数理解析研究所講究録 1205 巻 2001 年 7-12

A Primal-Dual Approximation Algorithm for the Survivable Network Design Problem in Hypergraphs

Liang Zhao †, Hiroshi Nagamochi ‡, and Toshihide Ibaraki †

† Dep. of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University
Kyoto 606-8501, Japan
{zhao,ibaraki}@amp.i.kyoto-u.ac.jp

‡ Dep. of Information and Computer Sciences, Toyohashi University of Technology
Toyohashi 441-8580, Japan
naga@ics.tut.ac.jp

趙 亮 †, 永持 仁 ‡, 茨木 俊秀 †

† 京都大学情報学研究科数理工学専攻
〒 606-8501 京都府京都市左京区吉田本町

‡ 豊橋技術科学大学情報工学系
〒 441-8580 愛知県豊橋市天伯町雲雀ヶ丘 1-1

Abstract: Given a hypergraph with nonnegative hyperedge cost and a function $r: 2^V \to \mathbf{Z}^+$, where V is the vertex set, we consider the problem of finding a minimum cost hyperedge subset E^* such that for all $S \subseteq V$, E^* contains at least $r(S)$ hyperedges incident to S. If r is weakly supermodular and the so-called *minimum violated sets* can be found in polynomial time, we present a primal-dual approximation algorithm with performance guarantee $d_{\max}\mathcal{H}(r_{\max})$, where $d_{\max}$ is the maximum degree of hyperedges with positive cost, $r_{\max} = \max\{r(S) | S \subseteq V\}$ and $\mathcal{H}(i) = \sum_{j=1}^{i} \frac{1}{j}$. In particular, it can be applied to the survivable network design problem in which the requirement is that there should be at least r_{st} hyperedge-disjoint paths between each pair of distinct vertices s and t, for which r_{st} is prescribed.

Keywords: approximation algorithm, primal-dual method, survivable network design problem, connectivity, graph, hypergraph

1 Introduction

Given an undirected graph with nonnegative edge cost, the network design problem is to find a minimum cost subgraph satisfying certain connectivity requirements. In the *survivable network design problem* (SNDP), the connectivity requirement is that there should be at least r_{st} *edge-disjoint* paths between each pair of distinct vertices s and t, for which r_{st} is prescribed. It is known that the SNDP is NP-hard even for unit cost and $r_{st} \in \{0,1\}$ [9]. Thus we focus on developing approximation algorithms. A ρ-approximation algorithm is a polynomial time algorithm which always outputs a feasible solution of cost at most ρ times the optimum.

The first approximation algorithm for the SNDP is given by Williamson *et al.* [9]. They formalize a basic mechanism for using the primal-dual method. It picks edge sets in $r_{\max} = \max\{r_{st}\}$ phases, and each phase tries to augment the size of cuts with deficiency by using an integer program, which is solved within factor 2 by a primal-dual approach. Their algorithm has a performance guarantee of $2r_{\max}$. In [2] Goemans *et al.* show that by augmenting the size of only those cuts with maximum deficiency, a $2\mathcal{H}(r_{\max})$-approximation algorithm can be obtained, where $\mathcal{H}(i) = \sum_{j=1}^{i} \frac{1}{j}$ is the harmonic function. For a detailed overview of these primal-dual algorithms, we refer the readers to the well-written surveys [4]. Recently, Jain [5] shows that there is an edge e with $x_e^* \geq \frac{1}{2}$ in any *basic* solution x^* to the LP relaxation of SNDP (where the constraint $x_e \in \{0,1\}$ is relaxed to $0 \leq x_e \leq 1$ for all edge e). Then it is shown that an iterative rounding process yields a 2-approximation algorithm.

In a recent paper [6], Jain *et al.* considered the *element connectivity problem* (ECP) in graphs. In that problem, the vertex set consists of two types of vertices: terminals and non-terminals. Edges and non-terminals are called the *elements*. And only each pair of terminals has connectivity requirement, the least number of *element-disjoint* paths to be realized. The objective is to find a minimum cost spanning subgraph satisfying the requirements. (Notice that only the edges have costs.) The SNDP is a special case of ECP with empty non-terminal set. Following the basic algorithmic schema established in [2, 9], they proposed a primal-dual approximation algorithm for the ECP. And by having verified that their algorithm satisfies three conditions stated in [7], they claim that it is a $2\mathcal{H}(r_{\max})$-

approximation algorithm. However, we note that even if the three conditions are satisfied it is still unclear whether the desired guarantee can be obtained or not. (As will be shown later, our result implies that the ECP is $2\mathcal{H}(r_{\max})$-approximable.)

