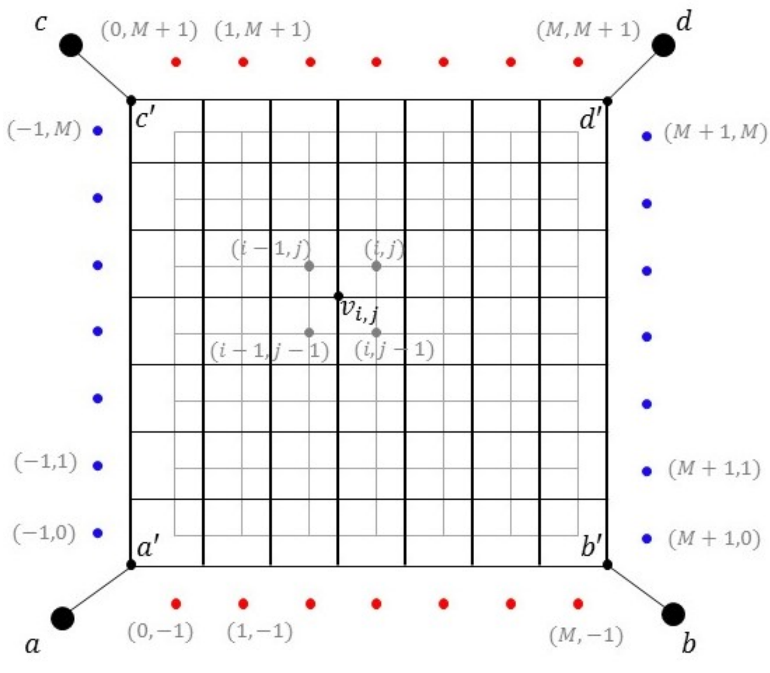

Figure 5: The diagonalized-grid of the graph G is shown in gray while the graph H is shown in black. For visual simplicity, we have not included the diagonal edges of the diagonalized-grid. The extra red and blue nodes are also shown. For node $v_{i,j}$, the nodes of G in the two diagonally opposite faces $D_1^{i,j}$ and $D_2^{i,j}$ are also shown.

We now modify H to obtain a directed subgraph $\mathcal{D}'$ as follows: for an edge $e \in E(H)$ with $e = \{v_{i,j}, v_{i+b_1,j+b_2}\}$ where $b_1, b_2 \in \{0,1\}$ and $b_1 + b_2 = 1$, we say that e is *bi-labeled* if $|\{(i,j),(i+b_1-1, j+b_2-1)\} \cap R| = 1$ and $|\{(i,j),(i+b_1-1,j+b_2-1)\} \cap B| = 1$ (i.e., the two faces of the edge contain a node from R and B). In addition, we will call the edges $\{a,a'\},\{b,b'\},\{c,c'\},\{d,d'\}$ to be *trivially bi-labeled*. We delete all edges of H that are *not* bi-labeled. We orient the trivially bi-labeled edges as $a' \to a$, $b \to b'$, $c \to c'$ and $d' \to d$ and orient the remaining bi-labeled edges by the following rule (see Figure 6):

1. for an edge $e = \{v_{i,j}, v_{i+1,j}\}$, we will orient the edge as
   (a) $v_{i,j} \to v_{i+1,j}$ if $(i,j) \in R$ and $(i,j-1) \in B$ and declare (i,j) to be the *left node* and $(i,j-1)$ to be the *right node* of the edge,
   (b) $v_{i+1,j} \to v_{i,j}$ if $(i,j) \in B$ and $(i,j-1) \in R$ and declare (i,j) to be the right node and $(i,j-1)$ to be the left node of the edge,
2. for an edge $e = \{v_{i,j}, v_{i,j+1}\}$, we will orient the edge as
   (a) $v_{i,j+1} \to v_{i,j}$ if $(i,j) \in R$ and $(i-1,j) \in B$ and declare (i,j) to be the left node and $(i-1,j)$ to be the right node of the edge,
   (b) $v_{i,j} \to v_{i,j+1}$ if $(i,j) \in B$ and $(i-1,j) \in R$ and declare (i,j) to be the right node and $(i-1,j)$ to be the left node of the edge,

We observe that this orienting rule ensures that the left and right nodes of every oriented edge are red and blue respectively (see Figure 6).

