Approximating Directed Weighted-Degree Constrained Networks

Zeev Nutov

The Open University of Israel, Raanana, Israel
nutov@openu.ac.il

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 $G = (V, E)$, edge-costs $\{c(e) : e \in E\}$, edge-weights $\{w(e) : e \in E\}$, an intersecting supermodular set-function f on V, and degree bounds $\{b(v) : v \in V\}$. 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 $(1, 4)$-approximation algorithm for arbitrary weights and a polynomial time algorithm for unit weights. Finally, we consider the problem of packing maximum number k of edge-disjoint arborescences so that their union satisfies weighted degree constraints, and give an algorithm that computes a solution of value at least $\lfloor k/36 \rfloor$.

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 [16, 4, 8, 10, 12].

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., [16]. 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.

The connectivity requirements can be specified by a set function f on V, as follows.

Definition 1. *For an edge set or a graph H and node set S let $\delta_H(S)$ ($\delta_H^{in}(S)$) denote the set of edges in H leaving (entering) S. Given a set-function f on subsets of V and a graph $H = (V, F)$, we say that H is* f-connected *if*

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

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

Definition 2. *A set function f on V is* intersecting supermodular *if for any $X, Y \subseteq V$, $X \cap Y \neq \emptyset$*

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

We consider *directed* network design problems with *weighted-degree* constraints. For simplicity of exposition, we will consider mainly out-degree constraints, but our results easily extend to the case when there are also in-degree constraints, see Section 6. The problem we consider is:

Directed Weighted Degree Constrained Network (DWDCN)
Instance: A directed graph $G = (V, E)$, edge-costs $\{c(e) : e \in E\}$, edge-weights $\{w(e) : e \in E\}$, set-function f on V, and degree bounds $\{b(v) : v \in V\}$.
Objective: Find a minimum cost f-connected subgraph $H = (V, F)$ of G that satisfies the weighted degree constraints

$$w(\delta_H(v)) \leq b(v) \quad \text{for all } v \in V \ . \tag{3}$$

We assume that f admits a polynomial time evaluation oracle. Since for most 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 (α, β)-approximation algorithm for DWDCN either computes an f-connected subgraph $H = (V, F)$ of G of cost $\leq \alpha \cdot \mathsf{opt}$ that satisfies $w(\delta_H(v)) \leq \beta \cdot b(v)$ for all $v \in V$, 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) by a factor of β.

A graph H is *k-edge-outconnected from r* if it has k-edge-disjoint paths from r to any other node. DWDCN includes as a special case the Weighted Degree Constrained k-Outconnected Subgraph problem, by setting $f(S) = k$ for all $\emptyset \neq S \subseteq V - r$, and $f(S) = 0$ otherwise. For $k = 1$ we get the Weighted Degree Constrained Arborescence problem. 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, this is equivalent to requiring that H is k-edge-otconnected from r and satisfies (3). This gives the following problem:

Weighted Degree Constrained Maximum Arborescence Packing (WDCMAP)
Instance: A directed graph $G = (V, E)$, edge-weights $\{w(e) : e \in E\}$, degree bounds $\{b(v) : v \in V\}$, and $r \in V$.
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 theorem. For an edge set I, let $x(I) = \sum_{e \in I} x(e)$. Let opt denote the optimal value of the following natural LP-relaxation for DWDCN that seeks to minimize $c \cdot x$ over the following polytope P_f:

$$\begin{aligned} x(\delta_E^{in}(S)) \geq f(S) \qquad & \text{for all } \emptyset \neq S \subset V \\ \sum_{e \in \delta_E(v)} x(e)w(e) \leq b(v) \qquad & \text{for all } v \in V \\ 0 \leq x(e) \leq 1 \qquad & \text{for all } e \in E \end{aligned}$$

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

Interistingly, we can show a much better result for the case of indegree constraints only (for the case of both indegree and outdegree constraints see Section 6).

