T-joins in Strongly Connected Hypergraphs

G. Calinescu *

Abstract

Given an edge-weighted undirected hypergraph $K = (V_K, E_K)$ and an even-sized set of vertices $R \subseteq V_K$, a T-cut is a partition of V_K into two parts Q and $\bar{Q} := V_K \setminus Q$ such that $|Q \cap R|$ is odd. A T-join in K for R is a set of hyperedges $M \subseteq E_K$ such that for every T-cut $(Q, \bar{Q})$ there is a hyperedge $e \in M$ intersecting both Q and $\bar{Q}$.

A *directed hypergraph* has for every hyperedge exactly one vertex, called the head, and several vertices, that are tails. A directed hypergraph is strongly connected if there exists at least one directed path between any two vertices of the hypergraph, where a directed path is defined to be a sequence of vertices and hyperedges for which each hyperedge has as one of its tails the vertex preceding it and as its head the vertex following it in the sequence. Orienting an undirected hypergraph means choosing a head for each hyperedge.

We prove that every edge-weighted undirected hypergraph that admits a strongly connected orientation has a T-Join of total weight at most $7/8$ times the total weight of all the edges of the hypergraph, and sketch an improvement to $4/5$. We also exhibit a series of example showing that one cannot improve the constant above to $2/3 - \epsilon$.

keywords: weighted hypergraph, directed hypergraph, T-join, T-cut

1 Introduction

Given an edge-weighted hypergraph $K = (V_K, E_K)$ and an even-sized set of vertices $R \subseteq V_K$, a T-cut is a partition of V_K into two parts Q and $\bar{Q} := V_K \setminus Q$ such that $|Q \cap R|$ is odd. A T-join in K for R is a set of hyperedges $M \subseteq E_K$ such that for every T-cut $(Q, \bar{Q})$ there is a hyperedge $e \in M$ intersecting both Q and $\bar{Q}$; such an hyperedge is said to *cross* the T-cut. See Figure 1 for intuition.

A minimum-weight T-join can be computed in polynomial-time if K is a graph (Chapter 29 of [9]). The generalization of Minimum Weight Graph T-join to hypergraphs, which we call *Hypergraph T-join*, is however NP-hard (Section 2).

A *directed hypergraph* has for every hyperedge exactly one vertex, called the tail, and several vertices, that are heads (to avoid having to rewrite most of the paper, we **reverse** the standard tail-head notation from the abstract above or from [5]; this reversal does not affect the results). A directed hypergraph is strongly connected if there exists at least one directed path between any two vertices

*Department of Computer Science, Illinois Institute of Technology, Chicago, IL 60616, USA. calinescu@iit.edu. Research supported in part by NSF grants CCF-0515088 and NeTS-0916743. Based on results from an extended abstract published in Springer LNCS 6520: *Proc.* 13^{th} *International Workshop on Approximation Algorithms for Combinatorial Optimization Problems*, 67–80 (APPROX 2010).

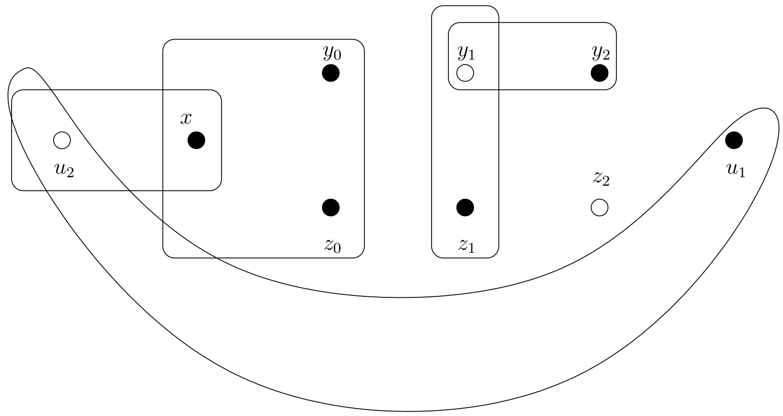

Figure 1: An example of a hypergraph T-join. R is given by the solid dark nodes. Another example appears later in Figure 3.

of the hypergraph, where a directed path between vertices u and v is defined to be an alternating sequence of vertices and hyperedges, starting with u and ending with v and such that each hyperedge has as tail the vertex preceding it and as one of its heads the vertex following it in the sequence. Orienting an undirected hypergraph means choosing a tail for each hyperedge. [5] characterizes the undirected hypergraphs that admit strongly connected orientations.

We study the supremum, over classes of hypergraphs and all possible R, of the minimum weight T-join divided by the weight of (all hyperedges in) the hypergraph. For (the class of) two-edge-connected graphs, this *T-ratio* is known (and not too hard to prove) to be $1/2$ [4].

Let $K = (V_K, E_K)$ be an undirected hypergraph. A path in a hypergraph consists of an alternating sequence of vertices and hyperedges for which each hyperedge contains the two vertices which precede and follow it in the sequence. Two vertices of K are in the same *connected component* if there exists a path with the two vertices as endpoints. It is easy to see that M is a T-join in K for R if and only if every connected component of the hypergraph (V_K, M) has an even number of vertices of R.

A hypergraph is two-edge-connected if there exist two hyperedge-disjoint paths between any two vertices. Note that Menger's theorem holds for hypergraphs and a hypergraph is two-edge-connected if and only if the removal of any single hyperedge does not disconnect the graph. It is not true [5] that every two-edge-connected hypergraph admits a strongly connected orientation (for graphs, this property holds and is known since at least 1939 [8]). However it is easy to prove that the undirected version of a strongly connected hypergraph is two-edge-connected.

In this paper we prove that for the class of edge-weighted undirected hypergraph that admit a strongly connected orientation, the T-ratio is at most $7/8$, and sketch an improvement to $4/5$. We also exhibit a series of example showing that the T-ratio is at least $2/3$. For the class of edge-weighted undirected hypergraphs that are two-edge-connected, we show that the T-ratio is 1.

The T-ratio for hypergraphs that admit strongly connected orientations was used by [2] to improve the approximation ratio for a power assignment problem, described below. A submitted journal version has an improved ratio (for the same algorithm) without using the T-ratio. In the first analysis of this algorithm, the T-ratio plays a similar role to the often used Steiner ratio [10, 1] for Steiner Tree. This paper still uses power assignment notation in addition to hypergraphs.

1.1 Power Assignment and Min-Power Strong Connectivity

Power Assignment problems take as input a directed simple graph $G = (V, E)$ and a cost function $c : E \to R_+$. The *power* of a vertex u in a directed spanning simple subgraph H of G is given by $p_H(u) = \max_{uv \in E(H)} c(uv)$, and corresponds to the energy consumption required for wireless node u to transmit to all nodes v with $uv \in E(H)$. The *power* (or *total power*) of H is given by $p(H) = \sum_{u \in V} p_H(u)$.

The study of Power Assignment was started by Chen and Huang [3], which consider, as we do, the case when E is bidirected, (that is, $uv \in E$ if and only if $vu \in E$, and if weighted, the two edge have the same cost; this case was sometimes called "symmetric" or "undirected" in the literature) while H is required to be strongly connected. See also [7]. We call this problem MIN-POWER STRONG CONNECTIVITY. We use with the same name both the (bi)directed and the undirected version of G. Hypergraphs arise naturally for MIN-POWER STRONG CONNECTIVITY (more so than for Steiner Tree), see Figure 2 for intuition.

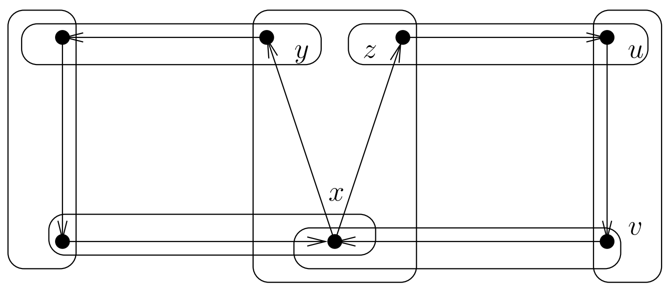

Figure 2: The arrows present a directed graph H. The rounded rectangles show an undirected hypergraph, obtained from H, by having a hyperedge (a subset of vertices) consisting of a vertex $v \in V$ and all $u \in V$ with $vu \in H$, with the weight of a hyperedge typically being $p_H(v)$. Thus in this example, the weights $w(\{x, y, z\}) = \max(c(xy), c(xz))$ and $w(\{u, v\}) = c(uv)$.

2 Preliminaries

In directed graphs, we use *arc* to denote a directed edge. In a directed graph K, an *incoming arborescence* rooted at $x \in V(K)$ is a spanning subgraph T of K such that the underlying undirected graph of T is a tree and every vertex of T other than x has exactly one outgoing arc in T. Given an arc xy, its *undirected version* is the undirected edge with endpoints x and y.

An alternative definition of MIN-POWER STRONG CONNECTIVITY (how it was originally posed) is: we are given a simple undirected graph $G = (V, E)$ and a cost function $c : E \to R_+$. A power assignment is a function $p : V \to R_+$, and it induces a simple directed graph $H(p)$ on vertex set V given by xy being an arc of $H(p)$ if and only if $\{x, y\} \in E$ and $p(x) \geq c(\{x, y\})$. The problem is to minimize $\sum_{u \in V} p(u)$ subject to $H(p)$ being strongly connected. To see the equivalence of the definition, given directed spanning subgraph H, define for each $u \in V$ the power assignment $p(u) = p_H(u)$.

