Vertex Downgrading to Minimize Connectivity

Hassene Aissi
Paris Dauphine University, France
aissi@lamsade.dauphine.fr

Da Qi Chen
Carnegie Mellon University, Pittsburgh, PA, USA
daqic@andrew.cmu.edu

R. Ravi
Tepper School of Business, Carnegie Mellon University, Pittsburgh, PA, USA
ravi@cmu.edu

Abstract

We consider the problem of interdicting a directed graph by deleting nodes with the goal of minimizing the local edge connectivity of the remaining graph from a given source to a sink. We introduce and study a general downgrading variant of the interdiction problem where the capacity of an arc is a function of the subset of its endpoints that are downgraded, and the goal is to minimize the downgraded capacity of a minimum source-sink cut subject to a node downgrading budget. This models the case when both ends of an arc must be downgraded to remove it, for example. For this generalization, we provide a bicriteria $(4, 4)$-approximation that downgrades nodes with total weight at most 4 times the budget and provides a solution where the downgraded connectivity from the source to the sink is at most 4 times that in an optimal solution. We accomplish this with an LP relaxation and rounding using a ball-growing algorithm based on the LP values. We further generalize the downgrading problem to one where each vertex can be downgraded to one of k levels, and the arc capacities are functions of the pairs of levels to which its ends are downgraded. We generalize our LP rounding to get a $(4k, 4k)$-approximation for this case.

2012 ACM Subject Classification Theory of computation → Routing and network design problems

Keywords and phrases Vertex Interdiction, Vertex Downgrading, Network Interdiction, Approximation Algorithm

Digital Object Identifier 10.4230/LIPIcs.SWAT.2020.5

Related Version https://arxiv.org/pdf/1911.11229.pdf

Funding *Hassene Aissi*: This research benefited from the support of the FMJH Program PGMO and from the support of EDF, Thales, Orange et Criteo.
R. Ravi: This material is based upon research supported in part by the U. S. Office of Naval Research under award number N00014-18-1-2099.

1 Introduction

Interdiction problems arise in evaluating the robustness of infrastructure and networks. For an optimization problem on a graph, the interdiction problem can be formulated as a game consisting of two players: an attacker and a defender. Every edge/vertex of the graph has an associated interdiction cost and the attacker interdicts the network by modifying the edges/vertices subject to a budget constraint. The defender solves the problem on the modified graph. The goal of the attacker is to hamper the defender as much as possible. Ford and Fulkerson initiated the study of interdiction problems with the maximum flow/minimum cut theorem [4, 10, 15]. Other examples of interdiction problems include matchings [17], minimum spanning trees [12, 20], shortest paths [7, 11], st-flows [14, 16, 18] and global minimum cuts [19, 3].

17th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2020).
Editor: Susanne Albers; Article No. 5; pp. 5:1–5:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Most of the interdiction literature today involves the interdiction of edges while the study of interdicting vertices has received less attention (e.g.[17, 18]). The various applications for these interdiction problems, including drug interdiction, hospital infection control, and protecting electrical grids or other military installations against terrorist attacks, all naturally motivate the study of the vertex interdiction variant. In this paper, we focus on vertex interdiction problems related to the minimum st-cut (which is equal to the maximum st-flow and hence also termed network flow interdiction or network interdiction in the literature).

For st-cut vertex interdiction problems, the set up is as follows. Consider a directed graph $G = (V(G), A(G))$ with n vertices, m arcs, an arc cost function $c : A(G) \to \mathbb{N}$, and an interdiction cost function $r : V(G) \setminus \{s,t\} \to \mathbb{N}$ defined on the set of vertices $V(G) \setminus \{s,t\}$. A set of arcs $F \subseteq A(G)$ is an st-cut if $G \backslash F$ no longer contains a directed path from s to t. Define the cost of F as $c(F) = \Sigma_{e \in F} c(e)$. For any subset of vertices $X \subseteq V(G) \setminus \{s,t\}$, we denote its interdiction cost by $r(X) = \sum_{v \in X} r(v)$. Let $\lambda_{st}(G \backslash X)$ denote the cost of a minimum st cut in the graph $G \backslash X$.

▶ **Problem 1.** ***Weighted Network Vertex Interdiction Problem (WNVIP) and its special cases.*** *Given two specific vertices s (source) and t (sink) in $V(G)$ and interdiction budget $b \in \mathbb{N}$, the Weighted Network Vertex Interdiction Problem* ***(WNVIP)*** *asks to find an interdicting vertex set $X^* \subseteq V(G) \setminus \{s,t\}$ such that $\sum_{v \in X^*} r(v) \leq b$ and $\lambda_{st}(G \backslash X^*)$ is minimum. The special case of WNVIP where all the interdiction costs are one will be termed* ***NVIP****, while the further special case when even the arc costs are one will be termed* ***NVIP with unit costs****.*

In this paper, we define and study a generalization of the network flow interdiction problem in digraphs that we call **vertex downgrading**. Since interdicting vertices can be viewed as attacking a network at its vertices, it is natural to consider a variant where attacking a node does not destroy it completely but partially weakens its structural integrity. In terms of minimum st-cuts, one interpretation could be that whenever a vertex is interdicted, instead of removing it from the network we partially reduce the cost of its incident arcs. In this context, we say that a vertex is *downgraded*. Specifically, consider a directed graph $G = (V(G), A(G))$ and a downgrading cost $r : V(G) \setminus \{s,t\} \to \mathbb{N}$. For every arc $e = uv \in A(G)$, there exist four associated nonegative costs $c_e, c_{eu}, c_{ev}, c_{euv}$, respectively representing the cost of arc e if 1) neither $\{u, v\}$ are downgraded, 2) only u is downgraded, 3) only v is downgraded, and 4) both $\{u, v\}$ are downgraded. Note that these cost functions are independent of each other so downgrading vertex v might affect each of its incident arcs differently. However, we do impose the following conditions: $c_e \geq c_{eu} \geq c_{euv}$ and $c_e \geq c_{ev} \geq c_{euv}$. These inequalities are natural to impose since the more endpoints of an arc are downgraded, the lower the resulting arc should cost. Given a downgrading set $Y \subseteq V(G) \setminus \{s,t\}$, define $c^Y : A(G) \to \mathbb{R}_+$ to be the arc cost function representing the cost of cutting e after downgrading Y.

	$u, v \notin Y$	$u \in Y, v \notin Y$	$u \notin Y, v \in Y$	$u, v \in Y$
$c^Y(e) =$	c_e	c_{eu}	c_{ev}	c_{euv}

Given a set of arcs $F \subseteq A(G)$, we define $c^Y(F) = \Sigma_{e \in F} c^Y(e)$.

▶ **Problem 2.** ***Network Vertex Downgrading Problem (NVDP).*** *Let $G = (V(G), A(G))$ be a directed graph with a source s and a sink t. For every arc $e = uv$, we are given non-negative costs $c_e, c_{eu}, c_{ev}, c_{euv}$ as defined above. Given a (downgrading) budget b, find a set $Y \subseteq V(G) \setminus \{s,t\}$ and an st-cut $F \subseteq A(G)$ such that $\Sigma_{v \in Y} r(v) \leq b$ and minimizes $c^Y(F)$.*

While it is not immediately obvious as it is for WNVIP, we can still show that detecting a zero solution for NVDP is easy.

▶ **Theorem 1.** *Given an instance of NVDP on graph G with budget b, there exists a polynomial time algorithm to determine if there exists $Y \subseteq V(G)$ and an st-cut $F \subseteq A(G)$ such that $\Sigma_{v \in Y} r(v) \leq b$ and $c^Y(F) = 0$.*

First we present some useful reductions between the above problems.

