Approximation Algorithms for Disjoint *st*-Paths with Minimum Activation Cost

Hasna Mohsen Alqahtani and Thomas Erlebach

Department of Computer Science, University of Leicester, Leicester, UK.
{hmha1|t.erlebach}@leicester.ac.uk

Abstract. In network activation problems we are given a directed or undirected graph $G = (V, E)$ with a family $\{f_{uv}(x_u, x_v) : (u, v) \in E\}$ of monotone non-decreasing activation functions from D^2 to $\{0, 1\}$, where D is a constant-size domain. The goal is to find activation values x_v for all $v \in V$ of minimum total cost $\sum_{v \in V} x_v$ such that the activated set of edges satisfies some connectivity requirements. Network activation problems generalize several problems studied in the network literature such as power optimization problems. We devise an approximation algorithm for the fundamental problem of finding the *Minimum Activation Cost Pair of Node-Disjoint st-Paths* (MA2NDP). The algorithm achieves approximation ratio 1.5 for both directed and undirected graphs. We show that a ρ-approximation algorithm for MA2NDP with fixed activation values for s and t yields a ρ-approximation algorithm for the *Minimum Activation Cost Pair of Edge-Disjoint st-Paths* (MA2EDP) problem. We also study the MA2NDP and MA2EDP problems for the special case $|D| = 2$.

1 Introduction

In this paper we consider network activation problems. In these problems we are given an *activation network*, which is a directed or undirected graph $G = (V, E)$ together with a family $\{f_{uv}(x_u, x_v) : (u, v) \in E\}$ of monotone non-decreasing activation functions from D^2 to $\{0, 1\}$, where D is a constant-size domain. The activation of an edge depends on the chosen values from the domain D at its endpoints. We say that an edge $(u, v) \in E$ is *activated* for chosen values x_u and x_v if $f_{uv}(x_u, x_v) = 1$. An activation function is called *monotone non-decreasing* if for every $(u, v) \in E$ we have that $f_{uv}(x_u, x_v) = 1$ implies $f_{uv}(y_u, y_v) = 1$ for any $y_u \geq x_u$, $y_v \geq x_v$. The goal is to determine activation values $x_v \in D$ for all $v \in V$ so that the total activation cost $\sum_{v \in V} x_v$ is minimized and the activated set of edges satisfies some connectivity requirements. Network activation problems were introduced by Panigrahi [11]. They generalize several known problems in wireless network design, e.g., minimum broadcast tree, installation cost optimization, and power optimization. For further applications and motivation for network activation problems we refer to [11, 9].

We assume in the remainder of the paper that G is a directed graph. For the problems under consideration, the case of undirected graphs can be modelled by replacing each undirected edge $\{u, v\}$ by two directed edges (u, v) and (v, u) with the same activation function, i.e., $f_{uv}(x_u, x_v) = f_{vu}(x_v, x_u)$.

As D is a constant-size domain, we assume that the activation functions are specified by lookup tables. For each edge $(u, v) \in E$ we can then compute in polynomial time the minimum cost $c_{uv} = x_u^{uv} + x_v^{uv}$, where x_u^{uv} is the activation value on u and x_v^{uv} the activation value on v such that $f_{uv}(x_u^{uv}, x_v^{uv}) = 1$ and $x_u^{uv} + x_v^{uv}$ is minimized.

Related work. The core objective of most network activation problems is to activate a network of minimum activation cost satisfying certain given connectivity requirements. The simplest connectivity requirement is to find an st-path for a specified pair of nodes s and t. Other examples of fundamental connectivity requirements are: spanning tree, k edge-disjoint st-paths, k node-disjoint st-paths, etc. Traditionally, these problems have been studied in a network model where each edge (or node) has a fixed cost.

In recent years, considerable work has been done on various network activation problems such as *Minimum Steiner Activation Network* (MSAN), *Minimum Spanning Activation Tree* (MSpAT), and *Minimum Activation Flow* (MAF). The problem of activating a network with k edge/node-disjoint paths between every pair of nodes is called *Minimum Edge/Node-connected Activation Network* (MEAN/MNAN). Panigrahi [11] gives an exact polynomial-time algorithm to solve the *Minimum Activation st-Path* (MAP) problem. However, he observes that the MAF problem (activating k edge-disjoint st-paths with minimum activation cost) is at least as hard as the ℓ-densest subgraph problem. As shown in [11], it is NP-hard to approximate MSpAT within a factor of $o(\log n)$. The MSpAT problem is a special case of the MSAN, MEAN and MNAN problems. Therefore, it is also NP-hard to approximate these problems within $o(\log n)$. Panigrahi presents $O(\log n)$-approximation algorithms for MSpAT, and also for MEAN and MNAN in the case of $k = 2$. Nutov [9] establishes a connection between network activation problems and edge-cost network design problems and shows that there exists a 2-approximation algorithm for the *Minimum Activation Cost k Node-Disjoint st-Paths* (MAkNDP) problem and a $2k$-approximation algorithm for the *Minimum Activation Cost k Edge-Disjoint st-Paths* (MAkEDP) problem.

