The Hypergraph Assignment Problem
The Hypergraph Assignment Problem

Ralf Borndörfer∗ Olga Heismann∗∗

Abstract

The hypergraph assignment problem (HAP) is the generalization of assignments from directed graphs to directed hypergraphs. It serves, in particular, as a universal tool to model several train composition rules in vehicle rotation planning for long distance passenger railways. We prove that even for problems with a small hyperarc size and hypergraphs with a special partitioned structure the HAP is NP-hard and APX-hard. Further, we present an extended integer linear programming formulation which implies, e.g., all clique inequalities.

1 Directed Hypergraph Terminology

Definition 1.1. A directed hypergraph \(D = (V, W, A)\) is a pair of two disjoint vertex sets \(V, W\) of the same size \(|V| = |W|\) and a set of hyperarcs \(A \subseteq 2^V \cup W\) such that every hyperarc \(a \in A\) has the same size \(|a \cap V| = |a \cap W| > 0\) of vertices in \(V\) and \(W\). If \(|a \cap V| = |a \cap W| = 1\), then we call the hyperarc \(a\) also an arc.

In the usual definition of a directed hypergraph we have only one vertex set and every hyperarc is a pair of two sets called its tail and its head [GLPN93]. Despite that we do not require a nonempty intersection of the tail and the head, our definition is equivalent to it. To get this form of the directed hypergraph identify for the vertex sets \(V = \{v_1, \ldots, v_n\}, W = \{w_1, \ldots, w_n\}\) the vertex \(v_1\) with \(w_1\), \(v_n\) with \(w_n\) and take for each hyperarc \(a \in A\), \(a \cap V\) as its tail and \(a \cap W\) as its head.

Definition 1.2. Let \(D = (V, W, A)\) be a directed hypergraph. A hyperassignment in \(D\) is a subset \(H \subseteq A\) of the hyperarcs such that for every \(v \in V \cup W\) there exists exactly one \(a \in H\) such that \(v \in a\).

For an example of a directed hypergraph and a hyperassignment see figure 1.

Definition 1.3. Given a set \(S\), a cost function is a function \(c_S : S \to \mathbb{R}\). For \(T \subseteq S\) we define

\[
c_S(T) := \sum_{s \in T} c_S(s).
\]

∗Zuse Institute Berlin, Takustraße 7, 14195 Berlin, Germany, borndoerfer@zib.de
∗∗Zuse Institute Berlin, Takustraße 7, 14195 Berlin, Germany, heismann@zib.de
Figure 1: Visualization of the directed hypergraph $D = (V, W, A)$ with $V = \{v_1, v_2, v_3\}$, $W = \{w_1, w_2, w_3\}$, $A = \{a_1, a_2, a_3, a_4\}$, $a_1 = \{v_1, w_1\}$, $a_2 = \{v_1, v_2, w_1, w_2\}$, $a_3 = \{v_1, v_3, w_2, w_3\}$, $a_4 = \{v_3, w_3\}$ with the hyperassignment $\{a_2, a_4\}$ drawn with thick hyperarcs.

**Problem 1.4 (Hypergraph Assignment Problem).**

**Input:** A pair $(D, c_A)$ consisting of a directed hypergraph $D = (V, W, A)$ and a cost function $c_A : A \to \mathbb{R}$.

**Output:** A minimum cost hyperassignment in $D$ w.r.t. $c_A$, i.e., a hyperassignment $H^*$ in $D$ such that

$$c_A(H^*) = \min\{c_A(H) : H \text{ is a hyperassignment in } D\},$$

or the information that no hyperassignment in $D$ exists if this is the case.

Analogically to the well-known minimum cost problem on directed graphs, the HAP can be viewed as a minimum cost flow problem on directed hypergraphs: We can add a source and a sink vertex, add arcs from the source to all vertices in $V$ and from all vertices in $W$ to the sink, all with cost 0, and then ask for a minimum cost flow from the source to the sink with $|V|$ units of flow which satisfies a maximum capacity of 1 on all hyperarcs.

