Construction of Minimum m-Time Relaxed Broadcast Hypergraphs I: $m = 1$

A. Averbuch * L. Kosachevski † Y. Roditty ‡

Abstract

Let $B_{k,m}(n)$ be the number of hyperedges in a $k-uniform$ sparsest possible hypergraph on n vertices in which broadcasting from any vertex can be accomplished in at most m additional time units to the optimal possible time. In this paper we extend our former result (see [1]) by prividing an upper bound to for $B_{k,m}(n)$ for $m = 1$. All results seem to be almost best possible.

1 Introduction

Let $B_{k,m}(n)$ be the number of hyperedges in a $k - uniform$ sparsest possible hypergraph on n vertices in which broadcasting from any vertex can be accomplished in at most $t_k(n) + m$ time units where $t_k(n)$ is defined in Lemma 1.7. In this paper we construct three different hypergraph based network models which achieve an upper bound for $B_{k,m}(n)$ for $m = 1$. The case $m > 1$ is analyzed in [2] and the broadcast models that provide an upper bound for $B_{k,m}(n)$ for $m > 1$ are extensions of the broadcast models that are constructed in this paper.

We begin with recalling the definition of the notation of the hypergraph on which the broadcasting models build in the next sections are based:

*School of Computer Science, Tel Aviv University, Ramat Aviv, Tel Aviv 69978 Israel

†Department of Computer Science, Bar-Ilan University, Ramat-Gan 52900, Israel

‡School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel and Department of Computer Science, The Academic College of Tel-Aviv-Yaffo, Tel-Aviv 61161, Israel. email: jr@post.tau.ac.il

Definition 1.1 *Let V be a set of n vertices, and let $E \subseteq V^k$ be a k-relation on V. We say that the pair (V, E) is a k-uniform hypergraph $HG = HG(V, E)$. A subset $U = \{u_1, ..., u_k\} \subseteq V$ is called a hyperedge in $HG(V, E)$ if there exists an element $< u_{\sigma(1)}, ..., u_{\sigma(k)} > \in E$ for some permutation σ. We will denote a hyperedge by $< u_1, ..., u_k >$ disregarding the permutation σ.*

We outline now some essential definitions we used in [1], which will help us to construct and describe broadcasting models, that are based on hypergraph.

Definition 1.2 *The degree of a vertex v (denoted $d(v)$) is the number of hyperedges that v participates in.*

Definition 1.3 *A hyperedge $e =< v_1, \cdots, v_k >$ of HG is called a leaf-hyperedge if only one of its vertices participates in another hyperedges.*

Definition 1.4 *A vertex v is called a leaf-vertex of the hyperedge e if e is a leaf-hyperedge and $d(v) = 1$ (i.e. participates only in e). The only vertex in e, which is connected to another hyperedge, is called the base-vertex of this hyperedge.*

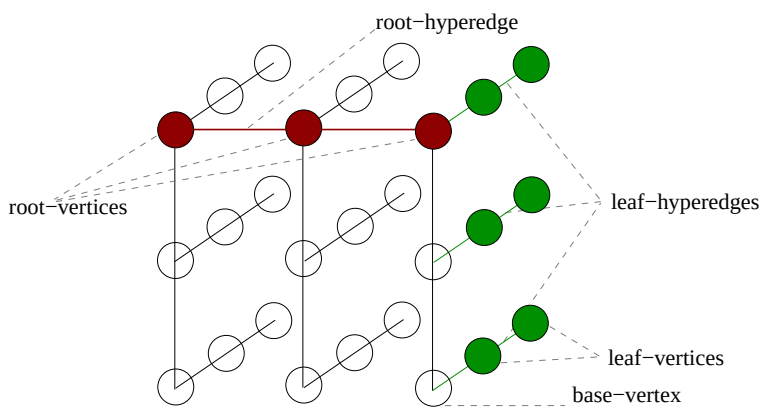


Figure 1: Illustrations of hypergraph definitions.

Definition 1.5 *Let $v_i, v_j \in V$ be two vertices in a hypergraph $HG(V, E)$ and let $E' = \{e_1, \cdots, e_m\} \subset E$ be a subset of hyperedges such that:*

1. *$e_l \cap e_{l+1} \neq \emptyset$ for $1 \leq l \leq m - 1$*

2. $v_i \in e_1$

3. $v_j \in e_m$

Then we call E' *a route between* v_i *and* v_j *with* m *as its length.*

This definition allows us to establish a distance notion between every two vertices of a hypergraph.

Definition 1.6 *The distance* $d(v, u)$ *between two vertices* $v, u \in V$ *is the length of the shortest route between them. The diameter of the hypergraph is denoted by*

$$diam(HG) \triangleq max\{d(u, v) \mid \forall v, u \in V\}.$$

Now we recall the basic rules of broadcasting in hypergraphs. Let $HG(V, E)$ be a k-uniform connected undirected finite hypergraph on n vertices. The hypergraph represents the communication network where the vertices are considered as processors and each hyperedge which connects k vertices assumed to be a direct communication link between these vertices. The *originators* are vertices that produce a mutual message that should be transmitted to all other vertices of the hypergraph according to the following rules:

1. Each conference call involves exactly k vertices.
2. Each call requires exactly one discrete time unit.
3. A vertex can participate in only one conference call per time unit.
4. A vertex can be idle at a specific time unit.

We deal only with a message that was originated in a single vertex. Let $t_k(n)$ be the time required to broadcast a message from one vertex in a complete k-uniform hypergraph on n vertices where conference calls within k vertices are allowed.

The following lemma, which was proved in [20], indicates the effect of the requirement for each call to include exactly k nodes. A tight estimation of $t_k(n)$ was proved in [20] to be

Lemma 1.7 *Given an integer $k \geq 2$. Let $n = k^i + p(k-1) + q$ with $1 \leq p < k^i$, and $r = k^i - p$. Then,*

$$t_k(n) = \begin{cases} \lceil log_k n \rceil, & max\{0, k-r\} \leq q < k \\ \lceil log_k n \rceil + 1, & 1 \leq q < k-r \end{cases}$$

As in general graphs, we generalized in [1] the notion of minimum broadcast graph, which appeared in [11], [12] and in [14] to k-uniform hypergraphs. Therefore, let $HG(V, E)$ be a k-uniform hypergraph with $|V| = n$ vertices.

Definition 1.8 *The broadcasting time $b(v)$ of $v \in V$ is the minimum time required to broadcast one message from v to all the other vertices in V. The broadcasting time of $HG(V, E)$ is $b(HG) \triangleq max\{b(v) \mid v \in V\}$.*

Denote by $b(n)$ the minimum message broadcasting time $b(HG)$ over all hypergraphs HG with n vertices.

Definition 1.9 *A hypergraph $HG(V, E)$ is said to be minimum broadcast hypergraph (MBH) if $b(HG) = b(n)$.*

In the next definition we extend the definitions from [14, 23, 24].

Definition 1.10 *Let $B_{k,m}(n)$ be the number of hyperedges in a $k-uniform$ sparsest possible hypergraph on n vertices in which broadcasting from any originator can be accomplished in at most $t_k(n) + m$ time units.*

A k-uniform hypergraph with broadcast time of $t_k(n) + m$ is called m-relaxed broadcast hypergraph ($m - RBH$), and a $m - RBH$ with $B_{k,m}(n)$ edges is called m-relaxed minimum broadcast hypergraph ($m - RMBH$).