Other relevant work has addressed power optimization [2, 7, 8]. In power optimization problems, each edge $(u, v) \in E$ has a threshold power requirement θ_{uv}. In the undirected case, edge (u, v) is activated for chosen values x_u and x_v if each of these values is at least θ_{uv}. In the directed case, edge (u, v) is activated if $x_u \geq \theta_{uv}$.

Power optimization is a special case of network activation problems. As mentioned in [11], in the power optimization setting the MEAN and MNAN problems have 4-approximation and 11/3-approximation algorithms, respectively, and it is known that the MSpAT problem is APX-hard. By a simple reduction to the shortest st-path problem, the *Minimum Power st-Path* problem is solvable in polynomial time for both directed and undirected networks [7]. Another problem that has been studied in the literature is finding the *Minimum Power k*

Edge-Disjoint st-Paths (MPkEDP). [6] shows that for both the directed and undirected variants, the MPkEDP problem is unlikely to admit even a polylogarithmic approximation algorithm. In contrast, the problem of finding Minimum Power k Node-Disjoint st-Paths in directed graphs can be solved in polynomial time [6,12].

The problem of finding node/edge disjoint st-paths with minimum cost in a network with edge costs is a well studied problem in graph theory. Polynomial-time algorithms have been known for decades [1,13,14]. These algorithms do not address the problem in the network activation setting, however. In this paper, we study the minimum activation cost pair of node/edge-disjoint st-paths problem. To the best of our knowledge, it is not yet known whether these problems are NP-hard. A *ρ-approximation algorithm* for a network activation problem is an algorithm that runs in polynomial time and always outputs a solution whose activation cost is at most ρ times the optimal activation cost for the given instance.

Our results. We give a 1.5-approximation algorithm for the MA2NDP problem. We also show that a ρ-approximation algorithm for the MA2NDP problem with fixed activation values of s and t implies a ρ-approximation algorithm for the MA2EDP problem. For the case where the domain D has size 2 and all edges of the network have the same activation function, we prove that the MAkNDP problem is polynomial-time solvable for four of five cases of the activation function, and that the MAkEDP problem is NP-hard.

We employ ideas and techniques from the theory of network flows in order to establish approximation algorithms for our problems. The idea of the MA2NDP algorithm is to first guess the optimal activation values for the nodes s and t by enumeration. For each choice of activation values for s and t, we construct an edge-cost network from G. We then use ideas similar to Suurballe's algorithm [13], with modifications in the construction of the residual graph, to find the two node-disjoint st-paths. For the connection between the MA2NDP and MA2EDP problems, we design an approximation algorithm for the MA2EDP problem by using a ρ-approximation algorithm for the MA2NDP problem for every pair of nodes in the graph and then iteratively combining disjoint paths to/from an intermediate node into edge-disjoint paths with common nodes. We prove that this algorithm has approximation ratio ρ. For the special case where the domain D has size 2 and all edges have the same activation function, we show the NP-hardness of the MAkEDP problem by giving a reduction from the decision version of the maximum balanced complete bipartite subgraph (MaxBCBS) problem [4].

The remainder of the paper is organized as follows. We start by presenting our algorithm for the MA2NDP problem in Section 2. In Section 3, we establish the connection between the MA2NDP and MA2EDP problems and obtain a 1.5-approximation algorithm for MA2EDP. We then discuss the problem of finding k node/edge-disjoint st-paths with minimum activation cost in the case where $|D| = 2$ in Section 4. Finally, we conclude with a short section on future work and open questions. Some proofs are omitted due to space constraints.

2 Minimum Activation Cost Node-Disjoint *st*-Paths

The minimum activation k node-disjoint st-paths (MAkNDP) problem can be stated as follows: Given an activation network $G = (V, E)$ and source-destination pair $s, t \in V$, find activation values x_v for all $v \in V$ such that k node-disjoint st-paths $P^{st} = \{P_1, P_2, \ldots, P_k\}$ are activated and the total activation cost $\sum_{v \in V} x_v$ is minimized. [9] gave a 2-approximation algorithm for the MAkNDP problem. In this section we consider the special case $k = 2$ and give a 1.5-approximation algorithm for the MA2NDP problem.

Let FMA2NDP denote the variant of the MA2NDP problem where values $d \in D$ and $d' \in D$ are specified as activation values of s and t, respectively, and do not count towards the objective value. In other words, only solutions with $x_s = d$ and $x_t = d'$ are considered feasible for FMA2NDP, and the activation cost of a solution is $\sum_{v \in V \setminus \{s,t\}} x_v$.

The MA2NDP algorithm takes as input an activation network $G = (V, E)$ and a source-destination pair, $s, t \in V$. Its output is a set of activation values x_v for all $v \in V$ that activate a pair of node-disjoint st-paths, $P^{st} = \{P_1, P_2\}$. The algorithm enumerates all pairs $d, d' \in D$. For each choice of d and d', it solves the FMA2NDP problem with the values d and d' chosen as the activation values of s and t, respectively. Let $C(s, d, t, d')$ represent the activation cost of the solution for d, d'. In the end, the algorithm outputs the solution of minimum activation cost among the feasible solutions obtained for all pairs of values $d, d' \in D$, i.e., the solution of activation cost $\min_{x_s, x_t \in D}\{C(s, x_s, t, x_t)\}$.