However, the usual setting for the minimum cost flow problem on directed hypergraphs, the minimum cost hyperflow problem, \cite{CGS92, CGS97, JMRW92} is a $B$-hypergraph $B = (N, E)$. There, we have one vertex set $N$ and a set of so-called $B$-hyperarcs (backward hyperarcs) $E$. A $B$-hyperarc $e = (T_e, h_e) \in E$ is pair of a vertex set $T_e \subseteq N$ and a vertex $h_e \in N$. Further, a demand vector $b \in \mathbb{R}^N$ is given. Also, weights can be associated with the $B$-hyperarcs but we we omit this here. A hyperflow $f \in \mathbb{R}^E$ is a function, which associates a flow value with each $B$-hyperarc such that the demand constraint

$$\sum_{e \in E : n = h_e} f_e - \sum_{e \in E : n \in T_e} f_e = b_n$$

is satisfied for all $n \in N$. The problem consists of finding a minimum cost hyperflow $f^*$, i.e.

$$\sum_{e \in E} c_E(e)f_e^* = \min \left\{ \sum_{e \in E} c_E(e)f_e : f \text{ is a hyperflow in } B \right\},$$
Figure 2: The corresponding B-hypergraph for the hypergraph in figure 1. The corresponding hyperflow is drawn with thick lines. The numbers next to the vertices are the values of the demand vector.

w. r. t. a given cost function \( c : E \rightarrow \mathbb{R} \) on the B-hyperarcs.

We can state the HAP in \( (D, c_A) \) with \( D = (V, W, A) \) as a minimum cost hyperflow problem in the following way. Let \( A = A_1 \cup A_2 \) where \( A_1 = \{ (h_a, t_a) : h_a \in V \cap A, t_a \in W, a \in A \} \) is the set of all arcs in \( A \) and \( A_2 = A \setminus A_1 \) are the other hyperarcs. We use the B-hypergraph \( B = (N, E) \) with \( N = V \cup W \cup A \) and \( E = E_1 \cup E_2 \cup E'_2, E_1 = \{ (\{ t_a \}, h_a) : a \in A_1 \}, E_2 = \{ (T_a, a) : T_a = a \cap V, h_a = a, a \in A_2 \}, E'_2 = \{ (T_a, h_a) : T_a = a \cap W, h_a = a, a \in A_2 \}. \) In the cost function, we assign \( c((t_a, h_a)) = c_a \) to the B-hyperarcs in \( E_1, c(T_a, h_a) = c_a \) to the B-hyperarcs in \( E_2 \) and cost 0 to all B-hyperarcs in \( E'_2 \). We define the demand vector \( b \) such that

\[
b_n = \begin{cases} 
-1 & \text{if } n \in V \\
1 & \text{if } n \in A \\
1 - |\{ a \in A_2 : w \in A \}| & \text{if } n \in W.
\end{cases}
\]

This leads to a cost-preserving bijection between the hyperassignments in \( D \) and hyperflows in \( B \) which assigns the following hyperflow values given a hyperassignment \( H \subseteq A \) in \( D \):

\[
f_e = \begin{cases} 
1 & \text{if } e = (\{ t_a \}, h_a) \in E_1, a \in H \\
0 & \text{if } e = (\{ t_a \}, h_a) \in E_1, a \in H \\
1 & \text{if } e = (T_a, h_a) \in E_2, a \in H \\
0 & \text{if } e = (T_a, h_a) \in E_2, a \notin H \\
0 & \text{if } e = (T_a, h_a) \in E'_2, a \in H \\
1 & \text{if } e = (T_a, h_a) \in E'_2, a \notin H.
\end{cases}
\]

For an example of the construction see figure 2.

To solve the minimum cost flow problem on directed hypergraphs, a hypergraph network simplex algorithm was introduced in [1]. In contrast to the problem on graphs, for integral input optimal integral solutions do not have to exist. The generalization of the network simplex algorithm can produce non-integral
solutions. Special cases where the set of feasible solutions is integral can be found in [JMRW92], but these usually do not apply to the instances arisen from the HAP.

