THE ROYAL SWEDISH ACADEMY OF SCIENCES

INSTITUT MITTAG-LEFFLER

Auravägen 17, SE-182 60 Djursholm, Sweden
Tel. +46 8 622 05 60 Fax. +46 8 622 05 89
info@mittag-leffler.se www.mittag-leffler.se

Balanced branchings in digraphs

J. Bang-Jensen and A. Yeo

REPORT No. 23, 2013/2014, spring

ISSN 1103-467X
ISRN IML-R- -23-13/14- -SE+spring

Balanced branchings in digraphs

J. Bang-Jensen* A. Yeo†

August 20, 2014

Abstract

We study out-branchings from a given root s which satisfy certain balance conditions, such as balancing the number of arcs out of s, the number of vertices in each subtree of the branching etc. We obtain polynomial algorithms for several of the problems and study the bordeline between polynomial and NP-complete instances of the problems.

Keywords: branchings, out-trees, balancing degrees, balancing vertices, NP-complete, polynomial algorithm.

1 Introduction

Notation follows [1]. We denote the vertex set and arc set of a digraph D by $V(D)$ and $A(D)$, respectively and write $D = (V, A)$ where $V = V(D)$ and $A = A(D)$. The digraphs may have parallel arcs but no loops. Paths and cycles are always directed unless otherwise specified. We will use the notation $[k]$ for the set of integers $\{1, 2, \ldots, k\}$.

An (s, t)-path in a digraph D is a directed path from the vertex s to the vertex t. The **underlying graph** of a digraph D, denoted $UG(D)$, is obtained from D by suppressing the orientation of each arc and deleting multiple edges. A digraph D is **connected** if $UG(D)$ is a connected graph. When xy is an arc of D we say that x **dominates** y and call y an **out-neighbour** of x. For a digraph $D = (V, A)$ the **out-degree**, $d^+_D(x)$ (resp. the **in-degree**, $d^-_D(x)$) of a vertex $x \in V$ is the number of vertices y in V such that xy (resp. yx) is an arc of A. For a vertex $v \in V(D)$ we denote by $N^+(v)$ the set of out-neighbours of v. When $X \subseteq V$ shall also write $d^+_X(v)$ to denote the number vertices in X that are dominated by v.

An **s-out-tree** is a connected digraph where one vertex (the root s) has in-degree zero and all other vertices have in-degree exactly one. An **s-out-branching** rooted at s in a digraph D is a spanning out-tree with *root* s. We will often use the notation B^+_s to denote an s-out-branching. See e.g. [1, Chapter 9] for a collection of results on branchings in digraphs.

Branchings in digraphs are important not only from a theoretical point of view but also in practical applications where messages are often sent via the arcs of a branching, Thus it is natural to study various quality measures of branchings and this has been done in a number of papers, see e.g. [3, 5, 6, 7]. In terms of protection against arc-faults, branchings are not a very good way of sending information from one source to all other vertices since deleting just one arc, we may in the worst case disconnect the root s from all other vertices. On the other hand the sparse structure of a branching is attractive and hence it is interesting to study various measures that consider how safe a given branching is against arc failures.

*Department of Mathematics and Computer Science, University of Southern Denmark, Odense DK-4230, Denmark (email:jbj@imada.sdu.dk). Parts of this work was done while the first author attended the program "Graphs, hypergraphs and computing" at Institute Mittag-Leffler, spring 2014. The research of Bang-Jensen was also supported by the Danish research council under grant number 1323-00178B

†Engineering Systems and Design, Singapore University of Technology and Design, 20 Dover Drive, 138682 Singapore, Singapore and Department of Mathematics, University of Johannesburg, Auckland Park, 2006, South Africa (email andersyeo@gmail.com).The research of Yeo was supported by the Danish research council under grant number 1323-00178B

Let $T^+_{s,1}, T^+_{s,2}$ be out-trees with only the vertex s in common. By the **union** of $T^+_{s,1}, T^+_{s,2}$ we mean the out-tree T^+_s whose arc-set is $A(T^+_{s,1}) \cup A(T^+_{s,2})$.

Definition 1.1 *Let k be a natural number and let $D = (V, A)$ be a digraph with a non-negative weight function w on V such that $w(s) = 0$. A branching B^+_s is* **k-safe** *if deleting any arc sv and every vertex in the out-tree rooted at v from $A(B^+_s)$ results in an out-tree T^+_s whose weight is at least k. Hence in the unweighted case the remaining s-out-tree has at least $k + 1$ vertices including s.*

Definition 1.2 *A branching B^+_s is* **k-balanced** *if it is the union of two out-trees $T^+_{s,1}, T^+_{s,2}$ with $V(T^+_{s,1}) \cap V(T^+_{s,2}) = \{s\}$ such that $|V(T^+_{s,i})| \geq k + 1$ for $i = 1, 2$. So when the weight function w above assigns weight 1 to every $v \neq s$, every k-balanced out-branching is k-safe but the other direction is not always true.*

In the first part of the paper we study problems concerning balanced branchings.

2 Balancing out-branchings

For a given pair of vertices p, q in a digraph D we denote by $\kappa(p, q)$ the maximum number of internally vertex disjoint (p, q)-paths. It is well known (see e.g. [1, Section 5.4]) that we can find $\kappa(p, q)$ as well as a set of $\kappa(p, q)$ internally disjoint (p, q)-paths in polynomial time using flows. Below we will only need to check whether $\kappa(p, q) \geq 2$ and this can be done in linear time.

Theorem 2.1 *Let $D = (V, A)$ be a digraph and let $s \in V$ be a vertex such that $D - s$ is strong and $\kappa(s, t) \geq 2$ for every $t \in V \setminus N^+(s)$. For every $i \in [n - 1]$ D contains an out-branching B^+_s which is the union of arc-disjoint s-out-trees $T^+_{s,1}, T^+_{s,2}$ with $|A(T^+_{s,1})| = i$ and $d^+_{T^+_{s,1}}(s) = 1$ (that is, there is precisely one arc out of s in $T^+_{s,1}$).*

Proof: We give an inductive construction for all i. For the base case, let v be any out-neighbour of s, let $A(T^{(1)+}_{s,1}) = \{sv\}$ and let $T^{(1)+}_{s,2}$ be an out-branching from s in $D - v$ which exists since $\kappa(s, t) \geq 2$ for all $t \in V \setminus N^+(s)$. Suppose we have constructed pairs $[T^{(1)+}_{s,1}, T^{(1)+}_{s,2}], \ldots, [T^{(k)+}_{s,1}, T^{(k)+}_{s,2}]$, where $|A(T^{(i)+}_{s,1})| = i$, s has out-degree one in $T^{(i)+}_{s,1}$ and the union of $T^{(i)+}_{s,1}$ and $T^{(i)+}_{s,2}$ forms an out-branching for $i \leq k$. Let $X = V(T^{(k)+}_{s,1}) - s$ and $Y = V(T^{(k)+}_{s,2})$. Because $D - s$ is strong there is at least one arc from X to Y. If D contains an arc ab with $a \in X$, $b \in Y$ such that b is a leaf in $T^{(k)+}_{s,2}$, then $T^{(k+1)+}_{s,1} = T^{(k)+}_{s,1} + \{ab\}$ and $T^{(k+1)+}_{s,2} = T^{(k)+}_{s,2} - b$ are the desired out-trees. Thus we may assume that there is no arc from X to a leaf in $T^{(k)+}_{s,2}$. Let $v \in V(T^{(k)+}_{s,2})$ be chosen so that D contains an arc uv with $u \in X$, but there is no arc from X to any vertex in $C = V(T^+_v) - v$, where T^+_v is the sub-out-tree rooted at v in $T^{(k)+}_{s,2}$.

Lemma 2.2 *Let $D = (V, A)$ be a digraph with a non-negative weight function w on its vertices and let s be a vertex of V such that $w(s) = 0$ and $\kappa(s, v) \geq 2$ for all $v \in V \setminus N^+(s)$. Let $w^* = \max_{v \in V} w(v)$ and $W = \sum_{v \in V} w(v)$. For every $k \leq \min\{\frac{W - w^*}{2}, w^*\}$ D contains a k-balanced out-branching B^+_s. Furthermore, we can construct such an out-branching in polynomial time.*

Proof: Let D, s and w be as above. Let $p \neq s$ be a vertex with $w(p) = w^*$. If p is an out-neighbour of s we get the desired branching by taking the union of a spanning s-out-tree in $D - p$ and the arc sp so we may assume the p is not an out-neighbour of s. As $\kappa(s, p) \geq 2$ there are internally disjoint (s, p)-paths P_1, P_2. Let $p_i \neq s$ be the predecessor of p on P_i, $i = 1, 2$. Consider the union of $P'_1 = P_1 - p$ and $P'_2 = P_2 - p$ as an out-tree T^+_s in $H = D - p$. By Menger's theorem and the assumption of the lemma, s can reach every vertex in $H - p$. Hence we can extend T^+_s to an out-branching of $H - p$ containing all arcs of T^+_s. This branching will be the union of two out-trees $T^+_{s,1}, T^+_{s,2}$ where P'_i is contained in $T^+_{s,i}$ for $i = 1, 2$. Let $M = \max\{w(V(T^+_{s,1})), w(V(T^+_{s,2}))\}$. Then $M \geq \frac{W - w^*}{2}$ so if we add the arc $p_j p$ to $T^+_{s,j}$, where $j \in [2]$ is chosen so that $w(V(T^+_{s,j})) \leq w(V(T^+_{s,3-j}))$, we obtain the desired out-branching.