The case $B_{k,0}(n)$ was dealt in [1].

2 k-nomial hypergraph tree

In this section we construct a hypergraph based network topology called k-nomial hypergraph tree, which provides an upper bound for $B_{k,m}(n)$ for $m = 1$. We will prove the following theorem:

Theorem 2.1 *Let $HG(V,E)$ be a k-uniform hypergraph on $n = k^t - p$ vertices, where $0 \le p < k^t - k^{t-1}$. Then,*

$$B_{k,1}(n) \le \frac{k^t-1}{k-1} + \frac{k^t-k}{k-1} - (t-1)k - \left(\left\lfloor \frac{p}{k-1} \right\rfloor + \left\lceil \frac{p}{k-1} \right\rceil - \left\lceil \frac{p}{k^{t-2}(k-1)} \right\rceil \right)$$

The proof is based upon the notion of the k-nomial hypergraph tree, which was described in details in [1]. However, in order to keep this paper almost self contained we give only a brief description of its properties.

The k-nomial hypergraph tree on $n = k^t$ vertices, denoted as kHT_{k^t}, has a unique root-hyperedge defined as $e_1 =< v_{0_1}, v_{0_2}, \cdots, v_{0_k} >$. The k vertices $v_{0_1}, v_{0_2}, \cdots, v_{0_k}$ are symmetric in terms of the construction, and are called root-vertices.

An important property of root-vertices is that their degrees are $d(v_{0_1}) = d(v_{0_2}) = \cdots = d(v_{0_k}) = t$. Based on this fact, it was shown and proved in [1] that the maximal degree of a vertex in a kHT_{k^t} is t, and the root-vertices are the only vertices that realize this value.

Another important property of the k-nomial hypergraph tree is presented by the following corollary:

Corollary 2.2 *Let $\{v_{0_1}, v_{0_2}, \cdots, v_{0_k}\}$ be root-vertices. Then, $b(v_{0_1}) = b(v_{0_2}) = \cdots = b(v_{0_k}) = t$ and for any $u \in V(kHT_{k^t})$, $u \notin \{v_{0_1}, v_{0_2}, \cdots, v_{0_k}\}$, $b(u) \ge t+1$.*

2.1 Construction of the broadcasting model

Now we outline the construction of a kHT based 1-relaxed broadcast model. Since we want to get an upper bound on the minimum number of hyperedges needed in a given time for any n, we need to show how to obtain from kHT_{k^t} a reduced kHT_n, which has only $k^{t-1} < n \le k^t$ vertices.

During the construction process we first create a complete k-uniform k-nomial hypergraph tree, then, the vertices are eliminated until we reach the desired n. This construction was described in [1]. Now our goal is to construct a 1-relaxed broadcast model. This is done by connecting an additional hyperedges in the way that is presented below.

Let $V_t' \subseteq V(kHT_{k^t})$ where $V_t' \triangleq \{v \in V \mid min\{d(v, v_{0_1}), d(v, v_{0_2}), \cdots, d(v, v_{0_k})\} \ge 2\}$.

In other words, V_t' contains all the vertices in kHT_{k^t} whose minimum route length (see definition 1.5) to some root-vertex is greater than one. Therefore, if $v \in V_t'$ originates the message, v will not be able to transmit it within one time unit to any of the root-vertices. For this reason, we have to add a new hyperedge, which connects every $k-1$ vertices from V_t' to one of the root-vertices. We will always connect these additional hyperedges to v_{0_1}. The reason for this is given in Proposition 2.3. Figure 2 illustrates how these additional hyperedges are added to the hypergraph.

Proposition 2.3 $(k-1)\big||V_t'|$.

Proof: This is a direct consequence from the construction. First, we consider a complete kHT_{k^t} and we prove the proposition by induction on t. For $t=1$ this is trivial. Assume now it is true for $V_{t-1}' \subseteq kHT_{k^{t-1}}$. Let A_t be the number of vertices added to $kHT_{k^{t-1}}$ at stage t in the construction of kHT_{k^t}. This is done by adding a leaf-hyperedge that has $k-1$ new vertices to each vertex in $kHT_{k^{t-1}}$. Hence, $(k-1)\big|A_t$. If one of the $k-1$ vertices of some new hyperedge is at a distance of at least 2 from the root-vertices, then all other $k-2$ new vertices of this hyperedge will be also at distance of at least 2 from all root-vertices. Let $A_t' \triangleq \{v \in A_t \mid min\{d(v,v_{0_1}), d(v,v_{0_2}), \cdots, d(v,v_{0_k})\} \geq 2\}$. $A_t' \subseteq A_t$ and it is also divisible by $(k-1)$. Since $V_{t-1}' \cap A_t' = \emptyset$, $(k-1)\big|V_{t-1}'$ and $(k-1)\big|A_t'$, $V_t' = V_{t-1}' \cup A_t'$ is also divisible by $(k-1)$.

Now we show that if we eliminate the vertices from the reduced BHT we still have that $|V_t'|$ is divisible by $k-1$. This is true since we eliminate all leaf-vertices from one hyperedge before starting to eliminate leaf-vertices from another hyperedge. And the rest of the leaf-vertices that are left in the hypergraph tree are connected directly to the root-vertex v_{0_1}. This means that if we eliminate a vertex from a hyperedge, all remaining leaf-vertices of this hyperedge will be connected to the root-vertex by the tail-hyperedge. Therefore, we subtract from the number of vertices in V_t' a number which is divisible by $k-1$. ■

In the sequel V' will denote the set V_t'.

As a consequence of proposition 2.3 we can add exactly $\frac{|V'|}{k-1}$ new edges which connect all the vertices in V' with v_{0_1}, as was described. Denote the additional set of hyperedges by E'.

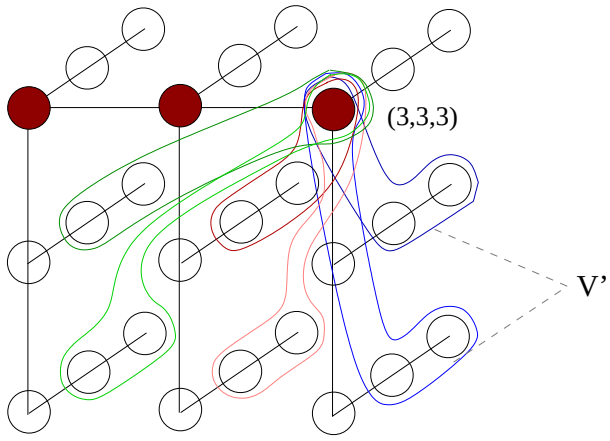


Figure 2: kHT_{3^3} with additional hyperedges, which connect vertices from V' to v_{0_1}.

Definition 2.4 *kHT_n with additional hyperedges, constructed as described above, is called $1 - kHT_n$.*

Now we show how the broadcasting algorithm operates.