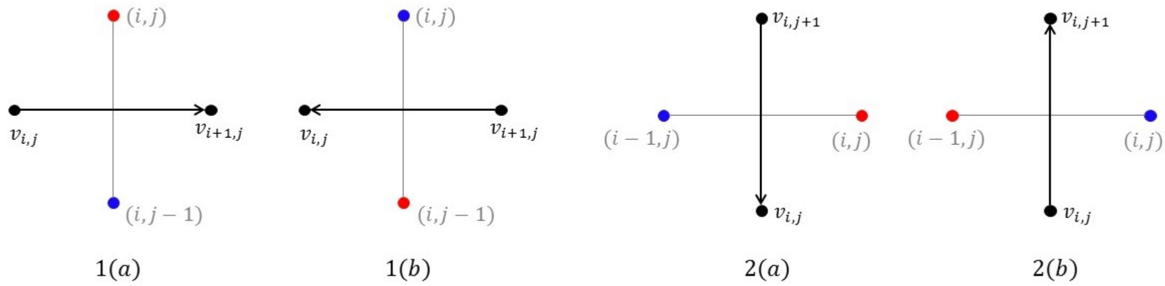

Figure 6: Orienting the bi-labeled edges of H.

We make one final modification to $\mathcal{D}'$ to obtain $\mathcal{D}$: for each vertex $v_{i,j}$ where $i,j \in \{1,\ldots,M\}$,

(I) if $D_1^{i,j} \subseteq B$ and $D_2^{i,j} \subseteq R$, then (see Figure 7) we replace the vertex $v_{i,j}$ by $v_{i,j}^1, v_{i,j}^2$, declare them to be the vertices in row i and column j, and replace the head of the incoming edge from the vertex in column $i-1$, row j by $v_{i,j}^1$, replace the head of the incoming edge from the vertex in column $i+1$, row j by $v_{i,j}^2$, replace the tail of the outgoing edge to the vertex in column i, row $j+1$ by $v_{i,j}^1$, and replace the tail of the outgoing edge to the vertex in column i, row $j-1$ by $v_{i,j}^2$, and

(II) if $D_1^{i,j} \subseteq R$ and $D_2^{i,j} \subseteq B$, then (see Figure 8) we replace the vertex $v_{i,j}$ by $v_{i,j}^1, v_{i,j}^2$, declare them to be the vertices in row i and column j, and replace the head of the incoming edge from the vertex in column i, row $j+1$ by $v_{i,j}^1$, replace the head of the incoming edge from the vertex in column i, row $j-1$ by $v_{i,j}^2$, replace the tail of the outgoing edge to the vertex in column $i+1$, row j by $v_{i,j}^1$, and replace the tail of the outgoing edge to the vertex in column $i-1$, row j by $v_{i,j}^2$.

We call the above operation to be a split operation. We emphasize that the operation separates the red nodes in a consistent manner. The left and right nodes of all oriented edges still remain the same after the split operation.

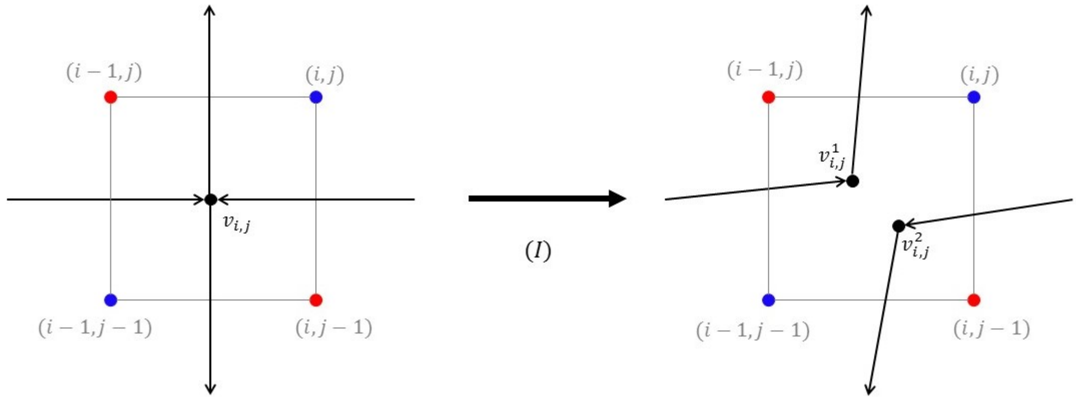

Figure 7: Splitting operation (I).

