Approximating Directed Weighted-Degree Constrained Networks*

Zeev Nutov

The Open University of Israel

nutov@openu.ac.il

Key-words: Directed network design; Intersecting supermodular requirements; Weighted degree constraints; Bicriteria approximation algorithms.

Abstract

Given a graph $H = (V, F)$ with edge weights $\{w_e : e \in F\}$, the *weighted degree* of a node v in H is $\sum\{w_{vu} : vu \in F\}$. We give bicriteria approximation algorithms for problems that seek to find a minimum cost *directed* graph that satisfies both *intersecting supermodular* connectivity requirements and *weighted degree* constraints. The input to such problems is a directed graph $G = (V, E)$ with edge-costs $\{c_e : e \in E\}$ and edge-weights $\{w_e : e \in E\}$, an intersecting supermodular set-function f on V, and degree bounds $\{b(v) : v \in B \subseteq V\}$. The goal is to find a minimum cost f-connected subgraph $H = (V, F)$ (namely, at least $f(S)$ edges in F enter every $S \subseteq V$) of G with weighted degrees $\leq b(v)$. Our algorithm computes a solution of cost $\leq 2 \cdot \mathsf{opt}$, so that the weighted degree of every $v \in V$ is at most: $7b(v)$ for arbitrary f and $5b(v)$ for a $0,1$-valued f; $2b(v) + 4$ for arbitrary f and $2b(v) + 2$ for a $0,1$-valued f in the case of unit weights. Another algorithm computes a solution of cost $\leq 3 \cdot \mathsf{opt}$ and weighted degrees $\leq 6b(v)$. We obtain similar results when there are both indegree and outdegree constraints, and better results when there are indegree constraints only: a $(1, 4b(v))$-approximation algorithm for arbitrary weights and a polynomial time algorithm for unit weights. Similar results are shown for crossing supermodular f. We also consider the problem of packing maximum number k of pairwise edge-disjoint arborescences so that their union satisfies weighted degree constraints, and give an algorithm that computes a solution of value at least $\lfloor k/36 \rfloor$. Finally, for unit weights and without trying to bound the cost, we give an algorithm that computes a subgraph so that the degree of every $v \in V$ is at most $b(v) + 3$, improving over the approximation $b(v) + 4$ of [2].

*Preliminary version in APPROX 2008, pp. 219–232.

1 Introduction

1.1 Problem definition

In many Network Design problems one seeks to find a low-cost subgraph H of a given graph G that satisfies prescribed connectivity requirements. Such problems are vastly studied in Combinatorial Optimization and Approximation Algorithms. Known examples are Min-Cost k-Flow, b-Edge-Cover, Min-Cost Spanning Tree, Traveling Salesperson, directed/undirected Steiner Tree, Steiner Forest, k-Edge/Node-Connected Spanning Subgraph, and many others. See, e.g., surveys in [18, 5, 10, 12, 14].

In Degree Constrained Network Design problems, one seeks the cheapest subgraph H of a given graph G that satisfies both prescribed connectivity requirements and degree constraints. One such type of problems are the matching/edge-cover problems, which are solvable in polynomial time, c.f., [18]. For other degree constrained problems, even checking whether there exists a feasible solution is NP-complete, hence one considers bicriteria approximation when the degree constraints are relaxed. For example, checking whether a directed graph contains an arborescence of maximum outdegree 1, or whether and undirected graph contains a spanning tree of maximum degree 2, is the directed/undirected Hamiltonian Path problem, which is NP-complete.

We consider *directed* network design problems, so all the graphs in this paper are assumed to be directed, unless stated otherwise. The connectivity requirements can be specified by a set-function f on V as follows. For an edge set or a graph H on node set V and node subset $S \subseteq V$ let $\delta_H(S)$ ($\delta_H^{in}(S)$) denote the set of edges in H leaving (entering) S. We will write $\delta_H(v)$ instead of $\delta_H(\{v\})$.

Definition 1.1 *Given a set-function f from subsets of V to non-negative integers, and a graph $H = (V, F)$ we say that H is* f-connected *if at least $f(S)$ edges in F enter every $S \subseteq V$, namely:*

$$|\delta_H^{in}(S)| \geq f(S) \quad \text{for all } S \subseteq V. \tag{1}$$

Several types of f are considered in the literature, among them the following known ones:

Definition 1.2 *A set-function f on V is* intersecting supermodular *if any $X, Y \subseteq V$ that intersect (namely, $X \cap Y \neq \emptyset$) satisfy the* supermodularity condition

$$f(X) + f(Y) \leq f(X \cap Y) + f(X \cup Y) \ . \tag{2}$$

If any $X, Y \subseteq V$ that cross (namely, the sets $X \cap Y, X \setminus Y, Y \setminus X, V \setminus (X \cup Y)$ are non-empty) satisfy the supermodularity condition (2), then f is crossing supermodular.

For a weight/cost function x on an edge set F of a graph H, let $x(H) = x(F) = \sum_{e \in F} x_e$ be the weight/cost of F (or of H). We consider directed network design problems with *weighted-degree constraints*. For simplicity of exposition, we will consider the cases of outdegree constraints and indegree constraints separately, but our results easily extend to the case with both indegree and outdegree constraints, see Section 6. The outdegree constrained version of our problem is:

Directed Weighted Degree Constrained Network (DWDCN)	
Instance:	A directed graph $G = (V, E)$ with edge-costs $\{c_e : e \in E\}$ and edge-weights $\{w_e : e \in E\}$, a set-function f on V, and degree bounds $b = \{b(v) : v \in B \subseteq V\}$.
Objective:	Find a minimum cost f-connected subgraph $H = (V, F)$ of G that satisfies the *weighted degree constraints* $$w(\delta_H(v)) \leq b(v) \quad \text{for all } v \in B \ . \qquad (3)$$

The set-function f may not be given explicitly but rather implicitly in a compact form. We need that some queries related to f can be answered in polynomial time. Specifically, we assume that f admits a polynomial time evaluation oracle. Since for most set-functions f even checking whether DWDCN has a feasible solution is NP-complete, we consider bicriteria approximation algorithms. Assuming that the problem has a feasible solution, an $(\alpha, \rho(b(v)))$-approximation algorithm for DWDCN either computes an f-connected subgraph H of G of cost $\leq \alpha \cdot \mathsf{opt}$ that satisfies $w(\delta_H(v)) \leq \rho(b(v))$ for all $v \in B$, or correctly determines that the problem has no feasible solution. Note that even if the problem does not have a feasible solution, the algorithm may still return a subgraph that violates the degree constraints (3) within $\rho(b(v))$.

Now we mention what connectivity types can be represented by intersecting supermodular and crossing supermodular functions. A graph H is *k-edge-outconnected from r* if it has k-edge-disjoint paths from r to any other node. A graph is *k-edge-connected* if it has k-edge-disjoint paths between every pair of its nodes. DWDCN with intersecting supermodular f includes as a special case the Weighted Degree Constrained k-Edge-Outconnected Subgraph problem, by setting $f(S) = k$ for all $\emptyset \neq S \subseteq V \setminus \{r\}$, and $f(S) = 0$ otherwise. For $k = 1$ we get the Weighted Degree Constrained Arborescence problem. DWDCN with crossing supermodular f includes as a special case the Weighted Degree Constrained k-Edge-Connected Subgraph problem, by setting $f(S) = k$ for all $\emptyset \neq S \subset V$, and $f(S) = 0$ otherwise.

We also consider the problem of packing maximum number k of edge-disjoint arborescences rooted at r so that their union H satisfies (3). By Edmond's Theorem [4], this is equivalent to requiring that H is k-edge-outconnected from r and satisfies (3). This gives the following problem:

Weighted Degree Constrained Maximum Arborescence Packing (WDCMAP)	
Instance:	A directed graph $G = (V, E)$ with edge-weights $\{w_e : e \in E\}$, degree bounds $b = \{b(v) : v \in B \subseteq V\}$, and a root $r \in V$.
Objective:	Find a k-edge-outconnected from r spanning subgraph $H = (V, F)$ of G that satisfies the degree constraints (3) so that k is maximum.