Theorem 2. DWDCN *with intersecting supermodular f and with indegree constraints only, admits a $(1, 4)$-approximation algorithm for arbitrary weights, and a polynomial time algorithm for unit weights.*

Theorem 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 such 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 which cost and weighted degrees are not much larger than the "expected" values $c(G) \cdot (\ell/k)$ and $w_G(v) \cdot (\ell/k)$. In particular, one can find an arborescence with both low weighted degrees and low cost.

Corollary 1. *Let $H_k = (V, F)$ be a k-outconnected from r directed graph with costs $\{c(e) : e \in F\}$ and weights $\{w(e) : e \in F\}$. Then for any $\ell \leq k$ the graph H_k contains an ℓ-outconnected from r 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-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 $\emptyset \neq S \subseteq V - r$, and $f(S) = 0$ otherwise. By Theorem 1, our algorithm computes a subgraph H_ℓ as required.

Another application is for the WDCMAP problem. Ignoring costs, Theorem 1 implies a "pseudo-approximation" algorithm for WDCMAP that computes the maximum number k of packed arborescences, but violates the weighted degrees. E.g., using the $(3, 6)$-approximation algorithm from Theorem 1, we can compute a k-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≠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 3. 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 very simple. We set $b'(v) \leftarrow b(v)/6$ for all $v \in V$ and apply the $(3, 6)$-approximation algorithm from Theorem 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 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.

We note that Theorem 3 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-outconnected subgraph.

1.3 Previous and related work

Fürer and Raghavachari [6] 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., [15, 11, 3]. Goemans [7] presented an algorithm that computes a spanning tree of cost $\leq \mathsf{opt}$ and with degrees at most $b(v)+2$ for all $v \in V$, where $b(v)$ is the degree bound of v. An optimal result was obtained by Singh and Lau [17]; their algorithm computes a spanning tree of cost $\leq \mathsf{opt}$ and with degrees at most $b(v)+1$ for all $v \in V$. The algorithm of Singh and Lau [17] uses the method of *iterative rounding*. This method was initiated in a seminal paper of Jain [9] 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 [17] 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 [13] 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 skew-supermodular. For directed graphs, they gave a $(4\mathsf{opt}, 4b(v)+6)$-approximation for intersecting supermodular f, and $(8\mathsf{opt}, 8b(v)+6)$-approximation for *crossing supermodular* f (when (2) holds for any X, Y that cross). Recently, in the full version of [13], these ratios were improved to $(3\mathsf{opt}, 3b(v)+5)$ for crossing supermodular f, and $(2\mathsf{opt}, 2b(v)+2)$ for $0,1$-valued intersecting supermodular f. For the latter case we obtain the same ratio, but our proof is simpler than the one in the full version of [13].

Bansal, Khandekar, and Nagarajan [2] gave for intersecting supermodular f a $(\frac{1}{\varepsilon} \cdot \mathsf{opt}, \lceil \frac{b(v)}{1-\varepsilon} \rceil + 4)$-approximation scheme, $0 \leq \varepsilon \leq 1/2$. They also showed, that this ratio cannot be much improved based on the standard LP-relaxation. For crossing supermodular f [2] gave a $(\frac{2}{\varepsilon} \cdot \mathsf{opt}, \lceil \frac{b(v)}{1-\varepsilon} \rceil + 4 + f_{\max})$-approximation scheme. For the degree constrained arborescence problem (without costs) [2] give an algorithm that computes an arborescence H with $|\delta_H(v)| \leq b(v)+2$ for all $v \in V$. Some additional results for related problems can also be found in [2].

For weighted degrees, Fukunaga and Nagamochi [5] considered *undirected* network design problems and gave a $(1,4)$-approximation for minimum spanning trees and a $(2,7)$-approximation algorithm for arbitrary weakly supermodular set-function f.