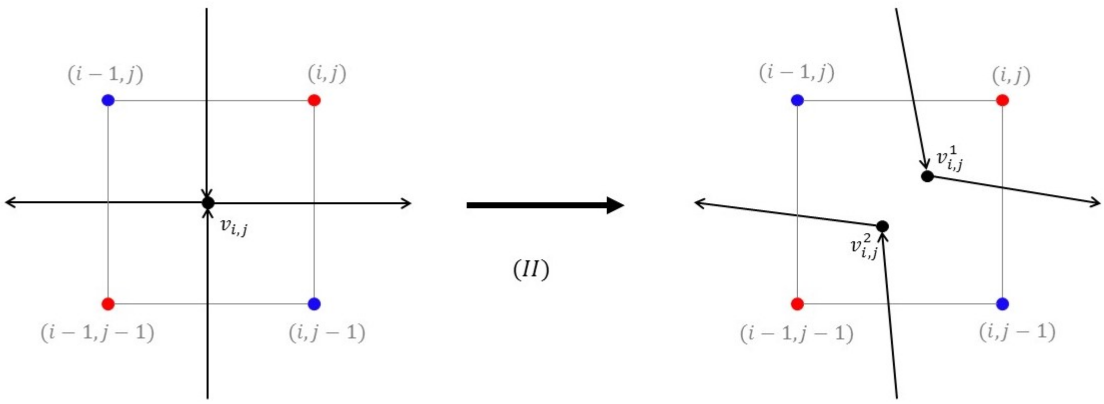

Figure 8: Splitting operation (II).

Claim 3.3. *Let $v \in V(\mathcal{D}) \setminus \{a, b, c, d\}$. Then, the incoming and outgoing degree of v are either both zero or are both 1.*

Proof. Vertices in $\mathcal{D}$ that were obtained by splitting a vertex in $\mathcal{D}'$ clearly satisfy the property since they have incoming and outgoing degree to be 1 after the split. So, we may assume that v is a vertex in $\mathcal{D}'$ as well as $\mathcal{D}$. Suppose v has incoming degree to be one in $\mathcal{D}'$ (the proof for outgoing degree being one is identical).

Suppose v is not a corner vertex. Let $v = v_{i,j}$. Without loss of generality, let the incoming edge be from a vertex in column $i-1$ and row j (see Figure 9). Then, $(i-1, j-1) \in B, (i-1, j) \in R$. Based on whether $(i, j-1)$ is in R or B and whether (i, j) is in R or B, we have four cases. One of the cases cannot happen since $v_{i,j}$ is a vertex in both $\mathcal{D}$ and $\mathcal{D}'$. The remaining three cases show that the outgoing degree from $v_{i,j}$ is also one in $\mathcal{D}'$.

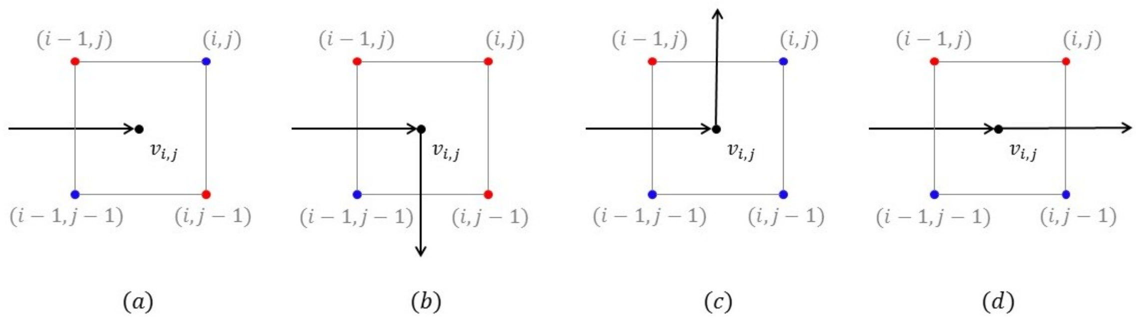

Figure 9: Degree of internal vertices: Case (a) is impossible. Cases (b), (c) and (d) have a unique outgoing edge as well.

Suppose v is a corner vertex. Without loss of generality, let $v = v_{0,M} = c'$ (see Figure 10). Now, depending on whether $(0, M)$ is in R or B, we have two cases. In both cases, the outgoing degree from $v_{0,M}$ is indeed one.

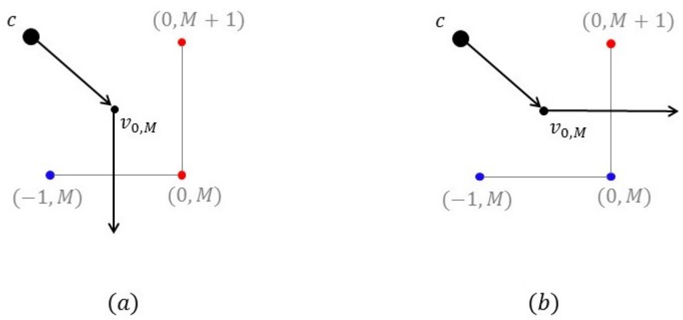

Figure 10: Degree of corner vertices.