In this paper we consider the SNDP in hypergraphs (SNDPHG). The difference between hypergraph and graph is that edges in hypergraph, called the *hyperedges*, may contain more than two vertices as their endpoints. The *degree* of a hyperedge e is the number of endpoints of e. The definition of SNDPHG is obtained by replacing *edges* with *hyperedges* in the definition of SNDP. Thus the SNDP is a special case of the SNDPHG in which the degrees of all the hyperedges are 2. We note that the ECP is also a special case of SNDPHG. To see this, consider a non-terminal w. Let $\{v_1, w\}, \ldots, \{v_k, w\}$ be the edges that are incident to w. For each $i = 1, \ldots, k$, replace edge $\{v_i, w\}$ with two edges $\{v_i, w_i\}$ and $\{w_i, w\}$, introducing a new terminal w_i. Let the cost of edge $\{v_i, w_i\}$ be the same as $\{v_i, w\}$. Let $r_{st} = 0$ if at least one of s and t is a new terminal. Then replace w and all the edges $\{\{w_i, w\} | i = 1, \ldots, k\}$ with hyperedge $e_w = \{w_1, \ldots, w_k\}$ of zero cost. Repeat this process until there is no non-terminal left. Clearly in this way we can reduce ECP to SNDPHG in linear time. In fact, let $d_{\max}$ denote the maximum degree of hyperedges with *positive* cost, we have shown that the ECP is a special case of SNDPHG with $d_{\max} = 2$.

In [8] Takeshita *et al.* extend the primal-dual approximation algorithm of [3] to the SNDPHG with $r_{st} \in \{0, 1\}$. They show a k-approximation algorithm, where k is the maximum degree of hyperedges. In this paper we design an approximation algorithm to the SNDPHG based on the primal-dual schema established in [2, 9]. As a result, we show that a performance guarantee of $d_{\max}\mathcal{H}(r_{\max})$ can be obtained. Thus our result includes or improves the former results of [2, 6, 9] (with $d_{\max} = 2$) and [8] (with $r_{\max} = 1$). (We note that the guarantee cannot be derived in a straightforward manner by simply combining the results of [2, 9] and [8].) Like the previous algorithms in [2, 8, 9], our algorithm is also applicable to more general problems, provided that they satisfy certain conditions which are described in the next two sections.

This paper is organized as follows. Section 2 contains some definitions and the formulation of the problem. Section 3 presents an algorithm for problems formulated in Sect. 2 that satisfy Conditions 1 and 2. Section 4 gives a proof of the performance guarantee. Section 5 shows that the SNDPHG satisfies the two conditions. Some proofs are omitted due to the page limit.

2 Definitions and Formulation

All hypergraphs treated in this paper are undirected unless stated otherwise. Let G be a hypergraph, and let $V(G)$ and $E(G)$ denote the vertex set and hyperedge set of G, respectively. A *hyperedge* e with endpoints $v_1, \ldots, v_k$ is denoted by $e = \{v_1, \ldots, v_k\}$ and it may be treated as the set $\{v_1, \ldots, v_k\}$ of vertices. The subgraph of G induced by a set $S \subseteq V(G)$ is denoted by $G[S]$ (i.e., $G[S] = (S, E(G) \cap 2^S)$). The neighbors of S in G is denoted by $\Gamma(S)$, i.e., $\Gamma(S) \triangleq \{v \in V(G) - S | \exists e \in E(G),\ v \in e,\ e \cap S \neq \emptyset\}$. The set of hyperedges incident to S is denoted by $\delta(S)$, i.e., $\delta(S) \triangleq \{e \in E(G) \mid \emptyset \neq e \cap S \neq e\}$. Let $\delta_A(S) \triangleq \delta(S) \cap A$ for a set $A \subseteq E(G)$. It is well known that for any subset $A \subseteq E(G)$, $|\delta_A| : 2^{V(G)} \rightarrow \mathbf{Z}^+$ is a symmetric and submodular function.

We first treat the SNDPHG with a function. Given a hypergraph H with nonnegative hyperedge cost and function $r : 2^{V(H)} \rightarrow \mathbf{Z}^+$, find a minimum cost $E^* \subseteq E(H)$ such that $|\delta_{E^*}(S)| \geq r(S)$ for all $S \subseteq V(H)$. We consider the problem by converting it into the next equivalent problem.