Definition 1.5. For a vertex subset \( U \subseteq V \cup W \) we define the adjacent hyperarcs \( \delta(U) := \{ a \in A : a \cap U \neq \emptyset \} \) to be the set of all hyperarcs having at least one vertex in \( U \) and write also \( \delta(v) \) for \( \delta(\{v\}) \) if \( v \) is a vertex.

Definition 1.6. \( D = (V,W,A) \) is called a partitioned directed hypergraph with maximum part size \( d \in \mathbb{N} \) if additionally there exist pairwise disjoint \( \leq d \)-element sets \( V_1, \ldots, V_p \) and \( W_1, \ldots, W_q \) called the parts of \( H \) such that \( \bigcup_{i=1}^{p} V_i = V, \bigcup_{i=1}^{q} W_i = W \) and for every \( a \in A \) there exist \( i \in \{1, \ldots, p\}, j \in \{1, \ldots, q\} \) such that \( a \cap V_i \subseteq V_j, a \cap W_i = \emptyset \forall k \neq j, a \cap W_k = \emptyset \forall k \neq i \).

We define the set of all configurations for some part \( P \in \{V_1, \ldots, V_p, W_1, \ldots, W_q\} \) to be

\[
\mathcal{C}_p = \left\{ C \subseteq A : a_1 \cap a_2 = \emptyset \ \forall a_1, a_2 \in C \text{ with } a_1 \neq a_2, \bigcup_{a \in C} a \in \{V \cup P, W \cup P\} \right\}.
\]

We write \( \mathcal{C}^V \) for \( \bigcup_{i=1}^{p} \mathcal{C}_{V_i} \) and \( \mathcal{C}^W \) for \( \bigcup_{i=1}^{q} \mathcal{C}_{W_i} \), respectively. Finally, let \( \mathcal{C} = \mathcal{C}^V \cup \mathcal{C}^W \).

For an example of a partitioned hypergraph see figure 3.

A hyperassignment in a partitioned hypergraph can be viewed as a system of disjoint representatives. To do this we can merge the all the vertices in each part of \( V \) to one vertex and use the configurations in \( \mathcal{C}^V \) to describe the HAP instead of hyperarcs. For this problem a generalization of Hall’s theorem was proved in [AH00]. This could be used to check whether there exists a hyperassignment in a given partitioned hypergraph. But even more than Hall’s theorem for bipartite matchings this is unusable for practical computations because of the very high exponential number of requirements one has to check.

2 Complexity and Structural Results

Theorem 2.1. Given a directed hypergraph \( D = (V,W,A) \) and a cost function \( c_A : A \to \mathbb{R} \) the hypergraph assignment problem is NP-hard and APX-hard, even if \( H \) is partitioned with maximum part size 2.

Proof. We will use the NP-complete [GJ79] page 46] and in its optimization version APX-hard [Kan91] 3-dimensional matching problem. The input of the 3-dimensional matching problem is an undirected hypergraph \( U \), i.e., a pair \( U = (X \cup Y \cup Z, E) \), \( E \subseteq 2^{X \cup Y \cup Z} \) with \( |X| = |Y| = |Z| \),

\[|e \cap X| = |e \cap Y| = |e \cap Z| = 1 \ \forall e \in E.\]

It asks whether a partitioning of \( U \), i.e., \( F \subseteq E \) such that each element from \( X \cup Y \cup Z \)
Figure 3: Visualization of a partitioned hypergraph with maximum part size $d = 3$ and parts $\{v_1\}, \{v_2, v_3\}, \{v_4, v_5, v_6\}$ and hyperarcs $\{v_1, w_2\}, \{v_2, v_3, w_4, w_5\}$ and hyperarcs $\{v_4, v_5, w_6\}$. The vertices of each part with more than one vertex are surrounded by an ellipse in the picture. For partitioned hypergraphs we visualize the hyperarcs which connect all the vertices from one part with all the vertices from another part by drawing just an arrow between the two ellipses surrounding the vertices of the part.

is contained in exactly one set in $F$, exists. Let

$$X = \{x_1, \ldots, x_n\},$$

$$Y = \{y_1, \ldots, y_n\},$$

$$Z = \{z_1, \ldots, z_n\},$$

$$E = \{\{x_{1i}, y_{1j}, z_{1k}\}, \ldots, \{x_{ni}, y_{nj}, z_{nk}\}\}.$$ 

To prove the theorem we construct an instance of the hypergraph assignment problem having polynomial size in the size of a given 3-dimensional matching problem such that there exists a hyperassignment if and only if there exists a partitioning of $U$.

