<table>
<thead>
<tr>
<th>Title</th>
<th>A Primal-Dual Approximation Algorithm for the Survivable Network Design Problem in Hypergraphs (New Developments of Theory of Computation and Algorithms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s)</td>
<td>Zhao, Liang; Nagamochi, Hiroshi; Ibaraki, Toshihide</td>
</tr>
<tr>
<td>Citation</td>
<td>数理解析研究所講究録 (2001), 1205: 7-12</td>
</tr>
<tr>
<td>Issue Date</td>
<td>2001-05</td>
</tr>
<tr>
<td>URL</td>
<td><a href="http://hdl.handle.net/2433/40992">http://hdl.handle.net/2433/40992</a></td>
</tr>
<tr>
<td>Type</td>
<td>Departmental Bulletin Paper</td>
</tr>
<tr>
<td>Textversion</td>
<td>publisher</td>
</tr>
</tbody>
</table>
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

Abstract: Given a hypergraph with nonnegative hyperedge cost and a function $r : 2^V \rightarrow \mathbb{Z}^+$, 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 (SNPD), 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 SNPD is NP-hard even for unit cost and $r_{st} \in \{0, 1\}$ [9]. Thus we focus on developing approximation algorithms. A $\rho$-approximation algorithm is a polynomial time algorithm which always outputs a feasible solution of cost at most $\rho$ times the optimum.

The first approximation algorithm for the SNPD 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 SNPD 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 $2\mathcal{H}(r_{\text{max}})$-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 $\{v_i, w_1\}$ and $\{v_i, w\}$, introducing a new terminal $w_i$. Let the cost of edge $\{v_i, w\}$ 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 $e_w = \{v_1, \ldots, v_k\}$ 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 $d_{\text{max}}$ denote the maximum degree of hyperedges with positive cost, we have shown that the ECP is a special case of SNDPHG with $d_{\text{max}} = 2$.

In [8] Takehita 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 $d_{\text{max}}\mathcal{H}(r_{\text{max}})$ can be obtained. Thus our result includes or improves the former results of [2, 6, 9] (with $d_{\text{max}} = 2$) and [8] (with $r_{\text{max}} = 1$). (We note that the guarantee cannot be derived in a straightforward manner by simply combining the results of [2, 9] and [8].)

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., $\Gamma(S) = \{v \in V(G) - S | \exists e \in E(G), v \in e, e \cap S \neq \emptyset\}$. The set of hyperedges incident to $S$ is denoted by $\delta(S)$, i.e., $\delta(S) = \{e \in E(G) \mid \emptyset \neq e \cap S \neq e\}$. Let $\delta_A(S) = \delta(S) \cap A$ for a set $A \subseteq E(G)$. It is well known that for any subset $A \subseteq E(G)$, $|\delta_A| : 2^{V(G)} \to \mathbb{Z}^+$ is a symmetric and submodular function.

We first treat the SNDPHG with a function. Given a hypergraph $H$ with nonnegative hyperedge cost and function $r : 2^{V(H)} \to \mathbb{Z}^+$, 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 $P$) 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 $c : W \to \mathbb{R}^+$ be a nonnegative cost function and $r : 2^T \to \mathbb{Z}^+$ 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 $\Gamma$ and $\Delta$ are defined with respect to the input bipartite graph $G$). Let $|\Delta_A(S)| = \Delta(S) \cap A$ 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 $|\Delta_A| : 2^T \to \mathbb{Z}^+$ is also a symmetric and submodular function.

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

(IP) \[ \min \sum_{w \in W} c_w x_w \]
\[ \text{s.t. } \pi(\Delta(S)) \geq r(S) \quad \forall S \subseteq T, \]
\[ x_w \in \{0, 1\} \quad \forall w \in W, \]
where we use the notation $\pi(A) \defeq \sum_{w \in A} x_w$. 

Let \( \mathcal{V}(A) \) denote the family of minimal violated sets w.r.t. \( A \). Clearly, \( A \) is feasible to (IP), 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 "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 \( M \subseteq W - W_{i-1} \), the family \( \mathcal{V}(A) \) of minimal violated sets can be computed in polynomial time.