Furthermore it is not difficult to see that all the above steps can be done in polynomial time. $\diamond$

Theorem 2.3 *Let $D = (V, A)$ be a digraph containing a vertex s such that $\kappa(s, t) \geq 2$ for every $t \in V \setminus N^+(s)$. Let w be a non-negative weight function on V such that $w(s) = 0$. Let $w^* = \max_{v \in V} w(v)$ and $W = \sum_{v \in V} w(v)$. If $w^* \leq W/3$ then D contains a $\frac{W}{3}$-balanced s-out-branching B_s^+. Furthermore, such a branching can be constructed in polynomial time.*

Proof: Suppose first that $D - s$ is strong and consider the algorithm which follows from the constructive proof of Theorem 2.1 above and let k be such that $w(T_{s,1}^{(k)+}) < \frac{W}{3}$ and $w(T_{s,1}^{(k+1)+}) \geq \frac{W}{3}$. Since no weight is larger than $\frac{W}{3}$, we will have $\min\{w(T_{s,1}^{(k+1)+}), w(T_{s,2}^{(k+1)+})\} \geq \frac{W}{3}$. Hence the branching formed by the union of these two out-trees is $\frac{W}{3}$-safe.

Now consider the case when $D - s$ is not strong. Then while following the algorithm above, we may reach a situation where there is no arc from the current set X to Y and $w(X) < \frac{W}{3}$ (otherwise we are done). In this case s will have two internally disjoint paths completely inside Y to any $t \in Y$ which is not an out-neighbour of s as no such path in D can use a vertex of X. In particular, the out-degree of s in $D\langle Y \cup \{s\}\rangle$ is at least 2. Let v be the unique out-neighbour of s in X and let T_v^+ be a spanning out-tree rooted at v in $D\langle X\rangle$.

If the maximum weight of a vertex in $D' = D\langle Y \cup \{s\}\rangle$ is at most $\frac{w(Y)}{3}$ then we can apply induction and get a k'-balanced s-out-branching $\hat{B}_s^+$ of D' where $k' \geq \frac{w(Y)}{3}$. Let $T_{s,1}^+, T_{s,2}^+$ be out-trees in D' such that $\hat{B}_s^+$ is the union of these, both of which have weight at least $\frac{w(Y)}{3}$ and at most $\frac{2w(Y)}{3}$. Furthermore, since $w(Y) \geq \frac{2W}{3}$ at least one of $T_{s,1}^+, T_{s,2}^+$, say $T_{s,j}^+$ has weight at least $\frac{W}{3}$ so by adding the out-tree consisting of T_v^+ and the arc sv to $T_{s,3-j}^+$ we obtain the desired s-out-branching.

Finally assume that the maximum weight of some vertex p in $D' = D\langle Y \cup \{s\}\rangle$ is larger than $\frac{w(Y)}{3}$. Note that, by the assumption of the theorem $w(p) \leq \frac{w(X)+w(Y)}{3}$. If $p \in N^+(s)$, then we just add p to the s-out-tree containing X which we constructed above and let $\tilde{T}_s^+$ be an s-out-tree covering exactly $Y \cup \{s\} - \{p\}$. Hence we may assume that $p \notin N^+(s)$. Apply the proof idea of Lemma 2.2: Let P_1, P_2 be internally disjoint (s, p)-paths in $D' = D\langle Y \cup \{s\}\rangle$ and let p_i be the predecessor of p on P_i. Consider the digraph $H' = D' - p$ and a pair of arc-disjoint out trees $\hat{T}_{s,1}^+, \hat{T}_{s,2}^+$ of H' whose union form an s-out-branching in H' with the property that $P_i - p$ is contained in $\hat{T}_{s,i}^+$ for $i = 1, 2$. In particular, we can add p to any of the two trees using one of the arcs p_1p, p_2p. We claim that adding all vertices of X to one of $\hat{T}_{s,1}^+, \hat{T}_{s,2}^+$ and p to the other or both to the same out-tree will produce the desired branching (as the union of these two s-out-trees). To see this first assume that vertices in one of the two trees $\hat{T}_{s,1}^+, \hat{T}_{s,2}^+$, say w.l.o.g. $\hat{T}_{s,1}^+$ has weight at least $\frac{W}{3}$. Then we add p and X to $\hat{T}_{s,2}^+$. The resuting out-tree will also have weight at least $\frac{W}{3}$ since $w(p) > \frac{w(Y)}{3}$ implying that the weight of $\hat{T}_{s,1}^+$ is at most $\frac{2w(Y)}{3} \leq 2\frac{W}{3}$. So we may assume that each of $\hat{T}_{s,1}^+, \hat{T}_{s,2}^+$ have weight less than $\frac{W}{3}$. Assume again that the weight of $\hat{T}_{s,1}^+$ is at least that of $\hat{T}_{s,2}^+$. Then adding p to $\hat{T}_{s,1}^+$ and all vertices of X to $\hat{T}_{s,2}^+$ produces the desired s-out-trees whose union is the out-branching we need (here we used again that $w(p) \leq \frac{W}{3}$).

It is not difficult to see that the desired branching can be found in polynomial time. ◇

Theorem 2.4 *For every constant c and every $k \leq c \log n$ the following problem is polynomially time solvable. If $D = (V, A)$ is a digraph on $n \geq 3c \log n + 1$ vertices and $s \in V$ then decide if D has a k-balanced out-branching B_s^+.*

Proof: Let D and s be given. We may assume that D has an s-out-branching as otherwise the desired out-branching cannot exist. Let w be the weight function that assigns weight 1 to all vertices of $D - s$ and zero to s.

If $\kappa(s, t) \geq 2$ for all $t \in V \setminus N^+(s)$, then Theorem 2.3 applies and we can find the desired branching in polynomial time.

Suppose that some vertex v separates s from a vertex $t \in V \setminus N^+(s)$ in D. Let A be the set of vertices that s can reach in $D - v$ and let $B = V - A - v$. Then there is no arc from A to B, implying that all vertices in $B \cup \{v\}$ will be part of one branch (rooted at v) of any out-branching from s in D. Note that, because s can reach every vertex in D there is a spanning out-tree T_v^+ in $D\langle B \cup \{v\}\rangle$. Contract $B \cup \{v\}$ into one vertex v', delete any multiple arcs appearing and assign weight $w(v') = \sum_{u \in B \cup \{v\}} w(u)$ to

v'. Repeat the process above as long as we do not have $\kappa(s,t) \geq 2$ for all $t \notin N^+(s)$ in the current digraph and call the resulting weighted digraph H. Let $X = \{t \in N^+(s) | \kappa_H(s,t) = 1\}$ be the set of out-neighbours of s in H which can only be reached from s by the direct arc. Let $Y = V(H) - X - s$. Note that we may have $X = \emptyset$.

The contractions above may result in many vertices having weight more than one, but the sum of all weights equals $n-1$ and $w(s) = 0$ still holds. Let $\hat{w}$ denote the largest weight and let p be a vertex with $w(p) = \hat{w}$.

If $\hat{w} > n - c\log n$ then D does not have the desired branching so we assume that $\hat{w} \leq n - c\log n$. Suppose first that $\hat{w} \leq n - 2c\log n$. If $\hat{w} \leq c\log n$ then we may apply Theorem 2.3 since the total weight is $n - 1 \geq 3\hat{w}$; so we may assume that $\hat{w} > c\log n$.

If $p \in Y$, then apply the algorithm of Lemma 2.2 to obtain a k-balanced out-branching $\hat{B}_s^+$ of H, where $k = \min\{\frac{n-\hat{w}}{2}, \hat{w}\} \geq c\log n$.

If $p \in X$ then p does not separate s from any other vertex (as the contraction process would have continued in that case). Thus we can take $\hat{B}_s^+$ to be any out-branching of H consisting of the union of the arc sp and a spanning out-tree from s in $H - p$.

By expanding contracted vertices in the reverse order of the contractions above we can turn $\hat{B}_s^+$ into a k-balanced out-branching of D.

Hence we may assume that we have $n - 2c\log n \leq \hat{w} \leq n - c\log n$. Now the digraph H has at most $2c\log n$ vertices other than s (as all weights are at least 1 and the total weight of H is at most $2c\log n$ since $\hat{w} \geq n - 2c\log n$).

For each partition R, S of $V(H) - s$ such that $\min\{w(R+s), w(S+s)\} \geq k$ we can decide in polynomial time whether $H_1 = H\langle R+s\rangle$ and $H_2 = H\langle S+s\rangle$ contain spanning out-trees $T_{s,1}^+$ and $T_{s,2}^+$ respectively. If that is the case the union of $T_{s,1}^+$ and $T_{s,2}^+$ forms a k-balanced s-out-branching of H. By expanding contracted vertices in the reverse order of the contractions above we can obtain the desired out-branching back in D.