Let $D = (V, W, A)$ be a partitioned hypergraph with parts

$$V^E_1 = \{v^E_{1x}, v^E_{1y}\}, \ldots, V^E_m = \{v^E_{mx}, v^E_{my}\},$$

$$V^Z_1 = \{v^Z_{1a}, v^Z_{1b}\}, \ldots, V^Z_n = \{v^Z_{za}, v^Z_{zb}\}$$

in $V$ and

$$W^E_1 = \{w^E_{1za}, w^E_{1zb}\}, \ldots, W^E_m = \{w^E_{mza}, w^E_{mzb}\},$$

$$W^X_1 = \{w^X_{1x}, w^X_{1y}\}, \ldots, W^X_n = \{w^X_{nx}, w^X_{ny}\}$$

in $W$. 

5
show that this also theoretically makes sense: Every directed hypergraph with the maximum number of vertices contained in a hyperarc can be transformed into a hypergraph modeling rotation planning problems for long distance passenger railways. Further, we can show that this also theoretically makes sense: Every directed hypergraph with the maximum number of vertices contained in a hyperarc can be transformed into a hypergraph modeling rotation planning problems for long distance passenger railways.
to a partitioned hypergraph $D'$ with maximum part size $d$ such that there exists a bijection between the hyperassignments in $D$ and $D'$.

The construction works as follows (for a formal proof see the next theorem). We duplicate each vertex $v$ of $D$ such that every hyperarc in $D$ which contains $v$ is transformed to a hyperarc of the same size in $D'$ using a different duplicate $v'$ of $v$. Then, we add arcs which ensure that we can always use only exactly one of the duplicates of the original vertex as an element of the corresponding original’s adjacent hyperarcs in a hyperassignment in $D'$.

**Theorem 2.2.** Let $D = (V, W, A)$ be a directed hypergraph with maximum hyperarc size $d$ and $c_A$ a cost function. Then there exists a partitioned hypergraph $D' = (V', W', A')$ with maximum part size $d$ and a cost function $c_{A'}$ such that there is a bijection between the set of hyperassignments in $D$ and the set of hyperassignments in $D'$ preserving the costs w. r. t. the cost functions $c_A$ and $c_{A'}$, respectively.

**Proof.** Let

$$V' = \{(v, a) : v \in V, a \in A, v \in a\} \cup \{(w, i) : w \in W, i \in \{1, \ldots, |\delta(w) - 1|\}\}$$

and

$$W' = \{(w, a) : w \in W, a \in A, w \in a\} \cup \{(v, i) : v \in V, i \in \{1, \ldots, |\delta(v) - 1|\}\}.$$ 

For $v \in V$ define $\{v_i', v_{\delta(v)}'\} := \{(v, a) \in V'\}$ to be an arbitrary but fixed order of the vertices constructed from $v$ and, analogically, the $w_i'$.

For each $a \in A$ we construct a corresponding hyperarc $a' = \{(v, a), (w, a) : v \in V, w \in W\} \in A'$ with cost $c_{A'}(a') = c_A(a)$. Further, we construct arc sets to control that we always choose exactly one of the hyperarcs $a'$ in a hyperassignment in $D'$ which contain a vertex from $V'(v)$ or $W'(w)$. We assign to all these other arcs the cost 0 by $c_{A'}$. We have

$$A' := \{a' : a \in A\} \cup \{(v_i', v_{j}) : (i - j) \in \{0, 1\}, v \in V, i \in \{1, \ldots, \delta(v)\}\} \cup \{(w_i', w_j') : (i - j) \in \{0, 1\}, w \in W, i \in \{1, \ldots, \delta(w)\}\}.$$

$D'$ can be partitioned: Every vertex has by construction only one adjacent hyperarc which is not an arc. The tail and head of an arc are always contained in only one part. The maximum hyperarc size of $D$ gives us the maximum part size of $D'$ in the finest possible partitioning.