The algorithm for the FMA2NDP problem with $x_s = d$ and $x_t = d'$ is as follows. For ease of presentation, we assume here that $D = \{0, 1, 2, \ldots, |D| - 1\}$. (The extension to arbitrary domains of constant size is straightforward.) We let $C(s, d, t, d')$ represent the total activation cost of the pair of node-disjoint activation paths P^{st} that the algorithm finds, or ∞ if such paths do not exist.

Step 1: Construct from G an edge-weighted graph $\bar{G}$ with two nodes s_d, $t_{d'}$ and $2|D|$ nodes $\{v_0^{in}, v_0^{out}, v_1^{in}, v_1^{out}, ..., v_{|D|-1}^{in}, v_{|D|-1}^{out}\}$ for every $v \in V \backslash \{s, t\}$. The edges of $\bar{G}$ are:

- For $a \in D$ and $v \in V \setminus \{s, t\}$, add a directed edge (v_a^{in}, v_a^{out}) with cost 0.
- For each $(u, v) \in E$ and $a, b \in D$ where $u, v \notin \{s, t\}$ and $f_{uv}(a, b) = 1$, add a directed edge (u_a^{out}, v_b^{in}) with cost b.
- For each $(s, v) \in E$ and $b \in D$ where $v \neq t$ and $f_{sv}(d, b) = 1$, add a directed edge (s_d, v_b^{in}) with cost b.
- For each $(v, t) \in E$ and $a \in D$ where $v \neq s$ and $f_{vt}(a, d') = 1$, add a directed edge $(v_a^{out}, t_{d'})$ with cost 0.
- If $(s, t) \in E$ and $f_{st}(d, d') = 1$, add a directed edge $(s_d, t_{d'})$ with cost 0.

Step 2: Run Dijkstra's algorithm on $\bar{G}$ to compute a shortest path P from s_d to $t_{d'}$. Let $C(P)$ be the edge-cost of P. If $\bar{G}$ has no such path, set $C(s, d, t, d') = \infty$ and skip Steps 3–5.

Step 3: Construct the residual network $\bar{G}_P$ induced by P:

- For each $v \in V \setminus \{s, t\}$ with $(v_a^{in}, v_a^{out}) \in P$ for some a, add a directed edge $(v_{\bar{a}}^{out}, v_{\bar{a}}^{in})$ with cost 0 for all $\bar{a} \in D$ with $\bar{a} \geq a$.

- For each $v \in V \setminus \{s,t\}$ such that $(v_a^{in}, v_a^{out}) \notin P$ for all $a \in D$, add (v_a^{in}, v_a^{out}) for all $a \in D$ with cost 0.
- For each $(u,v) \in E$ with $(u_a^{out}, v_b^{in}) \in P$ for some $a, b \in D$, add a directed edge $(v_{\bar{b}}^{in}, u_{\bar{a}}^{out})$ with cost $\bar{a} - a$ for all $\bar{a}, \bar{b} \in D$ where $\bar{a} \geq a$ and $\bar{b} \geq b$.
- For each $(u,v) \in E$ with $(u_{\underline{a}}^{out}, v_b^{in}) \notin P$ for all $a, b \in D$ where v is used in P with activation value $\bar{b}$ (i.e., $v_{\bar{b}}^{in} \in P$), add edges $(u_{a'}^{out}, v_{b'}^{in})$ with cost $b' - \bar{b}$ for all $a' \in D$, $b' \geq \bar{b}$ such that $f_{uv}(a', b') = 1$.
- For each $(u,v) \in E$ with $(u_a^{out}, v_b^{in}) \notin P$ for all $a, b \in D$ where v is not used in P, add edges $(u_{a'}^{out}, v_{b'}^{in})$ with cost b' for all $a', b' \in D$ such that $f_{uv}(a', b') = 1$.
- For each $v \in V \setminus \{t\}$ with $(s,v) \in E$ and $(s_d, v_b^{in}) \notin P$ for all $b \in D$, add edges $(s, v_{b'}^{in})$ with cost b' for all $b' \in D$ such that $f_{sv}(d, b') = 1$.
- For each $v \in V \setminus \{s\}$ with $(v,t) \in E$ and $(v_b^{out}, t_{d'}) \notin P$ for all $b \in D$, add edges $(v_{b'}^{out}, t_{d'})$ with cost 0 for all $b' \in D$ such that $f_{vt}(b', d') = 1$.
- If $(s,t) \in E$ and $f_{st}(d, d') = 1$ and $(s_d, t_{d'}) \notin P$, add a directed edge $(s_d, t_{d'})$ with cost 0.