□

Thus, the only vertices in $\mathcal{D}$ with outgoing degree 1 and incoming degree 0 are b and c while the only vertices in $\mathcal{D}$ with incoming degree 1 and outgoing degree 0 are a and d. Hence, by Claim 3.3, there exists a path from c to either a or d in $\mathcal{D}$.

Claim 3.4. *Suppose there exists a path from c to d in $\mathcal{D}$. Then there exists a path in $G - U_t^*$ from s to r.*

Proof. Suppose we have a path from c to d in $\mathcal{D}$. Let P denote the nodes of G along the right of the edges in this path. Thus, P induces a path from a node in the leftmost column to a node in the rightmost column in G. We recall that the right nodes along the edges in the path are blue nodes and are indeed not in U_t^*. Thus, we have a path from a node in the leftmost column to a node in the rightmost column in $G - U_t^*$ and hence a path from r to t in $G - U_t^*$. □

Claim 3.4 shows that a path from c to d in $\mathcal{D}$ contradicts the fact that U_t^* is a $r \to t$ cut in G. Thus, we must have a path from c to a in $\mathcal{D}$. Claim 3.5 below completes the proof of the lemma. □

Claim 3.5. *Suppose there exists a path from c to a in $\mathcal{D}$. Then there exists an axis parallel path from a node in the topmost row to a node in the bottommost row in U_t^*.*

Proof. Suppose we have a path Q from c to a in $\mathcal{D}$. Let P denote the nodes of G along the left of the edges in this path. We recall that the left nodes along the edges in the path are red nodes and hence are in U_t^*. Thus, P is a path from a node in the topmost row to a node in the bottommost row in G. It remains to show that P can be transformed into an axis-parallel path.

Suppose P uses a diagonal edge in G. Without loss of generality, let it be $(i-1, j) \to (i, j-1)$. Let Q' be the path Q projected on $\mathcal{D}'$—i.e., use the projected edges in $\mathcal{D}'$. Then, Q' traverses $v_{i-1,j} \to v_{i,j} \to v_{i,j-1}$. These edges imply that $(i-1, j), (i, j-1) \in R$ and $(i-1, j-1) \in B$. If $(i, j) \in B$, then the split operation to obtain $\mathcal{D}$ from $\mathcal{D}'$ shows that the edges in Q do not exist in $\mathcal{D}$, a contradiction (see Figure 11).

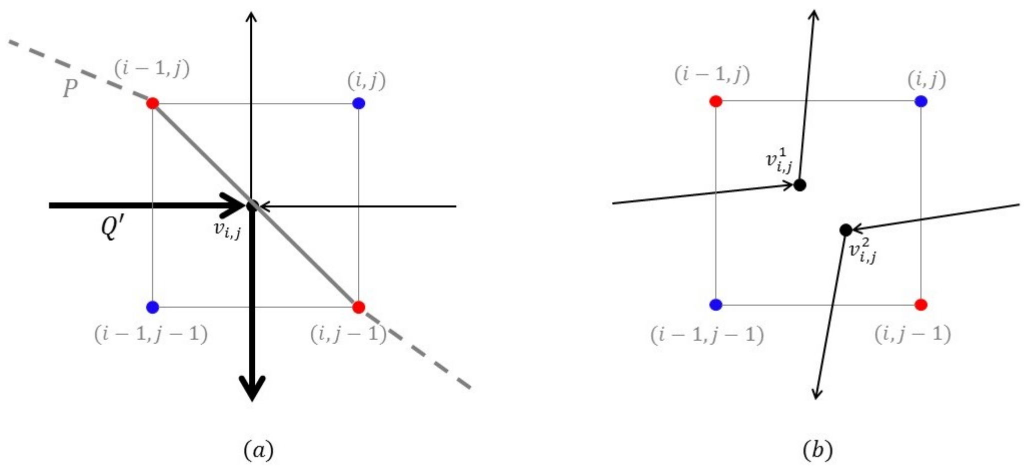

Figure 11: Diagonal path P. Path Q' is impossible owing to the split operation.