1. In the NFI (Network Flow Interdiction) problem defined in [4], the given graph is undirected instead of directed and the adversary interdicts edges instead of vertices. The goal is to minimize the cost of the minimum *st*-cut after interdiction. NFI can be reduced to the undirected version of WNVIP (where the underlying graph is undirected). Simply subdivide every undirected edge $e = uv$ with a vertex v_e. The interdiction cost of v_e remains the same as the interdiction cost of e while all original vertices have an interdiction cost of ∞ (or a very large number). The cut cost of the edges $uv_e, v_e v$ are equal to the original cost of cutting the edge e.
2. The undirected version of WNVIP can be reduced to the (directed) WNVIP by replacing every edge with two parallel arcs going in opposite directions. Each new arc has the same cut cost as the original edge.
3. WNVIP is a special case of NVDP with costs $c_{eu} = c_{ev} = c_{euv} = 0$ for all $e = uv$.

The first two observations above imply that any hardness result for NFI in [4] also applies to WNVIP. Based on the second observation, we prove our hardness results for the (more specific) undirected version of WNVIP. As a consequence of the third observation, all of these hardness results also carry over to the more general NVDP.

Our work also studies the following further generalization of NVDP. Every vertex has k possible levels that it can be downgraded to by paying different downgrading costs. Every arc has a cutting cost depending on what level its endpoints were downgraded to. More precisely, for each level $0 \leq i, j \leq k$, let $r_i(v)$ be the interdiction cost to downgrade v to level i and let $c_{i,j}(e)$ be the cost of cutting arc $e = uv$ if u, v were downgraded to levels i, j respectively. We assume that $0 = r_0(v) \leq r_1(v) \leq ... \leq r_k(v)$ since higher levels of downgrading should cost more and $c_{i,j}(e) \geq c_{i',j'}(e)$ if $i \leq i', j \leq j'$ since the more one downgrades, the easier it is to cut the incident arcs. Then, given a map $L : V(G) \to \{0, ..., k\}$, representing which level to downgrade each vertex to, one can talk about the cost of performing this downgrading: $r^L := \Sigma_{v \in V(G)} r_{L(v)}(v)$, and the cost of a cut F after downgrading according to L: $c^L(F) := \Sigma_{uv \in F} c_{L(u),L(v)}(uv)$. Now, we can formally define the most general problem we address.

▶ **Problem 3.** ***Network Vertex Leveling Downgrading Problem (NVLDP).*** *Let $G = (V(G), A(G))$ be a directed graph with a source s and a sink t. For every vertex v and $0 \leq i \leq k$, we have non-negative downgrading costs $r_i(v)$. For every arc $e = uv$ and levels $0 \leq i, j \leq k$, we are given non-negative cut costs $c_{i,j}(e)$. Given a (downgrading) budget b, find a map $L : V(G) \to \{0, ..., k\}$ and an st-cut $F \subseteq A(G)$ such that $r^L \leq b$ and minimizes $c^L(F)$.*

Note that when $k = 1$ we have NVDP.

Related Works

▶ **Definition 2.** *An (α, β)* ***bicriteria approximation*** *for the interdiction (or downgrading) problem returns a solution that violates the interdiction budget b by a factor of at most β and provides a final cut (in the interdicted graph) with cost at most α times the optimal cost of a minimum cut in a solution of interdiction budget at most b.*

Chestnut and Zenklusen [4] study the network flow interdiction problem (NFI), which is the undirected and edge interdiction version of WNVIP. NFI is also known to be essentially equivalent to the budgeted minimum st cut problem [13]. NFI is also a recasting of the k-route st-cut problem [5, 9], where a minimum cost set of edges must be deleted to reduce the node or edge connectivity between s and t to be k. The results of Chestnut and Zenklusen, and Chuzhoy et al. [5] show that an $(\alpha, 1)$-approximation for WNVIP implies a $2(\alpha)^2$-approximation for the notorious Densest k-Subgraph (DkS) problem. The results of Chuzhoy et al. [5] (Theorem 1.9 and Appendix section B) also imply such a hardness for NVIP even with unit edge costs. For the directed version, WNVIP is equivalent to directed NFI (by subdividing arcs or splitting vertices). As noted in [18], there is a symmetry between the interdicting cost and the capacity on each arc and thus it is also hard to obtain a $(1, \beta)$-approximation for WNVP. Furthermore, Chuzhoy et al. [5] also show that there is no $(C, 1+\gamma_C)$-bi-criteria approximation for WNVIP assuming Feige's Random κ-AND Hypothesis (for every C and sufficiently small constant γ_C). For example, under this hypothesis, they show hardness of $(\frac{11}{10} - \epsilon, \frac{25}{24} - \epsilon)$ approximation for WNVIP.

Chestnut and Zenklusen give a $2(n-1)$ approximation algorithm for NFI for any graph with n vertices. In the special case where the graph is planar, Philips [14] gave an FPTAS and Zenklusen [18] extended it to handle the vertex interdiction case.

Burch *et al.* [2] give a $(1+\varepsilon, 1), (1, 1+\frac{1}{\varepsilon})$ pseudo-approximation algorithm for NFI. Given any $\varepsilon > 0$, this algorithm returns either a $(1+\epsilon)$-approximation, or a solution violating the budget by a factor of $1 + \frac{1}{\epsilon}$ but has a cut no more expensive than the optimal cost. However, we do not know which case occurs *a priori*. In this line of work, Chestnut and Zenklusen [3] have extended the technique of Burch et al. to derive pseudo-approximation algorithms for a larger class of NFI problems that have good LP descriptions (such as duals that are box-TDI). Chuzhoy et al. [5] provide an alternate proof of this result by subdividing edges with nodes of appropriate costs.

Our Contributions

1. We define and initiate the study of multi-level node downgrading problems by defining the Network Vertex Leveling Downgrading Problem (NVLDP) and provide the first results for it. This problem extends the study in [18] of the vertex interdiction problem so as to consider a richer set of interdiction functions.
2. For the downgrading variant NVDP, we show that the problem of detecting whether there exists a downgrading set that gives a zero cost cut can be solved in polynomial time. (Section 2)
3. We design a new LP rounding approximation algorithm that provides a $(4, 4)$-approximation to NVDP. We use a carefully constructed auxiliary graph so that the level-cut algorithm based on ball growing for showing integrality of st-cuts in digraphs (See. e.g. [6]) can be adapted to choose nodes to downgrade and arcs to cut based on the LP solution. (Section 3)
4. For the most general version NVLDP with k levels of downgrading each vertex and k^2 possible downgraded costs of cutting an edge, we generalize the LP rounding method for NVDP to give a $(4k, 4k)$-approximation. The direct extension of the NVDP rounding to this case only gives an $O(k^2)$ approximation. However, we exploit the sparsity properties of a vertex optimal solution to our LP formulation to improve this guarantee to match that for the case of $k = 1$. Details are in the full version [1].
5. As noted before, many previous works showed hardness in obtaining a unicriterion approximation for WNVIP, which motivates the focus on finding bicriteria approximation results. We push the hardness result further to show that it is also "DkS hard" to obtain a

$(1, \beta)$-approximation for NVIP and NVIP with unit costs even in *undirected* graphs. Note that this is in sharp contrast to the edge interdiction case. NFI in undirected graphs with unitary interdiction cost and unitary cut cost can be solved by first finding a minimum cut and then interdicting b edges in that cut [19]. Details are in the full version [1].