By considering all of the at most $2^{|V(H)|} \leq 2^{2c\log n} = n^{2c}$ different subsets of $V(H)$ we either find a k balanced out-branching of D or conclude that none exists (here we use that all vertices contracted into one super vertex must belong to the same branch starting with an arc from s to an out-neighbour in every branching from s). $\diamond$

3 NP-complete instances

We will now prove that under the so-called exponential time hypothesis (ETH), for every $\epsilon > 0$ and every integer function $k(n)$ with $(\log(n))^{1+\epsilon} \leq k(n) \leq \frac{n}{2}$ (and $k(i+1) \leq k(i)$ for all large i) there is no polynomial algorithm for deciding whether a given digraph contains a $k(n)$-balanced out-branching from a given root. For example, given any $0 < c < 1/2$ there is no polynomial algorithm to decide whether a given digraph contains a (cn)-balanced out-branching from a given root (assuming ETH is true). Similarly, given any $\epsilon > 0$ there is no polynomial algorithm to decide whether a given digraph contains a $(\log(n))^{1+\epsilon}$-balanced out-branching from a given root (assuming ETH is true).

Exponential Time Hypothesis (ETH): [4] *There is a positive real s such that 3-SAT with n variables and m clauses cannot be solved in time $2^{sn}(n+m)O(1)$.*

If the ETH is true, then there is no algorithm that solves 3-SAT in subexponential time. That is, there is no $2^{o(n)}$ algorithm that solves a 3-SAT instances with n variables. Using the so-called Sparsification Lemma we can obtain the following lemma, where the n in the exponent of ETH has been replaced by m .

Lemma 3.1 *[4] Given ETH, there is a positive real s' such that 3-SAT with n variables and m clauses cannot be solved in time $2^{s'm}(n+m)O(1)$.*

Theorem 3.2 *Assume the ETH holds and let $\epsilon > 0$ and $I > 0$ be arbitrary and let $k(W)$ be an integer function, such that $(log(W))^{1+\epsilon} \leq k(W) \leq W/2$ for all $W > 0$ and for all positive integers i with $i \geq I$, there exists some integer W such that $k(W) = i$.*

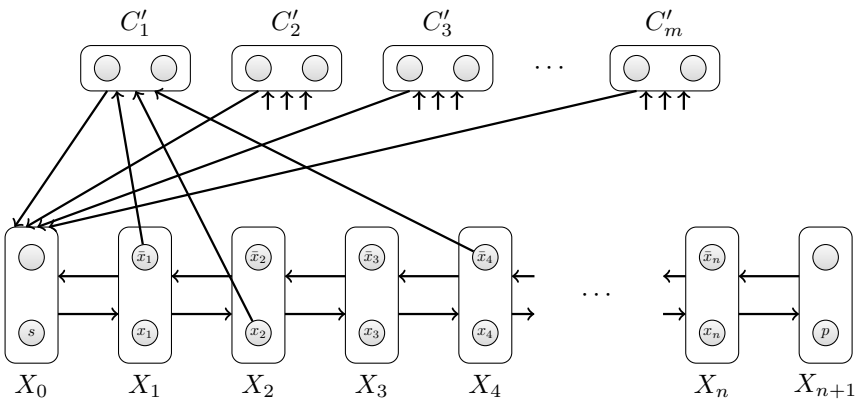

Figure 1: The digraph D, where $C_1 = (\bar{v}_1 \vee v_2 \vee \bar{v}_4)$

Then there is no polynomial (in $W+|V(D)|$) algorithm for finding a $k(W)$-balanced s-out-branching of a digraph D, where D is a 2-strong vertex-weighted digraph, with total vertex weight W and $s \in V(D)$.

Proof: Let Q be an instance of 3-SAT, with variables $v_1, v_2, \ldots, v_n$ and clauses $C_1, C_2, \ldots, C_m$. That is, $Q = C_1 \wedge C_2 \wedge \cdots \wedge C_m$. Assume that $n + 2m + 2 \geq I$. We will construct a vertex weighted digraph D as follows (see Figure 1).

Let $C'_1, C'_2, \ldots, C'_m$ and $X_0, X_1, X_2, \ldots, X_n, X_{n+1}$ be disjoint independent vertex sets of size two. Add all possible arcs between X_{i-1} and X_i (in both directions) to D for all $i = 1, 2, \ldots, n+1$. Let $X_i = \{x_i, \bar{x}_i\}$ for $i = 1, 2, \ldots, n$ and if clause C_j contains literal v_s then add the two arcs from x_s to C'_j and if C_j contains literal $\bar{v}_s$ then add the two arcs from $\bar{x}_s$ to C'_j (for all $j = 1, 2, \ldots, m$). Finally add all possible arcs from $C'_1 \cup C'_2 \cup \cdots C'_m$ to X_0. It is easy to check that that D is 2-strong.

We will now define the vertex-weights, w, of D. Let W be the integer, such that $k(W) = n+2m+2$, which is possible as $n + 2m + 2 \geq I$. By the definition of $k(W)$ we note that the following holds.

$$(log(W))^{1+\epsilon} \leq k(W) = n + 2m + 2 \leq \frac{W}{2}$$

Let $s \in X_0$ and $p \in X_{n+1}$ be arbitrary and let the weight of all vertices in $V(D) \setminus \{s, p\}$ be one and let $w(s) = 0$ and $w(p) = W - (|V(D)| - 2) = W - (2n + 2m + 2)$. Note that the sum of all weights in D is W and $w(p) \geq 2n + 4m + 4 - (2n + 2m + 2) = 2m + 2 > 3$ (as $n + 2m + 2 \leq W/2$).

We will now consider the problem of finding a $k(W)$-balanced s-out-branching of the digraph D. As any path from s to p in D has weight at least $n + w(p) \geq n + 2m + 2 = k(W)$, we note that this is equivalent to finding an s-out-tree, T_s^+, in D containing p, such that $D - (V(T_s^+) \setminus \{s\})$ has an s-out-branching B_s^+ of weight at least $k(W) = n + 2m + 2$.

Any s-out-tree, T_s^+, in D containing p must contain at least one vertex from each set $X_0, X_1, X_2, \ldots, X_{n+1}$. Furthermore if T_s^+ contains more than these $n + 2$ vertices then the weight of the vertices in $V(D) \setminus (V(T_s^+) \setminus \{s\})$ has weight less than $n + 2m + 2$, so we only need to consider out-trees, T_s^+, which form a path from $s \in X_0$ to $p \in X_{n+1}$ passing through $X_0, X_1, X_2, \ldots, X_n, X_{n+1}$ in this order picking up exactly one vertex in each set. Since the only connection to the vertices in the sets C'_i, $i \in [m]$ are from the sets $X_1, \ldots, X_n$ it is easy to see that T_s^+, B_s^+ exist if and only if D contains an (s, p)-path which avoids at least one of the X-vertices corresponding to each clause.

Suppose first that D has arc-disjoint s-out-trees $T_{s,1}^+, T_{s.2}^+$ whose union forms a $k(W)$-balanced s-out-branching and $p \in V(T_{s,1}^+)$. If $T_{s,1}^+$ does not contain the vertex x_i then let $v_i = True$ and if T_s^+ does not contain the vertex $\bar{x}_i$ then let $v_i = False$. If we have some clause, say $C_i = (v_2 \vee \bar{v}_4 \vee v_7)$, then as we argued above, since the clause vertices are all in $T_{s,2}^+$, either v_2 is true ($x_2 \notin V(T_s^+)$) or $\bar{v}_4$ is true ($\bar{x}_4 \notin V(T_s^+)$) or v_7 is true ($x_7 \notin V(T_s^+)$). Therefore the truth assignments given above will satisfy Q. Conversely if Q is satisfied by some truth assignment, then for each clause C'_i fix one of its litterals z_i which is true. Now let $T_{s,1}^+$ be the path from s to p such that if $v_i = True$ then $T_{s,1}^+$ contains $\bar{x}_i$ and if

$v_i = False$ then $T^+_{s,1}$ contains x_i (it avoids the vertices corresponding to true litterals). Furthermore let $T^+_{s,2}$ consist of the (X_0, X_{n+1}) path which picks up the remaining vertices of $X_0 \cup \ldots \cup X_{n+1}$ together with the two arcs from z_i to C'_i for $i \in [m]$, gives us the desired s-out-branching.

For the sake of contradiction assume that there is a polynomial (in $W + |V(D)|$) algorithm $\mathcal{A}$ for finding a $k(W)$-balanced s-out-branching of D. Note that $|V(D)| = 2n + 2m + 4 \leq W$ (as $W = |V(D)| - 2 + w(p)$ and $w(p) > 3$). Therefore the algorithm $\mathcal{A}$ would have complexity $O(W^c)$, for some constant c. As $(log(W))^{1+\epsilon} \leq k(W)$ we note that $log(W) \leq k(W)^{1/(1+\epsilon)}$, which gives us that $\mathcal{A}$ has the following complexity.

$$O(W^c) = O\left(\left(2^{(k(W)^{1/(1+\epsilon)})}\right)^c\right) = O\left(\left(2^{c((n+2m+2)^{1/(1+\epsilon)})}\right)\right)$$

As $n \leq 3m$, we have $n + 2 \leq 5m$ which implies that the running time of $\mathcal{A}$ at most $O(2^{c(7m)^{\epsilon'}})$, where $\epsilon' = 1/(1+\epsilon) < 1$. This is a contradiction to Lemma 3.1. $\diamond$

Note that the function $k(W) = \lfloor c \times W \rfloor$, where $0 < c \leq 0.5$ satisfies the statement of Lemma 3.2. Also the function $k(W) = \lceil (log(W))^{1+\epsilon} \rceil$ satisfies the statement of Lemma 3.2 for $\epsilon > 0$.