Thus, we may assume that $(i, j) \in R$ and is hence in U_t^*. In this case, we can ensure that P makes fewer axis-parallel turns by rerouting as $(i-1, j) \to (i, j) \to (i, j-1)\}$ (see Figure 12). By rerouting this way for each diagonal edge of P, we obtain the required axis-parallel path.

□

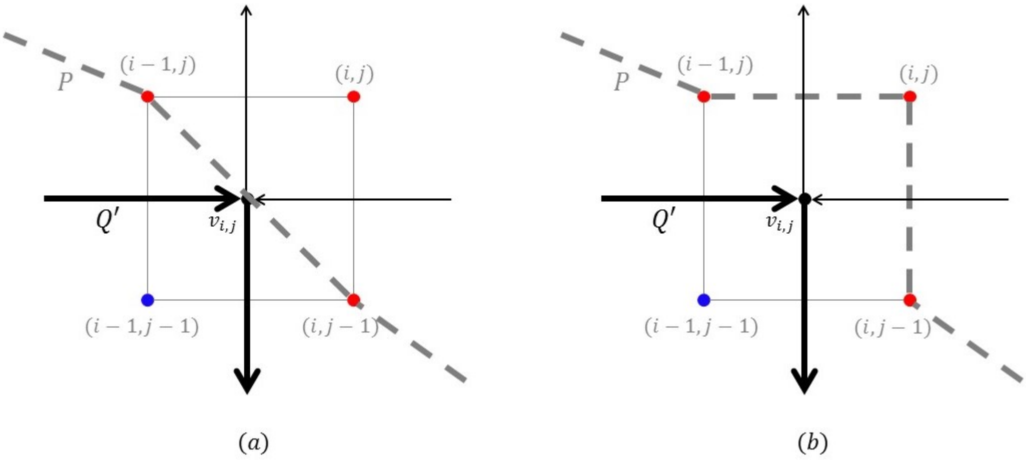

Figure 12: Diagonal path P can be made axis-parallel.

With Lemmas 3.1 and 3.2, we prove the main theorem of the section. We restate it below for convenience.

Theorem 3.1. *The integrality gap of the* Distance-LP *is at least* $\sqrt{2}$.

Proof. We will use the sequence of instances constructed at the beginning of the section. By Lemmas 3.1 and 3.2, it only remains to fix a choice of α and bound the sum of the node weights. We will pick an α that *minimizes* the sum of the node weights in order to get the largest possible integrality gap.

We now compute the sum of the node weights as a function of α. We have the following three claims from the definitions of the node weights.

Claim 3.6.

$$\sum_{i,j\in\{0,\dots,M\}:\ i+j\neq M} w_{ij} = \frac{(1-\alpha)(M+1)}{2}.$$

Claim 3.7.

$$\sum_{\substack{i,j\in\{0,\dots,M\}:\ i+j=M,\\ i<\left(1-\frac{1}{2(1-\alpha)}\right)M \ or \ i>\left(\frac{1}{2(1-\alpha)}\right)M}} w_{ij} = \left(\frac{(1-2\alpha)M}{4(1-\alpha)}\right)\left(\frac{1+2\alpha}{2}+\frac{1-\alpha}{M}\right).$$

Claim 3.8.

$$\sum_{\substack{i,j\in\{0,\dots,M\}:\ i+j=M,\\ \left(1-\frac{1}{2(1-\alpha)}\right)M\leq i\leq\left(\frac{1}{2(1-\alpha)}\right)M}} w_{ij} = \frac{\alpha^2 M}{1-\alpha}.$$

Using the above three claims, we have that

$$\sum_{i=0}^{M}\sum_{j=0}^{M} w_{ij} = \left(\frac{3-4\alpha+2\alpha^2}{4(1-\alpha)}\right)M + 1 - \frac{3\alpha}{2}$$

Now, the minimum value of the function $f(\alpha) := (3 - 4\alpha + 2\alpha^2)/(4(1-\alpha))$ in the domain $(0, 1/2)$ occurs at $\alpha = 1 - 1/\sqrt{2}$ and thus the minimum value of the function is $\min_{\alpha \in (0,1/2)} f(\alpha) = 1/\sqrt{2}$. Using this value of α shows that the objective value of an optimal solution to the linear program is at most $1/\sqrt{2} + \Theta(1/M)$ while the objective value of an optimal integral solution is at least 1. Consequently, the integrality gap of the sequence of instances constructed as above converges to $\sqrt{2}$ when M tends to infinity. $\square$

4 Blocking Arborescences

In this section, we show the equivalence between r-InOut-Node-Blocker and (s, r, t)-Node-Lin-3-Cut. We need the notion of the Strong-Node-Cut problem: the input is a directed graph with node weights, and the goal is to find a minimum weight subset of nodes whose deletion results in at least two disjoint weakly connected components. We observe that Strong-Node-Cut can be solved in polynomial-time.