2 Proof of Theorem 1

During the algorithm, F denotes the partial solution, I are the edges to add to F, and B is the set of nodes on which the outdegree bounds constraints are still present. The algorithm starts with $F = \emptyset$, $B = V$ and performs iterations. In any iteration, we work with the "residual problem" polytope $P_f(I, F, B)$ ($\alpha \geq 1$ is a fixed parameter):

$$
\begin{aligned}
x(\delta_I^{in}(S)) &\ge f(S) - |\delta_F^{in}(S)| && \text{for all } \emptyset \ne S \subset V \\
\sum_{e \in \delta_I(v)} x(e)w(e) &\le b(v) - w(\delta_F(v))/\alpha && \text{for all } v \in B \\
0 \le x(e) &\le 1 && \text{for all } e \in I
\end{aligned}
$$

Recall some facts from polyhedral theory. Let x belong to a polytope $P \subseteq R^m$ defined by a system of linear inequalities; an inequality is *tight* (for x) if it holds as equality for x. $x \in P$ is a *basic solution* for (the system defining) P if there exist 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 $\min\{c \cdot x : x \in P\}$ has an optimal solution, then it has an optimal solution which is basic, and that a basic optimal solution for $\{c \cdot x : x \in P_f(I,F,B)\}$ can be computed in polynomial time, c.f., [13].

Note that if $x \in P_f(I,F,B)$ is a basic solution so that $0 < x(e) < 1$ for all $e \in I$, then every tight equation is induced by either:

- *cut constraint* $x(\delta_I^{in}(S)) \ge f(S) - |\delta_F^{in}(S)|$ defined by some set $\emptyset \ne S \subset V$ with $f(S) - |\delta_F^{in}(S)| \ge 1$.
- *degree constraint* $\sum_{e \in \delta_I(v)} x(e)w(e) \le b(v) - w(\delta_F(v))/\alpha$ defined by some node $v \in B$.

A family $\mathcal{F}$ of sets is *laminar* 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 [13] for unit weights, which also holds in our setting.

Lemma 1. *For any basic solution x to $P_f(I,F,B)$ with $0 < x(e) < 1$ for all $e \in I$, there exist a laminar family $\mathcal{L}$ on V and $T \subseteq B$ such that x is the unique solution to the linear equation system:*

$$
\begin{aligned}
x(\delta_I^{in}(S)) &= f(S) - |\delta_F^{in}(S)| && \text{for all } S \in \mathcal{L} \\
\sum_{e \in \delta_I(v)} x(e)w(e) &= b(v) - w(\delta_F(v))/\alpha && \text{for all } v \in T
\end{aligned}
$$

where $f(S) - |\delta_F^{in}(S)| \ge 1$ for all $S \in \mathcal{L}$. In particular, $|\mathcal{L}| + |T| = |I|$ and the characteristic vectors of $\delta_I^{in}(S)$ for all $S \in \mathcal{L}$ are linearly independent.

Proof. Let $\mathcal{F} = \{\emptyset \ne S \subset V : x(\delta_E^{in}(S)) = f(S) - |\delta_F^{in}(S)| \ge 1\}$, (i.e., the tight sets) and $T = \{v \in B : \sum_{e \in \delta_I(v)} x(e)w(e) = b(v) - w(\delta_F(v))/\alpha\}$ (i.e., the 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 $\delta_I^{in}(S)$, $S \in \mathcal{F}'$. Similarly, $\mathsf{span}(T')$ is the linear space generated by the weight vectors of $\delta_I(v)$, $v \in T'$. In [9] (see also [14]) it is proved that a maximal laminar subfamily $\mathcal{L}$ of $\mathcal{F}$ satisfies $\mathsf{span}(\mathcal{L}) = \mathsf{span}(\mathcal{F})$. Since $x \in P_f(I,F,B)$ is a basic solution, and $0 < x(e) < 1$ for all $e \in I$, $|I|$ is at most 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.