1.2 Our results

Our main results are summarized in the following two theorems. Let τ^* denote the optimal value of the following natural LP-relaxation for DWDCN that for $x \in \mathbb{R}^E$ seeks to minimize $c \cdot x = \sum_{e \in E} c_e x_e$ over the following polytope $P(f, b)$:

$$\boxed{\begin{array}{rcll} x(\delta_E^{in}(S)) & \geq & f(S) & \text{for all } \emptyset \neq S \subset V \\ \displaystyle\sum_{e \in \delta_E(v)} x_e w_e & \leq & b(v) & \text{for all } v \in B \\ 0 \leq x_e & \leq & 1 & \text{for all } e \in E \end{array}}$$

Theorem 1.1 DWDCN *with intersecting supermodular f admits a polynomial time algorithm that computes an f-connected graph of cost $\leq 2\tau^*$ so that the weighted degree of every $v \in B$ is at most: $7b(v)$ for arbitrary f and $5b(v)$ for a $0,1$-valued f; for unit weights, the degree of every $v \in B$ is at most $2b(v) + 4$ for arbitrary f and $2b(v) + 2$ for a $0,1$-valued f. The problem also admits a $(3, 6b(v))$-approximation algorithm for arbitrary weights and arbitrary intersecting supermodular f.*

Interestingly, we can show a much better result for the version of DWDCN with indegree constraints $w(\delta_H^{in}(v)) \leq b^{in}(v)$ for all $v \in B$ (for the case of both indegree and outdegree constraints see Section 6).

Theorem 1.2 DWDCN *with indegree constraints and with intersecting supermodular f admits a polynomial time algorithm that computes an f-connected graph of cost τ^* so that the weighted indegree of every $v \in B$ is at most: $\min\{4, f_{\max}\} \cdot b^{in}(v)$ for arbitrary weights and $\min\{b^{in}(v), f_{\max}\}$ for unit weights, where $f_{\max} = \max_{S \subseteq V} f(S)$ is the maximum f-value. In particular, for a $0,1$-valued f and arbitrary weights, or for unit weights and arbitrary intersecting supermodular f, the problem admits an exact polynomial time algorithm.*

We leave an open question whether DWDCN with indegree constraints and arbitrary intersecting supermodular f admits a polynomial time algorithm for arbitrary weights. By combining Theorems 1.1 and 1.2 we can easily obtain approximation algorithms for DWDCN with crossing supermodular f. Any crossing supermodular set-function f can be naturally represented by two intersecting supermodular set-functions as follows, see, e.g. [8].

Fact 1.3 *Let f be a crossing supermodular set-function on V and let $r \in V$. Let $f^{in}(S) = f(S)$ if $r \notin S$ and $f(S) = 0$ otherwise, and let $f^{out}(S) = f(V \setminus S)$ if $r \notin S$ and $f(S) = 0$ otherwise. Then f^{in}, f^{out} are intersecting supermodular set-functions, and H is f-connected if, and only if, H is f^{in}-connected and the reverse graph of H is f^{out}-connected.*

Corollary 1.4 DWDCN *with crossing supermodular f admits a polynomial time algorithm that computes an f-connected graph of cost $\leq 3\tau^*$ so that the weighted degree of every $v \in B$ is at most: $(7 + \min\{4, f_{\max}\}) \cdot b(v))$ for arbitrary f and $6b(v)$ for a $0,1$-valued f; for unit weights, the degree* *of every $v \in B$ is at most $2b(v)+4+\min\{f_{\max}, b(v)\})$ for arbitrary f and $2b(v)+3$ for a $0,1$-valued f. The same ratios apply for the version with indegree constraints.*

Table 1 summarizes the ratios in Theorems 1.1, 1.2, and Corollary 1.4.

type of f	**intersecting supermodular**		**crossing supermodular**
constraints	*outdegree*	*indegree*	*outdegree/indegree*
any f, any w	$(2, 7b(v))$ $(3, 6b(v))$	$(1, \min\{4, f_{\max}\} \cdot b^{in}(v))$	$(3, (7+\min\{4, f_{\max}\}) \cdot b(v))$ $(4, (6+\min\{4, f_{\max}\}) \cdot b(v))$
$0,1$-f, any w	$(2, 5b(v))$	$(1, b^{in}(v))$	$(3, 6b(v))$
any f, $w \equiv 1$	$(2, 2b(v)+4)$	$(1, \min\{f_{\max}, b^{in}(v)\})$	$(3, 2b(v)+4+\min\{f_{\max}, b(v)\})$
$0,1$-f, $w \equiv 1$	$(2, 2b(v)+2)$	$(1, \min\{1, b^{in}(v)\})$	$(3, 2b(v)+3)$

Table 1: Bicriteria approximation ratios for DWDCN with indegree or outdegree constraints (but not both) for intersecting and crossing supermodular f. Each ratio in the right column (crossing supermodular f) is a sum of the corresponding ratios in the first two columns.

Theorem 1.1 has several applications. Bang-Jensen, Thomassé, and Yeo [1] conjectured that every k-edge-connected directed graph $G=(V,E)$ contains a spanning arborescence H so that $|\delta_H(v)| \leq |\delta_G(v)|/k+1$ for every $v \in V$. Bansal, Khandekar, and Nagarajan [2] proved that even if G is only k-edge-outconnected from r, then G contains a spanning arborescence H so that $|\delta_H(v)| \leq |\delta_G(v)|/k+2$. We prove that for any $\ell \leq k$, G contains an ℓ-outconnected from r spanning subgraph H whose cost and weighted degrees are not much larger than the "expected" values $c(G) \cdot (\ell/k)$ and $w(\delta_G(v)) \cdot (\ell/k)$. In particular, one can find an arborescence with both low weighted degrees and low cost.

Corollary 1.5 *Let $H_k=(V,F)$ be a k-edge-outconnected from r directed graph with edge costs $\{c_e : e \in F\}$ and edge weights $\{w_e : e \in F\}$. Then for any $\ell \leq k$ the graph H_k contains an ℓ-outconnected from r spanning subgraph H_ℓ so that $c(H_\ell) \leq c(H_k) \cdot (2\ell/k)$ and so that for all $v \in V$: $w(\delta_{H_\ell}(v)) \leq w(\delta_{H_k}(v)) \cdot (7\ell/k)$, and $w(\delta_{H_\ell}(v)) \leq w(\delta_{H_k}(v)) \cdot (5/k)$ for $\ell=1$; for unit weights, $|\delta_{H_\ell}(v)| \leq |\delta_{H_k}(v)| \cdot (2\ell/k)+2$. There also exists H_ℓ so that $c(H_\ell) \leq c(H_k) \cdot (3\ell/k)$ and $w(\delta_{H_\ell}(v)) \leq w(\delta_{H_k}(v)) \cdot (6\ell/k)$ for all $v \in V$.*

Proof: Consider the Weighted Degree Constrained ℓ-Edge-Outconnected Subgraph problem on H_k with degree bounds $b(v)=w(\delta_{H_k}(v)) \cdot (\ell/k)$. Clearly, $x_e=\ell/k$ for every $e \in F$ is a feasible solution of cost $c(H_k) \cdot (\ell/k)$ to the LP-relaxation $\min\{c \cdot x : x \in P_f\}$ where $f(S)=\ell$ for all $\emptyset \neq S \subseteq V \setminus \{r\}$, and $f(S)=0$ otherwise. By Theorem 1.1, our algorithm computes a subgraph H_ℓ as required. □

Another application is for the WDCMAP problem. Ignoring costs, Theorem 1.1 implies a "pseudo-approximation" algorithm for WDCMAP that computes the maximum number k of packed arborescences, but violates the weighted degree constraints. E.g., using the $(3,6)$-approximation algorithm from Theorem 1.1, we can compute a k-edge-outconnected H that violates the weighted degree bounds by a factor of 6, where k is the optimal value to WDCMAP. Note that assuming P$\neq$NP, WDCMAP cannot achieve a $1/\rho$-approximation algorithm for any $\rho > 0$, since deciding whether $k \geq 1$ is equivalent to the Degree Constrained Arborescence problem, which is NP-complete. We can however show that if the optimal value k is not too small, then the problem does admit a constant ratio approximation.

Theorem 1.6 WDCMAP *admits a polynomial time algorithm that computes a feasible solution H that satisfies (3) so that H is $\lfloor k/36 \rfloor$-outconnected from r.*

Proof: The algorithm is as follows. We set $b'(v) \leftarrow b(v)/6$ for all $v \in V$ and apply the $(3, 6b(v))$-approximation algorithm from Theorem 1.1. The degree of every node v in the subgraph computed is at most $6b'(v) \leq b(v)$, hence the solution is feasible. All we need to prove is that if the original instance admits a packing of size k, then the new instance admits a packing of size $\lfloor k/36 \rfloor$. Let H_k be an optimal solution to WDCMAP. Substituting $\ell = \lfloor k/36 \rfloor$ in the last statement of Corollary 1.5 and ignoring the costs we obtain that H_k contains a subgraph H_ℓ which is ℓ-outconnected from r so that $w(\delta_{H_\ell}(v)) \leq w(\delta_{H_k}(v)) \cdot (6\ell/k) \leq w(\delta_{H_k}(v))/6 \leq b(v)/6$ for all $v \in V$, as claimed. $\square$

We note that Theorem 1.6 easily extends to the case when edges have costs; the cost of the subgraph H computed is at most the minimum cost of a feasible k-edge-outconnected subgraph.

