A Note on Menger’s Theorem for Hypergraphs
A Note on Menger’s Theorem for Hypergraphs*

Ralf Borndörfer  Marika Karbstein

Abstract

We prove the companion Theorem to Menger’s Theorem for hypergraphs. This result gives rise to a new class of blocking pairs of ideal matrices, that generalize the incidence matrices of cuts and paths.

1 Introduction

Let \( H = (V, \mathcal{E}) \) be an undirected hypergraph with \( n \) vertices \( V = \{v_1, \ldots, v_n\} \) and \( m \) hyperedges \( \mathcal{E} = \{e_1, \ldots, e_m\} \), where \( e_i \subseteq V, i = 1, \ldots, m \). We want to allow for parallel edges, i.e., it is possible that \( e_i = e_j \) for two edges \( e_i \) and \( e_j, i \neq j \); we say that edges \( e_i \) and \( e_j, i \neq j \), are distinct. Let \( s \) and \( t \) be two different vertices of \( H \). An \( st \)-path \( (v_{i_0}, e_{j_1}, v_{i_1}, \ldots, e_{j_h}, v_{i_h}) \) in \( H \) is an alternating sequence of mutually different nodes \( v_{i_h}, h = 0, \ldots, k, \) and mutually distinct hyperedges \( e_{j_h}, h = 1, \ldots, k \), such that \( v_{i_0} = s, v_{i_h} = t, v_{i_h} \in e_{j_h} \) for all \( h = 1, \ldots, k \), and \( v_{i_k} = t \). The sequence \( \mathcal{P} = (e_{j_1}, \ldots, e_{j_h}) \) of hyperedges in an \( st \)-path is an \( st \)-hyperpath; we write \( e_{j_i} \in \mathcal{P}, i = 1, \ldots, k \), and we say that an \( st \)-hyperpath connects \( s \) and \( t \). A set of hyperedges \( \mathcal{E}' \subseteq \mathcal{E} \) is an \( st \)-hypercut if \( s \) and \( t \) are connected in \( H = (V, \mathcal{E}) \), but not in \( H' = (V, \mathcal{E} \setminus \mathcal{E}') \). Let \( \delta_H(W), W \subseteq V \), be the set of hyperedges that contain at least one node in \( W \) and one node in \( V \setminus W \), i.e., \( \delta_H(W) = \{e \in \mathcal{E} | e \cap W \neq \emptyset, e \cap (V \setminus W) \neq \emptyset\} \). Then \( \delta_H(W) \) is an \( st \)-hypercut for every \( s \in W \) and \( t \notin W \), provided that \( s \) and \( t \) are connected. A hypergraph is connected if each pair of nodes \( s, t \in V \) is connected. A hypergraph \( H = (V, \mathcal{E}) \) is \( k \)-hyperedge connected if \( H = (V, \mathcal{E} \setminus \mathcal{E}') \) is connected for every set of \( k - 1 \) hyperedges \( \mathcal{E}' \subseteq \mathcal{E} \), i.e., if the removal of \( k - 1 \) arbitrary hyperedges does not disconnect \( H \). Let \( c : \mathcal{E} \to \mathbb{N} \) be a weight function on the hyperedges with non-negative integer values. The capacity of a hypercut \( \mathcal{E}' \) w.r.t. \( c \) is \( c(\mathcal{E}') = \sum_{e \in \mathcal{E}'} c(e) / \text{length of a hyperpath } \mathcal{P} \) w.r.t. \( c \) is \( c(p) = \sum_{e \in p} c(e) \). An \( st \)-hypercut packing/st-hyperflow w.r.t. \( c \) is a set \( \{\mathcal{E}_1, \ldots, \mathcal{E}_k\} \) of \( st \)-hypercuts/a set \( \{p_1, \ldots, p_k\} \) of \( st \)-hyperpaths such that each hyperedge \( e \) is contained in at most \( c(e) \) hypercuts/hyperpaths; again, we allow for parallel hypercuts/hyperpaths. Two hypercuts/hyperpaths are disjoint if their edges are mutually distinct. The value or cardinality of an \( st \)-hypercut packing/st-hyperflow is the number \( k \) of hypercuts/hyperpaths it contains.

Figure 1 gives an example of a hypergraph with six hyperedges. Here, we illustrate a hyperedge by connecting its nodes in an arbitrary order (i.e., we represent

*Supported by the DFG Research Center “Mathematics for key technologies”
Adress of the authors: Zuse Institute Berlin, Takustr. 7, 14195 Berlin, Germany;
Email: {borndoerfer, karbstein}@zib.de
Figure 1: Example of a hypergraph $G = (V, \mathcal{E})$ with six hyperedges ($\mathcal{E} = \{e_1 = \{a, b, c, d\}, e_2 = \{e, f, g\}, e_3 = \{a, e\}, e_4 = \{e, f, c\}, e_5 = \{g, d\}, e_6 = \{f, g, c, d\}\}$). An ed-hyperpath $\mathcal{P} = (e_4, e_1)$ with associated ed-path $(e, e_4, c, e_1, d)$ contains the hyperedges $e_4$ and $e_1$; an example of an ed-hypercut is $\{e_2, e_3, e_4\}$ (hyperedges depicted as paths).