Step 4: Run Dijkstra's algorithm on the residual network $\bar{G}_P$ to identify a shortest path P' from s_d to $t_{d'}$. Let $C'(P')$ represent the edge-cost of P'. If no such path P' exists, set $C(s,d,t,d') = \infty$ and skip Step 5.

Step 5: Decompose P and P' into two node-disjoint paths, by removing from $P \cup P'$ the edge set which consists of the edges of P whose reverse edge is in P', and vice versa. Let P_1 and P_2 be the corresponding node-disjoint paths in G, and let $C(s,d,t,d')$ be the activation cost of $P^{st} = \{P_1, P_2\}$. Return $C(s,d,t,d')$ and P^{st}.

Note that the auxiliary graph $\bar{G}$ constructed in Step 1 has the property that any path Q from s_d to $t_{d'}$ in $\bar{G}$ with edge cost $C(Q)$ corresponds to an activated path Q' in G from s to t with activation cost $d + C(Q) + d'$, and vice versa. If the path Q uses an edge with head v_a^{in}, this corresponds to activating node v with activation value $x_v = a$ (and the cost of the edge 'pays' for this activation value). The shortest path constructed in Step 2 thus corresponds to a minimum activation cost st-path, under the constraint that $x_s = d$ and $x_t = d'$.

Let an instance of the FMA2NDP problem be given by a graph $G = (V, E)$ with designated nodes $s, t \in V$, a family F of activation functions from D^2 to $\{0, 1\}$, and values $d, d' \in D$. Let $P^{st} = \{P_1, P_2\}$ be the paths found by the FMA2NDP algorithm and let $P^{OPT} = \{P_1^{OPT}, P_2^{OPT}\}$ be an optimum solution for this instance. We define $C_{ALG}(Q)$ as the activation cost of a path Q in G in the solution generated by the algorithm and $C_{OPT}(Q)$ as the activation cost of a path Q in the optimum solution. The edge cost of a path Q in $\bar{G}$ is denoted by $C(Q)$, and the edge cost of a path Q in $\bar{G}_P$ is denoted by $C'(Q)$.

Lemma 1. *For any $x_s, x_t \in D$ for which there are two node-disjoint st-paths, let $\bar{G}_P$ be the residual network of $\bar{G}$ imposed by P (Step 3). Then there exists a path $P' \in \bar{G}_P$ from s_d to $t_{d'}$ with edge-cost $C'(P')$ such that:*

$$C'(P') \leq C_{OPT}(P_1^{OPT} \setminus \{s,t\}) + C_{OPT}(P_2^{OPT} \setminus \{s,t\}) \tag{1}$$

Proof. Let $\tilde{G} \subset G$ be the network generated by edges that belong to P^{OPT} and the path in G that corresponds to P. Let $\tilde{G}_P$ be the (standard) residual network of $\tilde{G}$ imposed by P. There exists a path P^* from s to t in $\tilde{G}_P$. We have $C_{OPT}(P^*) = \sum_{v \in P^*} x_v$, where x_v is the activation value of v in the optimal solution. Clearly, $C_{OPT}(P^*)$ is at most the total cost of the optimal solution P^{OPT}. Consequently, $C_{OPT}(P^*) - (d+d') \leq C_{OPT}(P_1^{OPT} \setminus \{s,t\}) + C_{OPT}(P_2^{OPT} \setminus \{s,t\})$.

We want to prove that there is a path corresponding to P^* in $\bar{G}_P$ with edge cost at most $C_{OPT}(P^*) - d - d'$. To prove this, one can show that for each edge (u,v) of P^* there exists a corresponding edge in $\bar{G}_P$ whose cost is bounded by the activation value of v in P^{OPT}. As the algorithm computes a path P' with minimum edge cost in $\bar{G}_P$, we get that $C'(P') \leq \sum_{v \in P^* \setminus \{s,t\}} x_v = C_{OPT}(P^*) - (d+d')$. □

Theorem 1. *The algorithm computes a* 1.5*-approximation for the FMA2NDP problem.*

Proof. Since P is an st-path of minimum activation cost, we get that the activation cost of its intermediate nodes, which is equal to its edge cost $C(P)$, is bounded by

$$\begin{aligned} C\left(P\right) &\leq \min\{C_{OPT}\left(P_1^{OPT} \setminus \{s,t\}\right), C_{OPT}\left(P_2^{OPT} \setminus \{s,t\}\right)\} \\ &\leq \frac{C_{OPT}\left(P_1^{OPT} \setminus \{s,t\}\right) + C_{OPT}\left(P_2^{OPT} \setminus \{s,t\}\right)}{2} \end{aligned} \tag{2}$$

From Step 5 in the algorithm we notice that:

$$C_{ALG}(P_1 \setminus \{s,t\}) + C_{ALG}(P_2 \setminus \{s,t\}) \leq C(P) + C'(P') \tag{3}$$

From Lemma 1, (2) and (3) we get that the solution computed by the algorithm has objective value at most 1.5 times the optimal objective value. □

As our MA2NDP algorithm enumerates all possibilities for the activation values of s and t and outputs the solution of minimum activation cost among all computed solutions, Theorem 1 implies the following corollary.