Finally, for unit weights and without trying to bound the cost, we obtain the following result that improves over the degree approximation $b(v) + 4$ of [2].

Theorem 1.7 DWDCN *with intersecting supermodular f and unit weights (and costs ignored) admits a polynomial time algorithm that computes an f-connected subgraph so that the degree of every $v \in B$ is at most $b(v) + 3$.*

1.3 Previous and related work

Fürer and Raghavachari [7] considered the problem of finding a spanning tree with maximum degree $\leq \Delta$, and gave an algorithm that computes a spanning tree of maximum degree $\leq \Delta + 1$. This is essentially the best possible since computing the optimum is NP-hard. A variety of techniques were developed in attempt to generalize this result to the minimum-cost case – the Minimum Degree Spanning Tree problem, c.f., [17, 13, 3]. Goemans [9] presented an algorithm that computes a spanning tree of cost $\leq$ opt and with degrees at most $b(v) + 2$ for all $v \in B$, where $b(v)$ is the degree bound of v. An optimal result was obtained by Singh and Lau [19]; their algorithm computes a spanning tree of cost $\leq$ opt and with degrees at most $b(v) + 1$ for all $v \in B$. The algorithm of Singh and Lau [19] uses the method of *iterative rounding*. This method was initiated in a seminal paper of Jain [11] that gave a 2-approximation algorithm for the Steiner Network problem. Without degree constraints, this method is as follows: given an optimal basic solution to an LP-relaxation for the problem, round at least one entry, and recurse on the residual instance. The algorithm of

Singh and Lau [19] for the Minimum Bounded Degree Spanning Tree problem is a surprisingly simple extension – either round at least one entry, or remove a degree constraint from some node v. The non-trivial part usually is to prove that basic fractional solution have certain "sparse" properties.

For unit weights, the following results were obtained recently. Lau, Naor, Salvatipour, and Singh [15] were the first to consider general connectivity requirements. They gave a $(2, 2b(v) + 3)$-approximation for undirected graphs in the case when f is weakly supermodular. For directed graphs, they gave a $(4, 4b(v)+6)$-approximation for intersecting supermodular f, and $(8, 8b(v)+6)$-approximation for crossing supermodular f. Recently, in the full version of [15], these ratios were improved to $(3, 3b(v)+5)$ for crossing supermodular f, and $(2, 2b(v)+2)$ for a $0,1$-valued intersecting supermodular f. For the latter case we have the same ratio, but our proof is simpler than the one in the full version of [15].

Bansal, Khandekar, and Nagarajan [2] obtained for an intersecting supermodular set-function f a $(\frac{1}{\varepsilon}, \lceil\frac{b(v)}{1-\varepsilon}\rceil + 4)$-approximation scheme, $0 \leq \varepsilon \leq 1/2$; substituting $\varepsilon = 1/2$ gives a $(2, 2b(v) + 4)$-approximation as in our Theorem 1.1, but our proof of this particular case is very simple. They also showed that this ratio cannot be much improved based on the standard LP-relaxation. For crossing supermodular f [2] gave a $(\frac{2}{\varepsilon}, \lceil\frac{b(v)}{1-\varepsilon}\rceil + 4 + f_{\max})$-approximation scheme. For the Degree Constrained Arborescence problem (without costs) [2] gave an algorithm that computes an arborescence H with $|\delta_H(v)| \leq b(v) + 2$ for all $v \in B$. Some additional results for related problems can also be found in [2].

For weighted degrees, Fukunaga and Nagamochi [6] considered *undirected* network design problems and gave a $(1, 4b(v))$-approximation for minimum spanning trees and a $(2, 7b(v))$-approximation algorithm for arbitrary weakly supermodular set-function f.

2 Proof of Theorem 1.1

Given an edge set (partial solution) J, let
$f_J(S) = f(S) - |\delta^{in}_J(S)|$ for all $\emptyset \neq S \subset V$
$b^\alpha_J(v) = b(v) - w(\delta_J(v))/\alpha$ for all $v \in B$ ($\alpha \geq 1$ is a fixed parameter).
It is known and easy to see that if f admits a polynomial time evaluation oracle and is intersecting supermodular, then so is f_J, for any J. The algorithm starts with $J = \emptyset$ and performs iterations. In any iteration, we work with the residual polytope $P(f_J, b^\alpha_J)$, and remove some edges from E and/or some nodes from B.

Let us recall some facts from polyhedral theory. Let x belong to a polytope $P \subseteq \mathbb{R}^m$ defined by a system of linear inequalities; an inequality is *tight* (for x) if it holds as equality for x. A point $x \in P$ is a *basic solution* for (the system defining) P if there exists a set of m tight inequalities in the system defining P such that x is the unique solution for the corresponding equation system; that is, the corresponding m tight equations are linearly independent. It is well known that if the

LP $\min\{c \cdot x : x \in P\}$ has an optimal solution, then it has an optimal solution which is basic, and that a basic optimal solution for $\min\{c \cdot x : x \in P(f_J, b_J^\alpha)\}$ can be computed in polynomial time for any J, c.f. [15].

Definition 2.1 *The polytope $P(f_J, b_J^\alpha)$ is (α, Δ)-*sparse *for integers $\alpha, \Delta \geq 1$ if any basic solution $x \in P(f_J, b_J^\alpha)$ has an edge $e \in E$ with $x_e = 0$, or satisfies at least one of the following:*

$$x_e \geq 1/\alpha \quad \text{for some } e \in E \tag{4}$$

$$|\delta_E(v)| \leq \Delta \quad \text{for some } v \in B \tag{5}$$

We prove the following two general statements that imply Theorem 1.1:

Theorem 2.1 *If for any J the polytope $P(f_J, b_J^\alpha)$ is (α, Δ)-sparse (if non-empty), then* DWDCN *admits an $(\alpha, \alpha + \Delta)$-approximation algorithm; for unit weights the algorithm computes a solution F so that $c(F) \leq \alpha \cdot \tau^*$ and $|\delta_F(v)| \leq \alpha b(v) + \Delta - 1$ for all $v \in V$.*

Theorem 2.2 *$P(f_J, b_J^\alpha)$ is $(2, 4)$-sparse and $(3, 3)$-sparse for intersecting supermodular f; if f is $0, 1$-valued, then $P(f_J, b_J^\alpha)$ is $(2, 3)$-sparse.*

3 The Algorithm (Proof of Theorem 2.1)

The algorithm performs iterations. Every iteration excludes at least one edge from E or at least one node from B. In the case of unit weights we assume that all the degree bounds are integers.

Algorithm for DWDCN with intersecting supermodular f
Initialization: $J \leftarrow \emptyset$, $E \leftarrow E \setminus \{vu \in E : w(vu) > b(v)\}$.
If $P(f, b) = \emptyset$, then return "UNFEASIBLE" and STOP.
While $E \neq \emptyset$ do:

1. Find a basic solution $x \in P(f_J, b_J^\alpha)$.
2. Remove from E all edges with $x_e = 0$.
3. Add to J and remove from E all edges with $x_e \geq 1/\alpha$.
4. Remove from B every $v \in B$ with $|\delta_E(v)| \leq \Delta$.

EndWhile
Return $F \leftarrow J$.

Lemma 3.1 *The algorithm has approximation ratio $(\alpha, \alpha+\Delta)$ if each polytope $P(f_J, b_J^\alpha)$ considered during the algorithm is (α, Δ)-sparse; furthermore, for unit weights, the algorithm computes a solution F such that $c(F) \leq \alpha \cdot \tau^*$ and $|\delta_F(v)| \leq \alpha b(v) + \Delta - 1$ for all $v \in V$.*

Proof: Clearly, if $P(f, b) = \emptyset$ then the problem has no feasible solution, and the algorithm indeed outputs "UNFEASIBLE". At every iteration of the while loop, if x is a feasible LP-solution found at the beginning of the iteration, then at the end of the iteration x remains a feasible solution to the residual LP. In particular, if $P(f,b) \neq \emptyset$ then $P(f_J, b_J^\alpha) \neq \emptyset$ throughout the subsequent iterations. Hence if the problem has a feasible solution, the algorithm returns an f-connected graph, and we need only to prove the approximation ratio. As for every edge added we have $x_e \geq 1/\alpha$, the algorithm indeed computes a solution of cost $\leq \alpha \cdot \tau^*$.