6. Burch *et al.* [2] gave a polynomial time algorithm that finds a $(1 + 1/\epsilon, 1)$ or $(1, 1 + \epsilon)$-approximation for any $\epsilon > 0$ for WNVIP in digraphs. This was reproved more directly by Chuzhoy et al [5] by converting both interdiction and arc costs into costs on nodes. We show that this strategy can also be extended to give a simple $(4, 4(1 + \epsilon))$-bicriteria approximation for the multiway cut generalization in directed graphs and a $(2(1 + \epsilon) \ln k, 2(1 + \epsilon) \ln k)$-approximation for the multicut vertex interdiction problem in undirected graphs, for any $\epsilon > 0$, where k is the number of terminal nodes in the multicut problem. Details are in the full version [1].

2 Detecting Zero in NVDP in Polynomial Time

In this section, we provide an algorithm to detect, in a given instance of NVDP, whether there exists nodes to downgrade such that the downgrading cost is less than the budget and the min cut after downgrading is zero, and hence prove Theorem 1.

In order to demonstrate the main idea of the proof, we first work on a special case of NVDP. Suppose for every arc $e = uv$, $c_e = c_{eu} = c_{ev} = 1$ and $c_{euv} = 0$. In other words, every arc is unit cost and requires the downgrading of both ends in order to reduce the cost down to zero. For every vertex $v \in V(G)$, we assume the interdiction cost $r(v) = 1$. We call this the **Double-Downgrading Network Problem** (DDNP). We first prove the following.

▶ **Lemma 3.** *Given an instance of DDNP on graph G with budget b, there exist a polynomial time algorithm to determine if there exists $Y \subseteq V(G)$ and an st-cut $F \subseteq A(G)$ such that $|Y| \leq b$ and $c^Y(F) = 0$.*

Proof. Let $X \subseteq V(G)$ be a minimum set of vertices to downgrade such that the resulting graph contains a cut of zero cost. Let F be the set of arcs in the graph induced by X (i.e., with both ends in X). Note that F are the only arcs with cost zero and hence F is an arc cut in G. Furthermore, since X is optimal, X is the set of vertices incident to F (there are no isolated vertices in the graph induced by X). Let V_s, V_t be the set of vertices in the component of $G \backslash F$ that contains s, t respectively.

Consider the graph G^2 where we add arc uv to G if there exists $w \in V(G)$ such that $uv, vw \in A(G)$. First we claim that X is a vertex cut in G^2. Suppose there is an st path in $G^2 \setminus X$ where the first arc crossing over from V_s to V_t is uv. Note that any such $u \in V_s \setminus X$ and $v \in V_t \setminus X$ are distance 3 apart and hence do not have an arc between them in G^2, a contradiction.

Given any vertex cut Y in G^2, we claim that downgrading Y in G creates an st-cut of zero cost, by deleting the arcs induced by Y from G. Suppose for a contradiction there is an st-path after downgrading Y and deleting the zero-cost arcs induced by Y. Then the path cannot have two consecutive nodes in Y. Let $y \in Y$ be a single node along the path with neighbors $y^-, y^+ \notin Y$. Note that $(y^-, y^+) \in G^2$, and shortcutting over all such single node occurrences from Y in the path gives us an st-path in $G^2 \setminus Y$, a contradiction.

This proves that a minimum size downgrading vertex set Y in G whose downgrading produces a zero-cost st-cut is also a minimum vertex-cut in G^2. Then, one can check if a zero-cut solution exists with budget b for DDNP by simply checking if the minimum vertex-cut in G^2 is at most b. ◀

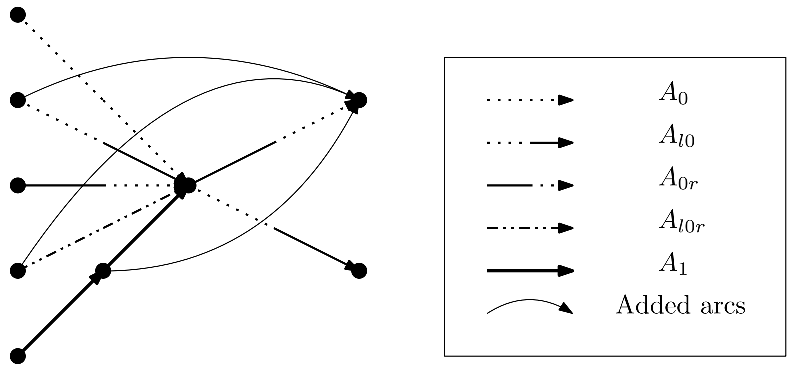

Figure 1 Example of Added Arcs in H.

Now, to prove Theorem 1, we have to slightly modify the graph G and the construction of G^2 in order to adapt to the various costs. Our goal is still to look for a minimum vertex cut in an auxiliary graph using $r(v)$ as vertex cost.

Proof. Given an instance of NVDP on G with a budget b, vertex downgrading costs $r(v)$ and arc costs $c_e, c_{eu}, c_{ev}, c_{eu}$, consider the following auxiliary graph H. First, we delete any arc e where $c_e = 0$ since they are free to cut anyways. For every arc $e = uv$ where $c_{euv} > 0$, subdivide e with a vertex t_e and let $r(t_e) = \infty$. In some sense, since $c_e, c_{eu}, c_{ev} \geq c_{euv} > 0$, downgrading u, v cannot reduce the cost of e to zero. Then, we should never be allowed to touch the vertex t_e. Let T be the set of all newly-added subdivided vertices. To finish constructing H, our next step is to properly simulate H^2.

We classify arcs into five types based on which of its costs are zero. Note that we no longer have any arcs where $c_e = 0$. Let $A_0 := \{e = uv : c_{eu} = c_{ev} = c_{euv} = 0\}$, the arcs where downgrading either ends reduce its cost to zero. Let $A_{l0} := \{e = uv : c_{eu} = c_{euv} = 0, c_{ev} > 0\}, A_{0r} := \{e = uv : c_{ev} = c_{euv} = 0, c_{eu} > 0\}, A_{l0r} := \{e = uv : c_{euv} = 0, c_{eu}, c_{ev} > 0\}$ respectively represent arcs that require the downgrading of its left tail, its right head, or both in order to reduce its cost. Let A_1 be all remaining arcs, those incident to the newly subdivided vertex t_e. Now, for every path uvw of length two, we consider adding the arc uw based on the following rules (see Figure 1 for example of newly added arcs):

If $v \notin T$					
Add uw?	$vw \in$				
$uv \in$	A_0	A_{l0}	A_{0r}	A_{l0r}	A_1
A_0	No	No	No	No	No
A_{l0}	No	No	Yes	Yes	Yes
A_{0r}	No	No	No	No	No
A_{l0r}	No	No	Yes	Yes	Yes
A_1	No	No	Yes	Yes	Yes
If $v = t_e \in T$, do not add uw					

The idea is similar to the proof for DDNP. If $uv, vw \in A_{l0r}$, downgrading v is not enough to cut uv, vw for free. Thus we add arc uw to keep the connectivity. If $uv \in A_{0r}$, then downgrading v should reduce the cost of uv to 0. Thus, we do not want to bypass v by adding an arc uw. If $v = t_e \in T$, since $r(v)$ has high cost, we never cut it so we do not need to strengthen the connectivity by adding arcs uw.

Let (X, F) be a solution to NVDP where $\Sigma_{v\in X} r(v)$ is minimum, F is an st-cut and $c^X(F) = 0$. Let V_s be all vertices connected to s in $G\backslash F$. We claim that X is a vertex cut in H. Suppose not and there exists an st-path in H and let uv be the first arc of the path leaves V_s. If $v = t_e \in T$, then arc $e \in F$, contradicting $c^X(F) = 0$. If $uv \in A(G)$, then $uv \in F$. Since $u, v \notin X$, $c^X(uv) > 0$, a contradiction. If uv is a newly added arc, then there exist $v' \in V(G)$ such that $uv'v$ is a path in G. By definition, $V_s \cap T = \emptyset$ so $u, v \notin T$. Then, there are only four cases where we add arc uw to create H. In all cases, downgrading v' does not reduce the cost of $uv'', v'v$ to 0. Since at least one of $uv', v'v \in F$, it contradicts $c^X(F) = 0$.