Now, let $H$ be a hyperassignment in $D$. For every vertex $v \in V$ let $i(v)$ be the index of $v_{\delta(v)} = (v, a)$ where $a$ is the unique hyperarc in $H$ containing $v$ and analogically define $i(w)$ for $w \in W$. Let

$$H' := \{a' : a \in H\} \cup \{v_i', (v, i) : v \in V, i < i(v)\} \cup \{v_{\delta(v)}', (v, i) : v \in V, i > i(v)\} \cup \{(v, i), v_i' : w \in W, i < i(w)\} \cup \{(w, i - 1), w_i' : w \in W, i > i(w)\}.$$
By construction, this is a hyperassignment in $D'$ with the same cost w.r.t. $c_A$ as $H$ in $D$ w.r.t. $c_A$.

On the other hand, given a hyperassignment $H'$ in $D'$ we can construct a hyperassignment $H$ in $D$ using exactly the corresponding hyperarcs from $A$ for the chosen $a' \in H'$ and thus having the same cost. To prove this, we need to show that for every vertex $v \in V \cup W$ of $D$, $H'$ has only one vertex $(v, a)$ which is contained in a hyperarc $a'$ in $H'$. For this purpose observe that by construction of $D'$ there is exactly one vertex less of the type $(v, i)$ than of the type $(v, a)$ and the vertices of the first type have only adjacent hyperarcs which are arcs and whose other end is one of the vertices of the second type. The only other adjacent hyperarc to $(v, a)$ is $a'$.

3 Clique Inequalities and an Extended Formulation

The canonical integer linear program for the HAP is the following.

\[
\begin{align*}
\text{minimize} & \quad \sum_{a \in A} c_A(a)x_a \\
\text{subject to} & \quad \sum_{a \in \delta(v)} x_a = 1 \quad \forall v \in V \cup W \quad (i) \\
& \quad x \geq 0 \quad (ii) \\
& \quad x \in \mathbb{Z}^A \quad (iii)
\end{align*}
\]

It is easy to see that (HAP) is a valid formulation for the hypergraph assignment problem. For a formal proof see [Hei10].

\textbf{Definition 3.1.} A \textit{clique} in (the conflict graph of) a directed hypergraph $D = (V, W, A)$ is a set $Q \subseteq A$ of hyperarcs such that every two hyperarcs $a_1, a_2 \in Q$ have at least one vertex in common, i.e., $a_1 \cap a_2 \neq \emptyset$. A clique $Q$ is a \textit{maximal clique} if there is no clique $Q' \supset Q$ containing $Q$ and additionally other hyperarcs.

We call $\sum_{a \in Q} x_a \leq 1$ a clique inequality.
Every feasible solution of (HAP) fulfills every clique inequality since by definition no two elements of a clique can be both in a hyperassignment.

We want to state an extended integer linear programming formulation for the HAP. After proving its correctness we will show that it implies all clique inequalities. Clique inequalities are important since for HAPs with real application data in partitioned hypergraphs (there the maximum part size is 7) the gap between the optimum solution and the solution of the linear programming relaxation of (HAP) could be highly reduced by inserting clique inequalities [BH11].

\[
\begin{align*}
\text{minimize} & \quad \sum_{a \in A} c_a(a)x_a & \quad \text{(HAP}_{\text{ext}}) \\
\text{subject to} & \quad \sum_{C \in \mathcal{C}^{V}: \delta_P \cap C = \{a\}} y_C = x_a \quad \forall a \in A \\
& \quad \sum_{C \in \mathcal{C}^{W}: \delta_P \cap C = \{a\}} y_C = x_a \quad \forall a \in A \\
& \quad \sum_{a \in \delta(v)} x_a = 1 \quad \forall v \in V \cup W \\
& \quad x, y \geq 0 \\
& \quad x \in \mathbb{Z}^A \\
& \quad y \in \mathbb{Z}^{\mathcal{C}^V} \times \mathbb{Z}^{\mathcal{C}^W} 
\end{align*}
\]