Theorem 3.3 *Let $1/3 < c < 1/2$ be arbitrary. Unless $P = NP$, there is no polynomial (in $W + |V(D)|$) algorithm for finding a (cW)-balanced s-out-branching of a digraph D, where D is a 2-strong vertex-weighted digraph, with total vertex weight W and maximum vertex-weight at most $W/3$ and $s \in V(D)$.*

Proof: We will reduce from the 2-linkage problem in 2-strong digraphs, which is NP-hard [8]. Let D_2 be an instance of 2-linkage, where D_2 is 2-strong, and $\{x_1, x_2, y_1, y_2\} \subseteq V(D_2)$. We want to decide if there exists vertex-disjoint (x_1, y_1)- and (x_2, y_2)-paths in D_2. Let $n_2 = |V(D_2)|$.

Create a new digraph D from D_2 by adding a new vertex s and arcs sx_1 and sx_2 and all arcs from $V(D_2)$ to s. Let $q = n_2 \times \max\{(1-c)/(6c-2), c/(3-6c)\}$. Let all vertices in $V(D) \setminus \{s, x_1, y_1, x_2, y_2\}$ have weight one and let $w(s) = 0$, $w(x_1) = q+1$, $w(y_1) = 2q+1$, $w(x_2) = 2q+1$ and $w(y_2) = q+1$. Let $W = \sum_{v \in V(D)} w(v) = n_2 + 6q$. We will now show that there exists a (cW)-balanced s-out-branching of D if and only if there are vertex-disjoint (x_1, y_1)- and (x_2, y_2)-paths in D_2. First we need the following equations which follows from the definition of q.

$$\begin{array}{rcll} (1-c)n_2 & \leq & q(6c-2) & \text{and} \qquad cn_2 \leq q(3-6c) \\ \Downarrow & & & \\ 2q + n_2 & \leq & c(6q+n_2) & \text{and} \qquad c(6q+n_2) \leq 3q \end{array}$$

As $W = n_2 + 6q$, this implies that $2q + n_2 \leq cW \leq 3q$.

First assume that there are vertex-disjoint (x_1, y_1)- and (x_2, y_2)-paths in D_2. Add the arcs sx_1 and sx_2 to these paths, resulting in an out-tree, T^+_s. Repeatedly add vertices to T^+_s until we obtain an s-out-branching of D. As $w(x_1) + w(y_1) > 3q \geq cW$ and $w(x_2) + w(y_2) > 3q \geq cW$ we have obtained a (cW)-balanced s-out-branching of D as desired.

Conversely assume that B^+_s is a (cW)-balanced s-out-branching of D. As $d^+_D(s) = 2$, we note that x_1 and x_2 are roots of different out-trees in $B^+_s - s$. If y_1 does not belong to the same out-tree as x_1 in $B^+_s - s$, then the out-tree containing x_1 can have weight at most $2q + (n_2 - 1) < cW$ (as x_2 does not belong to this out-tree), contradicting the fact that B^+_s is a k-balanced s-out-branchings of D. Therefore y_1 belongs to the same out-tree as x_1 and there is a (x_1, y_1)-path in B^+_s. Analogously y_2 belongs to the same out-tree as x_2 and there is a (x_2, y_2)-path in B^+_s. These two paths are vertex disjoint, so we have solved the 2-linkage problem.

As $\max\{(1-c)/(6c-2), c/(3-6c)\}$ is a constant (as $1/3 < c < 1/2$) we note that W is linear in n_2. Therefore any polynomial (in $W + |V(D)|$) algorithm for finding a (cW)-balanced s-out-branchings of D would solve our instance of 2-linkage in polynomial time, a contradiction (when $P \neq NP$). $\diamond$

Note that the reason we can prove a much stronger result in Theorem 3.2 than in Theorem 3.3 is that we allow the maximum vertex-weight to be arbitrary in Theorem 3.2 whereas in Lemma 3.3 it is bounded by $W/3$ (In the proof of Lemma 3.2 the vertex p has weight more than $W/3$).

4 Finding out-branchings with allowed sub-out-trees

Let P_k denote a directed path with k vertices. In this section, in order to simplify notation a bit, we sometimes use the name T for both an s-out-tree and an s-out-branching, The vertex s will always be clear from the context.

In this section, we shall determine the computational complexity of a family of problems which include the ones of deciding whether a given un-weighted digraph D contains a k-safe s-out-branching for k close to n. That is, we wish to decide whether some s-out-branching B_s^+ in the given digraph has the property that no matter which arc we delete from B_s^+, the remaining s-out-tree conatins almost all vertices of D.

Given an s-out-branching T and any $v \in V(T)$ let T_v be the maximal sub-out-tree of T rooted at v.

For a given s-out-tree T the **levels** of T are the distance classes from s in the digraph T. Let $L_i(T)$ denote the set of all vertices v in T for which the (s,v)-path in T has exactly i arcs. We say that v is at level i in T. For example, $L_0 = \{s\}$ and $L_1(s) = N^+(s)$.

We consider the following problem $\mathbf{P}_i(\mathcal{T})$ where i is a fixed integer and $\mathcal{T}$ is a fixed finite set of out-trees.

Input: a digraph D and a vertex s of D. Question: does D contain an s-out-branching T, such that for all vertices $v \in L_i(T)$ we have $T_v \in \mathcal{T}$?

Let $K^{\rightarrow}_{1,j}$ be the out-tree containing one vertex with j out-neighbours. Note that $K^{\rightarrow}_{1,0}$ is an isolated vertex and $K^{\rightarrow}_{1,1}$ is an arc.

Theorem 4.1 *The problem $\mathbf{P}_i(\mathcal{T})$ ($i \geq 1$) is polynomial-time solvable if $\mathcal{T} = \{K^{\rightarrow}_{1,0}, K^{\rightarrow}_{1,1}, K^{\rightarrow}_{1,2}, \ldots, K^{\rightarrow}_{1,a}\}$ for some $a \geq 0$ or if $i = 1$ and $\mathcal{T} = \{K^{\rightarrow}_{1,1}\}$.*

In all other cases $\mathbf{P}_i(\mathcal{T})$ is NP-complete.

We will prove Theorem 4.1 in Section 4.1. We now show some corollaries.

Corollary 4.2 *The problem of deciding if a digraph has a $(n-k)$-safe out-branching is polynomial-time solvable if $k \leq 2$ and NP-complete for all $k \geq 3$.*

Proof: If $k = 1$, then the problem is equivalent to $\mathbf{P}_1(\{K^{\rightarrow}_{1,0}\})$, which by Theorem 4.1 is polynomial-time solvable. Alternatively this is equivalent to deciding whether the vertex s has an arc into every other vertex of the digraph.

If $k = 2$, then the problem is equivalent to $\mathbf{P}_1(\{K^{\rightarrow}_{1,0}, K^{\rightarrow}_{1,1}\})$, which by Theorem 4.1 is polynomial-time solvable.

If $k \geq 3$, then the problem is equivalent to $\mathbf{P}_1(\mathcal{T}_k)$, where $\mathcal{T}_k$ is the set of all out-trees with at most k vertices. As $P_k \in \mathcal{T}_k$ we note that this problem is NP-complete by Theorem 4.1. $\diamond$

Let us now consider a generalization of $\mathbf{P}_i(\mathcal{T})$. We are given a fixed root s, a subset $I \subseteq \{1,2,3,\ldots\}$ ($I \neq \emptyset$) and for each $i \in I$ we are given $\mathcal{T}_i$ which is a finite set of out-trees. If $I = \{i_1, i_2, i_3 \ldots\}$, then let $\mathcal{F} = \{\mathcal{T}_{i_1}, \mathcal{T}_{i_2}, \mathcal{T}_{i_3}, \ldots\}$. We say that an s-out-tree T is **$\mathcal{F}$-good** if for all $i \in I$ and all vertices $v \in L_i(T)$ we have $T_v \in \mathcal{T}_i$. The problem $\mathbf{P}(I, \mathcal{F})$ is to decide if a given digraph D contains an $\mathcal{F}$-good s-out-branching.

If T is an s-out-tree and there exists an index $i \in I$ and a vertex $v \in L_i(T)$ with $T_v \notin \mathcal{T}_i$, then we say that T **violates** the $\mathcal{T}$-constraint at level i. If the level i is clear from the context, in particular, if we are dealing with problem $\mathbf{P}_i(\mathcal{T})$, or if we are considering the $\mathcal{T}_i$-constraint at level i in problem $\mathbf{P}(I, \mathcal{F})$, we often don't specify the level.

Theorem 4.3 *Any problem $\mathbf{P}(I, \mathcal{F})$ is equivalent to some problem $\mathbf{P}_i(\mathcal{T})$ for some integer i and some finite set $\mathcal{T}$ of out-trees.*

Proof: Assume we are given a problem $\mathbf{P}(I, \mathcal{F})$. Let i be the minimum value in I. Let T be any out-tree in $\mathcal{T}_i$ ($\in \mathcal{F}$). We will now decide if T can be deleted from $\mathcal{T}_i$, based on all the other $\mathcal{T}_j$'s.