Given a minimum vertex cut Y in H, we claim downgrading Y in G creates an st-cut of zero cost. Note that $Y \cap T = \emptyset$ since any vertex in T is too expensive to cut. Suppose for a contradiction there is an st-path P that does not cross an arc with cost 0 after downgrading Y. Let P' be the corresponding path in H. If P contains two consecutive vertices $u, v \in Y$, then $c_{euv} > 0$ and it would have been subdivided. This implies there are no consecutive vertices of Y in P'. Let uvw be a segment of P' where $v \in Y$. Since downgrading v does not reduce its incident arcs to a cost of 0, it follows that $uv \in A_{l0} \cup A_{l0r} \cup A_1$ and $vw \in A_{0r} \cup A_{l0r} \cup A_1$. Then it follows that $uw \in A(H)$. Then, every vertex $v \in Y \cap V(P')$ can be bypassed, a contradiction.

This implies a minimum vertex cut in H is a downgrading set that creates a zero-cost cut in G. Then, by checking the min-vertex cut cost of H, we can determine whether a zero-solution exists for G with budget b. ◀

3 Approximating Network Vertex Downgrading Problem (NVDP)

As an introduction and motivation to the LP model and techniques used to solve NVLDP, in this section, we focus on the special case NVDP, where there is only one other level to downgrade each vertex to. Our main goal is to show the following theorem.

▶ **Theorem 4.** *There exists a polynomial time algorithm that provides a $(4, 4)$-approximation to NVDP on an n-node digraph.*

3.1 LP Relaxation and Rounding

LP Model for Minimum st-cut. To formulate the NVDP as a LP, we begin with the following standard formulation of minimum st-cuts [8].

$$
\begin{aligned}
\min \quad & \sum_{e\in A(G)} c(e)x_e && \\
\text{s.t.} \quad & d_v \le d_u + x_{uv} && \forall uv \in A(G) \qquad (1)\\
& d_s = 0, d_t \ge 1 && \\
& x_{uv} \ge 0 && \forall uv \in A(G) \qquad (2)
\end{aligned}
$$

An integer solution of this problem can be seen as setting d to be 0 for nodes in the s shore and 1 for nodes in the t shore of the cut. Constraints (1) then insist that the x-value for arcs crossing the cut to be set to 1. The potential d_v at node v can also be interpreted as a distance label starting from s and using the nonnegative values x_{uv} as distances on the arcs. Any optimal solution of the above LP can be rounded to an optimal integer solution of no greater value by using the x-values on the arcs as lengths, growing a ball around s, and cutting it at a random threshold between 0 and the distance to t (which is 1 in this case). The expected cost of the random cut can be shown to be the LP value (See e.g., [6]), and the minimum such ball can be found efficiently using Dijkstra's algorithm. Our goal in this section is to generalize this formulation and ball-growing method to NVDP.

One difficulty in NVDP comes from the fact that every arc has four associated costs and we need to write an objective function that correctly captures the final cost of a chosen cut. One way to overcome this issue is to have a distinct arc associated with each cost. In other words, for every original arc $uv \in A(G)$, we create four new arcs $[uv]_0, [uv]_1, [uv]_2, [uv]_3$ with cost $c_e, c_{eu}, c_{euv}, c_{ev}$ respectively. Then, every arc has its unique cost and it is now easier to characterize the final cost of a cut. We consider the following auxiliary graph. See Figure 2.

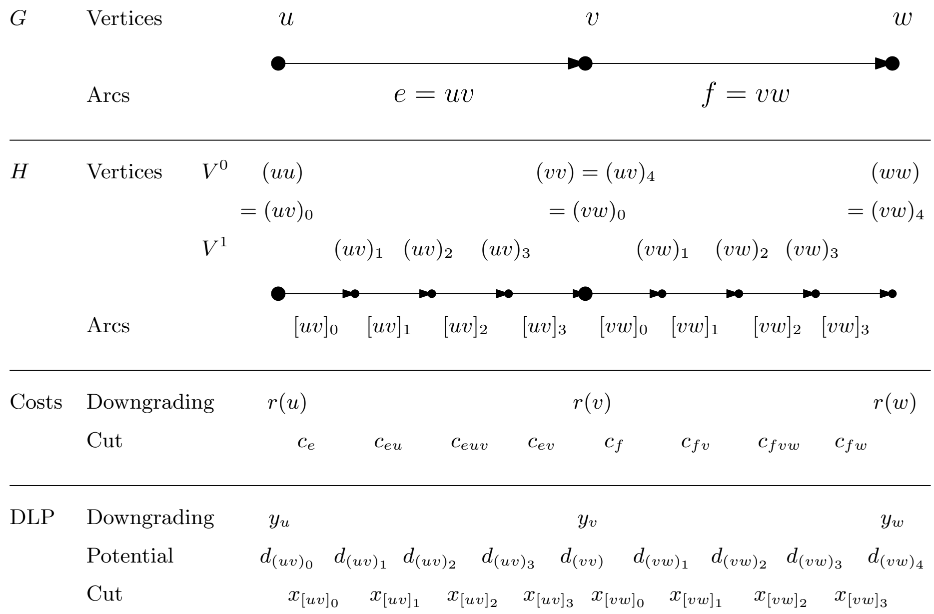

Figure 2 Construction of the auxiliary graph H.

Constructing the Auxiliary Graph H

Let $V(H) = V^0(H) \cup V^1(H)$ where $V^0(H) = \{(vv) : v \in V(G)\}$ and $V^1(H) = \{(uv)_i : uv \in A(G), i = 1, 2, 3\}$. Define $A(H) = \{[uv]_0 = (uu)(uv)_1, [uv]_1 = (uv)_1(uv)_2, [uv]_2 = (uv)_2(uv)_3, [uv]_3 = (uv)_3(vv) : uv \in A(G)\}$. Essentially, the vertices $(uu) \in V^0(H)$ correspond to the original vertices $u \in V(G)$ and for every arc $uv \in A(G)$, we replace it with a path $(uu)(uv)_1(uv)_2(uv)_3(vv)$ where the four arcs on the path are $[uv]_0, [uv]_1, [uv]_2, [uv]_3$. For convenience and consistency in notation, we define $(uv)_0 := (uu), (uv)_4 := (vv)$. Note that the vertices of H will always be denoted as two lowercase letters in parenthesis while arcs in H will be two lowercase letters in square brackets with subscript $i = 0, 1, 2, 3$. The cost function $c : A(H) \to \mathbb{R}_{\geq 0}$ is as follows: $c([uv]_0) = c_e, c([uv]_1) = c_{eu}, c([uv]_2) = c_{euv}, c([uv]_3) = c_{ev}$. Since we can only downgrade vertices in V^0, to simplify the notation, we retain $r(v)$ as the cost to downgrade vertex $(vv) \in V^0$. Note that $|V(H)| = 3|A(G)| + |V(G)| = O(n + m)$.

Downgrading LP

Given the auxiliary graph H, we can now construct an LP similar to the one for *st*-cuts. For vertices $(vv) \in V^0(H)$ corresponding to original vertices of G, we define a downgrading variable y_v representing whether vertex v is downgraded or not in G. For every arc $[uv]_i \in A(H)$, we have a cut variable $x_{[uv]_i}$ to indicate if the arc belongs in the final cut of the graph. Lastly for all vertices $(uv)_i \in V(H)$, we have a potential variable $d_{(uv)_i}$ representing its distance from the source (ss).