Definition 3. *The polytope $P_f(I, F, B)$ is (α, Δ)-*sparse *for integers $\alpha, \Delta \geq 1$ if any basic solution $x \in P_f(I, F, B)$ has an edge $e \in I$ with $x(e) = 0$, or satisfies at least one of the following:*

$$x(e) \geq 1/\alpha \quad \textit{for some } e \in I \tag{4}$$

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

We prove the following two general statements that imply Theorem 1:

Theorem 4. *If for any I, F the polytope $P_f(I, F, B)$ 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 $c(F) \leq \alpha \cdot \mathsf{opt}$ and $|\delta_F(v)| \leq \alpha b(v) + \Delta - 1$ for all $v \in V$.*

Theorem 5. *$P_f(I, F, B)$ is $(2, 5)$-sparse and $(3, 3)$-sparse for intersecting supermodular f; if f is $0, 1$-valued, then $P_f(I, F, B)$ is $(2, 3)$-sparse.*

3 The Algorithm (Proof of Theorem 4)

The algorithm perform iterations. Every iteration either removes at least one edges from I 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: $F \leftarrow \emptyset$, $B \leftarrow V$, $I \leftarrow E - \{vu \in E : w(vu) > b(v)\}$.
If $P_f(I, F, B) = \emptyset$, then return "UNFEASIBLE" and STOP.
While $I \neq \emptyset$ do:
1. Find a basic solution $x \in P_f(I, F, B)$.
2. Remove from I all edges with $x(e) = 0$.
3. Add to F and remove from I all edges with $x(e) \geq 1/\alpha$.
4. Remove from B every $v \in B$ with $|\delta_I(v)| \leq \Delta$.

EndWhile

Lemma 2. DWDCN *admits an $(\alpha, \alpha + \Delta)$-approximation algorithm if every polytope $P_f(I, F, B)$ constructed during the algorithm is (α, Δ)-sparse; furthermore, for unit weights, the algorithm computes a solution F so that $c(F) \leq \alpha \cdot \mathsf{opt}$ and $|\delta_F(v)| \leq \alpha b(v) + \Delta - 1$ for all $v \in V$.*

Proof. Clearly, if $P_f(I, F, B) = \emptyset$ at the beginning of the algorithm, then the problem has no feasible solution, and the algorithm indeed outputs "INFEASIBLE". It is also easy to see that if $P_f(I, F, B) \neq \emptyset$ at the beginning of the algorithm, then $P_f(I, F, B) \neq \emptyset$ 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 $\leq \alpha \cdot \mathsf{opt}$.

Now we prove the approximability of the degrees. Consider a node $v \in V$. Let F' be the set of edges in $\delta_F(v)$ added to F while $v \in B$, and let F'' be the set of edges in I leaving v at Step 3 when v was excluded from B. Clearly, $\delta_F(v) \subseteq F' \cup F''$. Note that at the moment when v was excluded from B we had

$$w(F') \leq \alpha \left(b(v) - \sum_{e \in F''} x(e)w(e) \right)$$

In particular, $w(F') \leq \alpha b(v)$. Also, $|F''| \leq \Delta$ and thus $w(F'') \leq |F''| \cdot b(v) \leq \Delta b(v)$. Consequently, $w(\delta_F(v)) \leq w(F')+w(F'') \leq \alpha b(v)+\Delta b(v) = (\alpha+\Delta)b(v)$.

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

4 Sparseness of $P_f(I, F, B)$ (Proof of Theorem 5)

Let $\mathcal{L}$ and T be as in Lemma 1. Define a child-parent relation on the members of $\mathcal{L} + T$ 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 members of T has a child. For every edge $uv \in I$ 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 $2|I|$.

Definition 4. *A token 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 $x \in P_f(I, F, B)$ is a basic solution so that $0 < x(e) < 1$ for all $e \in I$, then there exists $e \in I$ with $x(e) \geq 1/\alpha$ or there exists $v \in B$ with $|\delta_I(v)| \leq \Delta$. Assuming this is not so, we have:

The Negation Assumption:
- $|\delta_I^{in}(S)| \geq \alpha + 1$ for all $S \in \mathcal{L}$;
- $|\delta_I(v)| \geq \Delta + 1$ for all $v \in T$.