**Definition 3.2.** Let

\[ P_{\text{LP}}(\text{HAP}_{\text{ext}}) := \{(x, y) \in \mathbb{R}^A \times \mathbb{R}^{\mathcal{C}^V} \times \mathbb{R}^{\mathcal{C}^W} : \text{(HAP}_{\text{ext}}) \} \]

be the polyhedron associated with the LP relaxation of (HAP). Further, let

\[ \pi_x : \mathbb{R}^A \times \mathbb{R}^{\mathcal{C}^V} \times \mathbb{R}^{\mathcal{C}^W} \rightarrow \mathbb{R}^A, \quad (x, y) \mapsto x \]

be a mapping that produces a mapping onto the coordinates of all the hyperarc variables and

\[ \pi_{x^P} : \mathbb{R}^A \times \mathbb{R}^{\mathcal{C}^V} \times \mathbb{R}^{\mathcal{C}^W} \rightarrow \mathbb{R}^{\delta(P)}, \quad (x, y) \mapsto (x_a)_{a \in \delta(P)} \]

be a mapping that produces a projection onto the coordinates of all the variables for hyperarcs adjacent to the part \( P \).

The main idea for the correctness proof is the following.

**Lemma 3.3.** Let \( D = (V, W, A) \) be a partitioned hypergraph. Then, for every hyperassignment \( H \) of \( D \) there exist sets \( \mathcal{C}^V_{\mu} \subseteq \mathcal{C}^V \) and \( \mathcal{C}^W_{\mu} \subseteq \mathcal{C}^W \) of configurations such that for all \( a \in A \) \(|\mathcal{C} \in \mathcal{C}^V_{\mu} : a \in C\| = |\mathcal{C} \in \mathcal{C}^W_{\mu} : a \in C\| = |H \cap \{a\}|, \) i.e., \( \mathcal{C}^V_{\mu} \) and \( \mathcal{C}^W_{\mu} \) are sets of configurations whose disjoint union is exactly \( H \), respectively.

**Proof.** By the definition of a hyperassignment, for \( \mathcal{C}^V_{\mu} := \{C^V_P : P \in \{V_1, \ldots, V_p\}\} \) and \( \mathcal{C}^W_{\mu} := \{C^W_P : P \in \{W_1, \ldots, W_p\}\} \) with \( C^P = \delta(P) \cap H \) the required condition holds by construction. \( \square \)
This easily implies the correctness proof.

**Theorem 3.4.** Given a directed hypergraph $D = (V, W, A)$ and a cost function $c_A : A \to \mathbb{R}$, there are bijections between the feasible solutions of (HAP\_ext) and hyperassignments of $D$. The optimum value of (HAP\_ext) is equal to the cost of the minimum cost hyperassignment in $D$ w.r.t. $c_A$ if it exists and to $\infty$ otherwise.

**Proof.** Since the projection $\pi_x(x, y)$ of every feasible solution $(x, y)$ of (HAP\_ext) to its $x$-values is a feasible solution of (HAP) (the constraints from (HAP) are all contained in (HAP\_ext)) and the objective functions are the same, the previous lemma implies that (HAP\_ext) is a correct formulations for the HAP: For $C \in \mathcal{C}$ set $y_C = 1$ if $C \in \mathcal{C}_V \cup \mathcal{C}_W$ and $y_C = 0$ otherwise and use the same $x$-values as in (HAP).

**Lemma 3.5.** Let $D = (V, W, A)$ be a partitioned hypergraph. Then for every clique $Q$ in $D$ there exists a part $P$ such that $Q \subseteq \delta(P)$, i.e., every clique is a subset of the adjacent hyperarcs of the vertices of some part in $D$.