The idea is to construct an LP that forces s, t to be at least distance 1 apart from each other as before. This distance can only be contributed from the arc variables $x_{[uv]_i}$. The downgrading variables y_v imposes limits on how large these distances $x_{[uv]_i}$ of some of its incident arcs can be. The motivation is that the larger y_u and y_v are, the more we should allow arc $[uv]_2$ to appear in the final cut over the other arcs $[uv]_0, [uv]_1, [uv]_3$ in order to incur the cheaper cost of c_{euv}. We consider the following downgrading LP henceforth called DLP.

Figure 2 includes the list of variables associated with H. In the LP, our objective is to minimize the cost of the final cut. Constraint (3) corresponds to the budget constraint for the downgrading variables. Constraint (4) is analogous to Constraint (1) in the LP for min-cuts.

Constraint (5) relates cut and downgrade variables. If we do not consider any constraint related to downgrading variables for a moment, the LP will naturally always want to choose the cheapest arc $[uv]_2$ over $[uv]_0, [uv]_1, [uv]_3$ when cutting somewhere between (uu) and (vv). However, the cut should not be allowed to go through $[uv]_2$ if one of u, v is not downgraded. In other words $x_{[uv]_2}$ should be at most the minimum of y_u, y_v. This reasoning gives the constraint $x_{[uv]_2}, x_{[uv]_3}, x_{[vw]_1}$ and, $x_{[vw]_2}$ all need to be $\leq y_v$ for in-arcs uv and out-arcs vw. Now consider an arc $f = vw \in E(G)$. In an integral solution, if v is downgraded, the arc vw incurs a cost of either c_{fv} or c_{fvw} but not both, since v must lie on one side of the cut. This translates to a LP solution where only one of the arcs $[vw]_1, [vw]_2$ is in the final cut. Thus, a better constraint to impose is $x_{[vw]_1} + x_{[vw]_2} \leq y_v$. We can also similarly insist that $x_{[uv]_2} + x_{[uv]_3} \leq y_v$ for in-arcs uv. To push this even further, consider a path uvw in G. In an integral solution, at most one of the arcs uv, vw appears in the final cut. This implies that if v is downgraded, then only one of the costs $c_{ev}, c_{euv}, c_{fv}, c_{fvw}$ is incurred. This corresponds to the tighter constraint (5). Note that for every vertex $v \in V(G)$, for every pair of incoming and outgoing arcs of v, we need to add one such constraint. Then for every vertex in G, we potentially have to add up to n^2 many constraints. In total, the number of constraints would still only be $O(n^3)$. The last few constraints in DLP make sure s and t are 1 distance apart and cannot themselves be downgraded. The final LP relaxation is given below.

$$
\begin{aligned}
\min \quad & \sum_{[uv]_i \in A(H)} c([uv]_i) x_{[uv]_i} & & \\
\text{s.t.} \quad & \sum_{(vv) \in V^0(H)} r(v) y_v \leq b & & (3)\\
& d_{(uv)_{i+1}} \leq d_{(uv)_i} + x_{[uv]_i} & \forall \text{arc } [uv]_i, 0 \leq i \leq 3 \quad & (4)\\
& x_{[uv]_2} + x_{[uv]_3} + x_{[vw]_1} + x_{[vw]_2} \leq y_v & \forall \text{ path } (uv)_3(vv)(vw)_1 \quad & (5)\\
& d_{(ss)} = 0, d_{(tt)} = 1, y_s = 0, y_t = 0 & &
\end{aligned}
$$

The following lemmas shows the validity of our defined DLP for NVDP.

▶ **Lemma 5.** *An optimal solution to NVDP provides a feasible integral solution to DLP with the same cost.*

Proof. Given a digraph G with cost functions $c_e, c_{eu}, c_{ev}, c_{euv}$, a source s and a sink t, let $Y \subseteq V(G), F \subseteq A(G)$ be an optimal solution to NVDP where $r(Y) \leq b, F$ is an st-cut and $c^Y(F)$ is minimum. Then, a feasible solution (x, y, d) to DLP on the graph H can be constructed as follows:

- For the cut variables x, let
  - $x_{[uv]_0} = 1$ if $uv \in F$ and $u, v \notin Y$, 0 otherwise,
  - $x_{[uv]_1} = 1$ if $uv \in F$ and $u \in Y, v \notin Y$, 0 otherwise,

- $x_{[uv]_2} = 1$ if $uv \in F, u, v \in Y$, 0 otherwise,
- $x_{[uv]_3} = 1$ if $uv \in F, u \notin Y, v \in Y$, 0 otherwise.

- For the downgrading variables y, let $y_u = 1$ if $u \in Y$, 0 otherwise.
- For the potential variables d, let $d_{[uv]_i} = 0$ if $[uv]_i \in S$ and 1 otherwise,

where we define S, T as follows. Let F^* be the set of arcs in H whose x variable is 1. We claim that F^* is an st-cut in H. Note that every st-path Q in H corresponds to an st-path P in G. Then, there is an arc uv in P that is also in F. Then, it follows from construction that the x value for one of $[uv]_0, [uv]_1, [uv]_2, [uv]_3$ is 1 and thus there exists $i = 0, 1, 2, 3$ such that $[uv]_i \in F^*$. Note that $[uv]_i$ is also in Q. Therefore F^* is an st-cut in H. Then, let S be the set of vertices in $H \backslash F^*$ that is connected to the source s and let $T = V(H) \backslash S$.

Note that by construction, (x, y, d) is integral and is a feasible solution to DLP. The final objective value $\Sigma_{[uv]_i \in A(H)} c([uv]_i) x_{[uv]_i} = \Sigma_{[uv]_i \in F^*} c([uv]_i)$ and by construction, it matches the cost $c^Y(F^*)$. ◀

▶ **Lemma 6.** *An integral solution (x^*, y^*, d^*) to DLP with objective value c^* corresponds to a feasible solution (Y^*, E^*) to NVDP such that $c^{Y^*}(E^*) \leq c^*$.*

Proof. Given a directed graph G and its auxiliary graph H, let (x^*, y^*, d^*) be an optimal integral solution to DLP with an objective value of c^*. Let $F^* \subseteq A(H)$ be the set of arcs whose x^* value is 1. Let $Y^* \subseteq V^0(H)$ whose y value is 1. Let $E^* \subseteq A(G) = \{uv \in A(G) : [uv]_i \in F^*$ for some $i = 0, 1, 2, 3\}$ be the set of original arcs of those in F^*.

Note that by construction, Y^* does not violate the budget constraint. Every st-path in G corresponds directly to an st-path in H. Since F^* is an st-cut in H, it follows that E^* is an st-cut in G. Then it remains to show that $c^* \geq c^{Y^*}(E^*)$.

Note that

$$c^* = \Sigma_{[uv]_i \in A(H)} c([uv]_i) x^*_{[uv]_i} = \Sigma_{e=uv \in A(G)} c_e x^*_{[uv]_0} + c_{eu} x^*_{[uv]_1} + c_{euv} x^*_{[uv]_2} + c_{ev} x^*_{[uv]_3}.$$

Meanwhile, note that $c^{Y^*}(E^*) = \Sigma_{e=uv \in A(G)} c^{Y^*}(e)$. Thus, it suffices to prove the following claim.

▷ Claim 7. For every arc $e = uv \in A(G)$, $\Sigma_{i=0}^3 c([uv]_i) x^*_{[uv]_i} \geq c^{Y^*}(e)$.

First, note that if $e = uv \notin E^*$, then $c^{Y^*}(e) = 0$ by definition fo E^*. Then, the inequality is trivially true. Thus, we may assume $uv \in E^*$ which implies there exist $i = 0, 1, 2, 3$ such that $[uv]_i \in F^*$ and $x^*_{[uv]_i} = 1$. We will now break into cases depending on whether $u, v \in Y^*$.