We obtain the contradiction $|I| > |\mathcal{L}| + |T|$ by showing that for any $S \in \mathcal{L}$ we can assign the S-tokens so that every proper descendant of S in $\mathcal{L} + T$ 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 $\mathcal{L}+T$ gets 2 S-tokens;
- S gets $\alpha+1$ S-tokens.

Initial assignment:
For every $v \in T$, assign the $|\delta_I(v)|$ tail-tokens of the edges in $\delta_I(v)$.

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 $|\delta_I^{in}(S)| \geq \alpha+1$ head-tokens of the edges in $\delta_I^{in}(S)$. 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 be the set of edges entering some child of S, J the set of edges entering S or a child of S but not both, and D the set of edges that enter a child of S and their tail is in $T \cap S$ but not in a child of S. Formally:

$$\begin{aligned} C &= \bigcup\{\delta_I^{in}(R) : R \text{ is a child in } \mathcal{L} \text{ of } S\} \\ J &= (\delta_I^{in}(S) - C) \cup (C - \delta_I^{in}(S)) \\ D &= \{e = vu \in C - \delta_I^{in}(S) : v \in T\} \ . \end{aligned}$$

Lemma 3. *Let $S \in \mathcal{L}$ and suppose that $0 < x(e) < 1$ for all $e \in E$. Then $|J| \geq 2$, and every edge $e \in J - D$ has an endnode that owns an S-token that is not an R-token of any child R of S in $\mathcal{L}$.*

Proof. $C = \delta_I^{in}(S)$ contradicts linear independence, hence one of the sets $\delta_I^{in}(S) - C, C - \delta_I^{in}(S)$ is nonempty. If one of these sets is empty, say $\delta_I^{in}(S) - C = \emptyset$, then $x(C) - x(\delta_I^{in}(S))$ must be a *positive* integer. Thus $|C - \delta_I^{in}(S)| \geq 2$, as otherwise there is an edge $e \in C - \delta_I^{in}(S)$ with $x(e) = 1$. The proof of the case $C - \delta_I^{in}(S) = \emptyset$ is identical. The second statement is straightforward.

4.1 Arbitrary intersecting supermodular f

For $(2,5)$-sparseness the Negation Assumption is $|\delta_I^{in}(S)| \geq 3$ for all $S \in \mathcal{L}$, and $|\delta_I(v)| \geq 6$ for all $v \in T$. We prove that then the $(2,3)$-Scheme is feasible. First, for every $v \in T$, we reassign the $|\delta_I(v)|$ tail-tokens assigned to v as follows:
- 3 tokens to v;
- $1/2$ token to every edge in $\delta_I(v)$ (this is feasible since $|\delta_I(v)| \geq 6$).

Claim. 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. 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 $e \in J - D$ then S can get 1 token from an endnode of e, by Lemma 3. Else, $|D| = |J| \geq 2$. As every edge in D owns $1/2$ token, S can collect 1 token from edges in D.

Claim. 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. We can collect $|J-D|+|D|/2+|T\cap(S-R)|$ S-tokens that are not R-tokens, from edges in J and from the children of S in T, by Lemma 3 and our assignment scheme. We claim that $|J-D|+|D|/2+|T\cap(S-R)|\geq 2$. This follows from the observation that if $|J-D|\leq 1$ then $|T\cap(S-R)|\geq 1$, and if $|J-D|=0$ then $|D|=|J|\geq 2$, by Lemma 3.

It is easy to see that during our distribution procedure no token was assigned twice. For "node" tokens this is obvious. For 1/2 tokens on the edges, this follows from the fact that each time we assigned a 1/2 token of an edge, both endnodes of this edge were inside S, as this edge was connecting the two children of S.