Definition 1 (Problem $\mathcal{P}$) *Let $G = (T, W, E)$ be a bipartite graph with disjoint vertex sets T and W and an edge set E, where vertices in T and W are called* terminals *and* non-terminals, *respectively. Let $c : W \rightarrow \mathbf{R}^+$ be a nonnegative cost function and $r : 2^T \rightarrow \mathbf{Z}^+$ be a nonnegative requirement function. Find a minimum cost subset $W^* \subseteq W$ such that for all $S \subseteq T$, $|\Gamma(S) \cap \Gamma(T - S) \cap W^*| \geq r(S)$.*

The equivalence can be easily seen as follows. Let $T = V(H)$. Replace each hyperedge $e = \{v_1, \ldots, v_k\}$ with a new non-terminal vertex w_e and k edges $\{v_1, w_e\}, \ldots, \{v_k, w_e\}$. Assign w_e the same cost as the hyperedge e. Notice that $e \in \delta(S)$ in H if and only if $w_e \in \Gamma(S) \cap \Gamma(T - S)$ in G.

Let $\Delta(S) = \Gamma(S) \cap \Gamma(T - S)$ for $S \subseteq T$ in G (in what follows, notations Γ and Δ are defined with respect to the input bipartite graph G). Let $\Delta_A(S) \triangleq \Delta(S) \cap A$ for a set $A \subseteq W$. Notice that for any $X \subseteq T = V(H)$, $|\Delta(X)|$ defined in G equals to $|\delta(X)|$ in H. Thus $|\Delta_A| : 2^T \rightarrow \mathbf{Z}^+$ is also a symmetric and submodular function.

In what follows, we will consider problem $\mathcal{P}$ instead of the original form defined on hypergraph H. It can be written as the next integer program.

$$\begin{array}{lll} \text{(IP)} & \min & \sum_{w \in W} c_w x_w \\ & \text{s.t.} & x(\Delta(S)) \geq r(S) \quad \forall S \subseteq T, \\ & & x_w \in \{0, 1\} \quad \forall w \in W, \end{array}$$

where we use the notation $x(A) \triangleq \sum_{w \in A} x_w$.

W.l.o.g we assume that $r(\emptyset) = r(T) = 0$ and $r_{\max} = \max\{r(S) | S \subseteq T\} \leq |W|$. We assume that r satisfies two conditions. The first condition is as follows while the second will be stated in Sect. 3.

Condition 1 *r is weakly supermodular. That is, $r(T) = 0$ and for any $X, Y \subseteq T$*

$$r(X) + r(Y) \leq \max\{r(X \cap Y) + r(X \cup Y), \quad r(X - Y) + r(Y - X)\}. \quad (1)$$

3 A Primal-Dual Approximation Algorithm for (IP)

In this section we describe our algorithm for (IP) according to the primal-dual schema established in [2, 9]. We then show that it runs in polynomial time. The proof of the performance guarantee will be given in the next section.

For an $S \subseteq T$ and $A \subseteq W$, the *deficiency* of S with respect to A is defined as $r(S) - |\Delta_A(S)|$. Analogously with [2, 9], our algorithm consists of $r_{\max}$ phases. Let $W_0 = \emptyset$, and let $W_i \subseteq W$ denote the set of non-terminals picked so far by phase i. At the beginning of phase i, the maximum deficiency (with respect to W_{i-1}) is $r_{\max} - i + 1$. We decrease it by 1 in phase i, by adding a set $A_i \subseteq W - W_{i-1}$ to the current temporary solution W_{i-1}. We then set $W_i = W_{i-1} \cup A_i$ and proceed to phase $i + 1$ until $i = r_{\max}$. Finally we output $W_{r_{\max}}$, which is feasible to (IP) since our algorithm ensures that the maximum deficiency with respect to $W_{r_{\max}}$ is zero. In each phase i, we consider the next integer program to find such A_i with the minimum cost.

$$\begin{array}{lll} (\mathrm{IP})_i & \min \displaystyle\sum_{w \in W - W_{i-1}} c_w x_w & \\ \text{s.t.} & x(\Delta_{W - W_{i-1}}(S)) \geq h_i(S) & \forall S \subseteq T, \\ & x_w \in \{0, 1\} & \forall w \in W - W_{i-1}, \end{array}$$

where $h_i(\cdot)$ is defined as

$$h_i(S) = \begin{cases} 1 & \text{if } r(S) - |\Delta_{W_{i-1}}(S)| = r_{\max} - i + 1, \\ 0 & \text{otherwise} \end{cases}$$

In the following, we solve $(\mathrm{IP})_i$ approximately by a primal-dual approach based on that of [9]. We need a notation of *violated sets.*

Definition 2 (violated set) *Let $A \subseteq W - W_{i-1}$ be a non-terminal subset. A terminal subset $S \subseteq T$ is said to be* violated *with respect to A if $h_i(S) = 1$ and $\Delta_A(S) = \emptyset$. It is a* minimal violated set *if it is a violated set and minimal under set inclusion.*

Let $\mathcal{V}(A)$ denote the family of minimal violated sets w.r.t A. Clearly, A is feasible to $(\mathrm{IP})_i$ if and only if $\mathcal{V}(A) = \emptyset$. Under the assumption of Condition 1, the violated sets enjoy the following property.