Corollary 1. *There is a* 1.5*-approximation algorithm for MA2NDP.*

3 Minimum Activation Cost Edge-Disjoint *st*-Paths

The minimum activation cost k edge-disjoint st-paths problem (MAkEDP) can be stated as follows: Given an activation network $G = (V,E)$ and a source-destination pair $s, t \in V$, find activation values x_v for all $v \in V$ that activate a set of k edge-disjoint st-paths $P^{st} = \{P_1, P_2, \ldots, P_k\}$ such that the total cost $\sum_{v \in V} x_v$ is minimized. We consider the problem for $k = 2$, i.e., MA2EDP.

We observe that a pair of edge-disjoint st-paths can be viewed as the concatenation of pairs of node-disjoint paths between consecutive common nodes of the pair of edge-disjoint paths (see Fig. 1). This connection was used by Srinivas and Modiano [12] to derive a polynomial-time optimal algorithm for the special case of MA2EDP that arises in the power optimization setting. We generalize this method to the network activation setting and obtain the following theorem that connects the FMA2NDP and MA2EDP problems.

Theorem 2. *If there exist a ρ-approximation algorithm for FMA2NDP, then there exists a ρ-approximation algorithm for MA2EDP.*

The proof of this theorem is based on showing that the following MA2EDP algorithm computes a pair of edge-disjoint paths of activation cost at most ρ times the optimal activation cost. The MA2EDP algorithm takes as input an activation network $G = (V, E)$ and a source-destination pair, $s, t \in V$. Its output is a pair of edge-disjoint activated st-paths, $P^{st} = \{P_1, P_2\}$. The algorithm executes a ρ-approximation algorithm for FMA2NDP for each pair of nodes in G with specified activation values for that pair of nodes, and then iteratively combines disjoint paths to/from an intermediate node to obtain edge-disjoint paths. The MA2EDP algorithm can be specified via the following two steps:

Step 1: For every pair of nodes $u, u' \in V$ and every pair of activation values $d, d' \in D$, the algorithm runs the ρ-approximation algorithm for the FMA2NDP problem with source u, activated with $x_u = d$, and destination u', activated with $x_{u'} = d'$. This produces a pair of node-disjoint activation paths for each pair of nodes $u, u' \in V$ and specified activation values $x_u, x_{u'}$. Let $P^{(u,x_u,u',x_{u'})}$ denote this pair of node-disjoint paths (and the corresponding activation values of all nodes) and $C(u, x_u, u', x_{u'})$ its activation cost (or ∞, if such a pair of node-disjoint paths does not exist).

Step 2:

```
**for** each node $w \in V$:
    **for** each $x_w \in D$, each pair of nodes $u, u' \in V$ and each pair $x_u, x_{u'} \in D$:
        Combine the pairs of edge-disjoint paths $P^{(u,x_u,w,x_w)}$ and $P^{(w,x_w,u',x_{u'})}$
        into a pair of edge-disjoint paths $Q^{(u,x_u,u',x_{u'})}$ from $u$ to $u'$ and update
        the cost $C(u, x_u, u', x_{u'})$ via:
            $C(u, x_u, u', x_{u'}) =$
            $\min\{C(u, x_u, u', x_{u'}), C(u, x_u, w, x_w) + C(w, x_w, u', x_{u'}) - x_w\}$
        If the cost $C(u, x_u, u', x_{u'})$ changes by this update, set $P^{(u,x_u,u',x_{u'})}$ to
        $Q^{(u,x_u,u',x_{u'})}$.
```

The final output is the activation cost $\min_{x_s,x_t \in D}\{C(s, x_s, t, x_t)\}$ and the corresponding pair of edge-disjoint st-paths $P^{(s,x_s,t,x_t)}$.

To show that the MA2EDP algorithm actually finds a pair of edge-disjoint paths of activation cost at most $C(s, x_s, t, x_t)$, we have the following lemma.

Lemma 2. *Consider any time in the execution of the algorithm. Assume that at that time we have $C(u, x_u, u', x_{u'}) = T < \infty$. Then $P^{(u,x_u,u',x_{u'})}$ contains two edge-disjoint uu'-paths with activation cost at most T such that u has activation value at least x_u and u' has activation value at least $x_{u'}$.*

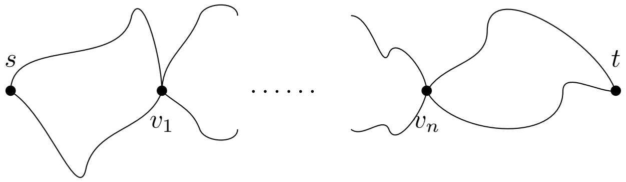

Fig. 1. A pair of edge-disjoint paths viewed as the concatenation of pairs of node-disjoint paths.