We now consider an algorithm to (IP), according to the primal-dual schema established in [2, 9]. Relax each constraint \( x_w \in \{0, 1\} \) to \( x_w \geq 0 \) in (IP). The dual of this relaxation is given by

\[
(D)_i: \quad \max \sum_{S \subseteq T} h_i(S)y_S \\
\text{s.t.} \quad \sum_{S \subseteq T: w \in \Delta(S)} y_S \leq c_w \quad \forall w \in W - W_{i-1}, \\
y \geq 0.
\]

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 \( \theta \) denote the time complexity to compute \( \mathcal{V}(A) \). Then steps 2, 4 and 11 can be done in \( \theta \) 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 (IP), takes \( O(|W|\theta + |T||W|) \) time to compute \( \tilde{A}_i \). Therefore the entire complexity to construct the solution \( W_{\max} = \bigcup_{i=1}^{r_{\max}} A_i \) 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 (IP)\
1 \( \zeta \leftarrow c, \quad A \leftarrow \emptyset, \quad y \leftarrow 0, \quad j \leftarrow 0 \)
2 WHILE A is not feasible
3 \( j \leftarrow j + 1 \)
4 \( V_j \leftarrow V(A) \)
5 IF \( \exists S \subseteq V_j, \Delta_{W-W_{j-1}-A}(S) = \emptyset \) THEN
6 \( \text{HALT} \) (IP) has no feasible solution
7 \( w_j \leftarrow \arg\min\{w \in W : w \in \Delta_{W-W_{j-1}-A}(S)\} \)
8 \( \varepsilon_j \leftarrow \{w \in \Delta_{W-W_{j-1}-A}(S) : w \in V_j\} \)
9 \( \varepsilon_j \leftarrow \varepsilon_j + \varepsilon_j \) for all \( S \subseteq V_j \)
10 FOR \( i = j \) DOWN TO 1
11 IF \( A - \{w_i\} \) is feasible THEN
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
\[
\sum_{w \in A_i} c_w \leq d_{\max} \sum_{S \subseteq T} h_i(S)y_S. \quad \Box
\]

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

Theorem 1 Let \( \text{opt}_{IP} \) 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
\[
\sum_{w \in W_{r_{\max}}} c_w \leq d_{\max} H(r_{\max}) \text{opt}_{IP}. \quad (2)
\]

Proof. Using Lemma 2. Other part is analogous to the proof in [2]. \( \Box \)

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 \( c_{w_l} = \sum_{l=1}^{r_{\max}} \{w \in W | w \not\in \Delta(S)\} \varepsilon_j \) for all \( l = 1, 2, \ldots, L \). Thus
\[
\sum_{w \in A_i} c_w = \sum_{w \in A_i} \sum_{1 \leq j \leq L} |\{w \in W | w \in \Delta(S)\}| \varepsilon_j
= \sum_{1 \leq j \leq L} \sum_{w \in V_j} |\Delta_{A_i}(S)| \varepsilon_j.
\]
On the other hand, since \( y_S = \sum_{j:S \in V_j} \varepsilon_j, \)
\[
\sum_{S \subseteq T} h_i(S)y_S = \sum_{S \subseteq T} \sum_{j:S \in V_j} \varepsilon_j
= \sum_{1 \leq j \leq L} \sum_{j:S \in V_j} \varepsilon_j = \sum_{j:S \in V_j} |V_j| \varepsilon_j.
\]
Thus to show Lemma 2 it suffices to show that
\[
\sum_{S \subseteq V_j} |\Delta_{A_i}(S)| \leq d_{\max} |V_j| \quad \text{for all } j = 1, \ldots, L. \quad (3)
\]
For a set \( A \subseteq W - W_{j-1} \) which is infeasible to (IP)\( _i \), \( B \subseteq W - W_{j-1} \) is called a minimal augmentation of \( A \) if \( A \subseteq B \) and \( B \) is feasible to (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
\[
\sum_{S \subseteq V_j} |\Delta_{B}(S)| \leq d_{\max} |V(A)|. \quad (4)
\]
Then (3) holds by (4) by letting \( A = \{w_1, \ldots, w_{j-1}\} \) and \( B = A \cup \{w_j\} \) for all \( j = 1, \ldots, L \). Thus we only need to show (4). For this, we introduce a notation of witness set. Let \( U \triangleq \bigcup_{S \subseteq V(A)} \Delta_{B}(S) \subseteq B - A \).