Menger’s theorem is known to hold for hypergraphs ([1, 2]).

**Proposition 1 (Menger’s Theorem for Hypergraphs).** The minimum cardinality of an $st$-hypercut is equal to the maximum number of hyperedge-disjoint $st$-hyperpaths.

**Corollary 2.** A hypergraph is $k$-hyperedge connected if and only if there are $k$ hyperedge-disjoint hyperpaths between each pair of nodes.

Multiplying hyperedges yields the following weighted version of Proposition 1.

**Corollary 3 (Max-Flow-Min-Cut Theorem for Hypergraphs).** The minimum capacity of an $st$-hypercut is equal to the maximum value of an $st$-hyperflow.

## 2 A Companion to Menger’s Theorem

We will prove in this section a companion theorem to Proposition 3, which arises from interchanging the roles of hyperpaths and hypercuts.

**Proposition 4.** The length of a shortest $st$-hyperpath is equal to the maximum value of an $st$-hypercut packing.

Considering unit costs yields our main result, a companion to Menger’s Theorem for hypergraphs.

**Proposition 5.** The minimum cardinality of an $st$-hyperpath is equal to the maximum number of hyperedge-disjoint $st$-hypercuts.

To prove Proposition 4 we first consider the following linear program

$$\begin{align*}
\text{min} & \quad \sum_{e \in \mathcal{E}} c_e x_e \\
\text{s.t.} & \quad \sum_{e \in \delta_H(W)} x_e \geq 1 & \forall s \in W \subseteq V \setminus \{t\} \\
& \quad x_e \geq 0 & \forall e \in \mathcal{E}.
\end{align*}$$

(1)
Here, $x_e$ is a variable which indicates how often the hyperedge $e$ is chosen. The inequalities guarantee that at least one hyperedge crosses each $st$-hypercut.

Proposition 4 follows from showing that the inequality system of program (1) plus the upper bounds $x_e \leq 1$ is TDI.

**Proposition 6.** The inequality system of program (1) is TDI.

**Proof.** It suffices to consider $st$-connected hypergraphs, because otherwise program (1) is infeasible, and a nonnegative cost vector $c$, because otherwise program (1) is unbounded. Consider the dual of program (1):

$$
\max \sum_{W \in \mathcal{W}} y_W \\
\text{s.t.} \sum_{W : e \in \delta_H(W)} y_W \leq c_e \quad \forall e \in \mathcal{E} \\
y_W \geq 0 \quad \forall W \in \mathcal{W},
$$

where $\mathcal{W} = \{ W \subseteq V \setminus \{ t \} \mid s \in W \}$. We use the primal-dual shortest hyperpath Algorithm 1 to construct optimal solutions $x$ and $y$ for the linear programs (1) and (2), respectively. The algorithm generalizes Dijkstra's algorithm to the hypergraph setting. It computes a shortest hyperpath from node $s$ to node $t$ with respect to the cost function $c$. Note that the algorithm does not compute a tree in a hypergraph (in contrast to Dijkstra's algorithm for graphs).

The distances from node $s$ are stored in node labels $d(v)$, and the nodes $v_i$ are marked in the order of increasing distance from the root; their unions $W_i = \bigcup_{1 \leq j \leq i} \{ v_j \}$ produce a sequence of nested hypercuts $\delta_H(W_i)$. The shortest $st$-hyperpath and the set of nested hypercuts give rise to primal and dual solutions $x$ and $y$ for programs (1) and (2), respectively. In the following we show that $x$ and $y$ are integral, feasible, and that the associated objective values are equal.

Consider the nodes $s = v_1, v_2, \ldots, v_h = t$ as marked by Algorithm 1 in line 18 and the node sets $W_0, \ldots, W_h$ as constructed in line 20. The following properties are easy to see:

1. $W_i = \{ v_1, \ldots, v_i \}$ for $i = 1, \ldots, h$ and $\emptyset = W_0 \subset W_1 \subset \ldots \subset W_h$. For each $W_i$, $i = 1, \ldots, h - 1$, we have $s \in W_i$ and $t \notin W_i$, i.e., $\delta_H(W_i)$ is an $st$-hypercut. (For $i = h$ we have $s, t \in W_h$, i.e., $\delta_H(W_h)$ is not an $st$-hypercut.)