2.2 The broadcasting algorithm

In this section we demonstrate how the broadcasting algorithm operates on the kHT based broadcast model. We distinguish between two cases:

Case i: The originator v is a root-vertex In this case the broadcasting is done as was described in [1], at the 1st time unit the root-vertex transmits the message to all other root-vertices and at the next $t-2$ time units the root-vertices broadcast the message to their subtrees and the tail hyperedge, if exists. The broadcasting concludes within at most $t_k(n)$ time units.

Case ii: The originator v is not a root-vertex

Since in the hypergraph, described above, every non root-vertex is at distance 1 from one of the root-vertices, then at the 1st time unit the originator v sends the message to the root-vertex it is connected to.

Then, from time $j = 2$ the broadcasting algorithm continues as in Case *i*.

Theorem 2.5 $1 - kHT_n$ *is* 1*-relaxed broadcast hypergraph.*

Proof: We have to prove that the broadcasting can be completed from any vertex in this hypergraph in at most $t_k(n) + 1$ time units. Therefore, it is sufficient to prove that we can broadcast a message from one of the root-vertices kHT_n in $t_k(n)$ as was computed in Lemma 1.7.

We prove that the cases where our broadcasting algorithm needs $t + 1$ time units to broadcast the message from a root-vertex to the the rest of the hypergraph are exactly the same as the cases when in Lemma 1.7 $t_k(n) = \lceil log_k n \rceil + 1$ (note that $\lceil log_k n \rceil = t$).

Suppose that instead of the reduced kHT_n we have a complete hypergraph. Obviously, the number of time units needed to complete broadcasting in a complete hypergraph is less than or equal to the number of time units needed in any sub-hypergraph of it. Then, according to the proof of Lemma 1.7 at the first $t - 1$ time units, every vertex that was informed of the message should transmit the message to a new hyperedge. So, we may as well assume that this was done on hyperedges that exist in our kHT_n. Therefore, we have k^{t-1} vertices that know the message after $t - 1$ time units. Denote by p the number of vertices which are the base vertices of a leaf-hyperedge that its members are still uninformed of the message. The number of the rest of the uninformed vertices will be denoted by q. Therefore, we have $n = k^{t-1} + p(k-1) + q$. The size of V'' is exactly $r = k^{t-1} - p$ because these are the vertices that have no uninformed leaf-hyperedge left to broadcast the message to. Now if $1 \leq q < k - r$ then according to Lemma 1.7 we will need two more time units to complete the broadcasting. This is exactly when the tail-hyperedge must contain at least one vertex which is not in V''. On the contrary, if $max\{0, k-r\} \leq q < k$ then the tail-hyperedge, if exists, contains only vertices from V''. Hence, broadcasting in the kHT_n can be performed within t time units, as in Lemma 1.7. This completes the proof. ■

Now, that we have a $1 - RBH$ topology we have to examine how many hyperedges are needed in order to obtain an upper bound for $B_{k,1}(n)$. This is resolved in section 2.3.

2.3 Bound for $B_{k,1}(n)$

In this section we compute how many hyperedges are needed in $1 - kHT_n$. Then, recall that the number of hyperedges in kHT_n is

$$E_{kHT_n} = \frac{k^t - 1}{k - 1} - \left\lfloor \frac{p}{k-1} \right\rfloor.$$

Now, we calculate the number of additional hyperedges from V' to the root-vertex v_{0_1}. As was already mentioned, according to Proposition 2.3 $|E'| = \frac{|V'|}{k-1}$. Hence, we have to calculate the number of vertices in V'. Since, V' consists of all the vertices in kHT_{k^t} that are located at distance of at least 2 from one of the root-vertices, $V' = k^t - l_1 - l_2$ where $l_1 = k$ is the number of vertices in the root-hyperedge, and l_2 which is the number of vertices at distance 1 from the root-hyperedge.

Lemma 2.6 $l_2 = (t-1)k(k-1)$.

Proof: Vertices at distance 1 from a root-vertex must be connected by a hyperedge to one of the root-vertices. Therefore, we count how many hyperedges are connected to a root-vertex. At each step of the construction, beside the first step in which all vertices are the root-vertices, we add to each root-vertex a new hyperedge, thus we gain k new hyperedges each consists of $k-1$ new vertices. Besides, no vertex that was at distance greater than 1 is connected to any of the root-vertices. So, $l_2 = (t-1)k(k-1)$. ■

Corollary 2.7 *The number of hyperedges added in order to make a kHT_n into $1 - kHT_n$ is $\frac{k^t-k}{k-1} - (t-1)k$.*

Proposition 2.8 *When deleting p vertices from the entire $1 - kHT_n$ we obtain that the number of eliminated hyperedges from V' to the root-vertices is $\left\lceil \frac{p}{k-1} \right\rceil - \left\lceil \frac{p}{k^{t-2}(k-1)} \right\rceil$.*

Proof: Let us assume that every leaf-vertex is connected by an additional hyperedge to a root-vertex. When we begin to eliminate vertices from a leaf-hyperedge, we replace it by a tail-hyperedge. This means that all the vertices in this hyperedge are now at distance 1 from v_{0_1} which is a root-vertex. Therefore, we can eliminate the additional hyperedge that was added. Hence, we eliminate exactly $\lceil \frac{p}{k-1} \rceil$.

However, this is not exactly the case that we are interested in, since we have some leaf-hyperedges that their base vertex is a root-vertex. In this case we did not add additional hyperedges to connect them to a root-vertex. Hence, we subtract the number of such eliminated leaf-hyperedges from the number given above.

The number of leaf-vertices in a kHT_{k^t} is $k^{t-1}(k-1)$. The special case treated here happens every time we eliminate a hyperedge connected to a root vertex. Note, that the number of leaf-vertices in each sub $kHT_{k^{t-1}}$ of kHT_{k^t} that is connected to one of the root-vertices is $k^{t-2}(k-1)$. Since all the leaf-vertices in a subtree $kHT_{k^{t-1}}$ rooted in one root-vertex are eliminated before we begin to eliminate leaf-vertices from another subtree kHT_{k^t}, then the number of such hyperedges is $\left\lceil \frac{p}{k^{t-2}(k-1)} \right\rceil$.

Therefore, the total number of deleted hyperedges when we eliminate p vertices is $\lceil \frac{p}{k-1} \rceil - \lceil \frac{p}{k^{t-2}(k-1)} \rceil$. ■

Proof of theorem 2.1: By combining the number of hyperedges edges in a complete kHT_{k^t}, and the number of added hyperedges in order to get a 1-relaxed broadcast model (from 2.7) with the number of removed hyperedges (from 2.8) the proof of theorem 2.1 is completed. ■

3 Hypergraph cube

Hypergraph cube is the optimal minimum broadcast model based on hypergraph topologies, as was shown in [1]. The hypergraph cube, denoted by HQ is similar in concept to the t-dimensional cube graph. We obtain now an upper bound for $B_{k,1}(n)$ using this hypergraph topology.

Moreover, the hypergraph cube, as in graphs, satisfies special symmetry properties, which are important in establishing broadcasting schemes.

In this section we establish upper bound on $B_{k,1}(n)$.