For $(3,3)$-sparseness the Negation Assumption is $|\delta_I^{in}(S)|\geq 4$ for all $S\in\mathcal{L}$ and $|\delta_I(v)|\geq 4$ 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 $D=\emptyset$ then S can get 2 tokens from endnodes of the edges in J. 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 $|\delta_I^{in}(S)|\geq 3$ for all $S\in\mathcal{L}$ and $|\delta_I(v)|\geq 4$ for all $v\in T$. Assign colors to members of $\mathcal{L}+T$ 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 $|\delta_I^{in}(S)|\geq 3$ head tokens of the edges in $\delta_I^{in}(S)$; if S is black, then S has a child in T and S gets 1 more token from this child.

Lemma 4. *If $J=D$ then S has a child in T or at least 2 black children in $\mathcal{L}$.*

Proof. Otherwise, all edges in J 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 $\delta_I^{in}(R)\subseteq\delta_I^{in}(S)$, and since $x(\delta_I^{in}(R))=x(\delta_I^{in}(S))=1$, we must have $\delta_I^{in}(R)=\delta_I^{in}(S)$. This contradicts linear independence.

Claim. If S has in $\mathcal{L}+T$ 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.

Claim. 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 4.2. 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 there is $e \in J - D$, and S can get a token from an endnode of e, by Lemma 3.

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

Proof. Suppose that $T \cap (S - R) = \emptyset$. Then S is black if, and only if, R is black. Thus S needs 2 S-tokens not from R. As every edge in D has tail in $T \cap (S - R)$ and head in R, $D = \emptyset$ so $|J - D| = |J| \geq 2$, and thus S can get 2 S-tokens from endnodes of the edges in J, by Lemma 3.

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

This finishes the proof of Theorem 5, and thus also the proof of Theorem 1 is complete.

5 Indegree constraints only (Proof of Theorem 2)

Here we prove Theorem 2. Consider the following polytope $P_f^{in}(I, F, B)$:

$$\begin{aligned} x(\delta_I^{in}(S)) &\geq f(S) - |\delta_F^{in}(S)| && \text{for all } \emptyset \neq S \subset V \\ \sum_{e \in \delta_I^{in}(v)} x(e)w(e) &\leq b(v) - w(\delta_F^{in}(v)) && \text{for all } v \in B \\ 0 \leq x(e) &\leq 1 && \text{for all } e \in I \end{aligned}$$

Theorem 6. *$P_f^{in}(I, F, B)$ is $(1,3)$-sparse for intersecting supermodular f. For unit weights and integral indegree bounds, any basic solution of $P_f^{in}(I, F, B)$ always has an edge e with $x(e) = 1$.*

In Lemma 1, we have a set T^{in} of nodes corresponding to tight in-degree constraints. We prove that if $x \in P_f^{in}(I, F, B)$ is a basic solution so that $x(e) > 0$ for all $e \in I$, then there exists $e \in I$ with $x(e) = 1$ or there exists $v \in T^{in}$ with $|\delta_I^{in}(v)| \leq 3$. Otherwise, we must have:

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

Assuming Theorem 5 is not true, we show that given $S \in \mathcal{L}$, we can assign the S-tokens so that (here token is an S-token if it is not a tail-token of an edge leaving S):

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

The contradiction $|I| > |\mathcal{L}| + |T^{in}|$ 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 $v \in T$, we assign the 4 tail-tokens of some edges in $\delta_I^{in}(v)$.

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 $|\delta_I^{in}(S)| \geq 2$ head-tokens, as claimed. If S has in $\mathcal{L} + T^{in}$ 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 J be as in Lemma 3, so $|J| \geq 2$. One can easily verify that S can collect 1 S-token from an endnode of every edge in J. Thus the $(2,2)$-Scheme is feasible.