Now we prove the approximability of the degrees. Consider a node $v \in V$. Let J' be the set of edges in $\delta_F(v)$ added to J while $v \in B$, and let J'' be the set of edges in E leaving v at Step 3 when v was excluded from B. Clearly, $\delta_F(v) \subseteq J' \cup J''$. Note that at the moment when v was excluded from B the set J'' of remaining edges satisfied the weighted degree constraints with bounds $b_{J'}^\alpha(v)$, namely, $\sum_{e \in J''} x_e w_e \leq b_{J'}^\alpha(v) = b(v) - w(\delta_{J'}(v))/\alpha$. By rearranging terms we have

$$w(J') \leq \alpha \left(b(v) - \sum_{e \in J''} x_e w_e \right) .$$

In particular, $w(J') \leq \alpha b(v)$. Also, $|J''| \leq \Delta$ and thus, by the initialization step of the algorithm, $w(J'') \leq |J''| \cdot b(v) \leq \Delta b(v)$. Consequently, $w(\delta_F(v)) \leq w(J') + w(J'') \leq \alpha b(v) + \Delta b(v) = (\alpha + \Delta) b(v)$.

Now consider the case of unit weights. We had $|J'| \leq \alpha \left(b(v) - \sum_{e \in J''} x_e \right)$ when v was excluded from B. Moreover, we had $x_e > 0$ for all $e \in J''$, since edges with $x_e = 0$ were removed at Step 2, before v was excluded from B. Hence if $J'' \neq \emptyset$ then $|J'| < \alpha b(v)$, and thus $|\delta_F(v)| \leq |J'| + |J''| < \alpha b(v) + \Delta$. Since all numbers are integers, this implies $|\delta_F(v)| \leq \alpha b(v) + \Delta - 1$. If $J'' = \emptyset$, then $|\delta_F(v)| = |J'| \leq \alpha b(v) \leq \alpha b(v) + \Delta - 1$. Consequently, in both cases $|\delta_F(v)| \leq \alpha b(v) + \Delta - 1$, as claimed. $\square$

4 Sparseness of $P(f_J, b_J^\alpha)$ (Proof of Theorem 2.2)

Note that if $x \in P(f_J, b_J^\alpha)$ is a basic solution so that $0 < x_e < 1$ for all $e \in E$, then every tight equation is induced by either:

- *cut constraint* $x(\delta_E^{in}(S)) \geq f_J(S)$ defined by some set $\emptyset \neq S \subset V$ with $f_J(S) \geq 1$.
- *degree constraint* $\sum_{e \in \delta_E(v)} x_e w_e \leq b_J^\alpha(v)$ defined by some node $v \in B$.

A family $\mathcal{F}$ of sets is *laminar* if its members are pairwise non-crossing, namely, if for every $S, S' \in \mathcal{F}$, either $S \cap S' = \emptyset$, or $S \subset S'$, or $S' \subset S$. We use the following statement observed in [15] for unit weights, which also holds in our setting.

Lemma 4.1 *For any basic solution x to $P(f_J, b_J^\alpha)$ with $0 < x_e < 1$ for all $e \in E$, there exist a laminar family $\mathcal{L}$ on V and $T \subseteq B$, such that $f_J(S) \geq 1$ for all $S \in \mathcal{L}$, and such that x is the* *unique solution to the linear equation system:*

$$\begin{array}{rcll} x(\delta_E^{in}(S)) & = & f_J(S) & \text{for all } S \in \mathcal{L} \\ \sum_{e \in \delta_E(v)} x_e w_e & = & b_J^\alpha(v) & \text{for all } v \in T \end{array}$$

In particular, $|\mathcal{L}| + |T| = |E|$ *and the characteristic vectors of* $\{\delta_E^{in}(S) : S \in \mathcal{L}\}$ *are linearly independent.*

Proof: Let $\mathcal{F} = \{\emptyset \neq S \subset V : x(\delta_E^{in}(S)) = f_J(S) \geq 1\}$ be the family of the tight sets and $T = \{v \in B : \sum_{e \in \delta_E(v)} x_e w_e = b(v) - w(\delta_J(v))/\alpha\}$ the set of tight nodes in B. For $\mathcal{F}' \subseteq \mathcal{F}$ let $\mathsf{span}(\mathcal{F}')$ denote the linear space generated by the characteristic vectors of $\delta_E^{in}(S)$, $S \in \mathcal{F}'$. Similarly, $\mathsf{span}(T')$ is the linear space generated by the weight vectors of $\delta_E(v)$, $v \in T'$. In [11] (see also [16]) it is proved that a maximal laminar subfamily $\mathcal{L}$ of $\mathcal{F}$ satisfies $\mathsf{span}(\mathcal{L}) = \mathsf{span}(\mathcal{F})$. Since $x \in P(f_J, b_J^\alpha)$ is a basic solution, and $0 < x_e < 1$ for all $e \in E$, $|E|$ is equal to the dimension of $\mathsf{span}(\mathcal{F}) \cup \mathsf{span}(T) = \mathsf{span}(\mathcal{L}) \cup \mathsf{span}(T)$. Hence repeatedly removing from T a node v so that $\mathsf{span}(\mathcal{L}) \cup \mathsf{span}(T - v) = \mathsf{span}(\mathcal{L}) \cup \mathsf{span}(T)$ results in $\mathcal{L}$ and T as required. $\square$

Let $\mathcal{L}$ and T be as in Lemma 4.1. Define a child-parent relation on the members of $\mathcal{L} \cup T$ as follows. For $S \in \mathcal{L}$ or $v \in T$, its parent is the inclusion minimal member of $\mathcal{L}$ properly containing it, if any. Note that if $v \in T$ and $\{v\} \in \mathcal{L}$, then $\{v\}$ is the parent of v, and that no member of T has a child. For every edge $uv \in E$ assign one *tail-token* to u and one *head-token* to v, so every edge contributes exactly 2 tokens. The number of tokens is thus $2|E|$.

Definition 4.1 *A tail-token or a head-token assigned to a node contained in S is an* S-token *if it is not a tail-token of an edge vu leaving S so that $v \notin T$ (so a tail-token of an edge vu leaving S is an S-token if, and only if, $v \in T$).*

Recall that we need to prove that if $x \in P(f_J, b_J^\alpha)$ is a basic solution so that $0 < x_e < 1$ for all $e \in E$, then there exists $e \in E$ with $x_e \geq 1/\alpha$ or there exists $v \in B$ with $|\delta(v)| \leq \Delta$. Assuming this is not so, we have:

The Negation Assumption:
- $0 < x_e < 1/\alpha$ for every $e \in E$; hence $|\delta^{in}(S)| \geq \alpha + 1$ for every $S \in \mathcal{L}$.
- $|\delta_E(v)| \geq \Delta + 1$ for every $v \in T$.

We obtain the contradiction $|E| > |\mathcal{L}| + |T|$ by showing that for any $S \in \mathcal{L}$ we can assign the S-tokens so that every proper descendant of S in $\mathcal{L} \cup T$ gets 2 S-tokens and S gets at least 3 S-tokens. Except the proof of $(2,3)$-sparseness of $0,1$-valued f, our assignment scheme will be:

The $(2, \alpha + 1)$-Scheme:
- Every proper descendant of S in $\mathcal{L} \cup T$ gets 2 S-tokens.
- S gets $\alpha + 1$ S-tokens.