3.1 k-hypergraph cube topology definition and construction

Hypergraph cube topology is based on a generalization of the t-dimensional cube. The t-dimensional cube graph topology is constructed inductively in a similar way to the cube graph (a detailed description see [1]). The hypergraph

created, based on this generalization, is called a k-Hypergraph Cube on k^t vertices and denoted by HQ_{k^t}. Figure 3 presents an HQ_{3^3}.

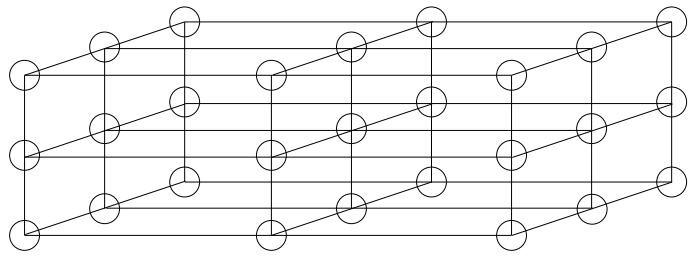


Figure 3: 3-Hypergraph Cube HQ_{3^3}.

3.1.1 Construction of the broadcast model based on hypergraph cube

In this section we describe the construction of the broadcast model which is based on the k-hypergraph cube. This paper extends the results that were obtained in [1].

Our main construction is based upon contracting vertices in the upper graph. Suppose that we want to remove $p = k^t - n$ vertices from a complete HQ_{k^t}. First, we contract $\lceil \frac{p}{k-1} \rceil$ sets of k vertices each into one vertex eliminating their common hyperedge. Therefore, after the completion of the contraction process we will have only $\lceil \frac{p}{k-1} \rceil$ vertices instead of p.

If there are hyperedges that connect the same vertices when contracting the vertices, then we merge these hyperedges into one. Figure 4 illustrates the reduction case when there are contracted vertices.

During the process of contraction described above we may eliminate more vertices than p, because we transform $\lceil \frac{p}{k-1} \rceil$ sets of k vertices into one vertex. If we eliminated $p \leq \lceil \frac{p}{k-1} \rceil (k-1)$ we need to restore $\lceil \frac{p}{k-1} \rceil (k-1) - p$ vertices back to the hypergraph. Since $0 \leq \lceil \frac{p}{k-1} \rceil (k-1) - p < k$ we can connect these *restored vertices* to the hypergraph using a hyperedge which is called a *tail-hyperedge*. Figure 4 illustrates the reduction case when there are restored vertices and a tail-hyperedge.

Definition 3.1 *The hypergraph obtained by the above process is called the* **reduced** *hypergraph cube and denoted HQ_n where n is the number of vertices in the hypergraph cube.*

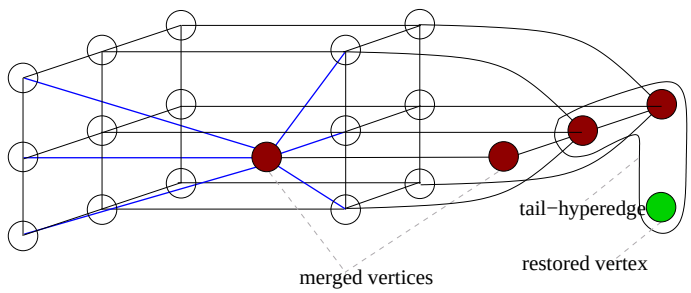


Figure 4: After elimination of 7 vertices from HQ_{3^3} (see Fig. 3) we get HQ_n where $n = 20$.

3.2 The broadcasting algorithm in HQ_n

In order to show the broadcasting scheme in HQ_n we distinguish between two cases:

Case i: The originator v is not one of the restored vertices

This means that v does not belong to a tail-hyperedge. The following proposition was presented and proved in [1]:

Proposition 3.2 *Let HQ_n be a hypergraph on $n = k^t - p$ vertices, where $(k-1) \mid p$. Then, $b(HQ_n) \leq t$.*

According to this proposition we can use the first t time units to broadcast the message to all the vertices in HQ_n. Then, by using one additional time unit we inform the restored vertices, if there are any, by using the tail-hyperedge.

Case ii: The originator v is one of the restored vertices

This means that v participates in the tail-hyperedge. We use the tail-hyperedge in the first time unit to inform at least one of the vertices in the reduced HQ_n, and then according to Proposition 3.2 we inform the rest of the vertices within additional t time units.

Theorem 3.3 *Let HQ_n be as described above. Then, it is a 1-relaxed broadcast hypergraph, denoted by $1 - HQ_n$.*

Proof: We show that if n is the number of vertices in a hypergraph and $k^{t-1} < n \leq k^t$, then, the time needed to complete a broadcasting of the message in such a hypergraph topology is at most $t+1$ time units. Since $t+1 \leq t_k(n)+1$ (according to Lemma 1.7) the given hypergraph, which is based on this topology, is a 1-relaxed hypergraph. ■

3.3 Bound for $B_{k,1}(n)$

In this section we compute the number of hyperedges in $1-HQ_{k^t}$. As was shown in [1], the number of hyperedges in a complete hypergraph cube on k^t vertices is

$$|E(HQ_{k^t})| = tk^{t-1}.$$

We compute now the number of hyperedges, which can be eliminated when contracting vertices and hyperedges. Let $k^{t-1} < n \leq k^t$ and $p = k^t - n$. According to the construction we need to perform $p' = \lceil \frac{p}{k-1} \rceil$ merges of k vertices into 1. We want to compute how many hyperedges were merged.

In [1] we proved the following propositions:

Proposition 3.4 *Let p' denote the number of contractions of k vertices into 1 performed while reducing the number of vertices in a HQ_{k^t}. Suppose that $p' = k^i$ for some integer $0 \leq i < t$. Then, the number of hyperedges eliminated during the reduction process is:*

$$T_a(p') = (i+1)k^i - ik^{i-1}$$

Proposition 3.4 describes the case when $p' = k^i$, for some integer $0 \leq i < t$. Proposition 3.5 considers the cases where p' is not a power of k.

Proposition 3.5 *Let p' denote the number of contractions of k vertices into 1 performed while reducing the number of vertices in a HQ_{k^t}. Let i be the integer such that $k^{i-1} < p' \leq k^i$. Then, the number of hyperedges that were reduced is calculated by the following recursive formula:*

$$T_b(0) = 0$$

$$T_b(1) = 1$$

$$T_b(p') = \left\lfloor \frac{p'}{k^{i-1}} \right\rfloor T_a(k^{i-1}) + T_b(p' \ mod \ k^{i-1}) + max\{0, p' - (k-1)k^{i-1}\}(k-1)$$

Combining the number of hyperedges in HQ_k^t with the number of removed hyperedges during the vertex elimination process we get the following (see [1]):

Theorem 3.6 *Let HQ_n be a k-uniform hypergraph cube on $n = k^t - p$ vertices, $(k-1) \mid p$. Denote $p' = \lceil \frac{p}{k-1} \rceil$ the number of contractions performed during vertex elimination. Then,*

$$|E(HQ_n)| = tk^{t-1} - T_b(p'),$$

where $T_b(p')$ is as in 3.5.

We close this section we one of our main results.