Proof (of Theorem 2). Let an instance of the MA2EDP problem be given by $G = (V, E)$, $s, t \in V$ and a family F of activation functions. Let $V = \{v_1, v_2, \ldots, v_n\}$, where the nodes are numbered in the order in which they are processed by the outer for-loop in Step 2. Define $C_k(u, x_u, v, x_v)$ as the value of $C(u, x_u, v, x_v)$ after $k \in \{0, ..., n\}$ iterations of the outer for-loop in Step 2 and $C_k^{OPT}(u, x_u, v, x_v)$ to be the optimal activation cost for two edge-disjoint paths from u, activated with x_u, to v, activated with x_v, for which the only intermediate nodes that are common between the paths are among $\{v_1, \ldots, v_k\}$. Let $\bar{C}_k(u, x_u, v, x_v) = C_k(u, x_u, v, x_v) - x_u - x_v$ and $\bar{C}_k^{OPT}(u, x_u, v, x_v) = C_k^{OPT}(u, x_u, v, x_v) - x_u - x_v$. By induction, we will prove that for all $u, v \in V$ and $x_u, x_v \in D$ after k iterations the following holds:

$$\bar{C}_k(u, x_u, v, x_v) \ \leq \ \rho\, \bar{C}_k^{OPT}(u, x_u, v, x_v) \tag{4}$$

Induction Base: If $k = 0$ (there is no common intermediate node), (4) holds as before the first iteration the algorithm uses a ρ-approximation for FMA2NDP. **Induction Step:** Assume that the statement (4) holds for the case where all common nodes between the two paths are among $\{v_1, v_2, ..., v_{k-1}\}$. This means that for all $u, v \in V$, $x_u, x_v \in D$ after $k - 1$ iterations of the algorithm, we have $\bar{C}_{k-1}(u, x_u, v, x_v) \ \leq \ \rho\, \bar{C}_{k-1}^{OPT}(u, x_u, v, x_v)$.

Now consider the k-th iteration, where v_k is considered as additional intermediate node for two edge-disjoint paths from u, activated with at least x_u, to v, activated with at least x_v:

If the optimum solution for the two edge-disjoint uv-paths with common nodes among $\{v_1, \ldots, v_k\}$ uses only nodes among $\{v_1, v_2, .., v_{k-1}\}$ as common nodes, by induction hypothesis, (4) holds as $\bar{C}_k(u, x_u, v, x_v) \leq \bar{C}_{k-1}(u, x_u, v, x_v)$.

If the optimum solution for the two edge-disjoint uv-paths with common nodes among $\{v_1, \ldots, v_k\}$ uses the node v_k with activation value x_{v_k} as common node, then we have:

$$\begin{aligned}
\bar{C}_k(u, x_u, v, x_v) &\leq \bar{C}_{k-1}(u, x_u, v_k, x_{v_k}) + \bar{C}_{k-1}(v_k, x_{v_k}, v, x_v) + x_{v_k} \\
&\leq \rho\, \bar{C}_{k-1}^{OPT}(u, x_u, v_k, x_{v_k}) + \rho\, \bar{C}_{k-1}^{OPT}(v_k, x_{v_k}, v, x_v) + x_{v_k} \\
&\leq \rho \left(\bar{C}_{k-1}^{OPT}(u, x_u, v_k, x_{v_k}) + \bar{C}_{k-1}^{OPT}(v_k, x_{v_k}, v, x_v) + x_{v_k} \right) \\
&= \rho\, \bar{C}_k^{OPT}(u, x_u, v, x_v)\,.
\end{aligned}$$

This completes the proof of the theorem. □

From Theorem 1 and Theorem 2, we obtain the following corollary.

Corollary 2. *The MA2EDP algorithm computes a* 1.5*-approximate solution for the MA2EDP problem.*

We remark that for the special case of power optimization, [6, 12] gave an exact polynomial-time algorithm for the MAkNDP problem in directed graphs. Theorem 2 thus implies that there is an exact polynomial-time algorithm for the directed MA2EDP problem for power optimization, as was already shown in [12]. Nutov [8] shows that for arbitrary k there exists a k-approximation algorithm for the directed case of the MAkEDP problem.

4 Activation Networks with $|D| = 2$

In this section, we restrict the domain D to have size 2. This case is interesting from a theoretical point of view because it is the smallest non-trivial case for the size of the domain. From a practical point of view, this case corresponds to a simple setting where nodes have just two different activation states, e.g., low power and high power. Let $D = \{a, b\}$ with $a < b$. Note that the cost of a solution that activates B nodes with activation value b and $|V| - B$ nodes with activation value a is $a|V| + B(b - a)$. This means that minimizing the activation cost is equivalent to minimizing the number of nodes that have activation value b. In the rest of this section, we assume that all edges of the activation network have the same activation function $f : D^2 \to \{0, 1\}$.