The rest of the proof is by induction on the number of descendants of S in $\mathcal{L}$. If S has no children/descendants in $\mathcal{L}$, it has at least $|\delta_E^{in}(S)| \geq \alpha + 1$ head-tokens of the edges in $\delta_E^{in}(S)$. We therefore assume that S has in $\mathcal{L}$ at least one child. Given $S \in \mathcal{L}$ with at least one child in $\mathcal{L}$, let C_S be the set of edges entering some child of S, A_S the set of edges entering S or a child of S but not both, and D_S the set of edges that enter a child of S and their tail is in $T \cap S$. Formally:

$$\begin{aligned} C_S &= \bigcup \{\delta_E^{in}(R) : R \text{ is a child in } \mathcal{L} \text{ of } S\} \\ A_S &= (\delta_E^{in}(S) \setminus C_S) \cup (C_S \setminus \delta_E^{in}(S)) \\ D_S &= \{vu \in A_S : v \in T \cap S\} = \{vu \in C_S : v \in T \cap S\} \ . \end{aligned}$$

Lemma 4.2 *Let $S \in \mathcal{L}$. Then every edge $e \in A_S \setminus D_S$ has an endnode which was assigned an S-token that is not an R-token of any child R of S in $\mathcal{L}$. Furthermore, if one of the sets $\delta_E^{in}(S) \setminus C_S, C_S \setminus \delta_E^{in}(S)$ is empty, then the other has at least $\alpha + 1$ edges. In particular, $|A_S| \geq 2$.*

Proof: The first statement is straightforward. Note that $C_S = \delta_E^{in}(S)$ contradicts linear independence, hence at least one of the sets $\delta_E^{in}(S) \setminus C_S, C_S \setminus \delta_E^{in}(S)$ is nonempty. It is easy to verify that

$$x(C_S) - x(\delta_E^{in}(S)) = \sum_{R \text{ is a child of } S} x(\delta_E^{in}(R)) - x(\delta_E^{in}(S)) = \sum_{R \text{ is a child of } S} f(R) - f(S) \ .$$

Thus $x(C_S) - x(\delta_E^{in}(S))$ is an integer. If one of the sets $\delta_E^{in}(S) \setminus C_S, C_S \setminus \delta_E^{in}(S)$ is empty, say $\delta_E^{in}(S) \setminus C_S = \emptyset$, then $x(C_S) - x(\delta_E^{in}(S))$ must be a *positive* integer since $C_S \setminus \delta_E^{in}(S) \neq \emptyset$ and since $x_e > 0$ for every $e \in E$. Consequently, $|C_S \setminus \delta_E^{in}(S)| \geq \alpha + 1$, as otherwise $x_e \geq 1/\alpha$ for some $e \in C_S \setminus \delta_E^{in}(S)$. The proof of the case $C_S \setminus \delta_E^{in}(S) = \emptyset$ is identical. □

4.1 Arbitrary intersecting supermodular f

For $(2,5)$-sparseness the Negation Assumption is $|\delta_E^{in}(S)| \geq 3$ for all $S \in \mathcal{L}$, and $|\delta_E(v)| \geq 6$ for all $v \in T$. We prove that then the $(2,3)$-Scheme is feasible. First, for every $v \in T$, we reassign the $|\delta_E(v)|$ tail-tokens of the edges in $\delta_E(v)$ as follows:

- 3 tokens to v;
- 1/2 token to every edge in $\delta_E(v)$ (this is feasible since $|\delta_E(v)| \geq 6$).

Claim 4.3 *If S has at least 3 children in $\mathcal{L}$, then the* $(2,3)$-Scheme *is feasible.*

Proof: By moving one token from each child of S to S we get an assignment as required. □

Claim 4.4 *If S has exactly 2 children in $\mathcal{L}$ then the* $(2,3)$-Scheme *is feasible.*

Proof: S can get 2 tokens by taking one token from each child, and needs 1 more token. If there is $e \in A_S \setminus D_S$ then S can get 1 token from an endnode of e, by Lemma 4.2. Else, $|D_S| = |A_S| \geq 2$. As every edge in D_S owns 1/2 token, S can collect 1 token from edges in D_S. □

Claim 4.5 *If S has exactly 1 child in $\mathcal{L}$, say R, then the* $(2,3)$-Scheme *is feasible.*

Proof: S gets 1 token from R, and needs 2 more tokens. By our assignment scheme, S gets $|A_S \setminus D_S| + |D_S|/2 + |T \cap (S \setminus R)|$ S-tokens that are not R-tokens from edges in A_S and from the children of S in T. We claim that $|A_S \setminus D_S| + |D_S|/2 + |T \cap (S \setminus R)| \geq 2$. If $|A_S \setminus D_S| \geq 2$, we are done. If $|A_S \setminus D_S| \leq 1$ then by Lemma 4.2 $|A_S \cap D_S| \geq 1$, which implies $|T \cap (S \setminus R)| \geq 1$. If $|A_S \setminus D_S| = 0$ then $|D_S| = |A_S| \geq 2$. In all cases, our claim holds. $\square$

It is not hard to verify that during our distribution procedure no token was assigned twice. For "node" tokens this is obvious. For 1/2 "edge" tokens, note that if a 1/2 token of an edge e was assigned S, then S is the unique minimal inclusion set that contains both endnodes of e.

For $(3,3)$-sparseness the Negation Assumption is $|\delta_E^{in}(S)| \geq 4$ for all $S \in \mathcal{L}$ and $|\delta_E(v)| \geq 4$ for all $v \in T$. In this case we can easily prove that the $(2,4)$-Scheme is feasible. If S has at least 2 children in $\mathcal{L}$, then by moving 2 tokens from each child to S we get an assignment as required. If S has exactly 1 child in $\mathcal{L}$, say R, then S gets 2 tokens from R, and needs 2 more tokens. If $D_S = \emptyset$ then S can get 2 tokens from endnodes of the edges in A_S. Else, S has a child in T, and can get 2 tokens from this child.

4.2 Improved sparseness for $0,1$-valued f

Here the Negation Assumption is $|\delta_E^{in}(S)| \geq 3$ for all $S \in \mathcal{L}$ and $|\delta_E(v)| \geq 4$ for all $v \in T$. Assign colors to members of $\mathcal{L} \cup T$ as follows. All nodes in T are black; $S \in \mathcal{L}$ is black if $S \cap T \neq \emptyset$, and S is white otherwise. We show that given $S \in \mathcal{L}$, we can assign the S-tokens so that:

The $(2,3,4)$-Scheme
- every proper descendant of S gets 2 S-tokens;
- S gets at least 3 S-tokens, and S gets 4 S-tokens if S is black.

As in the other cases, the proof is by induction on the number of descendants of S in $\mathcal{L}$. If S has no descendants in $\mathcal{L}$, then S gets $|\delta_E^{in}(S)| \geq 3$ head tokens of the edges in $\delta_E^{in}(S)$; if S is black, then S has a child in T and S gets 1 more token from this child.

Lemma 4.6 *If $A_S = D_S$ then S has a child in T or at least 2 black children in $\mathcal{L}$.*

Proof: If $A_S = D_S$ then the tail of every edge in A_S is in $T \cap S$, so it either belongs to a black child of S in $\mathcal{L}$ or is a child of S in T. Thus if the statement of the lemma does not hold, then all edges in A_S must have tails in $T \cap R$ for some child R of S, and every edge that enters S also enters some child of S. Thus $\delta_E^{in}(R) \subseteq \delta^{in}(S)$, and since $x(\delta_E^{in}(R)) = x(\delta_E^{in}(S)) = 1$, we must have $\delta_E^{in}(R) = \delta_E^{in}(S)$. This contradicts linear independence. $\square$

Claim 4.7 *If S has in $\mathcal{L} \cup T$ at least 3 children, then the* $(2,3,4)$-Scheme *is feasible.*

Proof: S gets 3 tokens by taking 1 token from each child; if S is black, then one of these children is black, and S can get 1 more token. $\square$

Claim 4.8 *If S has in $\mathcal{L}$ exactly 2 children, say R, R', then the* $(2,3,4)$-Scheme *is feasible.*

Proof: If S has a child $v \in T$, then we are in the case of Claim 4.7. If both R, R' are black, then S gets 4 tokens, 2 from each of R, R'. Thus we assume that S has no children in T, and that at least one of R, R' is white, say R' is white. In particular, S is black if, and only if, R is black. Thus S only lacks 1 token, that does not come directly from R, R'. By Lemma 4.6 there is $e \in A_S \setminus D_S$, and S can get a token from an endnode of e, by Lemma 4.2. $\square$

Claim 4.9 *If S has in $\mathcal{L}$ exactly one child, say R, then the* $(2,3,4)$-Scheme *is feasible.*

Proof: Suppose that $T \cap (S \setminus R) = \emptyset$. Then S is black if, and only if, R is black. Thus S needs 2 S-tokens not from R. By the definition of D_S, every edge in D_S has tail in $T \cap (S \setminus R)$, hence the assumption $T \cap (S \setminus R) = \emptyset$ implies $D_S = \emptyset$. Consequently, $|A_S \setminus D_S| = |A_S| \geq 2$, by Lemma 4.2. Thus S can get 2 S-tokens from endnodes of the edges in A_S.