For the case of unit weights (and integral degree bounds), we can prove that any basic solution to $P_f^{in}(I, F, B)$ has an edge e with $x(e) = 1$. 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 $\delta_I^{in}(v) \neq \delta_I^{in}(S)$, as otherwise we obtain a contradiction to the linear independence in Lemma 1. Thus assuming that there are at least 2 edges in I entering any member of $\mathcal{L} + T^{in}$, we obtain a contradiction in the same way as before, by showing that the $(2,2)$-Scheme is feasible. Initially, every minimal member of $\mathcal{L} + T^{in}$ gets 2 tail-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 3.

Remark: Note that we also showed the well known fact (c.f., [16]), that if there are no degree constraints at all, then there is an edge $e \in I$ with $x(e) = 1$.

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 1, we have sets T and T^{in} of nodes corresponding to tight out-degree and in-degree constraints, respectively. Let $S \in \mathcal{L}$ and suppose that S has in $\mathcal{L} + T + T^{in}$ a unique child $v \in T^{in}$ (possibly $S = \{v\}$).

Arbitrary weights: For arbitrary weights, we can show that an appropriate polytope 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 $|\delta_I^{in}| \geq 5$, and we do not put any tokens on the edges leaving v (unless their tail is in T). Even if $\delta_I^{in}(S) = \delta_I^{in}(v)$

(note that 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 ratio is $(\alpha, \alpha + \Delta, \alpha + \Delta^{in}) = (2, 7, 6)$. In a similar way we can also show the sparseness $(\alpha, \Delta, \Delta^{in}) = (3, 3, 4)$, and in this case the ratio is $(3, 6, 7)$.

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

$0, 1$*-valued* f*:* In the case of $0, 1$-valued f, we can show that the corresponding polytope 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 $|\delta_I^{in}| \geq 5$; 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 ratio is $(\alpha, \alpha + \Delta, \alpha + \Delta^{in}) = (2, 5, 6)$. If we have also unit weights, then $\delta_I^{in}(S) \neq \delta_I^{in}(v)$, as otherwise we obtain a contradiction to the linear independence; hence for unit weights we can obtain sparseness $(\alpha, \Delta, \Delta^{in}) = (2, 3, 3)$, and the ratio $(\alpha \cdot \mathsf{opt}, \alpha b(v) + \Delta - 1, \alpha b^{in}(v) + \Delta^{in} - 1) = (2 \cdot \mathsf{opt}, 2b(v) + 2, 2b^{in}(v) + 2)$.

Summarizing, we obtain the following result:

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

References

1. J. Bang-Jensen, S. Thomassé, and A. Yeo. Small degree out-branchings. *J. of Graph Theory*, 42(4):287–307, 203.
2. N. Bansal, R. Khandekar, and V. Nagarajan. Additive gurantees for degree bounded directed network design. To appear in STOC 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. 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. Elsvier, 1995.
5. T. Fukunaga and H. Nagamochi. Network design with weighted degree constraints. Manuscript, 2008.
6. M. Furer and B. Raghavachari. Approximating the minimum-degree steiner tree to within one of optimal. *Journal of Algorithms*, 17(3):409–423, 1994.
7. M. X. Goemans. Minimum bounded degree spanning trees. In *FOCS*, pages 373–282, 2006.

8. 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.
9. K. Jain. A factor 2 approximation algorithm for the generalized Steiner network problem. *Combinatorica*, 21(1):39–60, 2001.
10. 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.
11. 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.
12. 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.
13. 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.
14. V. Melkonian and E. Tardos. Algorithms for a network design problem with crossing supermodular demands. *Networks*, 43(4):256–265, 2004.
15. R. Ravi, M. V. Marathe, S. S. Ravi, D. J. Rosenkrantz, and H. B. H. III. Many birds with one stone: Multi objective approximation algorithms. In *STOC*, pages 438–447, 1993.
16. A. Schrijver. *Combinatorial Optimization, Polyhedra and Efficiency*. Springer-Verlag Berlin, Heidelberg New York, 2004.
17. M. Singh and L. C. Lau. Approximating minimum bounded degree spanning trees to within one of optimal. In *STOC*, pages 661–670, 2007.