While it may be already known, it is easy to check that Hypergraph T-join is indeed NP-hard, by a reduction from 4-D Matching (which asks if a 4-regular hypergraph K with $V(K)$ a multiple of 4, contains a perfect matching, that is, a set of disjoint hyperedges containing every vertex of the input hypergraph; see Garey and Johnson [6] problem SP1). It is easy to check that for $R = V(K)$, a T-join of size at most $|V(K)|/4$ must be a perfect matching.

For $u \in V$ and $r \in \{c(uv) \mid uv \in E\}$, let $S(u,r)$ be the directed star with center u containing all the arcs uv with $c(uv) \leq r$; note that r is the power of S. For a directed star S, let $E(S)$ be its set of arcs and $V(S)$ be its set of vertices. The vertices of the star other than the center are also called *leafs*. For a collection $\mathcal{A}$ of directed stars $S(u_i, r_i)$, define $w(\mathcal{A}) = \sum_{S(u_i,r_i)\in\mathcal{A}} r_i$, the total power used by the stars in $\mathcal{A}$.

Let $(S_v)_{v\in V}$ be the directed stars of OPT, with S_v centered at v, where OPT is the optimum feasible solution to a MIN-POWER STRONG CONNECTIVITY instance. As an aside, this and next section only use that OPT is feasible. Let $\mathcal{A}$ be collection of the stars of OPT. Let $K = (V_K, E_K)$ be the (undirected) hypergraph defined by $V_K = V$ and $E_K = \{V(S) \mid S \in \mathcal{A}\}$. Define the weight of an hyperedge to be the power of the corresponding directed star. Recall from the introduction that, with given $R \subseteq V$ with $|R|$ even, a T-cut is a partition of V into two parts Q and $\bar{Q} := V \setminus Q$ such that $|Q \cap R|$ is odd. A T-join in K for R is a set of hyperedges $M \subseteq E_K$ such that for every T-cut $(Q, \bar{Q})$ there is a hyperedge $e \in M$ intersecting both Q and $\bar{Q}$.

3 T-ratio in Hypergraphs that Admit Strongly Connected Orientation

Before the proof of our main result, Theorem 1 below, it is instructive to see why we cannot get a T-ratio of $1/2$, as it would be if we were dealing with graphs rather than hypergraphs. Below is a $3/5$ small example: (obtained from a Power Assignment instance, see Figure 3) nine vertices $x, y_0, y_1, y_2, z_0, z_1, z_2, u_1, u_2$, edges of cost 2: xy_0 and xz_0, edges of cost 1: y_1y_2, z_1z_2, and u_1u_2, and edges of cost 0: y_0y_1, z_0z_1, y_2u_1, z_2u_1, and u_2x. OPT has power 5: x has power 2 and y_1, z_1, and u_1 each have power 1. Thus K, obtained from OPT as above, has hyperedges of weight 0: $\{u_2, x\}$, $\{y_0, y_1\}$, $\{z_0, z_1\}$, $\{y_2, u_1\}$, and $\{z_2, u_1\}$, of weight 1: $\{u_1, u_2\}$, $\{y_1, y_2\}$, and $\{z_1, z_2\}$, and of weight 2: $\{x, y_0, z_0\}$. One can check by inspection that any T-join in K for $R = \{x, y_0, z_0, u_1\}$ has weight at least 3.

A series of examples where the ratio approaches $2/3$ is given in Subsection 5.1, in the appendix.

The theorem below is proved for K obtained from OPT as at the end of the previous section. However, the proof below never uses that OPT is an optimum. To get the theorem for an arbitrary hypergraph K' that admits a strongly connected orientation, construct from K' a strongly connected graph H as follows: $V_H = V_{K'} \cup E_{K'}$, and for any oriented hyperedge e of K', put in H arcs from e to all of e's heads, each with cost $w(e)$. Also. for any vertex v of K', put in H arcs of cost 0 from v to all $e \in E_{K'}$ such that v is the tail of e.

Use H as OPT (notice that H is strongly connected), and obtain K from OPT. Use the same $R \subseteq V_{K'}$; this is possible since $V_{K'} \subset V_K$. Note that $w(E_K) = w(E_{K'})$. If one takes M to be a T-join for R in K, removes from M the hyperedges with tail a vertex of $V_{K'}$, and replaces every hyperedge of M with tail a hyperedge $e \in E_{K'}$ by e, then one obtains a T-join for R in K' of the same weight, as it can be easily checked. Viceversa, if one starts with a T-join for R in K' (which we call M') then one can obtain a T-join M for R in K of the same weight as follows: put in M all the hyperedges of K corresponding to stars of power 0 in H, and all the hyperedges of K corresponding to stars centered

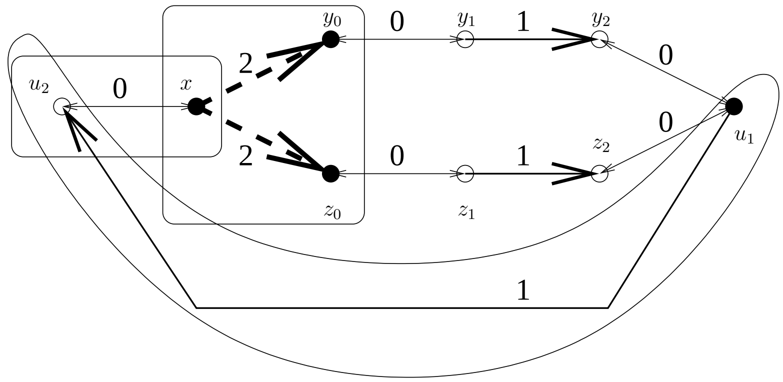

Figure 3: All edges have their cost written: thinnest edges have cost 0, medium thick have cost 1, and thickest edges have cost 2. Arrows indicate the optimum power assignment solution. Solid edges give the minimum spanning tree, and its vertices of odd degree are dark solid and form the set R. An example of a hypergraph T-join for R is given by the hyperedges represented by the three rounded shapes.

at vertices of H that are also hyperedges $e \in M'$, with power $w(e)$. One can check that indeed any connected component of the hypergraph (V_K, M) has an even number of vertices of R.

Theorem 1 *For K the hypergraph obtained from strongly connected graph OPT and for arbitrary $R \subseteq V$, there is a T-join in K with weight at most $(7/8)w(K)$.*

Proof. Recall that $(S_v)_{v \in V}$ are the directed stars of OPT, with S_v centered at v, and $\mathcal{A}$ is the collection of these stars. $K = (V_K, E_K)$ is the (undirected) hypergraph defined by $V_K = V$ and $E_K = \{V(S) \mid S \in \mathcal{A}\}$. The weight of an hyperedge is the power of the corresponding directed star. For technical reasons reset in OPT costs as follows: $c(e) := p(S_v)$ if e has tail v. This does not change powers or weights, since for any v, $p_{OPT}(S_v)$ does not change.

We do the following ear-decomposition of OPT (see Figure 4 for an illustration): start with one arbitrary directed cycle (graph) H_1 inside OPT. We will construct strongly connected H_{i+1} out of H_i, stopping only when $V(H_i) = V$, as follows: Since OPT is strongly connected, there exist $x_i \in V(H_i)$ such that $V(S_{x_i})$ contains vertices not in H_i. Let $\tilde{S}_i$ be the maximal substar of S_{x_i} whose leafs are not in H_i. Let $u_1, u_2, \ldots, u_{k_i}$ be the vertices of $V(\tilde{S}_i) \setminus \{x_i\}$. For $j = 1$ to k_i, find a minimal path P^i_j in OPT from u_j to either a vertex in H_i or a vertex on some P^i_q with $q < j$. (strong connectivity guarantees the existence of these paths. Intuitively, the nice thing about these paths (and arborescences) is that their power equals their cost.) Add $\tilde{S}_i$ and $B_i := \cup_j P^i_j$ to H_i to make H_{i+1}. Let $\bar{i}$ be such that $V(H_{\bar{i}}) = V$, our last subgraph H.

We have that H_i is a subgraph of OPT, but not necessarily H_i is exactly the subgraph of OPT induced by $V(H_i)$, as for example some u_j may have two arcs of OPT going to vertices of H_i, and only one is included in H_{i+1}.

Note that a vertex v has outdegree one when it joins its first H_i; we call e_v the unique arc out of v in this H_i. Also note that a v can appear as an x_i at most once above (v is not used twice in this role in the ear decomposition). Let e_i be e_{x_i} (also depicted in Figure 4). For such an x_i, let $\hat{S}_i = \hat{S}_{x_i}$ be the star that contains e_i and all the arcs of $\tilde{S}_i$.

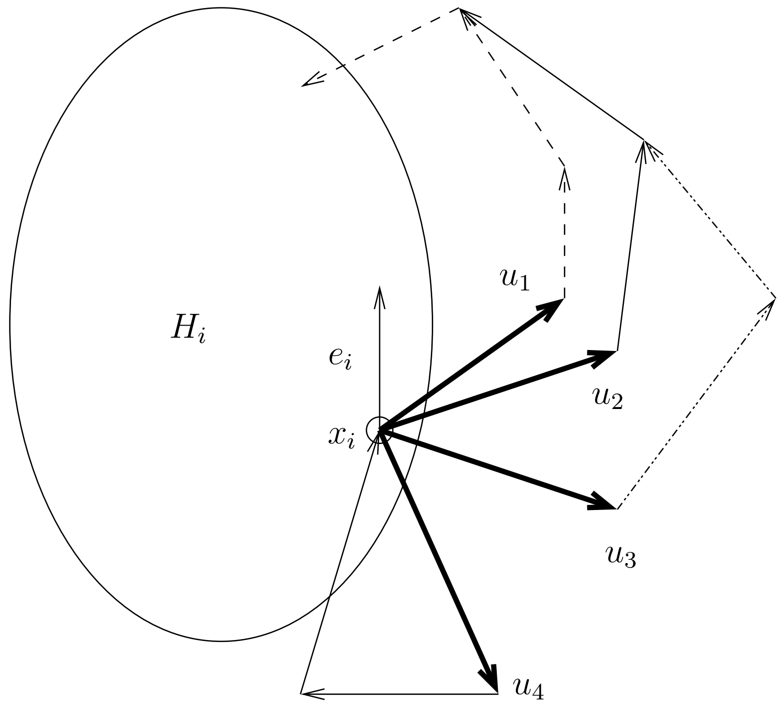

Figure 4: The vertices of H_i, a strongly connected subgraph, are in the ellipse. We select x_i to construct H_{i+1}. $\tilde{S}_i$ is represented by thick arcs, with four leafs u_1, u_2, u_3, u_4. The path P_1 is represented by dashed arrows, P_2 and P_4 use solid arrows, while P_3 uses dash-dots arrows. Altogether, $\tilde{S}_i$ and these paths are added to H_i to make H_{i+1}.