Theorem 4.1. *There exists an efficient α-approximation algorithm for r-InOut-Node-Blocker if and only if there exists an efficient α-approximation for (s, r, t)-Node-Lin-3-Cut.*

Proof. We first show that r-InOut-Node-Blocker is a combination of (s, r, t)-Node-Lin-3-Cut and Strong-Node-Cut.

Claim 4.1. *For every directed graph $D = (V, E)$ with $r \in V$, the optimal solution to r-InOut-Node-Blocker has value equal to*

$$\min \left\{ \min_{s,t \in V-r} \{(s, r, t)\text{-Node-Lin-3-Cut } \textit{in } D\}, \text{Strong-Node-Cut } \textit{in } D \right\}.$$

Proof. Let U be an optimal solution of r-InOut-Node-Blocker in $D = (V, E)$ with $r \in V$. The optimal values of both (s, r, t)-Node-Lin-3-Cut in D and Strong--Node-Cut in D are upper bounds for the weight of U. If the weight of U is strictly smaller than Strong-Node-Cut, then $D[V - U]$ is weakly connected. By the definition of U, we have that $D[V - U]$ does not contain an in-r-arborescence and hence it has a strongly connected component C_1 not containing r with $\delta^{out}_{D[V-U]}(C_1) = \emptyset$. Similarly, since $D[V - U]$ does not contain an out-r-arborescence, it has a strongly connected component C_2 not containing r with $\delta^{in}_{D[V-U]}(C_2) = \emptyset$. Since $D[V - U]$ is weakly connected, we have $C_1 \neq C_2$. Since C_1 and C_2 are strongly connected components, they are disjoint. For arbitrary nodes $s \in C_1$ and $t \in C_2$, there are no directed paths from s to r, from r to t and from s to t in $D[V - U]$. Thus U is a feasible solution to (s, r, t)-Node-Lin-3-Cut in D. $\square$

Now we turn to the proof of the theorem. The 'if' part follows from Claim 4.1 above. To see the other direction, consider an instance $D = (V, E)$ of (s, r, t)-Node-Lin-3-Cut. Clearly, we may assume that s, r and t have infinite weights. For each node $v \in V$, add an arc from t to v and an arc from v to s. This step does not affect the values of the feasible solutions to (s,r,t)-Node-Lin-3-Cut. Let D' denote the graph thus arising.

We claim that the feasible solutions with finite weight of (s,r,t)-Node-Lin-3-Cut and those of r-InOut-Node-Blocker coincide in D'. Indeed, assume first that U is a solution of (s,r,t)-Node-Lin-3-Cut in D'. As $D'[V-U]$ does not contain a directed path from s to r or from r to t, there exists no in-r-arborescence or out-r-arborescence in $D'[V-U]$, hence U is also a solution of r-InOut-Node-Blocker in D'. Now assume that U is a solution of r-InOut-Node-Blocker in D' with finite weight, that is, $s,t \notin U$. If $D'[V-U]$ contains a directed path from s to r or from r to t or from s to t, then the arcs that were added to D can be used to obtain either an in-r-arborescence or an out-r-arborescence, a contradiction. Hence no such path exists and so U is also a solution of (s,r,t)-Node-Lin-3-Cut in D'.

By the above, an α-approximate solution to r-InOut-Node-Blocker in the extended graph is also an α-approximate solution to (s,r,t)-Node-Lin-3-Cut in D, thus concluding the proof of the theorem. □

The above proof ideas also extend to show that there exists an efficient αapproximation algorithm for r-InOut-Edge-Blocker if and only if there exists an efficient α-approximation for (s,r,t)-Edge-Lin-3-Cut, which is equivalent to (s,r,t)-Node-Lin-3-Cut.

References

[1] K. Bérczi, K. Chandrasekaran, T. Király, E. Lee, and C. Xu, *Global and fixed-terminal cuts in digraphs*, Proceedings of the 20th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (to appear), Preprint: `https://arxiv.org/abs/1612.00156`, APPROX, 2017.

[2] A. Bernáth and G. Pap, *Blocking optimal arborescences*, Mathematical Programming **161** (2017), no. 1, 583–601.

[3] C. Chekuri, S. Kamath, S. Kannan, and P. Viswanath, *Delay-constrained unicast and the triangle-cast problem*, IEEE International Symposium on Information Theory, ISIT, 2015, pp. 804–808.