Lemma 1 *Let $X, Y \subseteq T$ be two violated sets with respect to some A. Then $X \cap Y, X \cup Y$ or $X - Y, Y - X$ are violated sets with respect to A.*

Proof. Analogously with [9]. Notice that $|\Delta_A|$ is a symmetric and submodular function. □

Two sets X and Y are said to *intersect* if $X \cap Y \neq \emptyset$, $X - Y \neq \emptyset$ and $Y - X \neq \emptyset$. An immediate conclusion is the next corollary, where for simplicity, the words of "with respect to A" are omitted.

Corollary 1 *Let X be a minimal violated set. Then any violated set Y does not intersect X, i.e., either $X \subseteq Y$ or $X \cap Y = \emptyset$. Moreover, if Y is also a minimal violated set then $X \cap Y = \emptyset$.* □

Another condition that r satisfies is the next.

Condition 2 *For any $A \subseteq W - W_{i-1}$, the family $\mathcal{V}(A)$ of minimum violated sets can be computed in polynomial time.*

We now consider an algorithm to $(\mathrm{IP})_i$ according to the primal-dual schema established in [9]. Relax each constraint $x_w \in \{0, 1\}$ to $x_w \geq 0$ in $(\mathrm{IP})_i$. The dual of this relaxation is given by

$$\begin{array}{lll} (\mathrm{D})_i & \max \displaystyle\sum_{S \subseteq T} h_i(S) y_S & \\ \text{s.t.} & \displaystyle\sum_{S \subseteq T : w \in \Delta(S)} y_S \leq c_w & \forall w \in W - W_{i-1}, \\ & y \geq 0. & \end{array}$$

In the algorithm (Table 1), we use $\bar{c}$, A, y and j to denote the reduced cost, primal solution, dual variable and number of iterations, respectively.

Let us consider the running time. We store only those y_S of positive value. Thus step 1 takes $O(|W|)$ time. Since $|A|$ increases by 1 after one WHILE iteration, there are at most $|W - W_{i-1}| \leq |W|$ WHILE iterations. Let θ denote the time complexity to compute $\mathcal{V}(A)$. Then steps 2, 4 and 11 can be done in θ time since A is feasible if and only if $\mathcal{V}(A) = \emptyset$. It is not difficult to see that step 6 can be done in $O(|T||W|)$ time since $|\mathcal{V}(A)| \leq |T|$ by Corollary 1, and this dominates other steps. Hence the algorithm for $(\mathrm{IP})_i$ takes $O(|W|(\theta + |T||W|))$ time to compute A_i. Therefore the entire complexity to construct the solution $W_{r_{\max}} = \bigcup_{i=1}^{r_{\max}} A_i$ to (IP) is in $O(r_{\max}|W|(\theta + |T||W|))$ time. This is polynomial since $r_{\max} \leq |W|$ by assumption.

Table 1: Approximation Algorithm for $(\mathrm{IP})_i$

1	$\bar{c} \leftarrow c, \quad A \leftarrow \emptyset, \quad y \leftarrow 0, \quad j \leftarrow 0$
2	WHILE A is not feasible
3	$\quad j \leftarrow j+1$
4	$\quad \mathcal{V}_j \leftarrow \mathcal{V}(A)$
5	$\quad$ IF $\exists S \in \mathcal{V}_j$, $\Delta_{W-W_{i-1}-A}(S) = \emptyset$ THEN HALT ((IP) has no feasible solution)
6	$\quad w_j \leftarrow \mathrm{argmin}\{\frac{\bar{c}_w}{\lvert\{S \in \mathcal{V}_j \mid w \in \Delta(S)\}\rvert} \mid w \in W - W_{i-1} - A\}$
7	$\quad \epsilon_j \leftarrow \frac{\bar{c}_{w_j}}{\lvert\{S \in \mathcal{V}_j \mid w_j \in \Delta(S)\}\rvert}$, $y_S \leftarrow y_S + \epsilon_j$ for all $S \in \mathcal{V}_j$
8	$\quad \bar{c}_w \leftarrow \bar{c}_w - \lvert\{S \in \mathcal{V}_j \mid w \in \Delta(S)\}\rvert \epsilon_j$, for all $w \in W - W_{i-1} - A$
9	$\quad A \leftarrow A \cup \{w_j\}$
10	FOR $l = j$ DOWN TO 1
11	$\quad$ IF $A - \{w_l\}$ is feasible THEN $A \leftarrow A - \{w_l\}$
12	Output A (as A_i).