Let K_i be the following hypergraph: $V(K_i) = V(H_i)$ and $E(K_i)$ consists of the undirected version of the arcs of $E(H_i)$ and, if $i > 1$, the hyperedges $V(\hat{S}_j)$, for $1 \leq j \leq i-1$.

We use recursion to obtain a T-join $J_{\bar{i}}$ in $K_{\bar{i}}$, and an accounting scheme to prove that $J_{\bar{i}}$ has low weight. When processing H_i, we are given the set R_i for which we must find a T-join J_i in hypergraph K_i, and costs c_i on the arcs of H_i; for $H_{\bar{i}}$, $R_{\bar{i}} := R$ and $c_{\bar{i}} := c$. Costs c_i give power function p_i on H_i, and as we will see when we set up the recursion, c_i may differ from c only on arcs e_j for $j \geq i$, for which c_i may be 0; if the recursion picks one such an arc e of cost $c_i(e) = 0$, then the proof (later) makes sure that e will be removed at some point and not used in the final T-join; moreover e does not appear in any star of H_i with more than e as arcs (as $e = e_v$ for some v).

Moreover, vertices v of $V(H_i))$ can each have ***debt***: $debt_i(v)$, where $debt_{\bar{i}}(v) = 0$ for all $v \in V$. For K_i, the weight w_i of a hyperedge is obtained with respect to cost function c_i. If $i = 1$, we will obtain (later):

$$w(J_1) + \sum_{v \in V(H_1)} debt_1(v) \leq (7/8)p_1(H_1) \qquad (1)$$

For $i > 1$, we will carefully (later) select R_{i-1} and c_{i-1}, and recourse. Then we will construct J_i, a

T-join in K_i for R_i from J_{i-1} and some hyperedges of $E(K_i) \setminus E(K_{i-1})$ to satisfy:

$$w_i(J_i) - w_{i-1}(J_{i-1}) + \sum_{v \in V(H_i) \setminus V(H_{i-1})} debt_i(v) \tag{2}$$

$$\leq (7/8)\left(p_i(H_i) - p_{i-1}(H_{i-1})\right) + \sum_{v \in V(H_{i-1})} \left(debt_{i-1}(v) - debt_i(v)\right). \tag{3}$$

By summing up Inequations (1) and (3), one gets for all i:

$$w_i(J_i) + \sum_{v \in V(H_i)} debt_i(v) \leq (7/8)p_i(H_i), \tag{4}$$

which is exactly what we need once we plug in $i = \bar{i}$. What actually happens when we look at the cases later is that only for $v = x_{i-1}$, we can have $debt_{i-1}(v) \neq debt_i(v)$, so one can also think as "x_{i-1} gets into debt for the operation (reduction in size) and for retiring the debt of those nodes in H_i but not H_{i-1}". This way of thinking is also correct since $x_i \neq x_j$ for $i \neq j$, so x_{i-1} had no debt before we recourse from H_i to H_{i-1}. Thus we think, when doing a recursive step, that we have $(7/8)\left(p_i(H_i) - p_{i-1}(H_{i-1})\right)$ cash in hand, to pay for the operation and retiring the debt of those nodes in $V(H_i) \setminus V(H_{i-1})$; if this cash is not enough we borrow from (or, in other words, place a debt on) x_{i-1}.

We will prove that our recursion also maintains the following invariant: vertices have no debt except for those $v \in V(H_i)$ (for some i) such that $v = x_j$ for some $j \geq i$, for which

$$debt_i(v) \leq \frac{1}{8} c_i(e_v), \tag{5}$$

where recall that e_v is the unique arc out of v in H_i, i.e. if $v = x_j$, $e_v = e_j$.

If v is added in $H_{\bar{i}}$ ($v \in V(H_{\bar{i}}) \setminus V(H_{\bar{i}-1})$, is in our last "ear"), then as implied before, v carries no debt. Also, recall that $c_{\bar{i}}(e) = c(e)$ for every arc e. For the maintenance of these invariants and the definition of c_i, we look at three cases.

In the **first case**, $i = 1$, and we deal with H_1, which is a directed cycle. We have $(7/8)p_1(H_1) = (7/8)c_1(H_1)$ cash (with outdegree 1 for every vertex, its power equals the cost of the outgoing arc). Exactly as in Christofides' analysis, the arcs of H_1 are partitioned into two T-joins, D_0 and F_0 of K_1: go around the cycle and change T-join whenever meeting a vertex of R_1. That is, start with an arc arbitrarily and put it in D_0, and then process each e arc of C as follows: if the preceding arc $e' \in D_0$ and the tail of e is not in R_1, put $e \in D_0$; if $e' \in D_0$ and the tail of e is in R_1, put $e \in F_0$; if $e' \in F_0$ and the tail of e is in R_1, put $e \in D_0$; if $e' \in F_0$ and the tail of e is not in R_1, put $e \in F_0$.

We use for our T-join: D_0 if $c_1(D_0) \leq c_1(F_0)$; otherwise we use F_0. Our cash pays for the hyperedges we use as well for retiring the debt of all $v \in V(H_1)$: indeed this debt does not exceed $(1/8)(c_1(D_0) + c_1(F_0))$ provided the invariant is maintained. In other words, we get Inequation (1) using Invariant (5).

In the **second case**, $i > 1$ and $p_i(\hat{S}_{i-1}) \geq 2(c_i(B_{i-1}))$. We pick J_i, the T-join in K_i for R_i, as follows: all the hyperedges of K_i obtained from B_{i-1} and all the hyperedges of J_{i-1}, a recursively-obtained a T-join in K_{i-1} for $R_{i-1} \subseteq V(K_{i-1})$, where R_{i-1} is constructed as follows: We set $R_{i-1} = R_i$, but then we modify it below, keeping in mind we must at the end have $R_{i-1} \subseteq V(H_{i-1})$ and

$|R_{i-1}|$ even. B_{i-1} consists of a collection of vertex-disjoint incoming arborescences A^j_{i-1}, each with its own distinct root r^j_{i-1} in $V(H_{i-1})$. If A^j_{i-1} has, including its root, an odd number of vertices of R_i, remove those vertices from R_{i-1} and add r^j_{i-1} in R_{i-1}. If A^j_{i-1} has, including its root, an even number of vertices of R_i, remove those vertices from R_{i-1}. Both transformation keep R_{i-1} even-sized. Also, the final R_{i-1} is a subset of $V(H_{i-1})$.

Moreover, the union of B_{i-1} and a T-join in K_{i-1} for this R_{i-1} is indeed a T-join in K_i for R_i, as we argue below. An arbitrary T-cut $(Q, \bar{Q})$ in K_i for R_i is crossed by (one of the arcs of) B_{i-1} unless, for each index j, Q contains all the arborescence A^j_{i-1} or $\bar{Q}$ contains all the arborescence A^j_{i-1}. If the T-cut $(Q, \bar{Q})$ has this property, then replacing R_i by R_{i-1} does not change the parity of $Q \cap R_{i-1}$. Thus $\big(Q \cap V(K_{i-1}), \bar{Q} \cap V(K_{i-1})\big)$ is a T-cut in K_{i-1} for R_{i-1}, and is therefore crossed by the recursively constructed T-join in K_{i-1} for R_{i-1}.

Also, in this second case, we keep $c_{i-1}(e) = c_i(e)$ for every $e \in E(H_{i-1})$, and in particular $c_{i-1}(e_{i-1}) = c(e_{i-1})$, as e_{i-1} was not considered for a costs modification before this recursive step. We need to pay for the hyperedges obtained from the arcs of B_{i-1} as well as debt accumulated by the vertices of $V(H_i) \setminus V(H_{i-1})$. The total payment is, using Invariant (5), at most $(9/8)c_i(B_{i-1})$.

Our cash in hand is $(7/8)\left(p_i(H_i) - p_{i-1}(H_{i-1})\right) = (7/8)c_i(B_{i-1})$. We also put on x_{i-1} a debt of $(1/8)c(e_{i-1})$ (thus satisfying Invariant (5)), and use this amount for the payment.

Using the fact that in this (second) case, $p_i(\hat{S}_{i-1}) \geq 2(c_i(B_{i-1}))$, and that $p_i(\hat{S}_{i-1}) = c_{i-1}(e_{i-1}) = c(e_{i-1})$ (the first equality follows from the fact that we reset the costs such that all the arcs leaving a vertex have the same cost c), we can immediately verify that that the cash in hand plus the one taken as a loan from the debt on x_{i-1} is enough to do the payment. Precisely, we verified that:

$$\frac{9}{8}c_i(B_{i-1}) \leq \frac{7}{8}c_i(B_{i-1}) + \frac{1}{8}c(e_{i-1}), \tag{6}$$

or in other words Inequation (3) holds.