Theorem 3.7 *Let $HG(V, E)$ be a k-uniform hypergraph on $n = k^t - p$ where $0 \leq p < k^t - k^{t-1}$ and $p' = \lceil \frac{p}{k-1} \rceil$. Then, for $m = 1$ we have:*

$$B_{k,1}(n) \leq \begin{cases} tk^{t-1} - T_b(p'), & \text{if } (k-1) \mid p \\ tk^{t-1} - T_b(p') + 1, & \text{otherwise} \end{cases}$$

where $T_b(n)$ is given at proposition 3.5.

Proof of the theorem 3.7: The proof follows immediately from the previous theorem and the broadcasting algorithm, except the fact that we have to consider whether there should be a tail-hyperedge. The tail-hyperedge is needed only when $k-1$ does not divide p, the number of eliminated vertices. In that case we add 1 hyperedge to the computed upper bound. ■

4 Hypergraph improved cube

In this section we investigate another hypergraph topology, the hypergraph improved cube. The hypergraph improved cube is slightly similar to the hypergraph cube from Section 3 in its structure and in the way vertices are removed. In Fig. 5 we illustrate construction of hypergraph improved cube.

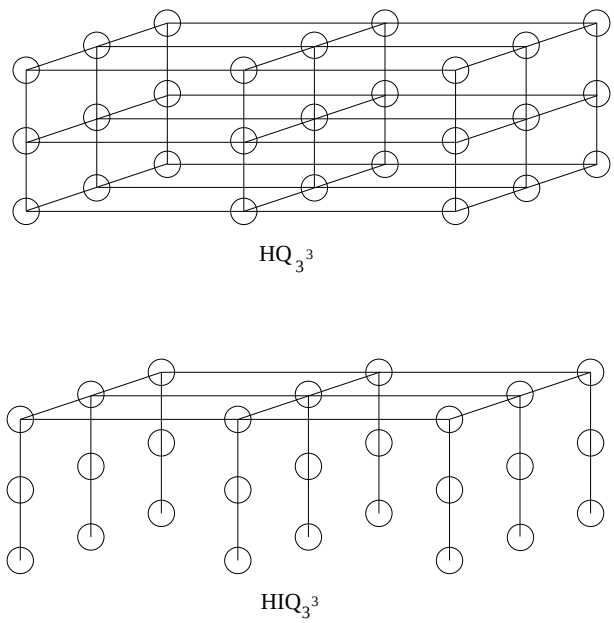


Figure 5: HQ_{3^3} versus hypergraph improved cube on 3^3 vertices.

Thus, the main result of this section is the following:

Theorem 4.1 *Let $HG(V,E)$ be a k-uniform hypergraph on $n = k^t - p$ where $0 \leq p < k^t - k^{t-1}$ and $p' = \lceil \frac{p}{k-1} \rceil$. Then, for $m = 1$ we have:*

$$B_{k,1}(n) \leq |E(HIQ_{k^t})| - Del(p),$$

where

$$|E(HIQ_{k^t})| = (t-1)k^{t-2} + k^{t-1}$$

and

$$Del(p) = \begin{cases} \left\lfloor \frac{p}{k-1} \right\rfloor + \sum_{i=1}^{t-2} \left\lfloor \frac{p}{(k-1)k^i} \right\rfloor k^{i-1}, & p \leq (k-1)^2 k^{t-2} \\ \left\lfloor \frac{p}{k-1} \right\rfloor + (k-1)k^{t-3}(t-2), & \textit{otherwise.} \end{cases}$$

In this section we define the hypergraph improved cube topology with k^t vertices, and show how to construct a hypergraph with $k^{t-1} < n \leq k^t$ vertices. We construct a 1-relaxed hypergraph based on hypergraph improved cube topology and compute an upper bound for $B_{k,1}(n)$.

4.1 The k-hypergraph improved cube topology

The k-hypergraph improved cube of dimension t is defined inductively. In the first step we take one vertex $t = 0$, $n = k^0 = 1$. In the next step we take k copies of this vertex and combine them with a mutual hyperedge.

Definition 4.2 *Let $HG(V, E)$ be a hypergraph. A $V' \subset V$ is called a* **base** *of HG where $V' \triangleq \{v \in V \mid d(v) > 1\}$.*

Given a k-hypergraph improved cube of dimension $t-1$, we label each of its vertices by a unique number. Then, take k copies of a hypergraph improved cube of dimension $t-1$. For every vertex v in the base of the first copy of the k-hypergraph improved cube of dimension $t-1$ we connect v in one-to-one way with the other such vertices in the bases of the rest $k-1$ copies of the k-hypergraph improved cubes of dimension $t-1$. It is done by introducing a new hyperedge for each k-tuple of vertices. Thus, we added k^{t-2} new hyperedges to the hypergraph.

Definition 4.3 *The hypergraph created by the process above is called a k-Hypergraph improved cube on k^t vertices, denoted by HIQ_{k^t}.*

4.1.1 Vertex labeling process

We assign a unique label to each of the vertices in a given HIQ_{k^t}. This process is done by giving to each vertex $v \in V$ in the hypergraph a t-tuple of positive integers. This t-tuple is the coordinate of v in HIQ_{k^t}. The labeling is performed inductively as follows:

First, we label an HIQ_{k^1} using one number for each vertex. Hence, the vertices in HIQ_{k^1} are labeled by $1, \cdots, k$.

Next, we extend the labeling of an $HIQ_{k^{t-1}}$ in order to achieve labeling of an HIQ_{k^t}. We do this by adding one more coordinate in front of the already existing coordinates in the $HIQ_{k^{t-1}}$. As was described above, in order to construct a HIQ_{k^t} we need k copies of $HIQ_{k^{t-1}}$, which are labeled from 1 to k, hence, we get $HIQ^1_{k^{t-1}}$, $\cdots$, $HIQ^k_{k^{t-1}}$. We add the copy index i for each of the vertices in the $HIQ^i_{k^{t-1}}$ as a new first coordinate.

Note, that the first coordinate of the vertex is the copy number of $HIQ_{k^{t-1}}$ it originally belonged to, and the rest of the coordinates mark its position within the $HIQ_{k^{t-1}}$. In Fig. 6 we demonstrate the labeling procedure.

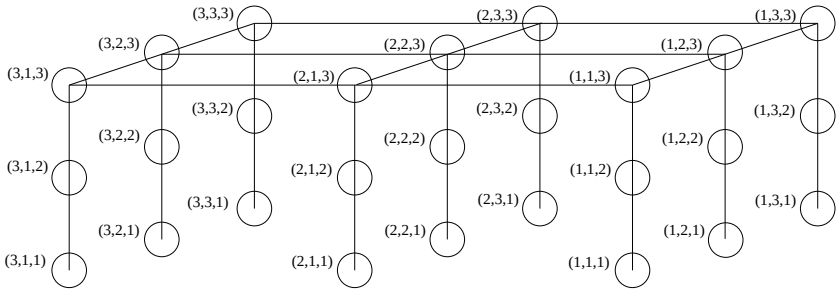


Figure 6: Vertex labeling in a HIQ_{3^3}.

4.1.2 Construction of the broadcast model based on hypergraph improved cube

In this section we describe how we eliminate vertices from HIQ_{k^t} in order to achieve a 1-relaxed hypergraph based on HIQ_{k^t} with $k^{t-1} < n \leq k^t$ vertices.