4 Proof of Performance Guarantee

Lemma 2 *Let A_i and y be the output and the corresponding dual variable obtained at the end of the primal-dual algorithm for $(IP)_i$, respectively. Then*

$$\sum_{w \in A_i} c_w \le d_{\max} \sum_{S \subseteq T} h_i(S) y_S. \ \square$$

Before proving Lemma 2, we show that it implies the claimed performance guarantee $d_{\max}\mathcal{H}(r_{\max})$.

Theorem 1 *Let opt_{IP} be the optimal value of (IP). Let $W_{r_{\max}} = \bigcup_{i=1}^{r_{\max}} A_i$ be the output of the $r_{\max}$-phases algorithm for (IP). Then*

$$\sum_{w \in W_{r_{\max}}} c_w \le d_{\max}\mathcal{H}(r_{\max}) opt_{\mathrm{IP}}. \qquad (2)$$

Proof. Using Lemma 2. Other part is analogous to the proof in [2]. $\square$

Proof of Lemma 2. (The proof ends at the end of this section.) First suppose that $c_w > 0$ for all $w \in W$. Then $d_{\max}$ is the maximum degree of non-terminals. Let L be the number of WHILE iterations. Notice that $c_{w_l} = \sum_{j=1}^{L} \lvert\{S \in \mathcal{V}_j \mid w_l \in \Delta(S)\}\rvert \epsilon_j$ for all $l = 1, 2, \ldots, L$. Thus

$$\begin{aligned} \sum_{w \in A_i} c_w &= \sum_{w \in A_i} \sum_{1 \le j \le L} \lvert\{S \in \mathcal{V}_j \mid w \in \Delta(S)\}\rvert \epsilon_j \\ &= \sum_{1 \le j \le L} \sum_{S \in \mathcal{V}_j} \lvert\Delta_{A_i}(S)\rvert \epsilon_j. \end{aligned}$$

On the other hand, since $y_S = \sum_{j: S \in \mathcal{V}_j} \epsilon_j$,

$$\begin{aligned} \sum_{S \subseteq T} h_i(S) y_S &= \sum_{S \subseteq T} \sum_{j: S \in \mathcal{V}_j} \epsilon_j \\ &= \sum_{1 \le j \le L} \sum_{S \in \mathcal{V}_j} \epsilon_j = \sum_{1 \le j \le L} \lvert\mathcal{V}_j\rvert \epsilon_j. \end{aligned}$$

Thus to show Lemma 2 it suffices to show that

$$\sum_{S \in \mathcal{V}_j} \lvert\Delta_{A_i}(S)\rvert \le d_{\max} \lvert\mathcal{V}_j\rvert \quad \text{for all } j = 1, \ldots, L. \qquad (3)$$

For a set $A \subseteq W - W_{i-1}$ which is infeasible to $(\mathrm{IP})_i$, $B \subseteq W - W_{i-1}$ is called a *minimal augmentation* of A if $A \subseteq B$ and B is feasible to $(\mathrm{IP})_i$ but the removal of any $w \in B - A$ violates the feasibility. We here claim that for any infeasible A and any minimal augmentation B of A, it holds

$$\sum_{S \in \mathcal{V}(A)} \lvert\Delta_B(S)\rvert \le d_{\max} \lvert\mathcal{V}(A)\rvert. \qquad (4)$$

Then (3) holds by (4) by letting $A = \{w_1, \ldots, w_{j-1}\}$ and $B = A \cup A_i$ for all $j = 1, \ldots, L$. Thus we only need to show (4). For this, we introduce a notation of *witness set*. Let $U \triangleq \bigcup_{S \in \mathcal{V}(A)} \Delta_B(S) \subseteq B - A$.

Definition 3 (witness set) *$C \subseteq T$ is a witness set of $w \in U$ if it satisfies* (i) $h_i(C) = 1$, *and* (ii) $\Delta_B(C) = \{w\}$.

By (i) and (ii), we see that C is a violated set (notice that $w \notin A$). For any $w \in U$, there must exist a witness set of w since the removal of w violates the feasibility of B. Call $\{C_w \mid w \in U\}$ a *witness set family*, in which for each $w \in U$, exact one witness set of w, C_w, is included.