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 \( w \not\in A \)). 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]. \( \Box \)

Let \( F = \{T \cup \{C_w \mid w \in U \} \} \) be the family obtained by adding \( T \) to a laminar witness set family. Construct a rooted tree \( T \) from \( F \) by set inclusion relationship as follows. (To avoid confusion, we will use the word "node" for the nodes of tree \( T \), and use the word "vertex" for the vertices of graph \( G \).) The node set of \( T \) consists of \( |F| \) nodes: \( u_C \in F \) for each \( C \). The root is \( w_T \), and for each non-root node \( u_C \) in \( T \), the parent of \( u_C \) is the node \( u_{C'} \) for the minimum \( C' \in F \) such that \( C \\subseteq C' \) (i.e., \( C \subseteq C' \) and \( C \neq C' \)). For each \( S \subseteq V(A) \), let \( u(S) \triangleq u_T \) for the minimum \( C \in F \) such that \( S \subseteq C \). An \( S \in V(A) \) is said to be associated with \( u(S) \). For each \( C \in F \), let \( n_C = |\{S \subseteq V(A) \mid u(S) = u_C \}| \) be the number of minimal violated sets that are associated with \( u_C \).
Let \( Q = \{u_C \in V(T) | n_C \geq 1\} \) be the subset of nodes with which at least one minimal violated set is associated. It is clear that

\[
|V(A)| = \sum_{u_C \in V(T)} n_C = \sum_{u_C \in Q} n_C.
\] (5)

Let \( d(u_C) \) denote the degree of a node \( u_C \) in \( T \). 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

\[
\sum_{u_C \in V(T)} d(u_C) \geq 2(|V(T)| - |Q|) - 1.
\] Since \( T \) is a tree, \( \sum_{u_C \in V(T)} d(u_C) = 2(|V(T)| - 1). \) Thus

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

We next show that

\[
\min_{u_C \in Q} \{d_{\max} - 1, n_C\} \leq \sum_{u_C \in Q} n_C.
\] (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 \( u_C \in Q, \)

\[
\sum_{S \in \mathcal{V}(A) \cup \mathcal{W} | u(S) = u_C} |\Delta_B(S)| \leq \min_{u_C \in Q} \{d_{\max} - 1, n_C\} d(u_C).
\] (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 \( F. \) By the definition of witness set family, we see that for any \( C' \subseteq F, C' \subseteq C_w \) implies \( \Gamma(w) \cap C' = \emptyset \), while \( C_w \subseteq C' \) 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 \( C' \subseteq F \) such that \( S \subseteq C' \subseteq C_w. \) Hence \( u(S) = u_{C_w}. \) Let \( u_C \) be the parent of \( u_{C_w}. \) Then \( \Gamma(w) \subseteq C_w. \) 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}\} \quad \text{(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 \n
\[
\min\{d_{\max} - 1, n_{C_w}\}.
\]

For any fixed \( u_C \in Q, \) the two cases may happen simultaneously. But we see that for one (undirected) edge \( \{u_C, u_{C_w}\} \) in tree \( T, \) there are at most \( \min\{d_{\max} - 1, n_{C_w}\} \) directed \( (u_C, u_{C_w}) \) edges that are produced in Case 1 or 2. Thus there are at most \( \min\{d_{\max} - 1, n_{C_w}\} d(u_C) \) 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) \cup \mathcal{W} | u(S) = u_C} |\Delta_B(S)|, \quad \text{where \( \Delta_B(S) \) is the set of nodes in \( S \) that are not connected to \( S \) by a \( (u_{C_w}, u_C) \) edge).}
\]

Thus (8) has been shown.

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 \( w \in \bigcup_{S \in \mathcal{V}(A)} \Delta_B(S) \).

Hence (8) has been shown.

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 \( c : W \to \mathbb{R}^+, \) and \( r_{st} \in \mathbb{Z}^+ \) 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 \( P \) (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 \( D \) from \( G \) to a digraph \( \overline{G} \) with edge capacities.

**Definition 4** \((D : G \to \overline{G}) \) 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 \( w^* \), change the tails of all directed edges \( (w, v) \) from \( w \) to \( w^* \), and add a new directed edge \( (w, w^*) \) of capacity \( 1 \).

In the following, a vertex in \( V(G) = T \cup W \) is also treated as a vertex in \( V(\overline{G}) \). However, the
notations of $\Gamma$ and $\Delta$ are only used with respect to $G$. Given $W' \subseteq W$, notice that in digraph $G[T \cup W']$, the capacity of a cut $C$ (i.e., $\emptyset \neq C \subseteq V(G[T \cup W'])$) is not $+\infty$ if and only if $C = S \cup \Gamma_W(S) \cup \{w \in \Gamma_W(S) - \Delta_W(S)\}$ 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 $G$, 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 $\bar{A}$. 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 $S = C_{st} \cup T$ for the minimum $s, t$-cut $C_{st}$ in digraph $G[T \cup \bar{A}]$ 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 $C_{st}$ in $G[T \cup \bar{A}]$ 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 minimal $s, t$-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 summarize 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 (IP)$^*$ 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