If there is $v \in T \cap (S \setminus R)$, then S can get 1 token from R, 2 tokens from v, and needs 1 more token. We claim that there is $e \in \delta^{in}_E(S) - \delta^{in}_E(R)$, and thus S can get the head-token of e. Otherwise, $\delta^{in}_E(S) \subseteq \delta^{in}_E(R)$, and since $x(\delta^{in}_E(S)) = x(\delta^{in}_E(R)) = 1$, we obtain $\delta^{in}_E(S) = \delta^{in}_E(R)$, contradicting linear independence. $\square$

This finishes the proof of Theorem 2.2, and thus also the proof of Theorem 1.1 is complete.

5 Indegree constraints (Proof of Theorem 1.2)

Here we prove Theorem 1.2. Consider the following polytope $P^{in}(f, b^{in})$:

$$\boxed{\begin{array}{rcll} x(\delta^{in}_E(S)) & \geq & f(S) & \text{for all } \emptyset \neq S \subset V \\ \displaystyle\sum_{e \in \delta^{in}_E(v)} x_e w_e & \leq & b^{in}(v) & \text{for all } v \in B^{in} \\ 0 \leq x_e & \leq & 1 & \text{for all } e \in E \end{array}}$$

Here we set $\alpha = 1$ and define $b^{in}_J(v) = b^{in}(v) - |\delta^{in}_J(v)|$. We say that $P(f_J, b^{in}_J)$ is Δ-sparse if any basic solution $x \in P(f_J, b^{in}_J)$ has an edge $e \in E$ with $x_e \in \{0,1\}$ or a node $v \in B^{in}$ with $|\delta^{in}_E(v)| \leq \Delta$. Lemma 3.1 and Theorem 2.1 easily extend to the indegree case, if in the algorithm at Step 4 we remove from B^{in} every $v \in B^{in}$ with $|\delta^{in}_E(v)| \leq \Delta$. We prove:

Theorem 5.1 *$P^{in}(f_J, b^{in}_J)$ is 3-sparse for intersecting supermodular f. For unit weights and integral indegree bounds, any basic solution of $P^{in}(f_J, b^{in}_J)$ always has an edge e with $x_e \in \{0,1\}$.*

Theorem 1.2 now easily follows by combining Theorem 5.1 with the following essentially known fact, for which we provide a proof-sketch for completeness of exposition.

Fact 5.2 *Let f be an intersecting supermodular set-function on V, and let F be an inclusion minimal edge set on V so that $H = (V,F)$ is f-connected. Then $|\delta_E^{in}(v)| \leq f_{\max}$ for all $v \in V$, where $f_{\max} = \max\{f(S) : S \subseteq V\}$.*

Proof: Consider the set-family $\mathcal{F} = \{S : |\delta_F^{in}(S)| = f(S) > 0\}$. It is well known and easy to see that $\mathcal{F}$ is an intersecting family, namely, $X \cap Y, X \cup Y \in \mathcal{F}$ for any $X, Y \in \mathcal{F}$ that intersect. For $e \in F$ let $\mathcal{F}^e = \{S \in \mathcal{F} : e \in \delta_F^{in}(S)\}$. Since F is an inclusion minimal edge set so that (V,F) is f-connected, $\mathcal{F}^e$ is non-empty for every $e \in F$. Furthermore, $\mathcal{F}^e$ is an intersecting family, hence among all sets $S \in \mathcal{F}^e$ there is a unique set S^e with $|S^e|$ minimal.

Suppose to the contrary that $|\delta_F^{in}(v)| \geq f_{\max} + 1$ for some $v \in V$. Let $e \in \delta_F^{in}(v)$ and let S^e as above. Since $|\delta_F^{in}(v)| \geq f_{\max} + 1$ there is $e' \in \delta_F^{in}(v)$ that does not enter S^e. Now consider the sets S^e and $S^{e'}$. These sets intersect and belong to $\mathcal{F}$, hence $S = S^e \cap S^{e'} \in \mathcal{F}$. It is also easy to see that $e, e' \in \delta_F^{in}(S)$. This contradicts the minimality of $|S^e|$ or of $|S^{e'}|$. $\square$

In the rest of this section we prove Theorem 5.1. In Lemma 4.1, we now have a set T^{in} of nodes corresponding to tight in-degree constraints. We prove that if $x \in P^{in}(f_J, b_J^{in})$ is a basic solution so that $x_e > 0$ for all $e \in E$, then there exists $e \in E$ with $x_e = 1$ or there exists $v \in T^{in}$ with $|\delta_E^{in}(v)| \leq 3$. Otherwise, we must have:

The Negation Assumption:
- $|\delta_E^{in}(S)| \geq 2$ for all $S \in \mathcal{L}$;
- $|\delta_E^{in}(v)| \geq 4$ for all $v \in T^{in}$.

Here an S-token is a token that is not a tail-token of an edge leaving S. Assuming Theorem 5.1 is not true, we show that given $S \in \mathcal{L}$, we can assign the S-tokens so that:

The $(2,2)$-Scheme:
S and every proper descendant of S in $\mathcal{L} \cup T^{in}$ gets 2 S-tokens.

The contradiction $|E| > |\mathcal{L}| + |T^{in}|$ is obtained by observing that if S is an inclusion maximal set in $\mathcal{L}$, then there are at least 2 edges entering S, and their tail-tokens are not assigned, since they are not S'-tokens for any $S' \in \mathcal{L}$.

Initial assignment:
For every $v \in T^{in}$, we assign the 4 head-tokens of some edges in $\delta_E^{in}(v)$.

The rest of the proof is by induction on the number of descendants of S, as before. If S has no children/descendants, it contains at least $|\delta_E^{in}(S)| \geq 2$ head-tokens, as claimed. If S has in $\mathcal{L} \cup T^{in}$

at least one child $v \in T^{in}$, then S gets 2 tokens from this child.

Thus we may assume that S has at least 1 child in $\mathcal{L}$ and no children in T^{in}. Let A_S be as in Lemma 4.2, so $|A_S| \geq 2$. One can easily verify that S can collect 1 S-token from an endnode of every edge in A_S. Thus the $(2,2)$-Scheme is feasible.

For the case of unit weights (and integral degree bounds), we can prove that any basic solution $x \in P^{in}(f_J, b^{in}_J)$ has an edge e with $x_e \in \{0,1\}$. This follows by the same proof as above, after observing that if $v \in T^{in}$ is a child of $S \in \mathcal{L}$, then $\delta^{in}_E(v) \neq \delta^{in}_E(S)$, as otherwise we obtain a contradiction to the linear independence in Lemma 4.1. Thus assuming that there are at least 2 edges in E entering any member of $\mathcal{L} \cup T^{in}$, we obtain a contradiction in the same way as before, by showing that the $(2,2)$-Scheme is feasible. Initially, every minimal member of $\mathcal{L} \cup T^{in}$ gets 2 head-tokens of some edges entering it. In the induction step, any $S \in \mathcal{L}$ can collect at least 2 S-tokens that are not tokens of its children, by Lemma 4.2.

Remark: Note that we also showed the well known fact (c.f., [18]), that if there are no degree constraints at all, then there is an edge $e \in E$ with $x_e \in \{0,1\}$.

6 The case of both indegree and outdegree constraints

Here we describe the slight modifications required to handle the case when there are both indegree and outdegree constraints. In this case, in Lemma 4.1, we consider the polytope $P = P(f_J, b^{\alpha}_J) \cap P^{in}(f_J, b^{in}_J)$. Then we have sets T and T^{in} of nodes corresponding to tight outdegree and indegree constraints, respectively. In the definition of $(\alpha, \Delta, \Delta^{in})$-sparseness we require that $x_e \geq 1/\alpha$ for some $e \in E$, or $|\delta_E(v)| \leq \Delta$ for some $v \in B$, or $|\delta^{in}_E(v)| \leq \Delta^{in}$ for some $v \in B^{in}$. Lemma 3.1 and Theorem 2.1 easily extend to the both indegree and outdegree constraints case.

Now we analyze the minor adjustment of the credit scheme. In what follows, let $S \in \mathcal{L}$, and suppose that S has in $\mathcal{L} \cup T \cup T^{in}$ a unique child $v \in T^{in}$ (possibly $S = \{v\}$).