[4] C. Chekuri and V. Madan, *Approximating multicut and the demand graph*, Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, 2017, pp. 855–874.

[5] A. Ene, J. Vondrák, and Y. Wu, *Local Distribution and the Symmetry Gap: Approximability of Multiway Partitioning Problems*, Proceedings of the Twenty-fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, 2013, pp. 306–325.

[6] R. Erbacher, T. Jaeger, N. Talele, and J. Teutsch, *Directed multicut with linearly ordered terminals*, Preprint: `https://arxiv.org/abs/1407.7498`, 2014.

[7] M. Goemans and D. Williamson, *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming*, Journal of the ACM **42** (1995), no. 6, 1115–1145.

[8] S. Khot, G. Kindler, E. Mossel, and R. O'Donnell, *Optimal Inapproximability Results for MAX-CUT and Other 2-Variable CSPs?*, SIAM Journal on Computing **37** (2007), no. 1, 319–357.

[9] E. Lee, *Improved Hardness for Cut, Interdiction, and Firefighter Problems*, Proceedings of the 44th International Colloquium on Automata, Languages, and Programming (to appear), Preprint: `https://arxiv.org/abs/1607.05133`, ICALP, 2017.

[10] D. Muthukumaran, S. Rueda, N. Talele, H. Vijayakumar, J. Teutsch, and T. Jaeger, *Transforming Commodity Security Policies to Enforce Clark-Wilson Integrity*, Proceedings of the 28th Annual Computer Security Applications Conference, ACSAC, 2012, pp. 269–278.

[11] N. Talele, J. Teutsch, T. Jaeger, and R. Erbacher, *Using security policies to automate placement of network intrusion prevention*, Proceedings of the 5th International Symposium on Engineering Secure Software and Systems, ESSoS, 2013, pp. 17–32.

Appendix A Necessary conditions for (2)

Claim A.1. *Condition* (2) *is satisfied by* ξ *if and only if the following hold for* ζ*:*

$$\zeta\left(\frac{\sqrt{2}-1}{\sqrt{2}}\right) = 0, \tag{11}$$

$$(1-y)\zeta(y) + \int_y^{1-y} \zeta(z)\,\mathrm{d}z = \frac{1}{2} \quad \text{if } \frac{\sqrt{2}-1}{\sqrt{2}} \le y \le \frac{1}{2}, \tag{12}$$

$$(1-y)\zeta(y) - \int_{1-y}^{y} \zeta(z)\,\mathrm{d}z = \frac{1}{2} \quad \text{if } \frac{1}{2} \le y \le \frac{1}{\sqrt{2}}. \tag{13}$$

Proof. The direction showing that (11)-(13) imply (2) was already shown in Claim 2.4. We now argue the necessity of (11), (12) and (13).

To see the necessity of (12), we consider $a = b = y$ for $\frac{\sqrt{2}-1}{\sqrt{2}} \le y \le \frac{1}{2}$. For this choice of a and b, condition (2) necessitates that

$$\begin{aligned}
1 &= 2\int_0^1 \xi(y,z)\,\mathrm{d}z + 2\int_y^1 \xi(z,y)\,\mathrm{d}z \\
&= 2\int_0^{1-y} \xi(y,z)\,\mathrm{d}z + 2\int_y^{1-y} \xi(z,y)\,\mathrm{d}z \\
&= 2\int_0^{1-y} \zeta(y)\,\mathrm{d}z + 2\int_y^{1-y} \zeta(z)\,\mathrm{d}z \\
&= 2(1-y)\zeta(y) + 2\int_y^{1-y} \zeta(z)\,\mathrm{d}z,
\end{aligned}$$

which shows the necessity of (12). The second equation above is because $\xi(y,z) = \xi(z,y) = 0$ for $z > 1-y$ since $y \le 1/2$.