In the **third case**, $i > 1$ and $p_i(\hat{S}_{i-1}) < 2c_i(B_{i-1})$. In this case we plan to use $\hat{S}_{i-1}$ as well as some arcs from B_{i-1} in addition to a T-join in K_{i-1} for carefully defined R_{i-1} and cost c_{i-1}, as described below. We set $R_{i-1} = R_i$, but then we modify it below, keeping in mind we must at the end have $R_{i-1} \subseteq V(H_{i-1})$ and $|R_{i-1}|$ even. Consider, one by one the vertex-disjoint arborescences in B_{i-1}, that is, for each j, A^j_{i-1}, and let $R^j_{i-1} = R_i \cap V(A^j_{i-1})$. Make A^j_{i-1} undirected, and add to it, if $r^j_{i-1} \neq x_{i-1}$, the vertex x_{i-1} and the edge of weight 0: $r^j_{i-1}x_{i-1}$. For an edge/arc of A^j_{i-1}, have its weight equal its cost c_i. Add to A^j_{i-1} the undirected version of the arcs of the star $\tilde{S}_{i-1}$ with head in A^j_{i-1} (tail is x_{i-1} for all such arcs), each with weight 0.

This way we create a two-edge-connected undirected graph Z^j_{i-1}. Indeed, there are two edge-disjoint paths between any two vertices of Z^j_{i-1}, as explained in the remainder of this paragraph. If one vertex is the ancestor of the other in A^j_{i-1}, one path is in A^j_{i-1}, and the other goes from the lower of the two vertices to a leaf of A^j_{i-1} to x_{i-1} to r^j_{i-1} to the highest of the two vertices. If none is the ancestor of the other, one path is obtained by going up from both vertices in A^j_{i-1} until the least common ancestor, the other path by going down to leafs of A^j_{i-1} and passing through x_{i-1}.

If $|R^j_{i-1}|$ is even, let $\hat{R}^j_{i-1} = R^j_{i-1}$, else $\hat{R}^j_{i-1} = R^j_{i-1} \otimes r^j_{i-1}$. In all cases, $\hat{R}^j_{i-1}$ is even-sized. There exists a *minimal* T-join Y^j_{i-1} in Z^j_{i-1} for $\hat{R}^j_{i-1}$ of weight at most $\frac{1}{2}w(E(Z^j_{i-1}))$. If this Y^j_{i-1} contains the edge (of weight 0) $r^j_{i-1}x_{i-1}$, then set $\hat{Y}^j_{i-1}$ be Y^j_{i-1} without this edge; otherwise $\hat{Y}^j_{i-1} := Y^j_{i-1}$. Also take out of $\hat{Y}_{i-1}^j$ the edges/arcs of $\tilde{S}_{i-1}$; we are left only with the undirected version of some of the arcs of A_{i-1}^j, a subgraph of B_{i-1}. Also, modify R_{i-1} as indicated in the four subcases below.

In **Subcase 1**, $|R_{i-1}^j|$ is even, and Y_{i-1}^j contains the edge (of weight 0) $r_{i-1}^j x_{i-1}$ (so $x_{i-1} \neq r_j^{i-1}$); then set $R_{i-1} = \left(R_{i-1} \setminus R_{i-1}^j\right) \otimes \{x_{i-1}\} \cup \{r_{i-1}^j\}$. Note that whether $r_{i-1}^j \in R_{i-1}^j$ or not, R_{i-1} stays even-sized.

In **Subcase 2**, $|R_{i-1}^j|$ is even and Y_{i-1}^j does not contain the edge (of weight 0) $r_{i-1}^j x_{i-1}$ (this is also the case when $r_{i-1}^j = x_{i-1}$) ; then set $R_{i-1} = R_{i-1} \setminus R_{i-1}^j$. Note that R_{i-1} stays even-sized.

In **Subcase 3**, $|R_{i-1}^j|$ is odd and Y_{i-1}^j contains the edge (of weight 0) $r_{i-1}^j x_{i-1}$ (so $x_{i-1} \neq r_j^{i-1}$); then set $R_{i-1} = \left(R_{i-1} \setminus R_{i-1}^j\right) \otimes \{x_{i-1}\}$. Note that whether $r_{i-1}^j \in R_{i-1}^j$ or not, R_{i-1} stays even-sized.

In **Subcase 4**, $|R_{i-1}^j|$ is odd and Y_j^{i-1} does not contain the edge (of weight 0) $r_{i-1}^j x_{i-1}$, (this is also the case when $r_{i-1}^j = x_{i-1}$); then set $R_{i-1} = \left(R_{i-1} \setminus R_{i-1}^j\right) \cup \{r_{i-1}^j\}$. Note that whether $r_{i-1}^j \in R_{i-1}^j$ or not, R_{i-1} stays even-sized.

In all four (sub)cases, the vertices of A_{i-1}^j other than r_{i-1}^j, are removed from R_{i-1}. Thus the final $R_{i-1} \subseteq V(H_{i-1})$. After we finish this for all j (x_{i-1} may enter and exit R_{i-1} several times), set $c_{i-1}(e_{i-1}) = 0$ (for all the other arcs e, keep $c_{i-1}(e) = c_i(e)$). Thus the final $R_{i-1} \subseteq V(H_{i-1})$.

Recourse in K_{i-1}, obtaining T-join J_{i-1}. Now we construct J_i, our desired (but not proven yet to be one) T-join in K_i for R_i, as follows: $J_i = (J_{i-1} \setminus \{e_{i-1}\}) \cup \{\hat{S}_{i-1}\} \cup \left(\cup_j \hat{Y}_{i-1}^j\right)$. That is, we use the whole star of x_{i-1}, and if recursion uses the arc out of x_{i-1} of cost c_{i-1} zero, we give it up (since it is included in the star anyway). Note that in the end, all the arcs selected at artificial (reduced by the procedure) cost 0 are removed and replaced by a bigger star/hyperedge.

We need the following fact, for which we could only find a very long proof by case analysis despite the fact that this fact may be intuitively clear to the reader. Again, it makes sense to delay reading the proof.

Claim 1 *In all cases, J_i is a T-join in K_i for R_i.*

Proof. If $e_{i-1} \in J_{i-1}$, we used $\hat{S}_{i-1}$ in J_i instead of e_{i-1} and $\tilde{S}_{i-1}$. However, with hyperedges e_{i-1} and $\tilde{S}_{i-1}$ sharing vertex x_{i-1}, using $\hat{S}_{i-1}$ is equivalent, for crossing T-cuts, to using e_{i-1} and $\tilde{S}_{i-1}$.

Let us look again at the construction of R_{i-1}. We started with $R_{i-1}(0) = R_i$ (please do not confuse $R_{i-1}(k)$ with R_{i-1}^k, they are not the same set). We processed one by one the arborescences A_{i-1}^j, for $j = 1, 2, \ldots, q$ (for some $q = q_i$), constructing set of edges Y_{i-1}^j, and $R_{i-1}(j)$ from $R_{i-1}(j-1)$, until $R_{i-1} = R_{i-1}(q)$ is the subset of $V(H_{i-1})$ used for the T-join J_{i-1} in K_{i-1}.

Thus it is enough to show that $J_i = J_{i-1} \cup \{\tilde{S}_{i-1}\} \cup \left(\cup_j \hat{Y}_{i-1}^j\right)$ is a T-join in K_i for R_i (since, if $e_{i-1} \notin J_{i-1}$, we make the proof with $\tilde{S}_{i-1}$ instead of the larger set $\hat{S}_{i-1}$ as a hyperedge). Let $M_l := J_{i-1} \cup \{\tilde{S}_{i-1}\} \cup \left(\cup_{j=q-l+1}^q \hat{Y}_{i-1}^j\right)$ (with $M_0 := J_{i-1} \cup \{\tilde{S}_{i-1}\}$), and note that we need to prove that M_q is a T-join in K_i for R_i. We prove by induction on l that: M_l is a T-join for $R_{i-1}(q-l)$ in K_i. Applying this with $l = q$ yields the claim.

For the base case ($l = 0$), let $(Q, \bar{Q})$ be an arbitrary T-cut for $R_{i-1}(q) = R_{i-1}$. Then $\left(Q \cap V(H_{i-1}), \bar{Q} \cap V(H_{i-1})\right)$ is a T-cut for R_{i-1} in K_{i-1}, and therefore a hyperedge of the T-join J_{i-1} crosses this T-cut, and it crosses $(Q, \bar{Q})$ in K_i as well. Thus M_0 is a T-join in K_i for $R_{i-1}(q-0)$.

For the inductive case, proving for $l+1$ assuming it holds for l, we must look at how Y_{i-1}^{q-l} and $R_{i-1}(q-l)$ are constructed from A_{i-1}^{q-l} and $R_{i-1}(q-l-1)$. To simplify notation, in the rest of the proof, let $x := x_{i-1}$, $r := r_{i-1}^{q-l}$, $Z := Z_{i-1}^{q-l}$, $R := R_{i-1}(q-l-1)$, $R' := R_{i-1}(q-l)$, and $Y := Y_{i-1}^{q-l}$, the minimal T-join in Z for $\hat{R}_{i-1}^{q-l}$. To prove below that M_{l+1} is a T-join for R in K_i, we use that M_l is a T-join for R' in K_i.