Let $p = k^t - n$ be the number of vertices that we delete from the HIQ_{k^t}. We will delete only leaf-vertices and leave the base-vertices untouched. In case $p = k^t - k^{t-1}$ set of base-vertices creates a $HIQ_{k^{t-1}}$.

We begin the elimination with leaf-vertices, which have the lowest index. The removal of leaf-vertices is performed in an increasing order. We have to describe which hyperedges are deleted from HIQ_{k^t} when we eliminate p vertices. The elimination process is divided into steps. Each step is considered when we delete all the leaf-vertices from k^i leaf-hyperedges. This means that a step is considered when we delete $k^i(k-1)$ leaf-vertices. After each step we delete all the hyperedges that contain only base-vertices of the former leaf-hyperedges that were deleted. The elimination of these hyperedges is terminated after we remove $(k-1)k^{t-2}$ leaf-hyperedges. From this point we do not remove any hyperedge from the remaining of the hypergraph, except of leaf-hyperedges.

This means that after we delete k leaf-hyperedges we delete one additional hyperedge. When we delete k^2 hyperedges we delete k hyperedges in addition to those that are removed after removal of each k hyperedges. In general, after we delete k^i leaf-hyperedges we remove an extra k^{i-1} hyperedges which are the base-hyperedges that were left without their leaf-hyperedges.

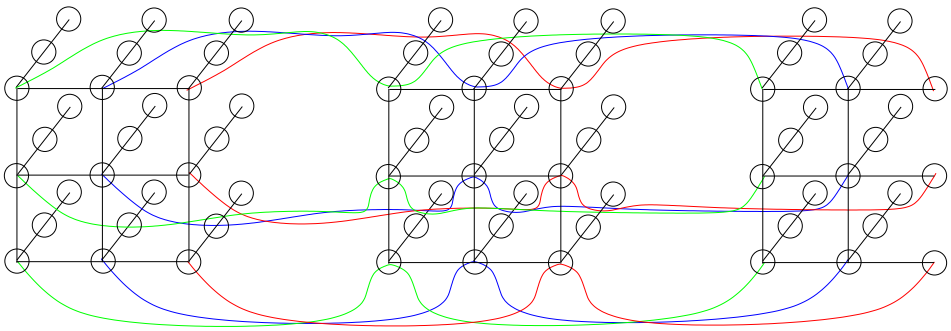
A close look shows that if $(k-1)$ does not divide p there are remaining vertices, which are left disconnected to the rest of the hypergraph. Denote

this set of vertices by U. Let $V'' \subset V'$ be the set of all base-vertices that their degree after reduction is less than their degree in the original HIQ_{k^t}.

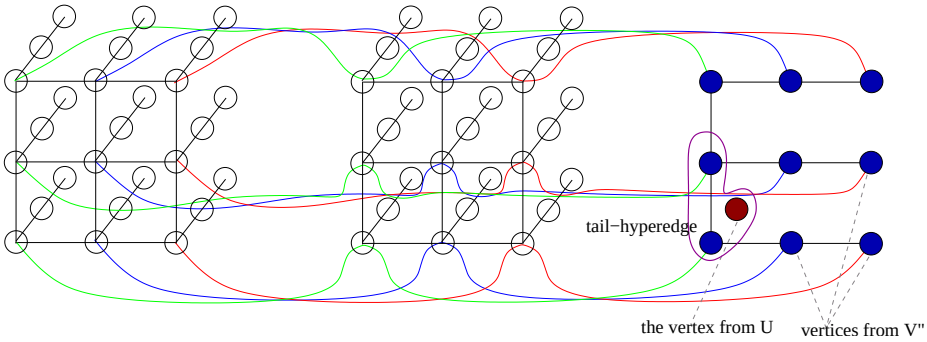
We connect the vertices from U using one additional hyperedge to the rest of the vertices by choosing $k - |U|$ vertices from V'' which have the highest indices. We call the new hyperedge a *tail-hyperedge*. Note that there is at most one tail-hyperedge in HIQ_n, and that we use the base-vertex of the former leaf-hyperedge of the vertices in U to connect them to the rest of the hypergraph.

Definition 4.4 *The hypergraph, that is obtained in this process, is called the* **reduced** *k-hypergraph improved cube and denoted HIQ_n.*

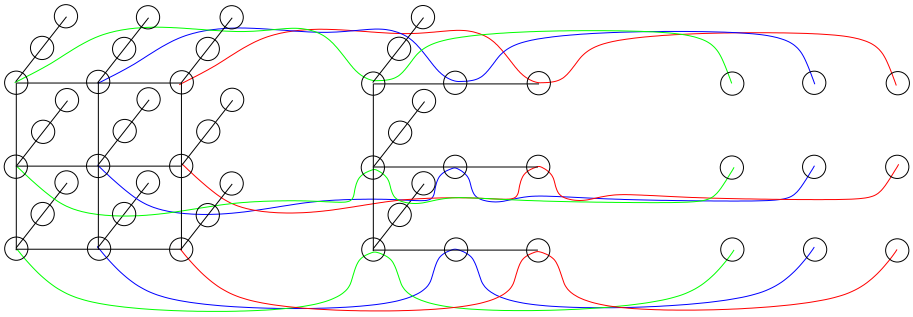
The reduction process is demonstrated in Figure 7(a-d). The resulted hypergraphs demonstrate and obtained after deletion of p vertices. We start with HIQ_{3^4} from which delete the p vertices. Figure 7(b) presents the case where a tail-hyperedge has to be connected. Figure 7(d) is the case where the maximal number of vertices is deleted, namely, $p = 3^4 - 3^3 = 54$ obtaining HIQ_{3^3}.



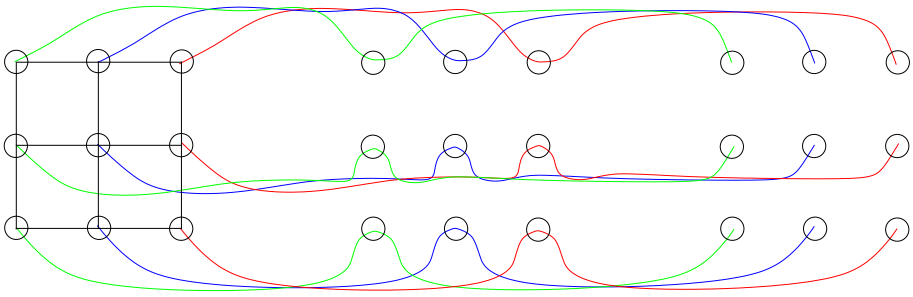
(a) p=6



(b) p=17



(c) p=30



(d) HIQ_{3^3}

Figure 7: Reduction in HIQ_{3^4} for different values of p.