Suppose $u, v \notin Y^*$. Then, $y^*_u = y^*_v = 0$ and by constraint (5) in DLP, it follows that the $^*x$ value for $[uv]_1, [uv]_2, [uv]_3$ are all 0. Then, $[uv]_0 \in F^*$ and $\Sigma_{i=0}^3 c([uv]_i) x^*_{[uv]_i} = c_e = c^{Y^*}(e)$.

Now, assume $u \in Y^*, v \notin Y^*$. By constraint (5), $x^*_{[uv]_2} + x^*_{[uv]_3} \leq y^*_v = 0$ and thus only the x^* value for $[uv]_0, [uv]_1$ can be 1. Since we have an integral solution, it follows that $x^*_{[uv]_0} + x^*_{[uv]_1} \geq 1$, since $e \in E^*$. Note that $c_e \geq c_{eu}$. Then $\Sigma_{i=0}^3 c_{[uv]_i} x^*_{[uv]_i} = c_e x^*_{[uv]_0} + c_{eu} x^*_{[uv]_1} \geq c_{eu}(x^*_{[uv]_0} + x^*_{[uv]_1}) \geq c_{eu} = c^{Y^*}(e)$. Note that a similar argument can be made for the case when $u \notin Y^*, v \in Y^*$.

Lastly, assume both $u, v \in Y^*$. Then $c^{Y^*}(e) = c_{euv}$. Note that $c_e, c_{eu}, c_{ev} \geq c_{euv}$. Then, $\Sigma_{i=0}^3 c([uv]_i) x^*_{[uv]_i} \geq \Sigma_{i=0}^3 c_{euv} x^*_{[uv]_i} \geq c_{euv}$. The last inequality is due to the fact that there exists $i = 0, 1, 2, 3$ such that $[uv]_i \in F^*$. This completes the proof of claim and thus also our lemma. ◀

Bicriteria Approximation for NVDP

We now prove Theorem 4. We will work with an optimal solution of DLP defined on the auxiliary graph H. The idea is to use a ball-growing algorithm that greedily finds cuts until one with the promised guarantee is produced. The reason this algorithm is successful is proved by analyzing a randomized algorithm that picks a number $0 \leq \alpha \leq 1$ uniformly at random and chooses a cut at distance α from the source s. Then we choose vertices to downgrade and arcs to cut based on arcs in this cut at distance α. By computing the expected downgrading cost and the expected cost of the cut arcs, the analysis will show the existence of a level cut that satisfies our approximation guarantee.

To achieve the desired result, we cannot work with the graph H directly. This is because the ball-growing algorithm works only if the probability of cutting some arc can be bounded within some range. This bound exists for the final cut arcs (as in the proof for st-cuts) but not for the final downgraded vertices. Consider a vertex v; it is downgraded if any arc of the form $[uv]_2, [uv]_3, [vw]_1, [vw]_2$ is cut in H. Thus it has the potential of being cut anywhere between the range of the vertices $(uv)_2$ and $(vw)_3$. We would like to use Constraint (5) to bound this range but we cannot do this directly since we do not know the length of the arc $[vw]_0$ which also lies in this range. To circumvent this difficulty and properly employ Constraint (5), we will construct a reduced graph H' obtained by contracting some arcs.

Let (x^*, y^*, d^*) be an optimal solution to DLP where the optimal cost is c^*. It follows from the validity of our model that c^* is at most the cost of an optimal integral solution.

Constructing Graph H'

For every arc $uv \in A(G)$, we compare the value $x^*_{[uv]_0}$ and $x^*_{[uv]_1} + x^*_{[uv]_2} + x^*_{[uv]_3}$. The reason we separate this way is because the variables in the second term are influenced by the downgrading values on u, v. Thus the more we downgrade u and v, the larger we are allowed to increase the second sum, and the more length we can place between u and v in these variables. For an arc $uv \in A(G)$, if $x^*_{[uv]_0} < x^*_{[uv]_1} + x^*_{[uv]_2} + x^*_{[uv]_3}$, we say uv is an *aided* arc since the majority of its length is contributed by the downgrading values on the u, v and thus the downgrading values help to generate its length. For all other arcs, we say uv is an *unaided* arc since more of its length would be contributed by the arc $[uv]_0$, corresponding to simply paying for the original cost of deletion c_e without the aid from downgrading. To construct H', if uv is an aided arc, then contract $[uv]_0$. Otherwise, contract $[uv]_1, [uv]_2, [uv]_3$.

Consider a path $P = (uv)_0(uv)_1(uv)_2(uv)_3(uv)_4$ in H. Note that the length of this path is shortened in H' depending on whether uv is an aided/unaided arc. However, since we always retain the larger of $x^*_{[uv]_0}$ and $x^*_{[uv]_1} + x^*_{[uv]_2} + x^*_{[uv]_3}$ in H', the path's length is at most halved. Then it follows that the distance between any two vertices in H' is reduced to at most half its original value in H. In particular, it follows that the shortest path between the source and the sink is at least $1/2$. This property will be crucial in arguing that the solution chosen by our algorithm has low cost relative to the LP optimum.

We make one last adjustments to the weight of aided arcs. Let $D^*((uv)_i)$ be the shortest path distance from the source (ss) to the vertex $(uv)_i$ viewing x^* as lengths in H'. Consider a path $[uv]_1[uv]_2[uv]_3$ of an aided arc. Note that the distances of nodes $(uu), (uv)_2, (uv)_3$ are strictly increasing but $D^*((vv))$ might be strictly less than $D^*((uv)_3)$ (e.g. via an alternate shorter path to (vv) avoiding (uu)). In fact $D^*((vv))$ might even be smaller than $D^*((uu))$. This makes the analysis of the usage of arcs of the form $[uv]_i$ in the cutting procedure difficult. To avoid this difficulty, for any aided arcs uv where $D^*((vv)) < D^*((uu))$, replace the path $[uv]_1[uv]_2[uv]_3$ with a single dummy arc. Then we redefine a new weight variable x': for every aided arc uv, where $0 \leq D^*((vv)) - D^*((uu)) < \Sigma_{i=1}^{3} x^*_{[uv]_i}$, let $x'_{[uv]_i} = x^*_{[uv]_i} \frac{D^*((vv)) - D^*((uu))}{\Sigma_{i=1}^{3} x^*_{[uv]_i}}$. The weight of all dummy arcs are 0. For all other arcs, the x' variables stay the same. This step guarantees that for all aided arcs that have not been replaced by dummy arcs, the distances of $(uu), (uv)_2, (uv)_3, (vv)$ are in non-decreasing order. For those that have been replaced by the dummy arc, we will ensure that these arcs do not occur in any cut chosen in our algorithm. Two important things to keep in mind: $x' \leq x^*$ for any arc while the distance of any vertex from the source remains unchanged. In particular $D^*((tt))$ remains at least $1/2$. Our ball growing algorithm uses the modified distances x'.

Algorithm 1 Ball-Growing Algorithm for NVDP.