To further simplify notation let $\tilde{S} := \tilde{S}_{i-1}$, $\hat{R} := \hat{R}_{i-1}^{q-l}$, $\tilde{R} := R_{i-1}^{q-l} \setminus \{r\}$, and $\hat{Y} := \hat{Y}_{i-1}^{q-l}$. Note that $M_{l+1} = M_l \cup \hat{Y}$, that $\tilde{R} = R \cap (V(Z) \setminus \{x, r\})$, that $\hat{R} = \tilde{R}$ or $\hat{R} = \tilde{R} \cup \{r\}$ (whichever makes $|\hat{R}|$ even), and that in all four subcases, $R' \subseteq (R \cup \{x, r\}) \setminus \tilde{R}$.

Let $(Q, \bar{Q})$ be an arbitrary T-cut for R, that is, a partition of V_{H_i} such that $|Q \cap R|$ has odd size. We need to find a hyperedge of M_{l+1} crossing the T-cut. First, we switch Q and $\bar{Q}$ if necessary such that $r \in Q$. If $R' \cap Q$ is odd, M_l has a hyperedge crossing $(Q, \bar{Q})$ and therefore M_{l+1} also has a hyperedge crossing $(Q, \bar{Q})$. So, from now on we assume $|R' \cap Q|$ is even (and so is $|R' \cap \bar{Q}|$).

We have, unfortunately, 16 cases based on whether $x \in R$ or not, $r \in R$ or not, $|\tilde{R}|$ even or not, and Y contains xr or not. One could combine cases, but for checking correctness one needs to split them again. In all cases, we find a hyperedge of M_{l+1} that crosses $(Q, \bar{Q})$: either $\tilde{S}$ or an edge of $\hat{Y}$. To do so, it is enough to find an edge e of Y, other than xr, crossing $\big(Q \cap V(Z), \bar{Q} \cap V(Z)\big)$. Indeed, $\hat{Y}$ is obtained from Y by removing the edges incident to x (if any), and all such edges other than xr are contained in the hyperedge $\tilde{S}$; so if e is incident to x, $\tilde{S}$ also crosses $(Q, \bar{Q})$. To find e, one reduces the 16 cases to one of the following three arguments:

Argument I. If Y does not contain xr and $Q \cap \hat{R}$ is odd-sized, then Y, being a T-join for $\hat{R}$ in Z, has an edge e of Y crossing in Z the cut $\big(Q \cap V(Z), \bar{Q} \cap V(Z)\big)$; note that $e \neq xr$ as $xr \notin Y$, and we are done.

Argument II. If Y contains xr, $x \in Q$, and $|Q \cap \hat{R}|$ odd, then Y, being a T-join for $\hat{R}$ in Z, has an edge e of Y crossing in Z the cut $\big(Q \cap V(Z), \bar{Q} \cap V(Z)\big)$; note that $e \neq xr$ since both x and r are in Q, and we are done.

Argument III. If Y contains xr, $x \notin Q$, and $|Q \cap \hat{R}|$ even, then we argue as follows. Recall that Y is a minimal T-join in the graph Z for $\hat{R}$. Let D be the connected component of $(V(Z), Y)$ containing both x and r, and split D in components D_r and D_x by removing the edge rx, which belongs to Y. Then both $|D_r \cap \hat{R}|$ and $|D_x \cap \hat{R}|$ are odd (or else, $Y \setminus \{xr\}$ would have an even number of elements of $\hat{R}$ in each connected component, and thus would also be a T-join for $\hat{R}$, contradicting the minimality of Y).

If $D_r \not\subseteq Q$, using that $r \in Q \cap D_r$, we get that an edge of Y other than xr crosses $\big(Q \cap V(Z), \bar{Q} \cap V(Z)\big)$, since D_r is connected and contains only edges of $Y \setminus \{xr\}$. Now assume that Q contains D_r. Using that $|D_r \cap \hat{R}|$ is odd, we get that $\hat{R} \cap ((Q \cap V(Z)) \setminus D_r)$ is an odd-sized subset of $\hat{R}$, and thus Y, being a T-join for $\hat{R}$ in Z, has an edge e crossing from $((Q \cap V(Z)) \setminus D_r)$. e cannot have x and r as endpoints, as neither of x, r is in $((Q \cap V(Z)) \setminus D_r)$ (recall that $x \notin Q$ and $r \in D_r$). The endpoint of e not in $((Q \cap V(Z)) \setminus D_r)$ cannot be in D_r by the maximality of the connected component D_r; indeed the only edge of Y crossing D_r is xr and we ruled out $e = xr$. Therefore $Y \setminus \{xr\}$ has the edge e crossing $\big(Q \cap V(Z), \bar{Q} \cap V(Z)\big)$, and we are done.

Here are the 16 cases:

1. $x \in R$, $r \in R$, $|\tilde{R}|$ even, Y contains xr (so $x \neq r$). Then $\hat{R} = \tilde{R}$, we are in Subcase 3, and $R' = R \setminus V(Z)$. If $x \in Q$, then in order to have $|Q \cap R|$ odd, we must have $|Q \cap \tilde{R}|$ odd (as we assumed $|R' \cap Q|$ is even, and $x, r \in R \cap Q$). Argument II applies. If $x \notin Q$, then in order to have $|Q \cap R|$ odd, we must have $|Q \cap \tilde{R}|$ even (as we assumed $|R' \cap Q|$ is even, and $r \in Q$, $x \notin Q$). Argument III applies

2. $x \in R$, $r \in R$, $|\tilde{R}|$ even, Y does not contain xr. It does not matter below whether $x \in Q$ ($x = r$ is possible) or $x \notin Q$ (so $x \neq r$). Then $\hat{R} = \tilde{R}$, we are in Subcase 4, and $R' = R \setminus \hat{R}$. In order to have $|Q \cap R|$ odd, we must have $|Q \cap \tilde{R}|$ odd (as we assumed $|R' \cap Q|$ is even). Argument I applies.

3. $x \in R$, $r \in R$, $|\tilde{R}|$ odd, Y contains xr (so $x \neq r$). Then $\hat{R} = \tilde{R} \cup \{r\}$, we are in Subcase 1, and $R' = (R \setminus V(Z)) \cup \{r\} = \left(R \setminus \tilde{R}\right) \setminus \{x\}$. If $x \in Q$, then in order to have $|Q \cap R|$ odd, we must have $|Q \cap \tilde{R}|$ even (as we assumed $|R' \cap Q|$ is even, and $x \in Q$), and therefore $|Q \cap \hat{R}|$ is odd. Argument II applies. If $x \notin Q$, then in order to have $|Q \cap R|$ odd, we must have $|Q \cap \tilde{R}|$ odd (as we assumed $|R' \cap Q|$ is even, and $x \notin Q$); thus $|Q \cap \hat{R}|$ is even. Argument III applies.

4. $x \in R$, $r \in R$, $|\tilde{R}|$ odd, Y does not contain xr. It does not matter below whether $x \in Q$ ($x = r$ is possible) or $x \notin Q$ (so $x \neq r$). Then $\hat{R} = \tilde{R} \cup \{r\}$, we are in Subcase 2, and $R' = R \setminus \hat{R}$. In order to have $|Q \cap R|$ odd, we must have $|Q \cap \hat{R}|$ odd (as we assumed $|R' \cap Q|$ is even). Argument I applies.

5. $x \in R$, $r \notin R$ (so $x \neq r$), $|\tilde{R}|$ even, Y contains xr. Then $\hat{R} = \tilde{R}$, we are in Subcase 1, and $R' = (R \setminus V(Z)) \cup \{r\} = \left((R \setminus \tilde{R}) \setminus \{x\}\right) \cup \{r\}$. If $x \in Q$, then in order to have $|Q \cap R|$ odd, we must have $|Q \cap \hat{R}|$ odd (as we assumed $|R' \cap Q|$ is even, and using that $x \in Q \cap R$ and $r \in (Q \cap R') \setminus R$). Argument II applies. If $x \notin Q$, then in order to have $|Q \cap R|$ odd, we must have $|Q \cap \hat{R}|$ even (as we assumed $|R' \cap Q|$ is even, and using that $x \notin Q$ and $r \in (Q \cap R') \setminus R$). Argument III applies.

6. $x \in R$, $r \notin R$ (so $x \neq r$), $|\tilde{R}|$ even, Y does not contains xr. It does not matter below whether $x \in Q$ or $x \notin Q$. Then $\hat{R} = \tilde{R}$, we are in Subcase 2, and $R' = R \setminus \tilde{R}$. In order to have $|Q \cap R|$ odd, we must have $|Q \cap \hat{R}|$ odd (as we assumed $|R' \cap Q|$ is even). Argument I applies.

7. $x \in R$, $r \notin R$ (so $x \neq r$), $|\tilde{R}|$ odd, Y contains xr. Then $\hat{R} = \tilde{R} \cup \{r\}$, we are in Subcase 3, and $R' = (R \setminus V(Z))$. If $x \in Q$, then in order to have $|Q \cap R|$ odd, we must have $|Q \cap \tilde{R}|$ even (as we assumed $|R' \cap Q|$ is even, and also $x \in Q \cap R$ and $r \notin (R \cup R')$). With $r \in Q$, we get $|Q \cap \hat{R}|$ odd and Argument II applies. If $x \notin Q$, in order to have $|Q \cap R|$ odd, we must have $|Q \cap \tilde{R}|$ odd (as we assumed $|R' \cap Q|$ is even, and using that $x \notin Q$ and $r \notin (R \cup R')$). Then $|Q \cap \hat{R}|$ is even, and Argument III applies.