To see the necessity of (13), we consider $a = y$, $b = 1-y$ for some y such that $\frac{1}{2} \le y \le \frac{1}{\sqrt{2}}$. For this choice of a and b, condition (2) necessitates that

$$\begin{aligned}
1 &= \int_0^1 \left(\xi(y,z) + \xi(1-y,z)\right)\,\mathrm{d}z + \int_y^1 \xi(z,1-y)\,\mathrm{d}z + \int_{1-y}^1 \xi(z,y)\,\mathrm{d}z \\
&= \int_0^1 \xi(y,z)\,\mathrm{d}z + \int_0^1 \xi(1-y,z)\,\mathrm{d}z + 0 + 0 \\
&= \int_0^{1-y} \xi(y,z)\,\mathrm{d}z + \int_0^{y} \xi(1-y,z)\,\mathrm{d}z \\
&= \int_0^{1-y} \zeta(y)\,\mathrm{d}z + \int_0^{y} \zeta(1-y)\,\mathrm{d}z \\
&= (1-y)\zeta(y) + y\zeta(1-y). \qquad (14)
\end{aligned}$$

We note that the bounds on y imply that $\frac{\sqrt{2}-1}{\sqrt{2}} \le 1-y \le \frac{1}{2}$. Hence, by (12) applied to $y' := 1-y$, we obtain that

$$y\zeta(1-y) = \frac{1}{2} - \int_{1-y}^{y} \zeta(z)\,\mathrm{d}z. \qquad (15)$$

Substituting (15) in (14) and rewriting in the required form shows the necessity of (13).

To see the necessity of (11), we consider $a = b < \frac{\sqrt{2}-1}{\sqrt{2}}$. For this choice of a and b, condition (2) necessitates that

$$1 = 2\int_0^1 \xi(a,z)\,\mathrm{d}z + 2\int_a^1 \xi(z,a)\,\mathrm{d}z = 0 + 2\int_a^1 \xi(z,a)\,\mathrm{d}z = 2\int_{\frac{\sqrt{2}-1}{\sqrt{2}}}^{\frac{1}{\sqrt{2}}} \zeta(z)\,\mathrm{d}z. \qquad (16)$$

Now, by (12) applied to $y = \frac{\sqrt{2}-1}{\sqrt{2}}$, we obtain that

$$\frac{1}{2} = \frac{1}{\sqrt{2}}\zeta\left(\frac{\sqrt{2}-1}{\sqrt{2}}\right) + \int_{\frac{\sqrt{2}-1}{\sqrt{2}}}^{\frac{1}{\sqrt{2}}} \zeta(z)dz = \frac{1}{\sqrt{2}}\zeta\left(\frac{\sqrt{2}-1}{\sqrt{2}}\right) + \frac{1}{2}$$

where the second equation is obtained by substituting (16). Hence, $\zeta(\frac{\sqrt{2}-1}{\sqrt{2}}) = 0$, showing the necessity of (11).

□

Appendix B Integral of ξ on the unit square

We show that if ξ is defined as in (10), then $\int_0^1 \int_0^1 \xi(z_1, z_2)\,\mathrm{d}z_1\,\mathrm{d}z_2 = \frac{1}{2\sqrt{2}}$. By substituting the function values, we get

$$\begin{aligned}
\int_0^1 \int_0^1 \xi(z_1, z_2)\,\mathrm{d}z_1\,\mathrm{d}z_2 &= \int_{(z_1,z_2)\in\mathcal{R}_1} \xi(z_1, z_2)\,\mathrm{d}z_2\,\mathrm{d}z_1 + \int_{(z_1,z_2)\in\mathcal{R}_2} \xi(z_1, z_2)\,\mathrm{d}z_2\,\mathrm{d}z_1 \\
&= \left(1 - \frac{1}{\sqrt{2}}\right)^2 \left(1 + \frac{1}{\sqrt{2}}\right) + \int_{z_1=\frac{\sqrt{2}-1}{\sqrt{2}}}^{\frac{1}{\sqrt{2}}} \int_{z_2=0}^{1-z_1} \xi(z_1, z_2)\,\mathrm{d}z_2\,\mathrm{d}z_1 \\
&= \left(1 - \frac{1}{\sqrt{2}}\right)^2 \left(1 + \frac{1}{\sqrt{2}}\right) + \int_{\frac{\sqrt{2}-1}{\sqrt{2}}}^{\frac{1}{\sqrt{2}}} (1 - z_1)\zeta(z_1)\,\mathrm{d}z_1 \\
&= \left(1 - \frac{1}{\sqrt{2}}\right)^2 \left(1 + \frac{1}{\sqrt{2}}\right) + \int_{\frac{\sqrt{2}-1}{\sqrt{2}}}^{\frac{1}{\sqrt{2}}} \frac{(1 - z_1)(2z_1(2 - z_1) - 1)}{4z_1(1 - z_1)^2}\,\mathrm{d}z_1 \\
&= \frac{\sqrt{2}-1}{2\sqrt{2}} + \frac{\sqrt{2}-1}{2} = \frac{1}{2\sqrt{2}}.
\end{aligned}$$