Lemma 3 *There exists a laminar (i.e., intersect-free) witness set family.*

Proof. Using Lemma 1, it is analogous with [9]. $\square$

Let $\mathcal{F} = \{T\} \cup \{C_w \mid w \in U\}$ be the family obtained by adding T to a laminar witness set family. Construct a rooted tree $\mathcal{T}$ from $\mathcal{F}$ by set inclusion relationship as follows. (To avoid confusion, we will use the word "node" for the nodes of tree $\mathcal{T}$, and use the word "vertex" for the vertices of graph G.) The node set of $\mathcal{T}$ consists of $\lvert\mathcal{F}\rvert$ nodes: u_C for $C \in \mathcal{F}$. The root is u_T, and for each non-root node u_C in $\mathcal{T}$, the parent of u_C is the node $u_{C'}$ for the minimum $C' \in \mathcal{F}$ such that $C \subset C'$ (i.e., $C \subseteq C'$ and $C \neq C'$). For each $S \in \mathcal{V}(A)$, let $u(S) \triangleq u_C$ for the minimum $C \in \mathcal{F}$ such that $S \subseteq C$. An $S \in \mathcal{V}(A)$ is said to be *associated* with $u(S)$. For each $C \in \mathcal{F}$, let $n_C = \lvert\{S \in \mathcal{V}(A) \mid u(S) = u_C\}\rvert$ be the number of minimal violated sets that are associated with u_C.

Let $Q = \{u_C \in V(\mathcal{T}) | n_C \geq 1\}$ be the subset of nodes with which at least one minimal violated set is associated. It is clear that

$$|\mathcal{V}(A)| = \sum_{u_C \in V(\mathcal{T})} n_C = \sum_{u_C \in Q} n_C. \quad (5)$$

Let $d(u_C)$ denote the degree of a node u_C in $\mathcal{T}$. For a non-root node u_C, C is a witness set (thus a violated set). Hence it must include some minimal violated set, implying that if u_C is a leaf then $C = u(S)$ for some $S \in \mathcal{V}(A)$. Hence all non-root nodes of degree 1 belong to Q. This observation shows that $\sum_{u_C \in V(\mathcal{T})-Q} d(u_C) \geq 2(|V(\mathcal{T})| - |Q|) - 1$. Since $\mathcal{T}$ is a tree, $\sum_{u_C \in V(\mathcal{T})} d(u_C) = 2(|V(\mathcal{T})| - 1)$. Thus

$$\sum_{u_C \in Q} d(u_C) \leq 2|Q| - 1. \quad (6)$$

We next show that

$$\sum_{u_C \in Q} \min\{d_{\max} - 1, n_C\} d(u_C) \leq d_{\max} \sum_{u_C \in Q} n_C. \quad (7)$$

Let $X = \{u_C \in Q | n_C \geq d_{\max} - 1\}$, $Y = \{u_C \in Q | n_C = 1\} - X$ and $Z = Q - X - Y$. It is easy to see (7) by comparing the two sides.

Finally we show that for each $u_C \in Q$,

$$\sum_{S \in \mathcal{V}(A): u(S) = u_C} |\Delta_B(S)| \leq \min\{d_{\max} - 1, n_C\} d(u_C). \quad (8)$$

Notice that (5), (7) and (8) imply (4). Therefore we only need to show (8) to complete the proof of Lemma 2. Consider an $S \in \mathcal{V}(A)$ and a $w \in \Delta_B(S)$. Let C_w be the witness set of w in the family $\mathcal{F}$. By the definition of witness set family, we see that for any $C' \in \mathcal{F}$, $C' \subset C_w$ implies $\Gamma(w) \cap C' = \emptyset$, while $C_w \subset C'$ implies $\Gamma(w) \subseteq C'$. By Corollary 1, either $S \subseteq C_w$ or $S \cap C_w = \emptyset$ holds.

Case 1: Suppose that $S \subseteq C_w$. Notice that there is no $C' \in \mathcal{F}$ such that $S \subseteq C' \subset C_w$. Hence $u(S) = u_{C_w}$. Let u_C be the parent of u_{C_w}. Then $\Gamma(w) \subseteq C$. We use an *directed* edge (u_{C_w}, u_C) to represent that this case occurs for an $S \in \mathcal{V}(A)$ and a $w \in \Delta_B(S)$ such that $u(S) = u_{C_w}$. The directed edge (u_{C_w}, u_C) may not be unique since there may exists some other $S' \in \mathcal{V}(A)$ such that $w \in \Delta_B(S')$ and $u(S') = u_{C_w}$. In such cases multiple directed edges (u_{C_w}, u_C) are allowed, but for each S' of such sets only one edge is assigned. Notice that such sets S' are disjoint by Corollary 1. Therefore the total number of the directed (u_{C_w}, u_C) edges is at most $\min\{|\Gamma(w)| - 1, n_{C_w}\} \leq \min\{d_{\max} - 1, n_{C_w}\}$ (notice that $\Gamma(w) \cap (T - C_w) \neq \emptyset$).