8. $x \in R$, $r \notin R$ (so $x \neq r$), $|\tilde{R}|$ odd, Y does not contains xr. It does not matter below whether $x \in Q$ or $x \notin Q$. Then $\hat{R} = \tilde{R} \cup \{r\}$, we are in Subcase 4, and $R' = \left(R \setminus \tilde{R}\right) \cup \{r\}$. In order

to have $|Q \cap R|$ odd, we must have $|Q \cap \tilde{R}|$ even (as we assumed $|R' \cap Q|$ is even, and using that $r \in R' \setminus R$). As $(Q \cap \hat{R}) = (Q \cap \tilde{R}) \cup \{r\}$, $|Q \cap \hat{R}|$ is odd, an Argument I applies.

9. $x \notin R$, $r \in R$ (so $x \neq r$), $|\tilde{R}|$ even, Y contains xr. Then $\hat{R} = \tilde{R}$, we are in Subcase 3, and $R' = (R \setminus V(Z)) \cup \{x\}$. If $x \in Q$, then in order to have $|Q \cap R|$ odd, we must have $|Q \cap \tilde{R}|$ odd (as we assumed $|R' \cap Q|$ is even, and using that $r \in R \setminus R'$ and $x \in (R' \cap Q) \setminus R$). Therefore $|Q \cap \hat{R}|$ is odd and Argument II applies. If $x \notin Q$, then in order to have $|Q \cap R|$ odd, we must have $|Q \cap \tilde{R}|$ even (as we assumed $|R' \cap Q|$ is even, and using that $r \in R \setminus R'$ and $x \in R' \setminus Q$). Therefore $|Q \cap \hat{R}|$ is even, $x \notin Q$, and Argument III applies.

10. $x \notin R$, $r \in R$ (so $x \neq r$), $|\tilde{R}|$ even, Y does not contains xr. It does not matter below whether $x \in Q$ or $x \notin Q$. Then $\hat{R} = \tilde{R}$, we are in Subcase 4, and $R' = (R \setminus V(Z)) \cup \{r\} = R \setminus \tilde{R}$. In order to have $|Q \cap R|$ odd, we must have $|Q \cap \tilde{R}|$ odd (as we assumed $|R' \cap Q|$ is even). With $Q \cap \hat{R} = Q \cap \tilde{R}$, Argument I applies.

11. $x \notin R$, $r \in R$ (so $x \neq r$), $|\tilde{R}|$ odd, Y contains xr. Then $\hat{R} = \tilde{R} \cup \{r\}$, we are in Subcase 1, and $R' = (R \setminus V(Z)) \cup \{r, x\}$. If $x \in Q$, then in order to have $|Q \cap R|$ odd, we must have $|Q \cap \tilde{R}|$ even (as we assumed $|R' \cap Q|$ is even, and using that $x \in Q \setminus R$) and thus $|Q \cap \hat{R}|$ odd. Argument II applies. If $x \notin Q$, then in order to have $|Q \cap R|$ odd, we must have $|Q \cap \tilde{R}|$ odd (as we assumed $|R' \cap Q|$ is even, and using that $x \notin Q \cap R$), and therefore $|Q \cap \hat{R}|$ even. Argument III applies.

12. $x \notin R$, $r \in R$ (so $x \neq r$), $|\tilde{R}|$ odd, Y does not contains xr. It does not matter below whether $x \in Q$ or $x \notin Q$. Then $\hat{R} = \tilde{R} \cup \{r\}$, we are in Subcase 2, and $R' = R \setminus \hat{R}$. In order to have $|Q \cap R|$ odd, we must have $|Q \cap \hat{R}|$ odd (as we assumed $|R' \cap Q|$ is even). Argument I applies.

13. $x \notin R$, $r \notin R$, $|\tilde{R}|$ even, Y contains xr (so $x \neq r$). Then $\hat{R} = \tilde{R}$, we are in Subcase 1, and $R' = (R \setminus V(Z)) \cup \{r, x\}$. If $x \in Q$, then in order to have $|Q \cap R|$ odd, we must have $|Q \cap \tilde{R}|$ odd (as we assumed $|R' \cap Q|$ is even, and using that $\{r, x\} \subseteq ((R' \setminus R) \cap Q)$). With $\hat{R} = \tilde{R}$ and $|Q \cap \tilde{R}|$ odd, Argument II applies. If $x \notin Q$, then in order to have $|Q \cap R|$ odd, we must have $|Q \cap \tilde{R}|$ even (as we assumed $|R' \cap Q|$ is even, and using that $r \in (R' \setminus R) \cap Q$ and $x \in (R' \setminus R) \setminus Q$). With $\hat{R} = \tilde{R}$ and $|Q \cap \tilde{R}|$ even and $x \notin Q$, Argument III applies.

14. $x \notin R$, $r \notin R$, $|\tilde{R}|$ even, Y does not contain xr. It does not matter below whether $x \in Q$ ($x = r$ is possible) or $x \notin Q$ (so $x \neq r$). Then $\hat{R} = \tilde{R}$, we are in Subcase 2, and $R' = R \setminus \hat{R}$. In order to have $|Q \cap R|$ odd, we must have $|Q \cap \hat{R}|$ odd (as we assumed $|R' \cap Q|$ is even). Argument I applies.

15. $x \notin R$, $r \notin R$, $|\tilde{R}|$ odd, Y contains xr (so $x \neq r$). Then $\hat{R} = \tilde{R} \cup \{r\}$, we are in Subcase 3, and $R' = (R \setminus V(Z)) \cup \{x\} = \left(R \setminus \tilde{R}\right) \cup \{x\}$. If $x \in Q$, then in order to have $|Q \cap R|$ odd, we must have $|Q \cap \tilde{R}|$ even (as we assumed $|R' \cap Q|$ is even, and using that $x \in ((R' \setminus R) \cap Q)$). Then $|Q \cap \hat{R}|$ is odd (as $r \in Q$), and Argument II applies. If $x \notin Q$, in order to have $|Q \cap R|$ odd, we

must have $|Q \cap \tilde{R}|$ odd, (as we assumed $|R' \cap Q|$ is even, and using that $x \in ((R' \setminus R) \setminus Q)$). Then $|Q \cap \hat{R}|$ is even (as $r \in Q$), and Argument III applies.

16. $x \notin R$, $r \notin R$, $|\tilde{R}|$ odd, Y does not contain xr. It does not matter below whether $x \in Q$ ($x = r$ is possible) or $x \notin Q$ (so $x \neq r$). Then $\hat{R} = \tilde{R} \cup \{r\}$, we are in Subcase 4, and $R' = \left(R \setminus \tilde{R}\right) \cup \{r\}$. To have $|Q \cap R|$ odd, we must have $|Q \cap \tilde{R}|$ even (as we assumed $|R' \cap Q|$ is even, and using that $r \in Q \cap (R' \setminus R)$). Then $|Q \cap \hat{R}|$ is odd, and Argument I applies.

This was the last case of the claim. ■

We resume the proof of Theorem 1 (we are in the third case). We must pay for $w_i(J_i) - w_{i-1}(J_{i-1})$, which is at most $\frac{1}{2}c_i(B_{i-1}) + p_i(\hat{S}_{i-1})$, since $c_i(\hat{Y}^j_{i-1}) \leq \frac{1}{2}c_i(A^j_{i-1})$ for all j. We must also retire debt accumulated by the vertices $v \in V(H_i) \setminus V(H_{i-1})$ (recall that each such vertex has $e_v \in B_{i-1}$), which is at most $(1/8)c_i(B_{i-1})$. Keep in mind that x_{i-1} does not contribute by going in debt (the only way to accumulate debt is the second case); here $c_{i-1}(e_{i-1}) = 0$ and $debt_{i-1}(x_{i-1}) = 0$, maintaining Invariant (5), as indeed for any vertex $v \in V(H_{i-1}) \setminus \{x_{i-1}\}$, $debt_{i-1}(v) = debt_i(v)$ and $c_{i-1}(e_v) = c_i(e_v)$.

The cash in hand is $(7/8)(p_i(H_i) - p_{i-1}(H_{i-1})) = (7/8)(c_i(B_{i-1}) + p(\hat{S}_{i-1}))$, keeping in mind that $c_i(e_{i-1}) = c(e_{i-1}) = p(\hat{S}_{i-1}) = p_i(\hat{S}_{i-1})$ and $c_{i-1}(e_{i-1}) = 0$. Therefore to maintain the debt invariant we need the inequality:

$$\frac{1}{2}c_i(B_{i-1}) + p_i(\hat{S}_{i-1}) + \frac{1}{8}c_i(B_{i-1}) \leq (7/8)\left(c_i(B_{i-1}) + p(\hat{S}_{i-1})\right), \tag{7}$$

which is true since in this (third) case $p_i(\hat{S}_{i-1}) < 2c_i(B_{i-1})$. In other words, Inequality (3) holds.

This is the last case of the recursion, finishing the proof of Theorem 1. ■

4 Conclusions

We proved that for the class of edge-weighted undirected hypergraph that admit a strongly connected orientation, the T-ratio (the supremum, over a class of hypergraphs and all possible R, of the minimum weight T-join divided by the weight of the hypergraph) is at most $7/8$.

A series of examples where the ratio approaches $2/3$ is given in the appendix. The appendix also sketches the following: hypergraphs that admit a strongly connected orientation have a T-ratio of at most $4/5$, and for the class of two-edge-connected hypergraphs, the supremum of T-ratios converges to 1 as the number of hyperedges increases.

References

[1] A. Borchers and D.-Z. Du. The k-Steiner ratio in graphs. *SIAM Journal on Computing*, 26(3):857–869, 1997.

[2] G. Calinescu. Min-power strong connectivity. In M. Serna, K. Jansen, and J. Rolin, editors, *Proceedings of the International Workshop on Approximation Algorithms for Combinatorial Optimization*, number 6302 in Lecture Notes in Computer Science, pages 67–80. Springer, 2010.