4.2 The broadcasting algorithm in HIQ_n

In this section we present the broadcasting algorithm in HIQ_n. We distinguish between three cases:

Case i: The originator v is a base-vertex

Since each base vertex $(i_1, \cdots, i_{t-1}, k)$ in the HIQ_n is connected to base vertices $(1, i_2, \cdots, i_{t-1}, k), \cdots, (k, i_2, \cdots, i_{t-1}, k)$ we use this hyperedge to transmit the message to vertex $(k, i_2, \cdots, i_{t-1}, k)$. According to the construction, we did not eliminate any hyperedges which are not leaf-hyperedges from the sub-$HIQ_{k^{t-1}}$ that consists of vertices with the indices $(k, x_2, \cdots, x_t)$, which is obtained by the labeling procedure described in section 4.1.1. Therefore, since the base-vertices in $HIQ^k_{k^{t-1}}$ form an $HQ_{k^{t-2}}$, we can inform all of them with additional $t-2$ time units.

We use the hyperedges $< (1, i_2, \cdots, i_{t-1}, k), \cdots, (k, i_2, \cdots, i_{t-1}, k) >$ where $1 \leq i_1, \cdots, i_{t-1} \leq k$ to transmit the message to all the base-vertices in our hypergraph.

At time unit $t+1$ we transmit the message using all the leaf-hyperedges and the tail-hyperedge to the rest of the hypergraph vertices. If there was a vertex in the tail-hyperedge that participated in broadcasting to his leaf-hyperedge, then we could not inform the vertices in U during time unit $t+1$. Therefore, we use the tail-hyperedge in the next time unit to inform them. In this case we result with $t+2$ time units.

Case ii: The originator v is a leaf-vertex

At the first time unit the leaf-vertex transmits the message to its base-vertex say $(i_1, \cdots, i_{t-1}, k)$. At the next time unit the base-vertex transmits the message using the hyperedge $< (i_1, \cdots, 1, k), \cdots (i_1, \cdots, k, k) >$. At the next time unit all the informed vertices from this hyperedge transmit the message using hyperedges

$\langle (i_1, \cdots, i_{t-3}, 1, j, k), \cdots, (i_1, \cdots, i_{t-3}, k, j, k) \rangle$, $1 \leq j \leq k$. This process continues while in the i-th time unit k^{i-1} vertices, that already know the message, transmit it to $k^i - k^{i-1}$ new vertices using hyperedges that contain vertices which differ in their index only on the $k-i$ coordinate. In other words, at the ith time unit we use the hyperedges $<$

$(i_1, \cdots, i_{t-i}, 1, j_1, \cdots, j_{i-2}, k), \cdots (i_1, \cdots, i_{t-i}, k, j_1, \cdots, j_{i-2}, k) \ >, \ 1 \leq j_1, \cdots, j_{i-2} \leq k$.

Since at each step of the broadcasting process every hyperedge we use contains at least one base-vertex on which there is still an attached leaf-hyperedge, no one of these hyperedges was deleted during the reduction of vertices from the hypergraph.

Due to the fact that there are only k^{t-1} base-vertices in the HIQ_n the process, which was described above, ends after t time units, where all the base-vertices were already informed of the message. Therefore, at the $t+1$ time unit we transmit the message using all the leaf-hyperedges and the tail-hyperedge to the rest of the hypergraph vertices. If there was a vertex in the tail-hyperedge that participated in broadcasting to his leaf-hyperedge, then we can not inform the vertices in U during time unit $t+1$. Therefore, we use the tail-hyperedge in the next time unit to inform them, resulting in a total of $t+2$ time units.

Case iii: The originator v participates in the tail-hyperedge

We use the tail-hyperedge at the first time unit exactly as we use the leaf-hyperedge in Case *ii* in order to inform its base-vertex. From this step on, the broadcasting is performed as was described in Case *ii*.

Proposition 4.5 *In order to broadcast a message from any vertex in the HIQ_n to the rest of the hypergraph we need at most $t+2$ time units.*

Proof: The proof is immediate from the description of the broadcasting algorithm.

■

Theorem 4.6 *HIQ_n is a 1-relaxed broadcast hypergraph.*

Proof: We have to prove that we can complete broadcasting from any vertex in this hypergraph in at most $t_k(n)+1$ time units. In all the cases above the only case that we may use $t+2$ time units is when we can not transmit the message using the tail-hyperedge at the $t+1$ time unit. Therefore, we should prove that if this happens, then n is such that $t_k(n) = t+1$.

This situation can happen only if $0 < |U| + |V'| < k$. Hence, we know that we can write $n = k^{t-1} + (k-1)l + m$ where $|V'| = k^{t-1} - l$ and $|U| = m$.

Therefore, $m + k^{t-1} - l < k$, but this is exactly the situation in Lemma 1.7 that causes $t_k(n)$ to be $t + 1$. ■

Our next goal is to compute the number of hyperedges in a HIQ_n and to obtain an upper bound for $B_{k,1}(n)$.

4.3 The bound for $B_{k,1}(n)$

In this section we calculate an upper bound for $B_{k,1}(n)$ while computing the number of hyperedges in HIQ_n. We begin by computing the number of hyperedges in a complete HIQ_{k^t}, and then subtract from it the number of hyperedges which we eliminate when p vertices are removed.

Proposition 4.7 *Let E be the number of hyperedges in a complete HIQ_{k^t} with $n = k^t$ vertices, $t \geq 1$. Then,*

$$|E(HIQ_{k^t})| = (t-1)k^{t-2} + k^{t-1}.$$

Proof: The proof follows from the fact that if $t > 1$ the HIQ_{k^t} can be viewed as a $HQ_{k^{t-1}}$ with an additional hyperedge added to each of its vertices, and if $t = 1$ the HIQ_{k^1} is exactly HQ_{k^1}.

Therefore, by using the construction of HIQ_{k^t} and the fact that there are k^{t-1} vertices in an $HQ_{k^{t-1}}$ we get $(t-1)k^{t-2} + k^{t-1}$ hyperedges in the HIQ_{k^t}. Moreover, for $t = 1$ we have $(t-1)k^{t-2} + k^{t-1} = 1$ as required. ■

Now, we have to consider how many hyperedges we have to delete while eliminating p vertices from the HIQ_{k^t} in order to get the HIQ_n.

Proposition 4.8 *The number of hyperedges deleted from an HIQ_{k^t} when we remove $0 \leq p < k^t - k^{t-1}$ vertices is*

$$Del(p) = \begin{cases} \left\lfloor \frac{p}{k-1} \right\rfloor + \sum_{i=1}^{t-2} \left\lfloor \frac{p}{(k-1)k^i} \right\rfloor k^{i-1}, & p \leq (k-1)^2 k^{t-2} \\ \left\lfloor \frac{p}{k-1} \right\rfloor + (k-1)k^{t-3}(t-2), & \textit{otherwise.} \end{cases}$$

Proof: The removal of hyperedges is terminated (besides the leaf-hyperedges that we delete while eliminating the leaf-vertices) after we remove $(k-1)k^{t-2}$ leaf-hyperedges or $(k-1)^2k^{t-2}$ leaf-vertices. Therefore, we consider the case where $p \leq (k-1)^2k^{t-2}$ and the case where $(k-1)^2k^{t-2} < p$.