Take a directed path P of length i and identify the end-vertex of P with the root of T and let the resulting out-tree be denoted by T^*. For each $i' \in I$ with i' at most the depth of T^* we check if T^* violates the $\mathcal{T}_{i'}$-constraint. If the $\mathcal{T}_{i'}$-constraint is violated for some i' then we may remove T from $\mathcal{T}_i$. Let $\mathcal{T}$ contain the remaining out-trees after performing the above pruning.

Note that any out-tree that satisfies the $\mathcal{T}$-constraint at level i also satisfies all the $\mathcal{T}_j$-constraints for all $j \in I$. Furthermore if a tree satisfied the $\mathcal{T}_i$-constraint but not the $\mathcal{T}$-constraint, then the tree must violate some $\mathcal{T}_j$-constraint, for $j > i$. Therefore $\mathbf{P}(I, \mathcal{F})$ is equivalent to $\mathbf{P}_i(\mathcal{T})$. $\diamond$

4.1 Proof of Theorem 4.1

In this section we prove a number of lemmas which together will complete the proof of Theorem 4.1.

Lemma 4.4 *Let $\mathcal{T}$ be a finite set of out-trees. If there exists a $T \in \mathcal{T}$ with $L_2(T) \neq \emptyset$ (that is, T contains a directed path of length at least two), then $\mathbf{P}_i(\mathcal{T})$ is NP-hard for every $i \geq 1$.*

Proof: Since we may subdivide all arcs out of s an appropriate number of times, it suffices to prove the lemma for $i = 1$. We will prove the lemma by reducing from the NP-complete 3-dimensional matching problem. Consider an instance, I, of 3-dimensional matching, where the edges of I are $E(I) = \{e_1, e_2, \ldots, e_l\}$ and the three partite sets of I are V_1, V_2, V_3. Define $r = |V_1| = |V_2| = |V_3|$ and recall that every edge in $E(I)$ contains exactly one vertex from each partite set.

Let T be the largest out-tree in $\mathcal{T}$ with $L_2(T) \neq \emptyset$. Let u_2 be a vertex of maximum depth in T and let $u_0u_1u_2$ be the path in T of length 2 ending in u_2. Let $k = |N_T^+(u_1)|$ and let $T^*(u_0)$ be the tree T after removing u_1 and all k out-neighbours of u_1 from T. We will now construct a digraph D, such that there exists an s-out-branching in D satisfying the $\mathcal{T}$-constraint at level 1 if and only if I has a perfect matching.

Let D contain $2l + 2r$ copies of $T^*(u_0)$ and add a vertex s and an arc from s to the root of each of the trees $T^*(u_0)$. Now add $3r$ vertices $V_1 \cup V_2 \cup V_3$, corresponding to the vertices in I. For each edge e_j in I we add a copy of the gadget $G(y_1, y_2, s_1, s_2, z_1, z_2, z_3)$ shown in Figure 2. We then identify the two vertices y_1 and y_2 in the gadget with two distinct u_0-vertices in two distinct trees $T^*(u_0)$. Next we identify the vertex z_i in the gadget with the vertex in V_i which belongs to the edge e_j for $i \in [3]$. Analogously we identify the vertex z_2 with the vertex in V_2 and z_3 with the vertex in V_3. The $2r$ copies of u_0 in the $T^*(u_0)$'s that have not been identified with a vertex from a gadget will now be split up into two sets U' and U'' each of size r. Add independent sets U and U^*, such that $|U| = r$ and $|U^*| = (k-1)r$ and add a perfect matching from U' to U and for each vertex in U add $k-1$ arcs to U^*, such that each vertex in U^* gets indegree one. Then add all possible arcs from U to s_1-vertices in the added gadgets. Now add an independent set W of order $(k-1)r$ and add all possible arcs from U'' to s_2-vertices in the added gadgets and all possible arcs from the s_2-vertices to W. This completes the description on D.

We first show that if there is a perfect matching in I, then there exists an out-branching in D satisfying the $\mathcal{T}$-constraint. For each edge e_j in the perfect matching we will add the arcs shown in Figure 3(a). That is, The tree $T^*(u_0)$ where y_1 was identified with the u_0-vertex will have the arcs y_1x_1, x_1z_1 and all $k-1$ arcs from x_1 to B added, resulting in a copy of T. Analogously the tree $T^*(u_0)$ where y_2 was identified with the u_0 will have the arcs y_2x_2, x_2z_2 and all $k-1$ arcs from x_2 to A added, resulting in a copy of T. Finally pick a vertex in U' and add an arc from this vertex to a vertex $u \in U$ and then add the arc us_1 and $k-1$ arcs from u to U^*, which results in a copy of T. Then pick a vertex in U'' and an arc from this vertex to s_2 and the arc s_2z_3 and $k-1$ arcs from s_2 to W. If e_j does not belong to the matching we add the arcs shown in Figure 3(b), which analogously to above creates copies of T. Therefore D contains an s-out-branching satisfying the $\mathcal{T}$-constraint.

We will now show that if there exists an s-out-branching, $\hat{T}$, in D satisfying the $\mathcal{T}$-constraint at level 1, then there is a perfect matching in I. By deleting s and all arcs out of s from $\hat{T}$ we get a collection of at most $2l + 2r$ out-trees $T_1, T_2, \ldots, T_a$. Let Y_i denote all y_i-vertices in the gadgets for $i = 1, 2$. As there is no directed path between two vertices in $Y_1 \cup Y_2 \cup U' \cup U''$ we note that each vertex in this set belongs to a distinct tree T_j. As $|Y_1 \cup Y_2 \cup U' \cup U''| = 2l + 2r$ we note that $a = 2l + 2r$.

Assume $\hat{T}$ has a path from a vertex $u'' \in U''$ to a vertex in $V_3 \cup W$. By the definition of U'' and $\hat{T}$ and by the maximality of T we note that u'' has at most $k+1$ vertices it can reach in T_j belonging

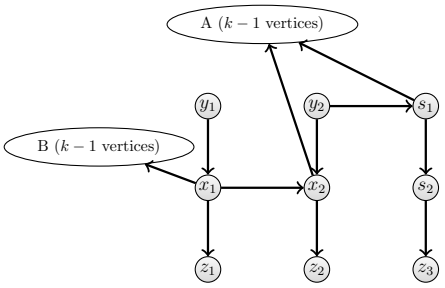

Figure 2: The gadget $G(y_1, y_2, s_1, s_2, z_1, z_2, z_3)$

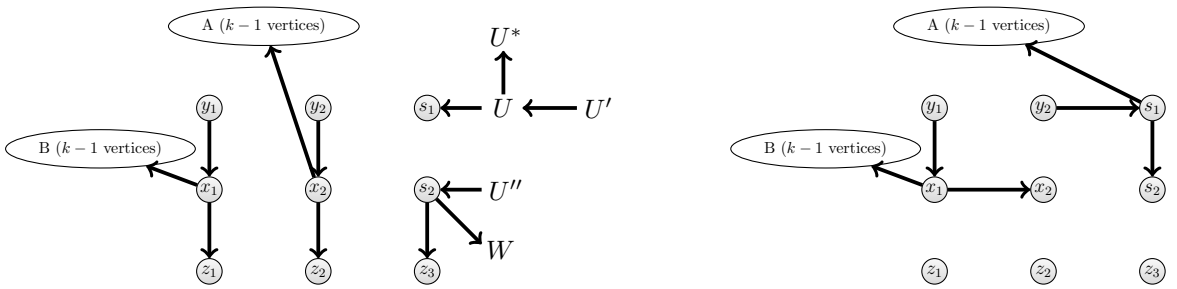

Figure 3: The possible ways of covering gadget $G(y_1, y_2, s_1, s_2, z_1, z_2, z_3)$.

to the set $V_3 \cup W \cup S_2$, where S_2 is the set of all s_2-vertices (recall that T was the largest out-tree in $\mathcal{T}$ with $L_2(T) \neq \emptyset$). Therefore u'' can reach at most k vertices in $V_3 \cup W$ in T_j, As all vertices in $V_3 \cup W$ need to be reached by paths through U'' and $|V_3 \cup W| = kr$ and $|U''| = r$ we note that each $u'' \in U''$ must reach exactly k vertices in $V_3 \cup W$ in $\hat{T}$. This means that no vertex in U'' has arcs out to two distinct s_2-vertices, as then it could reach at most $k-1$ vertices in $V_3 \cup W$.

The r vertices in V_3 need to be covered by $\hat{T}$ and the only way a vertex $v_3 \in V_3$ is covered is using the arcs $u''s_2z_3$, for some $u'' \in U''$ and some s_2 and z_3 in a gadget. However, by the remark above, the out-tree containing these arcs cannot cover any other vertex in V_3, implying that all out-trees, T_i, containing a vertex from U'' must have a path of length two from that vertex to a vertex in V_3.

We now consider the case when $s_2z_3 \notin \hat{T}$ for some gadget $G(y_1, y_2, s_1, s_2, z_1, z_2, z_3)$. This implies that there is no arc from U'' to s_2, by the above argument. Therefore we must use the arcs y_2s_1 and s_1s_2 in $\hat{T}$, in order to cover s_2. In order to cover x_2 we have $y_1x_1, x_1x_2 \in \hat{T}$. This implies that we must use the arcs shown in Figure 3(b). So in this case the gadget does not cover z_1, z_2 or z_3.