Require: a graph G and its auxiliary graph H' with non-negative arc-weights $x'_{[uv]_i}$, source (ss), sink (tt), arc cut costs $c([uv]_i)$ and vertex downgrading costs $r(v)$
Ensure: a vertex set V' and an arc cut E' of G such that $\Sigma_{v \in V'} r(v) \leq 4b, c^{V'}(E') \leq 4c^*$

initialization $V = \{(ss)\}, D((uv)_i) = 1$ for all $(uv)_i \in V(H')$
repeat
  let $X' \subseteq A(H')$ be the cut induced by V
  find $[uv]_i = (uv)_i(uv)_{i+1} \in X'$ minimizing $D((uv)_i) + x'_{[uv]_i}$
  update by adding $(uv)_{i+1}$ to V, update $D((uv)_{i+1}) = D((uv)_i) + x'_{[uv]_i}$
  let $E' = \{uv \in A(G) : \{[uv]_0, [uv]_1, [uv]_2, [uv]_3\} \cap X' \neq \emptyset\}$ and $V' = \{v \in V(G) : \{[uv]_2, [uv]_3, [vw]_1[vw]_2\} \cap X' \neq \emptyset$ for some $u, w \in V(G)\}$
until $\Sigma_{v \in V'} r(v) \leq 4b$ and $c^{V'}(E') \leq 4c^*$
output the set V', E'

Algorithm 1 is simply a restatement of Dijkstra's algorithm run on H'. It follows the general ball-growing technique and looks at cuts X' at various distances from the source. Note that the algorithm adds at least one vertex to a node set V at each iteration so it runs for at most $|V(H')| = O(m)$ steps when applied to the graph H' (Recall that m denotes the number of arcs in the original graph G).

At each iteration, the algorithm computes a cut $X' \subseteq A(H')$ and considers the set E' of original arcs associated to those in X' and the vertex set V' representing the set of vertices we should downgrade based on the arcs in X'. For example, if $[uv]_2 \in X'$, then we should downgrade both u and v. Note that every chosen cut only contains arcs $[uv]_i$ where $D((uv)_i) \leq D((uv)_{i+1})$ so they do not contain any dummy arcs. Thus we can essentially ignore dummy arcs in accounting for the cost of the chosen cut. Furthermore, since X' is a cut in H', it follows that E' is a cut in G.

To argue the validity of the algorithm, we show that there exists a cut X' at some distance $\alpha \leq D(tt)$ from the source such that the associated sets V', E' provides the approximation guarantee.

▶ **Lemma 8.** *There exists X', V', E' such that $\Sigma_{v \in V'} r(v) \leq 4b, c^{V'}(E') \leq 4c^*$*

The main idea of the proof is to pick a distance uniformly at random between zero and the distance of (tt) (which is at least half) and study the cut at that distance. We claim that the extent to which an arc is cut (chosen in E' above) in the random cut is at most twice its x^*-value, using the fact that the range of this arc is at most its x^*-value and the range of the cutting threshold is at least half. When nodes are chosen in the random cut (in V' above) to be downgraded, we argue that the range of cutting any node is at most the maximum of the values in the left hand side of the constraints (5) corresponding to this node, which in turn is at most its y^*-value. Again, since the range of the cutting threshold is at least half, we infer that the probability of downgrading a node in the cutting process is at most twice its y^*-value. To obtain a cut where we simultaneously do not exceed both bounds, we use Markov's inequality to argue a probability of at least half of being within twice these respective expectations, hence giving us a single cut with both bounds within four times their respective LP values. The detailed proof follows.

Proof of Lemma 8. Let $D((uv)_i)$ be the shortest-path distance from the source (ss) to any vertex $(uv)_i \in V(H')$ viewing the x' variables as lengths. Note that $D((tt)) \geq 1/2$ since the original distance is at least 1 and H' reduces the distance by at most $1/2$. Note that the triangle-inequality holds under this distance metric where $D((uv)_i) - D((u'v')_{i'})$ is at most the distance between $(uv)_i$ and $(u'v')_{i'}$.

Defining the Random Variables. Let α be chosen uniformly at random from the interval $[0, D((tt))]$. Consider $X_\alpha := \{[uv]_i \in A(H') : D((uv)_i) \leq \alpha < D((uv)_{i+1})\}$, the cut at distance α in H'. Let $E_\alpha = \{uv \in A(G) : [uv]_i \in X_\alpha$ for some $i = 0, 1, 2, 3\}$, representing the original arcs corresponding to those in X_α. Let $V_\alpha = \{v \in V(G) : \{[uv]_2, [uv]_3, [vw]_1, [vw]_2\} \cap X_\alpha \neq \emptyset$ for some $u, w \in V(G)\}$, representing the set of vertices we should downgrade so that the final cost of the arcs E_α matches the cost associated to X_α. More precisely, we want $c^{V_\alpha}(E_\alpha) = \Sigma_{[uv]_i \in X_\alpha} c([uv]_i)$. Note that by construction E_α is an st-cut in G. Let $\mathcal{V} = \Sigma_{v \in V_\alpha} r(v), \mathcal{E} = c^{V_\alpha}(E_\alpha)$. Our goal is to show that these two random variables $\mathcal{V}, \mathcal{E}$ have low expectations and obtain our approximation guarantee using Markov's inequality. In particular, we will prove that $\mathbb{E}[\mathcal{V}] \leq 2b$, and that $\mathbb{E}[\mathcal{E}] \leq 2c^*$ where c^* is the optimal value of DLP.

To understand $\mathcal{E}$, for every arc $e = uv \in A(G)$, we introduce the indicator variables $\mathcal{E}_e$ to be 1 if arc $e \in E_\alpha$ and 0 otherwise. Then $\mathcal{E} = \Sigma_{e \in A(G)} \mathcal{E}_e c^{V_\alpha}(e)$. To study the value of $\mathcal{E}_e c^{V_\alpha}(e)$, we can break into several cases depending on which arc $[uv]_i \in X_\alpha$. Note that if $[uv]_i \notin X_\alpha$ for $i = 0, 1, 2, 3$, then $e \notin E_\alpha$ and $\mathcal{E}_e c^{Y_\alpha}(e) = 0$. Next, if we assume $[uv]_i \in X_\alpha$, then one can check that $c^{V_\alpha}(e) \leq c([uv]_i)$ as in the proof of Claim 7.

Slightly abusing the notation, define the indicator variable $\mathcal{E}_{[uv]_i}$ for arc $[uv]_i \in A(H)$ to be 1 if $[uv]_i \in X_\alpha$ and 0 otherwise. Then, we can upper-bound the expectation of $\mathcal{E}$ using conditional expectations of the events $\mathcal{E}_{[uv]_i} = 1$ as follows.

$$\begin{aligned}
\mathbb{E}[\mathcal{E}] =& \Sigma_{e \in A(G)} \mathbb{E}[\mathcal{E}_e c^{V_\alpha}(e)] \\
=& \Sigma_{e \in A(G)} \Sigma_{i=0}^{3} \mathbb{E}[c^{V_\alpha}(e) | \mathcal{E}_{[uv]_i} = 1] \cdot Pr[\mathcal{E}_{[uv]_i} = 1] \\
\leq& \Sigma_{e \in A(G)} \Sigma_{i=0}^{3} c([uv]_i) Pr[\mathcal{E}_{[uv]_i} = 1]
\end{aligned}$$

To understand the probability of $\mathcal{E}_{[uv]_i} = 1$, note that an arc $[uv]_i \in X_\alpha$ if and only if $D((uv)_i) \leq \alpha < D((uv)_{i+1})$. Then, $Pr[[uv]_i \in X_\alpha] \leq (D((uv)_{i+1}) - D((uv)_i)/D((tt)) \leq 2x'_{[uv]_i} \leq 2x^*_{[uv]_i}$ since $D((tt)) \geq 1/2$. Combining with the previous inequalities, we see that

$$\begin{aligned}
\mathbb{E}[\mathcal{E}] \leq& \Sigma_{uv \in A(G)} \Sigma_{i=0}^{3} c([uv]_i) Pr[\mathcal{E}_{[uv]_i} = 1] \\
\leq& \Sigma_{uv \in A(G)} \Sigma_{i=0}^{3} c([uv]_i) 2x^*_{[uv]_i} = 2c^*.
\end{aligned}$$