2. $d(v_{i-1}) \leq d(v_i)$ for $i = 1, \ldots, h$. (For $i = 1$, note that $v_0 := s$ and $d(s) := 0$ is set in line 2, and $v_1 := s$ is set in the first pass through line 18.)

3. $d(t) < \infty$. ($H$ is $st$-connected.)

We first show that $y$ is a solution of program (2). Property 2 implies $y \geq 0$. In fact, the variables $y_W$ are zero for all $W \neq W_1, \ldots, W_{h-1}$. It remains to show that

$$\sum_{W : e \in \delta_H(W)} y_W \leq c_e \quad \forall e \in \mathcal{E}.$$  

Let $e \in \mathcal{E}$. If $v_i \notin e$ for all $i = 1, \ldots, h - 1$, then $e \notin \delta_H(W_i)$, $i = 1, \ldots, h - 1$, i.e., $\sum_{W : e \in \delta_H(W)} y_W = 0 \leq c_e$. Otherwise let $1 \leq i < h$ be the minimal
Algorithm 1: Primal-dual shortest hyperpath algorithm.

Input: An st-connected hypergraph \( H = (V, E) \), with costs \( c \in \mathbb{R}_+^E \) on the hyperedges, two nodes \( s, t \in V \).

Output: A minimum cost st-hyperpath \( P \subseteq E \). Values for \( x \) and \( y \) for programs (1) and (2).

1. \( d(s) := 0, d(v) := \infty \forall v \in V \setminus \{s\}, p(v) := \emptyset \forall v \in V \)
2. \( i := 0, v_0 := s, W_0 := \emptyset, y_W := 0 \forall W \in W \)
3. \( P := \emptyset, k := 1, u_1 := t, x_p := 0 \forall e \in E, k = 1 \)
4. All nodes are unmarked. All hyperedges are unmarked.
5. while \( u_i \neq t \) and \( \exists \) unmarked node \( w \) with \( d(w) < \infty \) do
6.   find \( v \) with \( d(v) = \min \{d(w) \mid w \text{ unmarked}\} \)
7.   for all unmarked \( e \in E \) with \( v \in e \) do
8.     for all unmarked \( w \) with \( w \in e \) do
9.       if \( d(w) > d(v) + c_e \) then
10.      \( d(w) := d(v) + c_e \)
11. \( p(w) := v \)
12. \( P(w) := e \)
13.   end
14. end
15. mark \( e \)
16. mark \( v \)
17. \( v_{i+1} := v \)
18. \( y_{W_i} := d(v_{i+1}) - d(v_i) \)
19. \( W_{i+1} := W_i \cup \{v_{i+1}\} \)
20. \( i := i + 1 \)
21. end
22. while \( u_k \neq s \) do
23.   \( P := P \cup P(u_k) \)
24. \( x_P(u_k) := 1 \)
25. \( u_{k+1} := p(u_k) \)
26. \( k := k + 1 \)
27. end
28. return \( P, x, y \)

index smaller than \( h \) such that \( v_i \in e \), i.e., \( e \notin \delta_H(W_j) \) for \( 1 \leq j < i < h \) but \( e \in \delta_H(W_i) \). Let similarly \( i \leq \ell < h \) be the maximal index smaller than \( h \) such that \( e \notin \mathcal{W}_j \), i.e., we have \( e \in \delta_H(W_j) \) for \( i \leq j \leq \ell < h \) and we have \( e \notin \delta_H(W_j) \) for \( \ell < j < h \). Then equation (3) becomes:

\[
\sum_{W \in W : e \in \delta_H(W)} y_W = \sum_{j=\ell}^{\ell} y_{W_j} = \sum_{j=1}^{\ell} d(v_{j+1}) - d(v_j)
= d(v_{\ell+1}) - d(v_\ell) \leq d(v_\ell) + c_e - d(v_\ell) = c_e.
\]

For the last inequality we distinguish the cases \( v_{\ell+1} \in e \) and \( v_{\ell+1} \notin e \). In the first case \( d(v_{\ell+1}) \leq d(v_\ell) + c_e \), because this value is considered in the computation of the distance label \( d(v_{\ell+1}) \) in line 9 when \( v_i \) is marked. In the second case, \( v_{\ell+1} = v_h = t \) and there exists a node \( w \in e \) with \( w \notin W_{h-1} \). Since \( v_{\ell+1} \) is
marked and not, we have \( d(v_{t+1}) \leq d(w) \). Since \( w \) can be reached from \( v_i \) via \( e \) we have \( d(w) \leq d(v_i) + c_e \). Again, \( d(v_{t+1}) \leq d(v_i) + c_e \), and inequality (3) is satisfied.