**Proof.** Let $Q$ be a nonempty clique in $D$ (otherwise the lemma in trivial) containing some hyperarc $a_1$ where $a_1 \cap V$ is contained in some part $V_1$ and $a_1 \cap W$ is contained in some part $W_1$. Thus, for every other hyperarc $a$ in $Q$ either $a \cap V \subseteq V_1$ or $a \cap W \subseteq W_1$, otherwise they would have an empty intersection. Assume that $Q$ contains some hyperarc $a_2$ with $a_2 \cap V \not\subseteq V_1$ and some hyperarc $a_3$ with $a_2 \cap W \not\subseteq W_1$. Now, $a_2 \cap a_3$ must be empty since both $a_2 \cap V$, $a_3 \cap V$ and $a_2 \cap W$, $a_3 \cap W$ are contained in different parts, which contradicts the assumption that $Q$ is a clique. Hence, either $V_1$ or $W_1$ contains $a \cap V$ or $a \cap W$ for all hyperarcs $a \in Q$. 

**Theorem 3.6.** Let $D = (V, W, A)$ be a partitioned hypergraph and $Q \subseteq A$ a clique. Then, the clique inequality

$$\sum_{a \in Q} x_a \leq 1$$

is a valid inequality for every feasible solution of the LP relaxation of (HAP\_ext).

**Proof.** First of all, observe that $|Q \cap C| \leq 1$ for every $C \in \mathcal{C}$ because if two hyperarcs are in $C$, then they are disjoint, which implies that they cannot be both in $Q$.

By Lemma 3.5 $Q \subseteq \delta(P)$ for some part $P$. W.l.o.g., let $P \subseteq V$ (the formulation is symmetric in $V, W$). Let $v$ be some vertex in $P$. Then, by (HAP\_ext) (iii)

$$1 = \sum_{a \in \delta(v)} x_a,$$

which is the sum of the right hand sides of (HAP\_ext) (i) for $a \in \delta(v)$. Considering the left hand sides we get

$$= \sum_{C \in \mathcal{C}} |\delta(v) \cap C| \cdot y_C.$$
Since every \( a \in \delta(v) \) is contained in \( V \) and exactly one such \( a \) is contained in every \( C \in \mathcal{C}_P \), we get

\[
\sum_{C \in \mathcal{C}_P} y_C.
\]

The observation from above implies

\[
\sum_{a \in Q} \left( \sum_{C \in \mathcal{C}_P} y_C \right) + \sum_{a \in \mathcal{C} \cap Q} y_C 
\]

by (HAP_ext) (iv). Finally, applying (HAP_ext) (i) again

\[
\sum_{a \in Q} x_a.
\]

The configuration formulation implies much more than the clique inequalities. It implies all valid inequalities one can know from considering all adjacent hyper-arcs to one part.

**Theorem 3.7.** Let \( D = (V, W, A) \) be a partitioned hypergraph and let \( P \) be some part. Then, \( \pi_x(P_{lp}(\text{HAP}\_\text{ext})) \) is a subset of

\[
X := \text{conv}\left\{ (x_a)_{a \in \delta(P)} \in \mathbb{Z}^{\delta(P)} : \sum_{a \in \delta(v) \cap \delta(P)} x_a \leq 1 \ \forall v \in V \cup W, \sum_{a \in \delta(v)} x_a = 1 \ \forall v \in P \right\}.
\]

**Proof.** Let \((x, y) \in P_{lp}(\text{HAP}\_\text{ext})\) be a feasible solution of the LP relaxation of (HAP_ext). For \( C \in \mathcal{C}_P \), define \( x(C) \in \mathbb{R}^{\delta(P)} \) by

\[
x(C)_a = \begin{cases} 1 & \text{if } a \in C \\ 0 & \text{otherwise.} \end{cases}
\]

This implies that every point in \( \pi_x(P_{lp}(\text{HAP}\_\text{ext})) \) is the linear combination of points \( x(C) \in X \), since \( x(C) \) has only integral coordinates 0 or 1 and satisfies by the definition of a configuration the equalities and inequalities defining \( X \).

---

**References**