So each of the r gadgets that contain the arc s_2z_3 must also cover z_1 and z_2. This implies that the r edges in I corresponding to these gadgets must form a perfect matching, which completes the proof. $\diamond$

Lemma 4.5 *Let $\mathcal{T}$ be a finite set of out-stars. If $K^{\rightarrow}_{1,a-1} \notin \mathcal{T}$ and $K^{\rightarrow}_{1,a} \in \mathcal{T}$ for some $a \geq 1$ and $K^{\rightarrow}_{1,b} \in \mathcal{T}$ for some $b \geq 2$, then $\mathbf{P}_i(\mathcal{T})$ is NP-hard.*

Proof: Again it suffices to prove the claim for $i = 1$. We will prove the lemma by reducing from 3-dimensional matching. Consider an instance, I, of 3-dimensional matching, where the edges of I are $E(I) = \{e_1, e_2, \ldots, e_l\}$ and the three partite sets of I are V_1, V_2, V_3. Define $r = |V_1| = |V_2| = |V_3|$ and recall that every edge in $E(I)$ contains exactly one vertex from each partite set. Let $V_j = \{v_1^j, v_2^j, \ldots, v_r^j\}$. We will now construct a digraph D, such that there exists an out-branching in D satisfying the $\mathcal{T}$-constraint at level 1 if and only if I has a perfect matching.

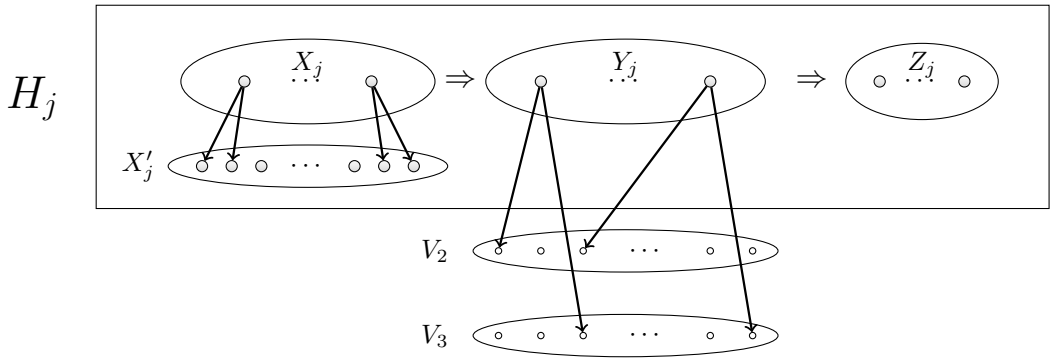

Figure 4: The gadget H_j

Let the partite sets V_2 and V_3 of I be vertices of D. Assume that q_j edges in I contain the vertex v_j^1 for all $j = 1, 2, \ldots, r$. We will now construct a gadget H_j, which we will use to determine which of the edges containing v_j^1 to use (see Figure 4). Let $V(H_j) = X_j \cup X'_j \cup Y_j \cup Z_j$, such that $|X_j| = q_j - 1$, $|X'_j| = (a-1)|X_j|$, $|Y_j| = q_j$ and $|Z_j| = b - 2$. Note that if $a = 1$ then $X'_j = \emptyset$ and if $b = 2$ then $Z_j = \emptyset$. For each vertex in X_j we add $a - 1$ arcs to X'_j, such that every vertex in X'_j gets indegree one. We then add all possible arcs from X_j to Y_j and from Y_j to Z_j. Finally, for each edge, e_k, in I containing the vertex v_j^1 we pick a vertex in Y_j and add arcs from this vertex to the two vertices in e_k from $V_2 \cup V_3$. We do this such that each of the q_j vertices in Y_j has two arcs to $V_2 \cup V_3$. This completes the construction on H_j.

In order to create D we add all the gadgets $H_1, H_2, \ldots, H_r$ to V_2 and V_3 and we also add a vertex s. Furthermore we add an arc from s to all vertices in the sets $X_j \cup Y_j$, for $j = 1, 2, \ldots, r$. This completes the construction of D.

We first show that if there is a perfect matching in I, then there exists an out-branching in D satisfying the $\mathcal{T}$-constraint. Assume that e is an edge in the perfect matching, such that $e = \{v_{j_1}^1, v_{j_2}^2, v_{j_3}^3\}$. We will now include the following arcs in the desired out-branching, T. Include all arcs from s to X_{j_1} and all arcs from X_{j_1} to X'_{j_1}. Let $y \in Y_{j_1}$ be the unique vertex with arcs to $v_{j_2}^2$ and $v_{j_3}^3$. Include a perfect matching from X_{j_1} to $Y_{j_1} \setminus \{y\}$. Include the arc from s to y and include the arcs $yv_{j_2}^2$, $yv_{j_3}^2$ and all arcs from y to Z_{j_1}. This implies that all vertices in X_{j_1} will have a out-neighbours and y will have b out-neighbours and all other vertices of H_{j_1} as well as the vertices $\{v_{j_2}^2, v_{j_3}^3\} \subseteq V_2 \cup V_3$ will be a leaf. Now it is not difficult to see that we obtain the desired out-branching.

Conversely assume that D contains an out-branching, $\hat{T}$, in D satisfying the $\mathcal{T}$-constraint at level 1. As each vertex in X_j belongs to $\hat{T}$, we note that all the arcs from s to X_j belong to $\hat{T}$. Furthermore each vertex in X'_j (if any) also belongs to $\hat{T}$, which implies that each vertex in X_j has $a - 1$ arcs to vertices in X'_j in $\hat{T}$. As $K^{\rightarrow}_{1,a-1} \notin \mathcal{T}$, we note that each vertex in X_j has at least one arc to Y_j in $\hat{T}$. This implies that at most one vertex, say y_j^*, in Y_j is not dominated by a vertex in X_j. This in turn implies that y_j^* is the only vertex from Y_j which can have arcs to $V_2 \cup V_3$ in $\hat{T}$.

As there are r vertices in V_2 and r gadgets, H_j, which each can dominate at most one vertex from V_2 in $\hat{T}$, we note that each gadget dominates exactly one vertex in V_2 in $\hat{T}$. Therefore each y_j^* dominates exactly one vertex from V_2 in $\hat{T}$. Analogously y_j^* dominates exactly one vertex from V_3 in $\hat{T}$. Let e_j^* be the edge in I containing the vertex from V_2 and the vertex from V_3 which is dominated by y_j^* as well as v_j^1. This implies that $\{e_1^*, e_2^*, \ldots, e_r^*\}$ is a perfect matching in I.

We have now shown that D contains the desired out-branching if and only if I contains a perfect matching. $\diamond$

Lemma 4.6 *If $\mathcal{T} = \{K^{\rightarrow}_{1,1}\}$ and $i \geq 2$, then $\mathbf{P}_i(\mathcal{T})$ is NP-hard.*

Proof: Let $\mathcal{T} = \{K^{\rightarrow}_{1,1}\}$ and $i \geq 2$. We can again reduce from 3-dimensional matching. Consider an instance, I, of 3-dimensional matching, where the edges of I are $E(I) = \{e_1, e_2, \ldots, e_l\}$ and the

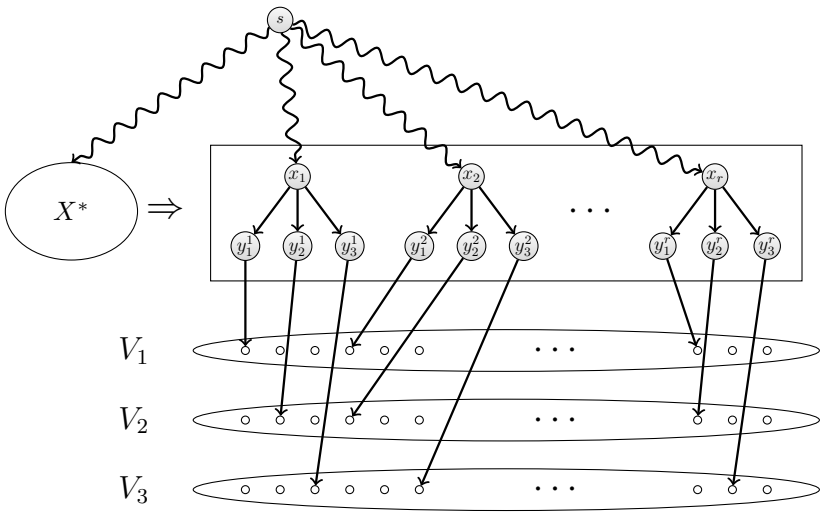

Figure 5: The digraph D in the proof of Lemma 4.6.

three partite sets of I are V_1, V_2, V_3 with $r = |V_1| = |V_2| = |V_3|$. We will construct a digraph D (see Figure 5), such that there exists an out-branching in D satisfying the $\mathcal{T}$-constraint at level i if and only if I has a perfect matching.

Let D contain the partite sets V_1, V_2 and V_3 of I. For each edge $e_j \in E(I)$ add the set X_j of four vertices $\{x_j, y_1^j, y_2^j, y_3^j\}$ to D together with all arcs $x_j y_s^j$ for $s = 1, 2, 3$ as well as an arc from y_s^j to the vertex from V_s belonging to e_j. Let X^* be a set of $4(l-r)$ independent vertices, which we also add to D together with all possible arcs from X^* to $X_1 \cup X_2 \cup \cdots X_l$. Finally add a vertex s, a directed path Q_j of length $i-1$ from s to x_j for all $1 \leq j \leq l$, and a directed path R_x of length i to each vertex x in X^*. This completes the construction of D.