4.1 Polynomial Cases of MAkNDP and MA2EDP

The following are all the different possibilities for a monotone non-decreasing activation function f with domain $D = \{a, b\}$:

1. $f(a, a) = 1$
2. $f(a, a) = f(a, b) = f(b, a) = 0, f(b, b) = 1$
3. $f(a, a) = f(a, b) = 0, f(b, a) = f(b, b) = 1$
4. $f(a, a) = f(b, a) = 0, f(a, b) = f(b, b) = 1$
5. $f(a, a) = 0, f(a, b) = f(b, a) = f(b, b) = 1$

The problem MAkNDP for activation function 1 is trivial as either the solution that activates all nodes with activation value a is optimal, or there is no feasible solution. For activation functions 2-4, we observe that the problem of minimizing the activation cost of node-disjoint paths from s to t is equivalent to the problem of minimizing the number of nodes used by the paths: For activation function 2, all nodes on all paths must be activated with value b. For activation functions 3, all nodes on all paths except node t must be activated with value b. For activation function 4, all nodes on all paths except node s must be activated with value b. To calculate the optimal solution in these cases, we first give unit cost to all edges of the graph, compute k node-disjoint paths of minimum edge-cost using a known polynomial-time algorithm for the minimum cost k-flow problem (with unit edge and node capacities), and finally activate the resulting network. Therefore, the MAkNDP problem with activation functions 1-4 can be solved in polynomial time for any k.

The problem with activation function 5 is polynomial-time solvable for $k = 2$. Assume that the minimum number of internal nodes (nodes excluding s, t) used by two node-disjoint paths from s to t is $M \geq 1$. We activate s with value b, and every other node on each of the two paths. If M is odd, one of the two paths must have an odd number o of internal nodes, the other an even number e of internal nodes, $M = o + e$. In total we activate $1 + (o + 1)/2 + e/2 = M/2 + 1.5$ nodes with value b, and this is optimal and independent of o, e. If M is even, there are two cases: If the two paths both have odd numbers of internal nodes, say o_1 and o_2, we activate $2 + (o_1 - 1)/2 + (o_2 - 1)/2 = M/2 + 1$ nodes with value b. If the two paths both have even numbers of internal nodes, say e_1 and e_2, we activate $1 + e_1/2 + e_2/2 = M/2 + 1$ nodes with value b. In both cases the optimal number of nodes activated with value b depends only on M. Thus, the MA2NDP problem for activation function 5 can also be solved by minimizing the number of nodes used by the two paths.

Note that minimizing the number of nodes used by the paths is not sufficient for activation function 5 and $k = 3$. If the three node-disjoint paths have 3, 1 and 1 internal nodes, respectively, the number of nodes that must be activated with value b is 3. If the three node-disjoint paths have 2, 2 and 1 internal nodes, respectively, the number of nodes that must be activated with value b is 4. In both cases the total number of nodes used by the three paths is the same, but only one of the two cases yields a solution with optimal activation cost.

For all five activation functions, it is easy to see that FMA2NDP is also polynomial-time solvable, and hence MA2EDP can be solved optimally in polynomial time by application of Theorem 2.

4.2 Hardness of MAkEDP

Panigrahi [11] showed that the MAkEDP problem is NP-hard since it generalizes the Node-Weighted k-Flow (NWkF) problem which is known to be NP-hard [10]. Nutov [10] proved the inapproximability of the NWkF problem by a reduction from the bipartite densest ℓ-subgraph problem to unit weight NWkF, and Panigrahi [11] observed that this inapproximability result can be adapted to MAkEDP as well. The reduction described in [10] uses parallel edges. Here, we use a similar approach that avoids parallel edges and establishes that the MAkEDP problem is NP-hard even in the case where $|D| = 2$, all edges have the same activation function, and there are no parallel edges. This is in contrast to the polynomial-time solvability of MAkNDP when $|D| = 2$ for activation functions 1–4 and arbitrary k. We show the hardness of the MAkEDP problem by giving a reduction from the decision version of the maximum balanced complete bipartite subgraph problem, which is NP-hard (Problem GT24 in [5]).

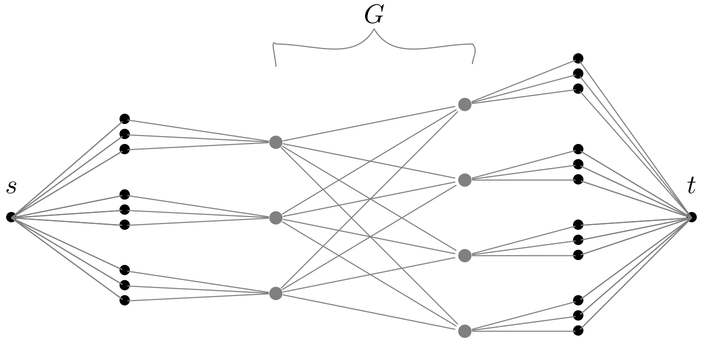

Fig. 2. A reduction of MaxBCBS (g=3)

Maximum Balanced Complete Bipartite Subgraph (MaxBCBS). Given a bipartite graph $G = (V_1 \cup V_2, E)$, find a maximum balanced complete bipartite subgraph (i.e. with the maximum number of nodes). Here, a balanced complete bipartite subgraph H is a complete bipartite subgraph such that $|H \cap V_1| = |H \cap V_2|$.