We now show that \( x \) is a solution of program (1). Due to the definition of \( x \) we have \( x \geq 0 \). We have to show that
\[
\sum_{e \in \delta_H(W)} x_e \geq 1 \quad \forall s \in W \subseteq V \setminus \{t\}.
\] (4)

Consider the nodes \( t = u_1, \ldots, u_k = s \) computed in the while loop starting in line 23 and an \( st \)-hypercut \( \delta_H(W) \). Let \( i \) be the largest index with \( u_i \notin W \) and \( u_{i+1} \in W \). This index exists since \( u_1 = t \notin W \) and \( u_k = s \in W \). Then we have \( x_P(u_i) = 1 \), \( P(u_i) \in \delta_H(W) \), and inequality (4) is satisfied.

The objective value of program (2) is
\[
\sum_{i=1}^{h-1} y_{W_i} = \sum_{i=1}^{h-1} (d(v_{i+1}) - d(v_i)) = d(v_h) - d(v_1) = d(t) - d(s) = d(t).
\]

Using lines 23 to 28 and 9 to 13 in Algorithm 1 we get
\[
d(t) = d(u_1) = d(u_2) + c_{P(u_1)} = d(u_3) + c_{P(u_2)} + c_{P(u_1)} = \ldots
\]
\[
= d(u_k) + \sum_{i=1}^{k} c_{P(u_i)} = 0 + \sum_{e \in \mathcal{P}} c_e = \sum_{e \in \mathcal{P}} c_e x_e,
\]
i.e., the objective values of (1) and (2) are equal. The integrality of \( x \) is obvious. Since \( c_e \) is integral, it follows that \( d(v_i) \) is integral for \( i = 0, \ldots, h-1 \). Therefore \( y_{W_i}, i = 1, \ldots, h-1, \) is also integral (line 19). This shows the claim.

\[
\min \sum_{e \in \mathcal{E}} c_e x_e
\] s.t.
\[
\sum_{e \in \delta_H(W)} x_e \geq 1 \quad \forall s \in W \subseteq V \setminus \{t\}
\] (5)
\[1 \geq x_e \geq 0 \quad \forall e \in \mathcal{E}.
\]

Corollary 7. The inequality system of program (5) is TDI.

Proof. Inequality system (5) adds upper bounds \( x \leq 1 \) to the inequality system (1); this adds variables \( z \) to the dual program. For solutions \( x \) and \( y \) as constructed by Algorithm 1 (note that \( x \leq 1 \)) the vectors \( x \) and \((y, 0)\) are primal and dual integer solutions of the extended systems with the same objective value.

\section{3 A New Class of Ideal Matrices}

\( st \)-hypercuts and \( st \)-hyperpaths form a blocking pair similar to \( st \)-cuts and \( st \)-paths. Likewise, the incidence matrices of all (inclusion wise) minimal \( st \)-hypercuts \( A_c \) and the incidence matrices of all (inclusion wise) minimal \( st \)-hyperpaths \( A_p \) form blocking pairs of matrices. By Propositions 3 and 4, these
matrices are ideal, like the incidence matrices of $st$-cuts and $st$-paths. Note that these matrices are in general not totally unimodular and also not balanced, see Figure 2 for an example. The $st$-hypercut incidence matrix $A_c$ for the hypergraph associated with this example is

$$
\begin{pmatrix}
1 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 \\
1 & 1 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 1 \\
\end{pmatrix}
\begin{pmatrix}
e_1 \\
e_2 \\
e_3 \\
e_4 \\
e_5 \\
e_6 \\
\end{pmatrix}
= A_c.
$$

The $3 \times 3$ matrix in the upper left corner has determinant $-2$.

The example also shows that the incidence matrices of $st$-hypercuts and $st$-hyperpaths form a new class of blocking pairs of ideal matrices which generalizes the class of incidence matrices of $st$-cuts and $st$-paths. This can be seen as follows. If $A_c$ would be an incidence matrix of $st$-cuts in an undirected graph $G = (V, E)$, the columns of $A_c$ have to correspond to the edges of $G$, i.e., such a graph would have six edges. Each cut of the graph contains exactly three edges, i.e., the edge-degrees of $s$ and $t$ have to be three. Furthermore, there can not be an edge connecting $s$ and $t$ since this edge would be contained in every cut. Therefore, the only possible graph is shown on the upper left of Figure 2. But this graph has seven minimal $st$-cuts instead of the six in matrix $A_c$. If $A_c$ would be an incidence matrix of $st$-paths in an undirected graph $G = (V, E)$, this graph would have six edges and each (minimal) path from $s$ to $t$ would have to contain exactly three edges. The only possible graph of this type is shown on the lower left of Figure 2. But this graph has eight minimal $st$-paths.
References