Next, we show a similar result for $\mathcal{V}$. Note that $\mathbb{E}[\mathcal{V}] = \Sigma_{v \in V(G)} r(v) \cdot Pr[v \in V_\alpha]$. Recall that $v \in V_\alpha$ if and only if there exists a vertex u or w such that at least one of $[uv]_2, [uv]_3, [vw]_1, [vw]_2 \in X_\alpha$. Note that if uv is an unaided arc, then $[uv]_2, [uv]_3$ would have been contracted in H' and would never be chosen in X_α. If uv is an aided arc that was turned into a dummy arc, it would also never be chosen in the final cut. Therefore, we only need to consider aided arcs that have not been turned into dummy arcs. In order to upper-bound the probability of choosing v into V_α, we thus need to find the range of possible α that might affect v. For any vertex $v \in V(G)$, it follows that we only need to examine aided arcs incident to the vertex v. Let $u \in V(G)$ such that $uv \in A(G), uv$ is an aided arc and $D((uv)_2)$ is minimum. Let $w \in V(G)$ such that vw is an aided arc and $D((vw)_3)$ is maximum. Note that for any aided arcs zv, vz' that are not replaced by a dummy arc, $D((zz)) \leq D((zv)_2) \leq D((zv)_3) \leq D((vv)) \leq D((vz')_2) \leq D((vz')_3) \leq D((z'z'))$ by our choice of x'. For all such arcs of the form $[zv]_2, [zv]_3, [vz']_1, [vz']_2$, their extremities are in the distance range $D((uv)_2), D((vw)_3)$. Then, v is chosen only if α is between $D((uv)_2)$ and $D((vw)_3)$. The distance between $(uv)_2$ and $(vw)_3$ is upper-bounded by the length of a shortest path in H'. Since vw is an aided arc, $[vw]_0$ is contracted in H'. Then $(uv)_2(uv)_3(vv)(vw)_1(vw)_2$ is a path in H'. Thus $D((vw)_3) - D((uv)_2) \leq x'_{[uv]_2} + x'_{[uv]_3} + x'_{[vw]_1} + x'_{[vw]_2} \leq x^*_{[uv]_2} + x^*_{[uv]_3} + x^*_{[vw]_1} + x^*_{[vw]_2} \leq y^*_v$ where the last inequality follows from Constraint (5)[1]. Thus, $Pr[D((uv)_2) \leq \alpha < D((vw)_3)] \leq y^*_v / D((tt)) \leq 2y^*_v$. Therefore

$$\begin{aligned}\mathbb{E}[\mathcal{V}] =&\Sigma_{v\in V(G)} r(v) \cdot Pr[v \in V_\alpha]\\ \leq&\Sigma_{v\in V(G)} r(v) 2y^*_v \leq 2b.\end{aligned}$$

Lastly, by Markov's inequality, $Pr[\mathcal{V} \leq (2+\epsilon)2b] \geq 1 - 1/(2+\epsilon), Pr[\mathcal{E} \leq 4c^*] \geq 1/2$ for any $\epsilon > 0$. Then it follows there exists $0 \leq \alpha \leq D((tt))$ such that $\Sigma_{v\in V_\alpha} r(v) \leq 4b + 2\epsilon b$ and $c^{V_\alpha}(E_\alpha) \leq 4c^*$. One can choose ϵ such that $2\epsilon b < 1$. Since $r(v)$ is always integral, it follows that $\Sigma_{v\in V_\alpha} r(v) \leq 4b$, proving Lemma 8. ◀

It is well known that the Ball Growing algorithm (which is Djikstra's algorithm run on H') selects a linear number of nested cuts that represent the set of all cuts at all distances between zero and $D((tt))$ from the source. It follows from Lemma 8 that one of these cuts meets the desired guarantees. Theorem 4 is then proved by simply running Algorithm 1 on the auxiliary graph H'.

References

1 Hassene Aissi, Da Qi Chen, and R. Ravi. Downgrading to minimize connectivity, 2019. arXiv:1911.11229.

2 C. Burch, R. Carr, S. Krumke, M. Marathe, C. Phillips, and E. Sundberg. A decomposition-based pseudoapproximation algorithm for network flow inhibition. In Woodruff D. L., editor, *Network Interdiction and Stochastic Integer Programming*, volume 26, pages 51–68. springer, 2003.

3 Stephen R Chestnut and Rico Zenklusen. Interdicting structured combinatorial optimization problems with {0, 1}-objectives. *Mathematics of Operations Research*, 42(1):144–166, 2016.

4 Stephen R Chestnut and Rico Zenklusen. Hardness and approximation for network flow interdiction. *Networks*, 69(4):378–387, 2017.

5 Julia Chuzhoy, Yury Makarychev, Aravindan Vijayaraghavan, and Yuan Zhou. Approximation algorithms and hardness of the k-route cut problem. *ACM Transactions on Algorithms (TALG)*, 12(1):2, 2016.

6 Julia Chuzoy. Flows, cuts and integral routing in graphs - an approximation algorithmist's perspective. In *Proc. of the International Congress of Mathematicians*, pages 585–607, 2014.

[1] This is the main reason why we distinguish between aided and unaided arcs and contract the appropriate one to construct H'. Without the contraction, the distance between $(uv)_2$ and $(vw)_3$ includes the arc $[vw]_0$ and thus could be arbitrarily larger than y^*_v. Also, without rescaling x^* to x', it is possible that some $D((zv)_3) > D((vw)_3)$. Then, the range in which v is downgraded can go much further past y^*_v.

7 Bruce Golden. A problem in network interdiction. *Naval Research Logistics Quarterly*, 25(4):711–713, 1978.

8 Bertrand Guenin, Jochen Könemann, and Levent Tuncel. *A gentle introduction to optimization*. Cambridge University Press, 2014.

9 Guru Guruganesh, Laura Sanita, and Chaitanya Swamy. Improved region-growing and combinatorial algorithms for k-route cut problems. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 676–695. Society for Industrial and Applied Mathematics, 2015.

10 T. E. Harris and F. S. Ross. Fundamentals of a method for evaluating rail net capacities. Technical report, RAND CORP SANTA MONICA CA, Santa Monica, California, 1955.

11 Eitan Israeli and R Kevin Wood. Shortest-path network interdiction. *Networks: An International Journal*, 40(2):97–111, 2002.

12 André Linhares and Chaitanya Swamy. Improved algorithms for mst and metric-tsp interdiction. *Proceedings of 44th International Colloquium on Automata, Languages, and Programming*, 32:1–14, 2017.

13 Christos H Papadimitriou and Mihalis Yannakakis. On the approximability of trade-offs and optimal access of web sources. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 86–92. IEEE, 2000.

14 Cynthia A. Phillips. The network inhibition problem. In *Proceedings of the Twenty-fifth Annual ACM Symposium on Theory of Computing*, STOC '93, pages 776–785, New York, NY, USA, 1993. ACM. `doi:10.1145/167088.167286`.

15 Alexander Schrijver. On the history of the transportation and maximum flow problems. *Mathematical Programming*, 91(3):437–445, 2002.

16 R. Wood. Deterministic network interdiction. *Mathematical and Computer Modeling*, 17(2):1–18, 1993.

17 R. Zenklusen. Matching interdiction. *Discrete Applied Mathematics*, 145(15), 2010.

18 R. Zenklusen. Network flow interdiction on planar graphs. *Discrete Applied Mathematics*, 158(13), 2010.

19 R. Zenklusen. Connectivity interdiction. *Operations Research Letters*, 42(67):450–454, 2014.

20 R. Zenklusen. An $\mathcal{O}(1)$ approximation for minimum spanning tree interdiction. *Proceedings of 56th Annual IEEE Symposium on Foundations of Computer Science*, pages 709–728, 2015.