In the decision version of MaxBCBS, we are additionally given a parameter g and the question is to decide whether G contains a balanced complete bipartite subgraph with $2g$ nodes (and g^2 edges).

The reduction. Let a bipartite instance of the decision version of MaxBCBS be given by $G = (V_1 \cup V_2, E)$ and parameter g. We construct an instance $\bar{G} = (\bar{V}, \bar{E})$ of MAkEDP as follows: We add to G as new nodes a source s and a target t. For each $v \in V_1 \cup V_2$ we add g new nodes $\{v_i : 1 \leq i \leq g\}$. For each $v \in V_1$, we add the edges $\{sv_i : 1 \leq i \leq g\} \cup \{v_i v : 1 \leq i \leq g\}$. For each $v \in V_2$, we add the edges $\{vv_i : 1 \leq i \leq g\} \cup \{v_i t : 1 \leq i \leq g\}$. The domain is $D = \{0, 1\}$, and for all $uv \in \bar{E}$, let $f_{uv}(1,1) = 1$ and $f_{uv}(0,0) = f_{uv}(1,0) = f_{uv}(0,1) = 0$. See Fig. 2. The reduction can be used for both directed and undirected graphs. We can show the following lemma and theorem.

Lemma 3. *There exists a balanced complete bipartite subgraph $K_{g,g}$ with $2g$ nodes in G if and only if there exist $k = g^2$ edge-disjoint paths in $\bar{G}$ of activation cost $2g^2 + 2g + 2$.*

Theorem 3. *The MAkEDP problem is NP-hard even for activation networks where the domain D is $\{0, 1\}$ and all edges $uv \in E$ have the same activation function $f : D^2 \rightarrow \{0, 1\}$.*

5 Conclusion

We have investigated the problem of finding disjoint *st*-paths of minimum activation cost in a given activation network. We gave a 1.5-approximation algorithm for the MA2NDP problem and showed that a ρ-approximation algorithm for the

FMA2NDP problem (MA2NDP with fixed activation values of s and t) can be used to obtain a ρ-approximation algorithm for the MA2EDP problem. For the restricted version of activation networks with $|D| = 2$ and a single activation function for all edges, we showed that MAkNDP can be solved in polynomial-time for arbitrary k (for $k = 2$ in one of the cases for the activation function). In addition, we showed that this restricted version of the MAkEDP problem is NP-hard.

The main open problem is to determine whether the MA2NDP problem is NP-hard. Our results show that a polynomial-time optimal algorithm for MA2NDP would imply a polynomial-time optimal algorithm for MA2EDP. It would also be interesting to study the case of domain size 2 in a setting where different edges can have different activation functions.

References

1. Ahuja, R. K., Magnanti, T. L., Orlin, J. B.: Network Flows: Theory, Algorithms, and Applications. Prentice Hall, New Jersey (1993)
2. Calinescu, G., Kapoor, S., Olshevsky, A., Zelikovsky, A.: Network lifetime and power assignment in ad hoc wireless networks. In: ESA 2003, LNCS, vol. 2832, pp. 114–126. Springer, Heidelberg (2003)
3. Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C.: Introduction to Algorithms. Second Edition, The MIT Press and McGraw-Hill Book Company (2001)
4. Feige, U., Kogan, S.: Hardness of approximation of the balanced complete bipartite subgraph problem. Technical report MCS04-04, Department of Computer Science and Applied Math., The Weizmann Institute of Science (2004).
5. Garey, M. R., Johnson, D. S.: Computers and Intractability. A Guide to the Theory of NP-Completeness. W. H. Freeman and Company, New York–San Francisco (1979)
6. Hajiaghayi, M. T., Kortsarz, G., Mirrokni, V. S., Nutov, Z.: Power optimization for connectivity problems. In: IPCO XI, LNCS, vol. 3509, pp. 349–361. Springer, Heidelberg (2005)
7. Lando, Y., Nutov, Z.: On minimum power connectivity problems. In: ESA 2007, LNCS, vol. 4698, pp. 87–98. Springer, Heidelberg (2007)
8. Nutov, Z.: Approximating minimum power covers of intersecting families and directed connectivity problems. In: APPROX 2006, LNCS, vol. 4110, pp. 236–247. Springer, Heidelberg (2006)
9. Nutov, Z.: Survivable network activation problems. In: LATIN 2012, LNCS, vol. 7256, pp. 594–605. Springer, Heidelberg (2012)
10. Nutov, Z.: Approximating Steiner networks with node-weights. SIAM J. Comput. 39(7), 3001–3022 (2010)
11. Panigrahi, D.: Survivable network design problems in wireless networks. In: 22nd Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1014–1027. SIAM (2011)
12. Srinivas, A., Modiano, E.: Finding Minimum Energy Disjoint Paths in Wireless Ad-Hoc Networks. Wireless Networks 11(4), 401–417 (2005)
13. Suurballe, J. W.: Disjoint paths in a network. Networks 4, 125–145 (1974)
14. Suurballe, J. W., Tarjan, R. E.: A quick method for finding shortest pairs of disjoint paths. Networks 14, 325–336 (1984)