Case 2: Otherwise $S \cap C_w = \emptyset$. Similarly, we see that $u(S) = u_C$ for the parent u_C of u_{C_w}. We use a directed edge (u_C, u_{C_w}) to represent this case. The total number of these (u_C, u_{C_w}) edges is at most $\min\{d_{\max} - 1, n_C\}$.

For any fixed $u_C \in Q$, the two cases may happen simultaneously. But we see that for one (undirected) edge $\{u_C, u_{C'}\}$ in tree $\mathcal{T}$, there are at most $\min\{d_{\max} - 1, n_C\}$ directed $(u_C, u_{C'})$ edges that are produced in Case 1 or 2. Thus there are at most $\min\{d_{\max} - 1, n_C\} d(u_C)$ directed edges with tail u_C. On the other hand, the way that the directed edges are produced ensures that the total number of the directed edges with tail u_C (over all $S \in \mathcal{V}(A)$ and all $w \in \Delta_B(S)$) equals to $\sum_{S \in \mathcal{V}(A): u(S) = u_C} |\Delta_B(S)|$. Hence (8) has been shown.

Thus we have proved Lemma 2 under the assumption that $c_w > 0$ for all $w \in W$. It is easy to see that it is also true when there exists some $w \in W$ of zero cost. To see this, notice that we only need to show (3) for j with $\epsilon_j > 0$, which implies that $c_w > 0$ for all $w \in \bigcup_{S \in \mathcal{V}_j} \Delta_{A_i - A}(S)$. Thus $|\Gamma(w)| \leq d_{\max}$ for all $w \in \bigcup_{S \in \mathcal{V}_j} \Delta_{A_i - A}(S)$, and the proof goes in a straightforward way. □

5 Survivable Network Design Problem in Hypergraphs

In this section we consider the SNDPHG. By replacing the hyperedges with non-terminals as we did in Sect. 2, the SNDPHG can be converted to the next equivalent problem defined in a bipartite graph $G = (T, W, E)$. Given $c : W \to \mathbf{R}^+$, and $r_{st} \in \mathbf{Z}^+$ for each pair of distinct terminals $s, t \in T$, find a minimum cost subset $W^* \subseteq W$ such that, for each pair of terminals s and t, $G[T \cup W^*]$ has at least r_{st} paths which are *W-disjoint* (i.e., no $w \in W$ belongs to two or more paths). We first show that it is equivalent to problem $\mathcal{P}$ (Sect. 2) with $r(S) = \max\{r_{st} | s \in S, t \in T - S\}$ for all $S \subseteq T$ ($r(\emptyset) = r(T) = 0$).

A useful idea when considering W-disjoint paths in G is the following transformation $\mathcal{D}$ from G to a digraph $\overrightarrow{G}$ with edge capacities.

Definition 4 ($\mathcal{D} : G \to \overrightarrow{G}$) *Replace all undirected edge $\{v, w\}$ by two directed edges (v, w) and (w, v) of capacity $+\infty$. Then for each non-terminal w, make a copy named w^c, change the tails of all directed edges (w, v) from w to w^c, and add a new directed edge (w, w^c) of capacity* 1.

In the following, a vertex in $V(G) = T \cup W$ is also treated as a vertex in $V(\overrightarrow{G})$. However, the notations of Γ and Δ are only used with respect to G. Given $W' \subseteq W$, notice that in digraph $\overrightarrow{G[T \cup W']}$, the capacity of a cut C (i.e., $\emptyset \neq C \subset V(\overrightarrow{G[T \cup W']})$) is not $+\infty$ if and only if $C = S \cup \Gamma_{W'}(S) \cup \{w^c | w \in \Gamma_{W'}(S) - \Delta_{W'}(S)\}$ for the $S = C \cap T$, where the capacity is exactly $|\Delta_{W'}(S)|$.

Notice that for any pair of terminals s and t, any h W-disjoint s,t-paths in G are transformed to an integer s,t-flow of value h in $\overrightarrow{G}$, vice versa. Thus by the maxflow-mincut theorem it is not difficult to show that the SNDPHG is equivalent to problem $\mathcal{P}$ with $r(S) = \max\{r_{st} | s \in S,\ t \in T - S\}$.