[3] W.T. Chen and N.F. Huang. The strongly connecting problem on multihop packet radio networks. *IEEE Transactions on Communications*, 37(3):293–295, 1989.

[4] Jack Edmonds and Ellis L. Johnson. Matching, Euler tours and the Chinese postman. *Mathematical Programming*, 5(1):88–124, 1973.

[5] András Frank, Tamás Király, and Zoltán Király. On the orientation of graphs and hypergraphs. *Discrete Applied Mathematics*, 131(2):385 – 400, 2003.

[6] M. R. Garey and D. S. Johnson. *Computers and Intractability*. W.H. Freeman and Co., NY, 1979.

[7] L. M. Kirousis, E. Kranakis, D. Krizanc, and A. Pelc. Power consumption in packet radio networks. *Theoretical Computer Science*, 243:289–305, 2000.

[8] H.E. Robbins. A theorem on graphs with an application to a problem of traffic control. *Amer. Math. Monthly*, 46:81–83, 1939.

[9] A. Schrijver. *Combinatorial Optimization*. Springer, 2003.

[10] A. Zelikovsky. An 11/6-approximation algorithm for the network Steiner problem. *Algorithmica*, 9:463–470, 1993.

5 Appendix

5.1 Lower bound for the T-ratio

For a series of examples where the T-ratio approaches $2/3$, start (see Figure 5 for an illustration) with a complete binary tree of height h with nodes i, $1 \leq i \leq 2^{h+1} - 1$ (as in a binary heap, the children of node j, with $j < 2^h$, are $2j$ and $2j + 1$).

Replace each node i with nodes y_i, z_i, connected by an edge of cost 0, and, for $i < 2^h$, add edges of cost 2: $z_i y_{2i}$ and $z_i y_{2i+1}$. Call the resulting tree B; assume it is rooted at y_1, and for each y_i, let B_i be (the vertex set of) the subtree of B consisting of y_i and all its descendants. Add vertex u and edge of cost 1 uy_1. Add another 2^h vertices $x_1, \ldots, x_{2^h}$ and edges of cost 1: $x_i z_{i+2^h-1}$ and edges of cost 0: $x_i u$.

For this MIN-POWER STRONG CONNECTIVITY instance, OPT has, for $i = 1, 2, \ldots, 2^h - 1$, $p(z_i) = 2$, for $i = 2^h, \ldots, 2^{h+1} - 1$, $p(z_i) = 1$, and $p(u) = 1$, with all the other vertices having power 0. The total power of this solution is $2 \cdot (2^h - 1) + 2^h + 1 = 3 \cdot 2^h - 1$.

As an aside, we prove that OPT is indeed an optimum feasible solution, as its power is only 2 more than the cost of minimum spanning tree cost, described below, and one has, for every input graph and any feasible solution FS, that given the most costly edge e of the MST (the minimum spanning tree of G), $p(FS) \geq c(MST) + c(e)$, as proved in the remaining of this paragraph. Let U, W be the partition of the vertex set defined by MST after removing e. Since FS is strongly connected, there must exist an arc $a = (u, w)$ in $E(FS)$ with $u \in U$ and $w \in W$ and such e is not the undirected version of a. If $p_{E(FS)}(u) < c(e)$ then $c(a) \leq p_{E(FS)}(u) < c(e)$, so the tree $T' = (MST \setminus e) \cup a$ is cheaper than MST, a contradiction. Hence $p_{E(FS)}(u) \geq c(e)$. Using u as the root of a spanning

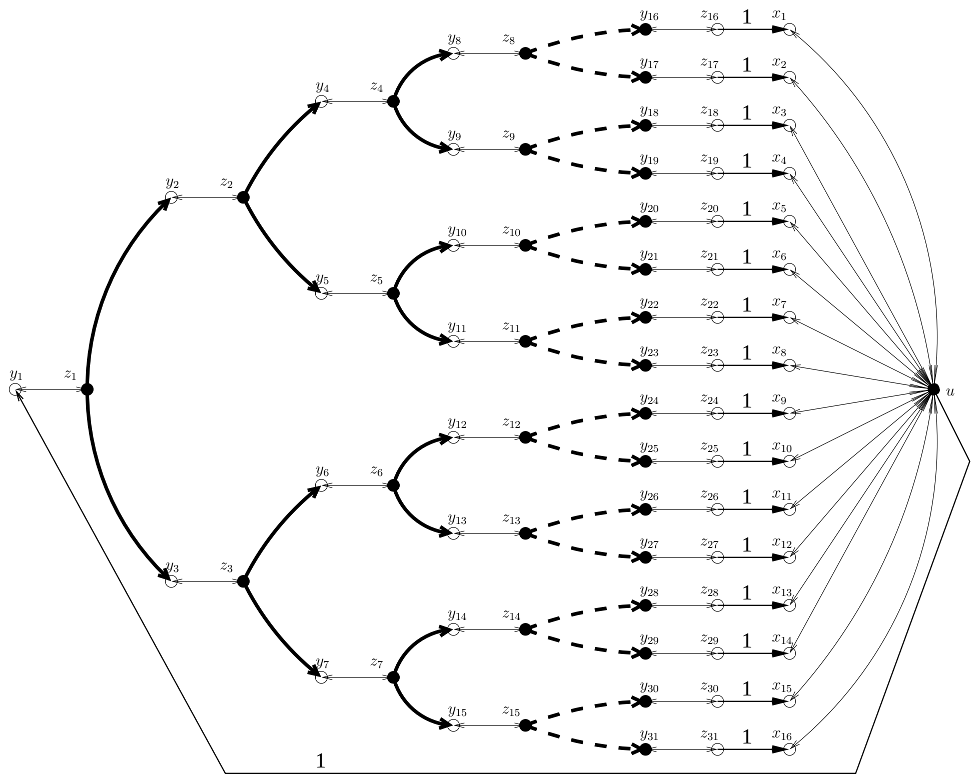

Figure 5: Thinnest edges have cost 0, medium thick have their cost written: 1, and thickest edges have cost 2. Arrows indicate the optimum power assignment solution. Solid edges give the minimum spanning tree, and its vertices of odd degree are solid and form the set R. Here $h = 4$.

in-arborescence inside FS, we obtain that for all $v \in V \setminus \{u\}$, $p_{FS}(v)$ is at least the cost of the arc connecting v to its parent in this incoming arborescence, whose total cost is at least the cost of an undirected minimum spanning tree of G. Thus $\sum_{v\in V\setminus\{u\}} p_{E(FS)}(v) \geq c(MST)$.

One minimum spanning tree T includes all the edges of cost 1 and 0, as well as the edges of cost 2 except the last level of the complete binary tree, that is T contains the edges $z_i y_{2i}$ and $z_i y_{2i+1}$, for $i < 2^{h-1}$. We choose R to be the set of vertices of odd degree of the MST (this set was important in MIN-POWER STRONG CONNECTIVITY algorithms). Here, R consists of u, z_i, for $i < 2^h$, and y_i, for $2^h \leq i < 2^{h+1}$.

Now we show that every T-join M for R has weight at least $2(2^h - 1)$. Assume that for some i $1 \leq i < 2^h$, M does not use the star with power 2 rooted at z_i (or else, we are done). Look at the subtrees $B = B_{2i}$ and $B' = B_{2i+1}$, each having an odd number of vertices of R. Now apply to B the following "pruning" procedure: if some $z_j \in B$ and $j < 2^h$ also is such that M does not use the star with power 2 rooted at z_j, then remove from B the vertices of $(B_j \setminus \{y_j, z_j\})$; note that B continues to have an odd number of vertices of R, since B_j has an odd number of vertices of R, and therefore $B_j \setminus \{y_j, z_j\}$ had an even number of vertices of R. After doing this for all possible j, B still has an odd number of vertices of R and thus a hyperedge of K must have an endpoint in B and one outside - this cannot be the star rooted at z_i or some pruned z_j (it must be the edge $z_k x_{k-2^h+1}$ for some k with

z_k descendant of i in B), and this hyperedge must have weight at least 1. Similarly, after pruning, another hyperedge of weight at least 1 is obtained crossing B'; associate these two hyperedges to z_i. Notice that for $j \neq i$, we cannot associate the same hyperedge to both z_i and z_j since such an (hyper)edge $z_k x_{k-2^h+1}$ will have z_k as descendant in B of both z_i and z_j - but then pruning will make sure that the higher (in B) of z_i and z_j cannot use $z_k x_{k-2^h+1}$.

Thus whenever M does not use the star with power 2 rooted at z_i, for some $1 \leq i < 2^h$, it must use two (hyper)edges of weight 1, not shared with another i. We conclude that indeed $w(M) \geq 2(2^h - 1) \geq (2/3 - \epsilon)opt$. Note that this $2/3$ lower bound holds for the T-ratio in hypergraphs that admit strongly connected orientations.

5.2 Upper bound on the T-ratio

Theorem 2 *There exists a collection of stars $\mathcal{B}$ with $f(\mathcal{B}) = c(T)$ and $w(\mathcal{B}) \leq (4/5)opt$, where opt is the power of the optimum solution.*

Proof sketch. The proof is as in Theorem 1 before the charging/accounting scheme. However, we allow debt $debt_i(v) \leq (1/5)c_i(e_v)$ instead of $(1/8)c_i(e_v)$, and we recourse in a similar but more complicated way.

The base case needs to pay $(1/2)c_1(H_1) + (1/5)c_1(H_1)$ for the T-join J_1 and retiring the debt of all vertices, using cash of $(4/5)p_1(H_1)$, which is enough.