Recall that when $p \leq (k-1)^2k^{t-2}$ it follows according to the construction that when we eliminate k^i leaf-hyperedges (which means elimination of $(k-$

$1)k^i$ leaf-vertices), we delete k^{i-1} additional hyperedges from the hypergraph. Therefore, we first calculate $\lfloor \frac{p}{k-1} \rfloor$ which is the number of leaf-hyperedges we delete. Each time we delete k^i leaf-vertices we have to remove k^{i-1} additional hyperedges, as described in Section 4.1.2. Since $\lfloor \frac{p}{(k-1)k^i} \rfloor$ is the number of times we removed set of k^i vertices, the number of hyperedges that where deleted and which are not leaf-hyperedges is

$$\sum_{i=1}^{t-2} \left\lfloor \frac{p}{(k-1)k^i} \right\rfloor k^{i-1}.$$

We get

$$\left\lfloor \frac{p}{(k-1)} \right\rfloor + \sum_{i=1}^{t-2} \left\lfloor \frac{p}{(k-1)k^i} \right\rfloor k^{i-1}$$

as the total number of hyperedges that where deleted in this case.

When $(k-1)^2 k^{t-2} < p$. Then, the number of leaf-hyperedges that we delete is still $\lfloor \frac{p}{k-1} \rfloor$, but the number of additional hyperedges that we delete is a constant, after we remove $(k-1)^2 k^{t-2}$ vertices. Therefore, we get

$$\sum_{i=1}^{t-2} \left\lfloor \frac{(k-1)^2 k^{t-2}}{(k-1)k^i} \right\rfloor k^{i-1} = (k-1)k^{t-3} \sum_{i=1}^{t-2} 1 = (k-1)k^{t-3}(t-2).$$

We get exactly

$$\left\lfloor \frac{p}{k-1} \right\rfloor + (k-1)k^{t-3}(t-2)$$

deleted hyperedges in this case. ■

These results are summarized in the following theorem:

Theorem 4.9 *Let $HG(V,E)$ be a hypergraph on n such that $k^{t-1} < n \leq k^t$ vertices based on the $m - HIQ_{k^t}$, as defined above. Let $p = k^t - n$. Then,*

$$|E(HQ_n)| = |E(HIQ_{k^t})| - Del(p).$$

■

Proof of the theorem 4.1: $|E(HQ_n)|$ provides an upper bound for $B_{k,1}(n)$. The proof follows immediately from the previous theorem and the broadcasting algorithm. ■

5 Conclusions

We investigated 1-relaxed models that are described above for many values n and ks. We observed that for small values of n the HIQ and the kHT give us the same results. This happens since for small values of n these models have similar structures.

We also observed that as n grows the kHT and the HIQ alternately provides the minimum number of hyperedges. We call the two consecutive values of n for which this happens an *alternation point* and for larger values of n the distance between alternation points increases.

A close look at the formulas shows that when $t \to \infty$ for powers k^t the kHT gives better results than the HIQ. Moreover, since the graph of the HIQ is concave and the graph of the kHT is almost linear when n grows, we conjecture that for very large values of n the kHT will always give better bounds.

Moreover, the HQ gives worse results than the other models. In opposite, when constructing m-relaxed models in [2] we will see that $m - RBH$ based on the HQ, provide us smaller upper bounds than hypergraphs that are based on kHT or HIQ.

In addition, from computed results for $k = 3, 4, 5$ we see that as k grows, alternation points occur for significantly larger values of n. Thus, if we divide the possible values of n into intervals $(k^{t-1}, k^t]$, $t \geq 1$, we see that for larger values of k the alternations happen in intervals that have larger t. Consequently, it follows from our research that for different values of n, different models provide the best upper bound.

References

[1] Averbuch A., Kosachevski L. and Roditty Y., *Construction of Minimum Broadcast Hypergraphs*, submitted.

[2] Averbuch A., Kosachevski L. and Roditty Y., *Construction of Minimum m-Time Relaxed Broadcast Hypergraphs II:* $m \geq 1$, preprint.

[3] Averbuch A., Roditty Y. and Shoham B., *Construction of Minimum Time-Relaxed Broadcasting Communication Networks*, J. of Combin. Math. Combin. Comput., **43**(2002), 123-134.

[4] Averbuch A., Roditty Y. and Shoham B., *Efficient Line Broadcasting in a d-dimensional Grid*, Discrete Appl. Math. **113** (2001), 129-141.

[5] Averbuch A., Roditty Y. and Shoham B., *On Broadcasting Multiple Messages in a Positive Weighted Tree*, J. of Combin. Math. Combin. Comput., **35** (2000),161-184.

[6] Berge C., *Hypergraphs*, North-Holland Mathematical Library, (1989).

[7] Berge C., *Graphs and Hypergraphs*, North-Holland Mathematical Library, (1973).

[8] Cockayene E.J. and Thomason A., *Optimal Multi-message Broadcasting in Complete Graphs*, Proceeding of the 11th SE Conference on Combinatorics, Graph Theory and Computing. Utilitas Math. Winnipeg (1980), 181-199

[9] Chau S.C. and Liestman A., *Construction Minimum Broadcast Networks*, J. Combin. Inform. System Sci. **10** (1985), 110-122.

[10] Farely A.M., *Minimum-time Line Broadcast Networks*, Networks **10** (1980), 59-70.

[11] Farely A.M., *Minimum Broadcast Networks*, Networks **9** (1979), 313-332.

[12] Farely A.M., Hedetniemi S.T., Mitchel.S and Proskurowski A, *Minimum Broadcast Graphs*, Discrete Appl. Math. **25** (1979), 189-193.

[13] Fraigniaud P. and Lazard E., *Methods and Problems of Communication in Usual Networks*, Discrete Appl. Math. **53** (1994), 79-133.

[14] Grigni M. and Peleg D. *Tight Bounds on Minimum Broadcast Networks*, SIAM J. Disc. Math. **4** (1991), 207-222.

[15] Hedetniemi S.T., Hedetneimi S.T. and Liestman A.L., *A Survey of Gossiping and Broadcasting in Communication Networks*, Networks **18** (1988), 319-349.

[16] Hell P. and Seyffarth M.R., *Broadcasting in Planar Graphs*, preprint.

[17] Johanson D.S. and Garey M.R., *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, San Francisco, (1979).

[18] Kwon Oh-Heum and Chwa Kyung-Yong, *Multiple Messages Broadcasting in Communication Networks*, Networks **26** (1995), 253-261.

[19] Labahn R, *Information Flows on Hypergraphs*, Discrete Math. **113** (1993), 71-97.

[20] Liestman A.L., and Richards D. *Generalization of Broadcasting and Gossiping*, Networks **18** (1988), 125-138.

[21] Proskurowski A. *Minimum Broadcast Trees*, IEEE Trans. on Computers **5** (1981), 363-366.

[22] Roditty Y. an Shoham B., *On broadcasting multiple messages in a d-dimensional grid*, Discrete Appl. Math. **75** (1997), 277-284.

[23] Shastri A., *Broadcasting in General Networks I: Trees*, Proceeding of Cocoon (1995), Xian, China.

[24] Shastri A., *Time-relaxed Broadcasting in Communication Networks*, Discrete Appl. Math. **83** (1998), 263-278.

[25] Slater P.J., Cockayne E.J. and Hedetniemi S.T., *Information Dissemination in Trees*, SIAM J. Comput. **10** (1981) No. 4, 692-701.