Arbitrary weights: For arbitrary weights, we can show that P has sparseness $(\alpha, \Delta, \Delta^{in}) = (2,5,4)$, in the same way as in Section 4.1. The Negation Assumption for $v \in T^{in}$ is $|\delta^{in}_E(v)| \geq 5$, and we do not put any tokens on the edges leaving v (unless their tail is in T). Even if $\delta^{in}_E(S) = \delta^{in}_E(v)$ (in the case of arbitrary weights this may not contradict linear independence), the head-tokens of at least 5 edges entering v suffice to assign 2 tokens for v and 3 tokens to S. Hence in this case the approximation is $(\alpha, (\alpha+\Delta) \cdot b(v), \min\{\alpha + \Delta^{in}, f_{\max}\} \cdot b^{in}(v)) = (2, 7b(v), \min\{6, f_{\max}\} \cdot b^{in}(v))$. In a similar way we can also show the sparseness $(\alpha, \Delta, \Delta^{in}) = (3,3,4)$, and in this case the ratio is $(3, 6b(v), \min\{7, f_{\max}\} \cdot b^{in}(v))$.

Unit weights: In the case of unit weights, we must have $\delta_E^{in}(S) \neq \delta_E^{in}(v)$, as otherwise the equations of S and v are linearly dependent. Hence in this case, it is sufficient to require $|\delta_E^{in}(v)| \geq 4$, and the sparseness is $(\alpha, \Delta, \Delta^{in}) = (2, 5, 3)$. Consequently, the approximation is $(\alpha, \alpha b(v) + \Delta - 1, \min\{\alpha b^{in}(v) + \Delta^{in} - 1, f_{\max}\}) = (2, 2b(v) + 4, \min\{2b^{in}(v) + 2, f_{\max}\})$.

$0,1$**-valued** f**:** In the case of $0,1$-valued f, we can show that P has sparseness $(\alpha, \Delta, \Delta^{in}) = (2, 3, 4)$, in the same way as in Section 4.2. The negation assumption for a node $v \in T^{in}$ is $|\delta_E^{in}| \geq 5$; a member in $\mathcal{L}$ containing a node from T^{in} only is *not* black, unless it also contains a node from T. Hence in this case the approximation is $(\alpha, (\alpha+\Delta)\cdot b(v), \min\{\alpha+\Delta^{in}, f_{\max}\}\cdot b^{in}(v)) = (2, 5b(v), \min\{6, f_{\max}\}\cdot b^{in}(v)) = (2, 5b(v), b^{in}(v))$. If we also have unit weights, then $\delta_E^{in}(S) \neq \delta^{in}(v)$, by the linear independence; hence for unit weights we obtain the sparseness $(\alpha, \Delta, \Delta^{in}) = (2, 3, 3)$; the approximation in this case is $(\alpha, \alpha b(v) + \Delta - 1, \min\{\alpha b^{in}(v) + \Delta^{in} - 1, f_{\max}\}) = (2, 2b(v) + 2, \min\{2b^{in}(v) + 2, f_{\max}\}) = (2, 2b(v) + 2, 1)$.

Summarizing, we obtain the following result (see Table 2):

Theorem 6.1 DWDCN *with intersecting supermodular f admits a polynomial time algorithm that computes an f-connected graph H of cost $\leq 2 \cdot \tau^*$ so that the weighted (degree,indegree) of every $v \in V$ is at most: $(7b(v), \min\{6, f_{\max}\} \cdot b^{in}(v))$ for arbitrary f, and $(5b(v), b^{in}(v))$ for a $0,1$-valued f. Furthermore, for unit weights, the (degree,indegree) of every $v \in V$ is at most $(2b(v) + 4, \min\{2b^{in}(v) + 2, f_{\max}\})$ for arbitrary f, and $(2b(v) + 2, 1)$ for a $0,1$-valued f. The problem also admits a $(3, 6b(v), \min\{7, f_{\max}\}\cdot b^{in}(v))$-approximation algorithm for arbitrary weights and arbitrary intersecting supermodular f.*

type of f	intersecting supermodular	crossing supermodular
any f, any w	$(2, 7b(v), \min\{6, f_{\max}\} \cdot b^{in}(v))$ $(3, 6b(v), \min\{7, f_{\max}\} \cdot b^{in}(v))$	$(4, (7 + \min\{6, f_{\max}\}) \cdot b(v))$ $(6, (6 + \min\{7, f_{\max}\}) \cdot b(v))$
$0,1$-f, any w	$(2, 5b(v), b^{in}(v))$	$(4, 6b(v))$
any f, $w \equiv 1$	$(2, 2b(v) + 4, \min\{2b^{in}(v) + 2, f_{\max}\})$	$(4, 2b(v) + 4 + \min\{2b(v) + 2, f_{\max}\})$
$0,1$-f, $w \equiv 1$	$(2, 2b(v) + 2, 1)$	$(4, 2b(v) + 3)$

Table 2: Bicriteria approximation ratios for DWDCN with both indegree and outdegree constraints for intersecting and crossing supermodular f. For crossing supermodular f, only the approximation for outdegree are given, and the approximation for indegrees is the same with $b(v)$ replaced by $b^{in}(v)$; e.g., in the last row ($0,1$-f, $w \equiv 1$) the approximation is $(4, 2b(v) + 3, 2b^{in}(v) + 3)$. In general, each degree ratio in the right column (crossing supermodular f) is a sum of the corresponding indegree+outdegree ratios for intersecting supermodular f.

Finally, we can combine Theorem 6.1 with Fact 1.3 to deduce (see Table 2):

Corollary 6.2 DWDCN *with crossing supermodular f admits a polynomial time algorithm that* *computes an f-connected graph H of cost $\leq 4\tau^*$ so that the weighted (degree,indegree) of every $v \in V$ is at most: $(7+\min\{f_{\max},6\} \cdot b(v), 7+\min\{f_{\max},6\} \cdot b^{in}(v))$ for arbitrary f, and $(6b(v), 6b^{in}(v))$ for $0,1$-valued f. Furthermore, for unit weights, the (degree,indegree) of every $v \in V$ is at most $(2b(v)+4+\min\{2b(v)+2, f_{\max}\}, 2b^{in}(v)+4+\min\{2b^{in}(v)+2, f_{\max}\})$ for arbitrary f, and $(2b(v)+3, 2b^{in}(v)+3)$ for a $0,1$-valued f.*

7 A $(b(v)+3)$-approximation (Proof of Theorem 1.7)

The key statement in the proof of Theorem 1.7 is the following.

Theorem 7.1 *Let f be intersecting supermodular and let $x \in P(f,b)$, be a basic feasible solution such that $0 < x < 1$ and such that all edges in E have their tail in B. Then there exists $v \in B$ with $|\delta_E(v)| \leq b(v)+3$.*

For a partial solution J let $b_J(v) = b(v) - |\delta_J(v)|$, while f_J is defined as before. Using Theorem 7.1, it is now a routine to show that the following algorithm computes a solution as in Theorem 1.7.

Algorithm for DWDCN with intersecting supermodular f and unit weights
Initialization: $J \leftarrow \emptyset$;
While $E \neq \emptyset$ do:

1. Compute a basic feasible solution to $P(f_J, b_J)$.
2. If there is $e \in E$ with $x_e = 0$ set $E \leftarrow E \setminus \{e\}$.
3. If there is $e \in E$ with $x_e = 1$ set $J \leftarrow J \cup \{e\}$, $E \leftarrow E \setminus \{e\}$.
4. If there is $e = uv \in E$ with $u \notin B$ set $J \leftarrow J \cup \{e\}$, $E \leftarrow E \setminus \{e\}$.
5. If there is $v \in B$ with at most $b(v)+3$ edges in E leaving v set $B \leftarrow B \setminus \{v\}$.

EndWhile
Return $F \leftarrow J$.

In the rest of this section we prove Theorem 7.1.

Claim 7.2 *Let $\mathcal{L}$ and T be as in Lemma 4.1. If $2|\mathcal{L}| < |E| + x(E) + (q-1)|B|$ for an integer q, then there exists $v \in B$ so that $|\delta_E(v)| - b(v) \leq q$.*

Proof: Note that $|E| = |T| + |\mathcal{L}| \leq |B| + |\mathcal{L}|$. Thus by the assumption of the claim

$$|\mathcal{L}| < |E| + x(E) + (q-1)|B| - |\mathcal{L}| = x(E) + q|B| + |E| - |B| - |\mathcal{L}| \leq x(E) + q|B| \ .$$

Thus it is sufficient to show that if $|\mathcal{L}| < x(E) + q|B|$ holds, then there exists $v \in B$ so that $|\delta_E(v)| - b(v) \leq q$. As every $uv \in E$ has its tail in B, it follows that $\sum_{v \in B} |\delta_E(v)| = |E|$ and

$\sum_{v\in B} x(\delta_E(v)) = x(E)$. Since x is a feasible solution, $b(v) \geq x(\delta_E(v))$. Thus

$$\sum_{v\in B}(|\delta_E(v)|-b(v)) \leq |E|-\sum_{v\in B} x(\delta_E(v)) = |E|-x(E) = |\mathcal{L}|+|T|-x(E) \leq |\mathcal{L}|+|B|-x(E) < (q+1)|B| \, .$$

This implies that there is $v \in B$ with $|\delta_E(v)| - b(v) < q + 1$, and since $|\delta_E(v)| - b(v)$ is an integer, we must have $|\delta_E(v)| - b(v) \leq q$. □

We apply Claim 7.2 with $q = 3$, namely, we will prove that $2|\mathcal{L}| < |E| + x(E) + 2|B|$ using a counting argument. We assign tokens to edges in E and nodes in B of total amount $|E|+x(E)+2|B|$, and show that these tokens can be redistributed among the sets of $\mathcal{L}$ so that every set gets at least 2 tokens, and at least one set gets at least 3 tokens.