For the recursion, as before, we have H_i, and follow the third case of the proof of Theorem 1. We construct Z^j_{i-1} as there, but then instead of settling for one Y^j_{i-1} of weight at most $\frac{1}{2}w(Z^j_{i-1})$, find (next paragraph) two T-joins $\tilde{Y}^j_{i-1}$ and $\bar{Y}^j_{i-1}$ such that $\tilde{Y}^j_{i-1}$ contains $r^j_{i-1}x_{i-1}$ (assuming this edge exists, i.e. $r^j_{i-1} \neq x_{i-1}$) and $\bar{Y}^j_{i-1}$ does not contain $r^j_{i-1}x_{i-1}$ and such that $w\left(\bar{Y}^j_{i-1}\right) + w\left(\tilde{Y}^j_{i-1}\right) \leq w\left(Z^j_{i-1}\right)$; this is indeed possible as argued below.

If the edge $r^j_{i-1}x_{i-1}$ does not exist (that is, if $r^j_{i-1} = x_{i-1}$) then set $\bar{Y}^j_{i-1} = \tilde{Y}^j_{i-1} = Y^j_{i-1}$, where Y^j_{i-1} comes from the proof of Theorem 1. Otherwise, do an ear decomposition of Z^j_{i-1} with the first cycle containing the edge $r^j_{i-1}x_{i-1}$. For every ear other than the first cycle, traverse it changing sides each time you meet a vertex of $\hat{R}^j_{i-1}$ - then pick the cheapest of the two edge sets. Set up recursion R - like in the Theorem 1, but simpler; we pay half of the cost reduction when we recourse. Finally, in the last cycle, partition it into two sets of edges as in the base case of Theorem 1, making sure the edge $r^j_{i-1}x_{i-1}$ is in $\tilde{Y}^j_{i-1}$ and not $\bar{Y}^j_{i-1}$.

Let $B' = \cup_j \bar{Y}^j_{i-1}$ and $B'' = \tilde{Y}^j_{i-1}$; thus $w(B') + w(B'') \leq c_i(B_i)$. Edges of B' and B'' come from either arcs of B_{i-1} or arcs of $\tilde{S}_{i-1}$ or are of type $r^j_{i-1}x_{i-1}$ for some j, with all edges of this later type in B''. Let $\bar{B}'$ be the arcs of B_{i-1} which give rise to edges of B', and $\bar{B}''$ be the arcs of B_{i-1} which give rise to edges of B''. We do not have that $\bar{B}''$ and $\bar{B}'$ are disjoint but we do have

$$c_i(\bar{B}') + c_i(\bar{B}'') \leq c_i(B_{i-1}). \tag{8}$$

In a first case, $p_i(\hat{S}_{i-1}) \leq c_i(\bar{B}'')$. Then we proceed as in the third case of Theorem 1. The cash in hand is $(4/5)\left(p_i(\hat{S}_{i-1}) + c_i(B_{i-1})\right)$. We use it to pay $p_i(\hat{S}_{i-1}) + \min\left(c_i(\bar{B}'), c_i(\bar{B}'')\right)$, the cost of upgrading J_{i-1} to J_i, and another $(1/5)c_i(B_{i-1})$ to pay for retiring the debt of vertices in

$V(H_i) \setminus v(H_{i-1})$. Thus to maintain the credit invariant it will be enough if

$$\frac{3}{5}c_i(B_{i-1}) \geq \frac{1}{5}c_i(\bar{B}'') + \min\left(c_i(\bar{B}'), c_i(\bar{B}'')\right), \tag{9}$$

where we used $p_i(\hat{S}_{i-1}) \leq c_i(\bar{B}'')$. Then, if $c_i(B'') \leq c_i(\bar{B}')$, then the inequality above becomes $\frac{3}{5}c_i(B_{i-1}) \geq \frac{1}{5}c_i(\bar{B}'') + c_i(\bar{B}'')$, which is indeed true in this subcase, using Inequation (8). If $c_i(\bar{B}'') > c_i(\bar{B}')$, then Inequality (9) becomes $\frac{3}{5}c_i(B_{i-1}) \geq \frac{1}{5}c_i(\bar{B}'') + c_i(\bar{B}')$, which is, using Inequation (8), true in this second subcase.

So from now we assume $p_i(\hat{S}_{i-1}) > c_i(B'')$. Also, if

$$\frac{3}{5}c_i(B_{i-1}) \geq \frac{1}{5}p_i(\hat{S}_{i-1}) + c_i(B'') \tag{10}$$

then we proceed as above, and the credit invariant is maintained.

So from now on, Inequality (10) does not hold, and $p_i(\hat{S}_{i-1}) > c_i(\bar{B}'')$. Set $c_{i-1}(e_{i-1}) = c_i(e_{i-1}) - c_i(B'')$; recall that $c_i(e_{i-1}) = p_i(\hat{S}_{i-1})$. Set R_{i-1} as in the third case in the proof of Theorem 1 using for each j, $\bar{Y}^j_{i-1}$ instead of Y^j_{i-1}. We are either in Subcase 2 (with R^j_{i-1} even-sized) or Subcase 4 (when on can check that r^j_{i-1} is in the final R_{i-1}). It is important to observe that R_{i-1} is reset the same way as in the second case of the proof of Theorem 1. We recourse in H_{i-1} with cost c_{i-1}, obtaining T-join J_{i-1} in K_{i-1} for R_{i-1}. If J_{i-1} does not contain e_{i-1}, we set $J_i = J_{i-1} \cup B_{i-1}$, which is indeed a T-join in K_i for R_i as argued in the second case of the proof of Theorem 1. Otherwise, J_{i-1} contains e_{i-1}, we set $J_i = J_{i-1} \setminus \{e_{i-1}\} \cup \{\hat{S}_{i-1}\} \cup \bar{B}'$, which is indeed a T-join in K_i for R_i as argued in the third case (Subcases 2 and 4, see Claim 1) of the proof of Theorem 1. Note that in the end, all the arcs selected at artificial (reduced by the procedure) cost are removed and replaced by a bigger star/hyperedge, with its original cost.

In both subcases, we have:

$$w_i(J_i) - w_{i-1}(J_{i-1}) \leq c_i(B_{i-1}), \tag{11}$$

using in the second subcase that $c_{i-1}(e_{i-1}) = c_i(e_{i-1}) - c_i(\bar{B}'') = p_i(\hat{S}_{i-1}) - c_i(\bar{B}'')$ and Inequality (8).

Thus we need to pay at most $(6/5)c_i(B_{i-1})$ for the operation, including retiring the debt of the vertices of $V(H_i) \setminus V(H_{i-1})$. The cash in hand is $(4/5)\left(p_i(H_i) - p_{i-1}(H_{i-1})\right) = (4/5)\left(c_i(B_{i-1}) + c_i(\bar{B}'')\right)$. In addition, we put a debt of $(1/5)c_{i-1}(e_{i-1})$ on x_{i-1} (previously, debt-free). Thus, to maintain the credit invariant, it is enough that

$$\frac{6}{5}c_i(B_{i-1}) \leq \frac{4}{5}\left(c_i(B_{i-1}) + c_i(\bar{B}'')\right) + \frac{1}{5}\left(p_i(\hat{S}_{i-1}) - c_i(\bar{B}'')\right). \tag{12}$$

This is equivalent to

$$\frac{2}{5}c_i(B_{i-1}) \leq \frac{1}{5}p_i(\hat{S}_{i-1}) + \frac{3}{5}c_i(\bar{B}''). \tag{13}$$

Since Equation 10 does not hold in this subcase, we obtain:

$$\begin{aligned}
\frac{2}{5}c_i(B_{i-1}) &= \frac{2}{3}\cdot\frac{3}{5}c_i(B_{i-1}) \\
&< \frac{2}{3}\left(\frac{1}{5}p_i(\hat{S}_{i-1}) + c_i(\bar{B}'')\right) \\
&\leq \frac{2}{15}p_i(\hat{S}_{i-1}) + \frac{2}{3}c_i(\bar{B}'') \\
&< \frac{2}{15}p_i(\hat{S}_{i-1}) + \frac{2}{3}c_i(\bar{B}'') + \frac{1}{15}p_i(\hat{S}_{i-1}) - \frac{1}{15}c_i(\bar{B}'') \\
&= \frac{1}{5}p_i(\hat{S}_{i-1}) + \frac{3}{5}c_i(\bar{B}''),
\end{aligned}$$

with the last inequality holding since we are in the case $p_i(\hat{S}_{i-1}) > c_i(\bar{B}'')$. Thus there is enough cash to maintain the credit invariant and pay for the operation. ■

5.3 T-ratio in two-edge-connected hypergraphs

For the class of two-edge-connected hypergraphs, the supremum of T-ratios converges to 1 as the number of hyperedges increases, as we see in the following series of examples. For integer k multiple of 8, have $\binom{k}{2}$ vertices u_{ij}, where $1 \leq i < j \leq k$. The k hyperedges are $e_1, e_2, \ldots, e_k$ (all with weight 1), and e_i contains, for all j with $1 \leq j < i$, u_{ji}, and for all j with $i < j \leq k$, u_{ij}. This hypergraph is two-edge-connected: two edge-disjoint paths connecting u_{12} and u_{34} are $u_{12}, e_1, u_{13}, e_3, u_{34}$ and $u_{12}, e_2, u_{24}, e_4, u_{34}$, two edge-disjoint paths connecting u_{12} and u_{13} are u_{12}, e_1, u_{13} and $u_{12}, e_2, u_{23}, e_3, u_{13}$, with all the other pairs of vertices being connected in a cases symmetric to one of these two cases. With R given by V, missing any two hyperedges (say, e_i and e_j with $i < j$) results in an isolated vertex (u_{ij}), and then the T-cut with this vertex on one side is not crossed; thus any T-join has size/weight at least $k - 1$.