We next show that Condition 1 and 2 are satisfied. It is easy to verify that this r satisfying Condition 1. We show that in phase i of the algorithm in Table 1, the minimum violated sets with respect to any $A \subseteq W - W_{i-1}$ can be found in polynomial time (i.e., Condition 2).

Lemma 4 *Denote $W_{i-1} \cup A$ by $\tilde{A}$. Let S be a minimal violated set, and let s and t be two terminals such that $r_{st} = r(S)$ and $s \in S$, $t \in T - S$. Then $S = C_{st} \cap T$ for the minimum s,t-cut C_{st} in digraph $\overrightarrow{G[T \cup \tilde{A}]}$ that is minimal under set inclusion.*

Proof. It is not difficult and omitted here. □

Lemma 4 shows that we can identify the minimal violated sets by computing the minimal minimum s,t-cut in $\overrightarrow{G[T \cup \tilde{A}]}$ for all pairs of distinct terminals s and t and checking if they are violated and minimal among these $O(|T|^2)$ cuts. It is well known that for each pair of s and t, the minimal minimum s,t-cut can be found by one maxflow computation in $O(p^3)$ time for a digraph with p vertices [1]. Thus the total running time of finding the minimal violated sets is dominated by $O(|T|^2)$ maxflow computations. Thus our algorithm for the SNDPHG can be implemented to run in $O(r_{\max}|W||T|^2(|T| + |W|)^3)$ time. We summary this as the next theorem.

Theorem 2 *Let $d_{\max}$ be the maximum degree of hyperedges with positive cost and $r_{\max}$ be the maximum requirement. The SNDPHG can be approximated within factor $d_{\max}\mathcal{H}(r_{\max})$ in $O(r_{\max}mn^2(n+m)^3)$ time, where m and n are the numbers of hyperedges and vertices respectively.* □

6 Remark

We remark that the performance guarantee $d_{\max}$ in Lemma 2 of the primal-dual algorithm for $(\text{IP})_i$ is tight. A tight example will be given in the full paper. Notice that in [2] Goemans *et al.* have shown that for the SNDP in graphs the performance guarantee $2\mathcal{H}(r_{\max})$ is tight up to factor 2. It is thus interesting to know whether an algorithm with improved performance guarantee can be developed, e.g., via an iterative rounding process for the SNDP in graphs as used by Jain in [5].

References

[1] A. V. Goldberg and R. E. Tarjan. A New Approach to the Maximum Flow Problem. In Proc. STOC'86 (1986) 136–146. Full paper in J. ACM **35** (1988) 921–940.

[2] M. X. Goemans, A. V. Goldberg, S. Plotkin, D. Shmoys, E. Tardos, and D. P. Williamson. Improved Approximation Algorithms for Network Design Problems. In Proc. SODA'94 (1994) 223–232.

[3] M. X. Goemans and D. P. Williamson. A General Approximation Technique for Constrained Forest Problems. In Proc. SODA'92 (1992) 307–315.

[4] M. X. Goemans and D. P. Williamson. The Primal-Dual Method for Approximation Algorithms and its Application to Network Design Problems, in Approximation Algorithms for NP-hard Problems, (D. Hochbaum, eds.), PWS, (1997) 144–191.

[5] K. Jain. A Factor 2 Approximation Algorithm for the Generalized Steiner Network Problem. In Proc. FOCS'98 (1998) 448–457.

[6] K. Jain, I. Măndoiu, V. V. Vazirani, and D. P. Williamson. A Primal-dual Schema Based Approximation Algorithm for the Element Connectivity Problem. In Proc. SODA'99 (1999) 484–489.

[7] R. Ravi and D. P. Williamson. An Approximation Algorithm for Minimum-Cost Vertex-Connectivity Problems. In Proc. SODA'95 (1995) 332–341. Full paper in Algorithmica **18** (1997) 21–43.

[8] K. Takeshita, T. Fujito, and T. Watanabe. On Primal-Dual Approximation Algorithms for Several Hypergraph Problems (in Japanese). IPSJ Mathematical Modeling and Problem Solving **23-3** (1999) 13–18.

[9] D. P. Williamson, M. X. Goemans, M. Mihail, and V. V. Vazirani. A Primal-dual Approximation Algorithm for Generalized Steiner Network Problems. In Proc. STOC'93 (1993) 708–717. Full paper in Combinatorica **15** (1995) 435–454.