Initial token assignment:
$1 + x_e$ tokens to every $e \in E$ placed at the head of e, 2 tokens to every $v \in B$.

Definition 7.1 *A set $S \in \mathcal{L}$ is* black *if $S \cap B \neq \emptyset$ and S is* white *otherwise.*

The assignment scheme
- Every proper descendant of S in $\mathcal{L}$ gets 2 tokens;
- S gets 3 tokens if S is a white leaf, and S gets 4 tokens otherwise.

We prove that the above assignment scheme is feasible by induction on the number of descendants of S in $\mathcal{L}$.

Claim 7.3 *If S is a leaf or if S is white then the above assignment scheme is feasible.*

Proof: Let E_S be the set of edges entering S that do not enter the children of S. From linear independence and the integrality of cuts it follows that if S is a leaf, or if S is white, then $|E_S| \geq 2$ and $x(E_S)$ is a positive integer. Hence S gets $|E_S| + x(E_S) \geq 3$ tokens from the edges in E_S. If S is a black leaf, then it gets 1 more token from some $v \in B \cap S$. If S is not a leaf then S gets 1 more token from its child. □

Claim 7.4 *If S is black and has at least 3 children then the above assignment scheme is feasible.*

Proof: S gets at least 3 tokens from its children, and if one of them is black then S gets another token from this child. If all the children of S are white, then there is $v \in B \cap S$ so that v does not belong to a child of S; hence S gets 1 token from v. □

Claim 7.5 *If S is black and has exactly 2 children then the above assignment scheme is feasible.*

Proof: If no child of S is a white leaf, then S gets 2 tokens from each child, a total of 4 tokens. If both children of S are white leaves, then S gets 1 token from each child, and also 2 tokens from some $v \in B \cap S$. The remaining case is when one child R is black and the other R' is a white leaf. S gets 2 tokens from R and 1 token from R'. S gets 1 more token if there is an edge entering

S but not a child of S, or if there is $v \in B \cap S$ so that v does not belong to a child of S. In the remaining case, $\delta_E(S) \cup \delta_E(R) \cup \delta_E(R')$ is a disjoint union of the three sets: $\delta_E(S) \cap \delta_E(R)$, $\delta_E(S) \cap \delta_E(R')$, and the set $\delta_E(R, R')$ of edges that go from R to R'. By linear independence and the integrality of cuts, each one of these sets contains at least 2 edges, and its x-value is an integer. Thus $x(\delta_E(R')) = x(\delta_E(S) \cap \delta_E(R')) + x(\delta_E(R, R')) \geq 2$. Consequently, $|\delta_E(R')| + x(\delta_E(R')) \geq 4 + 2 = 6$, hence the white leaf R' has 2 (and in fact at least 4) spare tokens for S. $\square$

Claim 7.6 *If S is black and has a unique child R then the above assignment scheme is feasible.*

Proof: Let $E_S = \delta_E(S) \backslash \delta_E(R)$ and $E_R = \delta_E(R) \backslash \delta_E(S)$. By the linear independence and integrality of cuts $|E_S| + |E_R| \geq 2$. If $|E_S| \geq 2$, then S gets 3 tokens from E_S and an additional token comes from R. If $|E_S| \leq 1$ then $|E_R| \geq 1$, and thus there is $v \in B \cap (S \setminus R)$ (any tail of an edge in E_R). If $|E_S| \leq 1$ then S gets 2 tokens from v, and, unless $|E_S| = 0$ and R is a white leaf, S can collect 2 tokens from R and the edge in E_S. The remaining case is $|E_S| = 0$ and R is a white leaf. Then $\delta_E(R)$ is a disjoint union of two non-empty sets $\delta_E(S)$ and $\delta_E(R) \setminus \delta_E(S)$. By linear independence and the integrality of cuts, each one of these sets contains at least 2 edges, and its x-value is an integer. Thus $|\delta_E(R)| \geq 4$ and $x(\delta_E(R)) \geq 2$. Consequently, $|\delta_E(R)| + x(\delta_E(R)) \geq 4 + 2 = 6$, hence the white leaf R has 2 (and in fact at least 4) spare tokens for S. $\square$

The proof of Theorem 7.1, and thus also of Theorem 1.7 is now complete.

Acknowledgment: I thank an anonymous referee for many useful comments.

References

[1] J. Bang-Jensen, S. Thomassé, and A. Yeo. Small degree out-branchings. *J. of Graph Theory*, 42(4):287–307, 2003.

[2] N. Bansal, R. Khandekar, and V. Nagarajan. Additive guarantees for degree bounded directed network design. In *STOC*, pages 769–778, 2008.

[3] K. Chaudhuri, S. Rao, S. Riesenfeld, and K. Talwar. A push-relabel algorithm for approximating degree bounded MSTs. In *ICALP*, pages 191–201, 2006.

[4] J. Edmonds. Edge-disjoint branchings. *Combinatorial Algorithms*, pages 91–96, 1972.

[5] A. Frank. Connectivity and network flows. In R. L. Graham, M. Grötschel, and L. Lovász, editors, Chapter 2 in *Handbook of Combinatorics*, pages 111–177. Elsevier, 1995.

[6] T. Fukunaga and H. Nagamochi. Network design with weighted degree constraints. In *WALCOM*, pages 214–225, 2009.

[7] M. Furer and B. Raghavachari. Approximating the minimum-degree Steiner tree to within one of optimal. *Journal of Algorithms*, 17(3):409–423, 1994.

[8] H. N. Gabow. A representation for crossing set families with application to submodular flow problems. In *SODA*, pages 202–211, 1993.

[9] M. X. Goemans. Minimum bounded degree spanning trees. In *FOCS*, pages 373–282, 2006.

[10] M. X. Goemans and D. P. Williamson. The primal-dual method in approximation algorithms and its applications to network design problems. In D. S. Hochbaum, editor, Chapter 4 in*Approximation Algorithms For NP-hard Problems*. PWS, 1997.

[11] K. Jain. A factor 2 approximation algorithm for the generalized Steiner network problem. *Combinatorica*, 21(1):39–60, 2001.

[12] S. Khuller. Approximation algorithm for finding highly connected subgraphs. In D. S. Hochbaum, editor, Chapter 6 in *Approximation Algorithms For NP-hard Problems*. PWS, 1997.

[13] J. Könemann and R. Ravi. A matter of degree: Improved approximation algorithms for degree bounded minimum spanning trees. *SIAM Journal on Computing*, 31(3):1783–1793, 2002.

[14] G. Kortsarz and Z. Nutov. Approximating minimum-cost connectivity problems. In T. F. Gonzalez, editor, Chapter 58 in *Approximation Algorithms and Metaheuristics*. Chapman & Hall/CRC, 2007.

[15] L. C. Lau, J. Naor, M. R. Salavatipour, and M. Singh. Survivable network design with degree or order constraints. In *STOC*, pages 651–660, 2007.

[16] V. Melkonian and E. Tardos. Algorithms for a network design problem with crossing supermodular demands. *Networks*, 43(4):256–265, 2004.

[17] R. Ravi, M. V. Marathe, S. S. Ravi, D. J. Rosenkrantz, and H. B. Hunt III. Many birds with one stone: Multi objective approximation algorithms. In *STOC*, pages 438–447, 1993.

[18] A. Schrijver. *Combinatorial Optimization, Polyhedra and Efficiency*. Springer-Verlag Berlin, Heidelberg New York, 2004.

[19] M. Singh and L. C. Lau. Approximating minimum bounded degree spanning trees to within one of optimal. In *STOC*, pages 661–670, 2007.