We first show that if there is a perfect matching, M, in I, then there exists an s-out-branching, B_s^+, in D satisfying the $\mathcal{T}$-constraint at level i. Assume that $e_j \in M$. We will then include all the arcs in D out of x_j and out of y_1^j, y_2^j and y_3^j in B_s^+. Now include a perfect matching from X^* to all vertices in X_j, for which $e_j \notin M$. Include R_x for all $x \in X^*$, Q_j for all $e_j \in M$, and $Q_j - x_j$ for all $e_j \notin M$. This gives us the desired out-branching B_s^+.

Conversely assume that D contains an s-out-branching B_s^+ in D satisfying the $\mathcal{T}$-constraint at level i. As the vertices in X^* belong to B_s^+ we note that the path R_x belongs to B_s^+ for all $x \in X^*$. Furthermore as $\mathcal{T} = \{K_{1,1}^{\rightarrow}\}$, we note that each vertex in X^* must have exactly one arc into some distinct vertex in some X_j, which must be a leaf in B_s^+. As all vertices in $V_1 \cup V_2 \cup V_3$ have an arc into them in B_s^+, we note that at least $3r$ y_p^q-vertices have no arc into them from X^* and furthermore each of these $3r$ vertices has an arc into them from a x_q-vertex. Hence at least r x_q vertices have no arc into them from X^*. Therefore exactly r X_j-sets have no arc into them from X^* and $l-r$ X_j-sets are totally dominated by X^*. This implies that the r edges in $E(I)$ which correspond to the X_j-sets which are not dominated by X^* form a perfect matching in I. This completes the proof. $\diamond$

Lemma 4.7 *If $\mathcal{T} = \{K_{1,1}^{\rightarrow}\}$ then $\mathbf{P}_1(\mathcal{T})$ is polynomial-time solvable.*

Proof: Let D be a digraph and let s be an arbitrary vertex in D. Define an undirected graph G, such that $V(G) = V(D) \setminus \{s\}$ and $uv \in E(G)$ if and only if $uv \in D$ and $u \in N^+(s)$. If G contains a perfect matching, M, then by definition, for each $uv \in M$ we can add an arc su, which results in an s-out-branching of D satisfying the $\mathcal{T}$-constraint at level 1. Conversely, if D contains an s-out-branching, B_s^+, satisfying the $\mathcal{T}$-constraint atlevel 1, then by removing all arcs out of s in B_s^+ we obtain a perfect matching in G.

By using any polynomial algorithm for maximum matching, we can therefore decide in polynomial time whether the desired out-branching exists with s as a root. $\diamond$

Lemma 4.8 *If $\mathcal{T} = \{K^{\rightarrow}_{1,0}, K^{\rightarrow}_{1,1}, K^{\rightarrow}_{1,2}, \ldots, K^{\rightarrow}_{1,a}\}$ ($i \geq 1$) for some $a \geq 0$, then $\mathbf{P}_i(\mathcal{T})$ is polynomial-time solvable.*

Proof: Let D be a digraph and let s be an arbitrary vertex in D. Let L_i be all vertices in D at distance i from s. Note that if $L_j \neq \emptyset$ for any $j \geq i+2$, then the desired out-branching does not exist in D if s is to be the root. We may therefore assume that $V(D) = \{s\} = L_0 \cup L_1 \cup \cdots \cup L_{i+1}$. Define an undirected graph G, such that G contains a copies of every vertex in L_i and one copy of every vertex in L_{i+1}. For every arc $uv \in A(D)$ with $u \in L_i$ and $v \in L_{i+1}$ add an edge from all copies of u to v in G. This defines G.

First assume that G contains a matching, M, that saturates every vertex in L_{i+1}. Let T^+_s be the out-tree in D obtained as follows. First pick any s-out tree in $D \setminus L_{i+1}$, such that every vertex in L_i is a leaf. Then add every arc uv, where some copy of u is connected to v by an edge in M. This implies that every vertex in L_i has at most a outneighbours in T (as we made a copies of each vertex in L_i in G). Therefore T is the desired s-out-branching.

Conversely assume that B^+_s is an s-out-branching in D satisfying the $\mathcal{T}$-constraint at level i. By only considering the arcs from L_i to L_{i+1} in B^+_s, we note that we get an out-forest where every center-vertex has out-degree at most a and every vertex of L_{i+1} is a leaf. This implies that G contains a matching saturating L_{i+1}. So, for a given root s we can decide the existence of a good s-out-branching using any polynomial algorithm for bipartite matching. ◇

We will now complete the proof of Theorem 4.1, which we recall below.

Theorem 4.1: *A problem $\mathbf{P}_i(\mathcal{T})$ ($i \geq 1$) is polynomial if $\mathcal{T} = \{K^{\rightarrow}_{1,0}, K^{\rightarrow}_{1,1}, K^{\rightarrow}_{1,2}, \ldots, K^{\rightarrow}_{1,a}\}$ for some $a \geq 0$ or if $i = 1$ and $\mathcal{T} = \{K^{\rightarrow}_{1,1}\}$.*

In all other cases $\mathbf{P}_i(\mathcal{T})$ is NP-complete.

Proof: Clearly, $\mathbf{P}_i(\mathcal{T})$ is in NP, since an s-out-branching B^+_s such that $T_v \in \mathcal{T}$ for all $v \in L_i(B^+_s)$ is clearly a certificate.

The polynomial time cases follow from Lemma 4.7 and Lemma 4.8. So now assume that $\mathcal{T}$ is not of the form $\{K^{\rightarrow}_{1,0}, K^{\rightarrow}_{1,1}, K^{\rightarrow}_{1,2}, \ldots, K^{\rightarrow}_{1,a}\}$ for some $a \geq 0$ and also not of the form $\mathcal{T} = \{K^{\rightarrow}_{1,1}\}$ when $i = 1$.

By Lemma 4.4 we may assume that $\mathcal{T}$ only contains out-stars. Let q be the minimum value such that $K^{\rightarrow}_{1,q} \in \mathcal{T}$ and let b be the maximum value such that $K^{\rightarrow}_{1,b} \in \mathcal{T}$. If there exists a t such that $q < t < b$ and $K^{\rightarrow}_{1,t} \notin \mathcal{T}$, then let t be the largest such value and note that the problem in NP-hard by Lemma 4.5 (using $a = t+1 \geq 1$ and $b \geq t+1 \geq q+2 \geq 2$). Therefore we may assume that $\mathcal{T} = \{K^{\rightarrow}_{1,q}, K^{\rightarrow}_{1,q+1}, K^{\rightarrow}_{1,q+2}, \ldots, K^{\rightarrow}_{1,b}\}$. If $q = 0$, then the problem is polynomial-time solvable by Lemma 4.8, so assume that $q \geq 1$. If $b \geq 2$, then the problem is NP-hard by Lemma 4.5 (using $a = q \geq 1$ and $b \geq 2$). So we may assume that $b = 1$, which implies that $\mathcal{T} = \{K^{\rightarrow}_{1,1}\}$. We have already seen that when $i = 1$ we have a polynomial-time solvable problem and if $i \geq 2$ then the problem is NP-hard by Lemma 4.6. ◇

5 Balancing branchings at the root

We now turn to another problem concerning balancing branchings. In [2] the following result was shown.

Theorem 5.1 *[2] Let D be a digraph which is the union of two arc-disjoint s-out-branchings. There exists an s-out-branching B^+_s such that $d^+_{B^+_s}(v) \leq \frac{d^+_D(v)}{2} + 1$ for every vertex $v \in V(D)$. Furthermore, such an out-branching can be found in polynomial time.*

Hence, if a digraph has two arc-disjoint s-out-branchings, then we may find one s-out-branching in which no vertex has more that roughly half as many out-neighbours as it has in D. The natural

(and seemingly much harder) question is whether the existence of two arc-disjoint s-branchings together with perhaps some extra condition guarantees the existence of two arc-disjoint s-out-branchings $B^+_{s,1}, B^+_{s,2}$ in which no vertex has out-degree more than some fraction (less than 1) of its original out-degree. Below we consider the much more modest case, where we just want to balance out-degrees at the root s. It turns out that this is always possible in the strongest way possible.

Lemma 5.2 *Let D be a digraph (parallel arcs are allowed) and let $s \in V(D)$ be arbitrary. If D has two arc-disjoint s-out-branchings, then there exists two arc-disjoint s-out-branchings, B_1 and B_2, such that $d^+_{B_1}(s), d^+_{B_2}(s) \leq \lceil d^+(s)/2 \rceil$. Furthermore we can construct such a pair of branchings in polynomial time.*

Proof: We will prove the lemma using induction on the order of D. Clearly the lemma is true if $|V(D)| \leq 2$. Let D be a digraph (parallel arcs are allowed), let $s \in V(D)$ and let B^+_1 and B^+_2 be two arc-disjoint s-out-branchings in D. By our induction hypothesis the lemma is true for all digraphs of smaller order than D. Let $A^+ = A(B^+_1) \cup A(B^+_2)$ and let $D^+ = (V(D), A^+)$ be the spanning subgraph of D which only contains the arcs from B^+_1 and B^+_2. If we can show that the lemma is true for D^+, then it is also true for D, as $d^+_{D^+}(s) \leq d^+_D(s)$. Let $n = |V(D)| = |V(D^+)|$ and note that $|A^+| = 2(n-1)$. As $\sum_{v \in V(D)} d^+(v) = |A^+| < 2n$ we note that some vertex, $x \in V(D^+)$, has $d^+_{D^+}(x) \leq 1$. We now consider the following two cases.

Case 1, $d^+(x) = 0$. In this case let $D' = D^+ - x$. Since x is a leaf in both branchings both $B^+_1 - x$ and $B^+_2 - x$ are arc-disjoint s-out-branchings in D'. By induction there exists arc-disjoint s-out-branchings, T^+_1 and T^+_2, in D' such that $d^+_{T^+_1}(s), d^+_{T^+_2}(s) \leq \lceil d^+_{D'}(s)/2 \rceil$. If $d^+_{D'}(s) = d^+_{D^+}(s)$ then we can add x as a leaf to T^+_1 and T^+_2 using distinct arcs, thereby obtaining the desired s-out-branchings. So assume that $d^+_{D'}(s) < d^+_{D^+}(s)$. If there are two distinct (parallel) arcs from s to x in D^+ then use these two to add x as a leaf to T^+_1 and T^+_2, respectively. In this case both $d^+_{T^+_1}(s)$ and $d^+_{T^+_2}(s)$ increase by one, but $d^+_{D'}(s) \leq d^+_{D^+}(s) - 2$, so the resulting s-out-branchings satisfy the condition in the lemma. So we may assume that there is exactly one arc from s to x in D^+. Use sx to add x to the s-out-branching, T^+_i, with minimum $d^+_{T^+_i}(s)$ ($i \in \{1,2\}$). If $\lceil d^+_{D'}(s)/2 \rceil = \lceil d^+_{D^+}(s)/2 \rceil$ then $d^+_{D'}(s)$ is odd, implying that $d^+_{T^+_i}(s) < d^+_{T^+_{3-i}}(s)$. So in this case $d^+_{T^+_i}(s) + 1, d^+_{T^+_{3-i}}(s) \leq \lceil d^+_{D^+}(s)/2 \rceil$ and we are done. So, $\lceil d^+_{D'}(s)/2 \rceil < \lceil d^+_{D^+}(s)/2 \rceil$, which again implies that $d^+_{T^+_i}(s)+1, d^+_{T^+_{3-i}}(s) \leq \lceil d^+_{D'}(s)/2 \rceil + 1 \leq \lceil d^+_{D^+}(s)/2 \rceil$, completing the proof of Case 1.

Case 2, $d^+(x) = 1$. Without loss of generality assume that the arc, xu, out of x in D^+ belongs to B^+_1. Let $wx \in A(B^+_1)$ and $vx \in A(B^+_2)$ be the two arcs into x in A^+. Let D^* be the digraph obtained from $D - x$ by adding the arc wu. Note that removing x from B^+_2 and removing wx, xu from B^+_1 and adding wu, gives us two arc-disjoint s-out-branchings in D^*. By induction, there exists arc-disjoint s-out-branchings, $\tilde{T}^+_1$ and $\tilde{T}^+_2$, in D^* such that $d^+_{\tilde{T}^+_1}(s), d^+_{\tilde{T}^+_1}(s) \leq \lceil d^+_{D^*}(s)/2 \rceil$. As $|A(D^*)| = 2(|V(D^*)| - 1)$ we note that $A(D^*) = A(\tilde{T}^+_1) \cup A(\tilde{T}^+_2)$. Without loss of generality assume that $wu \in A(\tilde{T}^+_1)$.

if $v \neq s$, then $d^+_{D^*}(s) = d^+_{D^+}(s)$ and inserting x into the arc wu in $\tilde{T}^+_1$ (by deleting wu and adding wx and xu) and adding x as a leaf in $\tilde{T}^+_2$ (using vx) gives us the desired s-out-branchings (as $d^+_{\tilde{T}^+_1}(s)$ and $d^+_{\tilde{T}^+_2}(s)$ remain unchanged). So assume that $v = s$. In this case $d^+_{D^*}(s) = d^+_{D^+}(s) - 1$. As before inserting x into the arc wu in $\tilde{T}^+_1$ and let the resulting out-branching be denoted by $\hat{T}^+_1$ and adding x as a leaf in $\tilde{T}^+_2$ (using vx) and let the resulting out-branching be denoted by $\hat{T}^+_2$. If $d^+_{\hat{T}^+_1}(s), d^+_{\hat{T}^+_2}(s) \leq \lceil d^+_{D^+}(s)/2 \rceil$, then we are done, so assume that this is not the case. This implies that $d^+_{\hat{T}^+_2}(s) = d^+_{T^*_2}(s) + 1 > \lceil d^+_{D^+}(s)/2 \rceil$, which implies that $d^+_{\hat{T}^+_2}(s) \geq d^+_{\hat{T}^+_1}(s) + 2$. Now delete the arc vx from $\hat{T}^+_2$ and add the arc wx instead, which results in an s-out-branching in D^+. In $\hat{T}^+_1$ we remove wx and add vx instead, which again results in an s-out-branching (as $v = s$). These two arc-disjoint s-out-branchings satisfy the condition in the lemma, thereby completing the proof. Starting form an arbitrary pair of arc-disjoint s-out-branchings (which can be found in polynomial time using flows, see e.g. [1, Section 9.3]) the inductive proof is easily turned into a polynomial algorithm. $\diamond$

Corollary 5.3 *Let D be a digraph, where up to two parallel arcs are allowed between any pair of vertices and let $s \in V(D)$ be arbitrary. If D has two arc-disjoint s-out-branchings, then there exists two arc-disjoint s-out-branchings, B_1 and B_2, such that $d^+_{B_1}(s) = \lfloor d^+(s)/2 \rfloor$ and $d^+_{B_2}(s) = \lceil d^+(s)/2 \rceil$.*

Proof: Let Q_1 and Q_2 be two arc-disjoint s-outbranchings with as many arcs out of s as possible in $A(Q_1) \cup A(Q_2)$. If some arc out of s, say sx is not in $A(Q_1) \cup A(Q_2)$, then assume without loss of generality that no arc from s to x belongs to Q_1. Let $a \in A(Q_1)$ be the arc into x. By removing a from Q_1 and adding sx instead we obtain a s-out-branchings in D, contradicting the maximality of the number of arcs out of s in $A(Q_1) \cup A(Q_2)$. Therefore all arcs out of s belong to $A(Q_1) \cup A(Q_2)$. Let $D' = (V(D), A(Q_1) \cup A(Q_2))$ be the digraph only containing the $2(|V(D)| - 1)$ arcs $A(Q_1) \cup A(Q_2)$.

By using Lemma 5.2 on D' there exists two arc-disjoint s-outbranchings, B_1 and B_2, in D' such that $d^+_{B_1}(s), d^+_{B_2}(s) \leq \lceil d^+_{D'}(s)/2 \rceil = \lceil d^+_D(s)/2 \rceil$. Furthermore $A(B_1) \cup A(B_2) = A(Q_1) \cup A(Q_2)$, which implies that $d^+_{B_1}(s) + d^+_{B_2}(s) = d^+_{D'}(s) = d^+_D(s)$. Therefore, by possibly renaming B_1 and B_2, the corollary holds. $\diamond$

The following result can be proved by repeated applications of Corollary 5.3 to appropriate subdigraphs formed by the union of two s-out-branchings. We leave the easy details to the intersted reader.

Corollary 5.4 *Let D be a digraph containing k arc-disjoint s-out-branchings where $k > 0$ is an integer. Then there exists arc-disjoint s-out-branchings, $B_1, B_2, \ldots, B_k$ in D such that $d^+_{B_i}(s) \in \{\lfloor \frac{d^+(s)}{k} \rfloor, \lceil \frac{d^+(s)}{k} \rceil\}$.*

References

[1] J. Bang-Jensen and G. Gutin. *Digraphs: Theory, Algorithms and Applications, 2nd Edition.* Springer-Verlag, London, 2009.

[2] J. Bang-Jensen, S. Thomassé, and A. Yeo. Small degree out-branchings. *J. Graph Theory*, 42(4):297–307, 2003.

[3] J Daligault, G. Gutin, E.J. Kim, and A. Yeo. FPT Algorithms and Kernels for the Directed k-Leaf Problem. *Journal of Computer and System Sciences*, pages 144–152, 2010.

[4] R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, pages 512–530, 2001.

[5] M. Las Vergnas. Sur les arborescences dans un graphe orienté. *Discrete Math.*, 15(1):27–39, 1976.

[6] D. Lokshtanov, V. Raman, S. Saurabh, and S. Sikdar. On the directed degree-preserving spanning tree problem. 5917:276–287, 2009.

[7] S. Thomassé. Covering a strong digraph by $\alpha - 1$ disjoint paths: a proof of Las Vergnas' conjecture. *J. Combin. Theory Ser. B*, 83(2):331–333, 2001.

[8] C. Thomassen. Highly connected non-2-linked digraphs. *Combinatorica*, 11(4):393–395, 1991.