Routing Permutations and 2-1 Routing Requests in the Hypercube [1]

Olivier Baudon, Guillaume Fertin

LaBRI U.M.R. CNRS 5800, Université Bordeaux I, 351 Cours de la Libération, F33405 Talence Cedex
{baudon,fertin}@labri.u-bordeaux.fr

Ivan Havel [2]

Mathematical Institute, Academy of Sciences of the Czech Republic, Žitná 25, 115 67 Praha 1, Czech Republic
haveli@math.cas.cz

Abstract

Let H_n be the directed symmetric n-dimensional hypercube. Using the computer, we show that for any permutation of the vertices of H_4, there exists a system of pairwise arc-disjoint directed paths from each vertex to its target in the permutation. This verifies Szymanski's conjecture [8] for $n = 4$.

We also consider the so-called 2-1 routing requests in H_n, where any vertex can be used twice as a source but only once as a target ; we construct for any $n \geq 3$ a 2-1 request that cannot be routed in H_n by arc-disjoint paths : in other words, for $n \geq 3$, H_n is not (2-1)-rearrangeable.

Key words: Hypercubes, routing permutations, Szymanski's conjecture, 2-1 routing requests.

[1] This work has been supported by the Barrande Cooperative Research Grant #97137. Additional supports by the Région Aquitaine Project #98024002 and by Grant #201/98/1451 of GACR.
[2] We are very sad to inform you that Ivan Havel passed away in November 1999, while this article was submitted.

1 Introduction

The directed symmetric hypercube H_n of dimension $n \geq 1$ has the set of vertices V_n with $|V_n| = 2^n$ and the set of arcs A_n with $|A_n| = n \cdot 2^n$. From several possible equivalent definitions we choose the following : V_n consists of all the integers i such that $0 \leq i \leq 2^n - 1$ and for $i, j \in V_n$, (i, j) is an arc of H_n from i to j iff the binary representations $B(i)$ and $B(j)$ of i and j differ in exactly one bit. Here the binary representation $B(k)$ of $k \in V_n$ is the binary string $b_{n-1} \ldots b_0$, where $b_\nu \in \{0, 1\}$ and $k = \Sigma_{\nu=0}^{n-1} 2^{b_\nu}$. If $(i, j) \in A_n$ and $B(i)$, $B(j)$ differ in the bit b_ν, $0 \leq \nu \leq n-1$, we say that (i, j) is in dimension ν. For every ν, $0 \leq \nu \leq n-1$, in H_n there are 2^n arcs in dimension ν . Observe that H_n is symmetric, i.e. for each $i, j \in V_n$, $(i, j) \in A_n$ iff $(j, i) \in A_n$. Examples of H_n for $1 \leq n \leq 3$ are given in Figure 1.

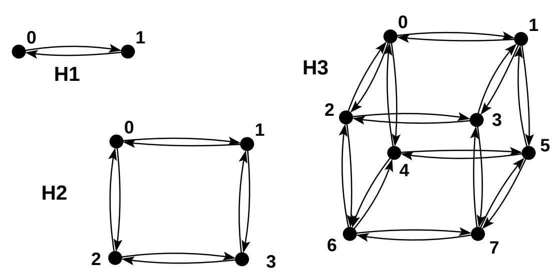

Fig. 1. Examples of hypercubes

For $h, k \geq 1$ we define a $h - k$ *routing request* (cf. [5]) :

Definition 1 (Routing Request) *A* routing request *on a directed graph G is a multi-set R of ordered pairs of vertices of G. For each pair $[s, t]$ in a routing request, s is called the* source *and t is called the* target *of the pair. A routing request R is said to be $h - k$ if each vertex appears in R at most h times as a source and at most k times as a target. A routing request on G is called a* partial permutation *if it is 1-1, and a* permutation *if it is 1-1 and has exactly $|V(G)|$ pairs.*

Note that in the following, if it is clear from the context, a routing request $[s, t]$ might be denoted by $s \to t$.

Observe that if $h = 1$ or $k = 1$ then R contains no two identical source-target pairs. Hence R can be considered to be an injection of $V(G)$ into $2^{V(G)}$, the set of all subsets of $V(G)$, fulfilling the condition that the size of an image of every vertex of G is at most h.

Szymanski [8] considered the following problem : given a hypercube H_n and a permutation R on the vertices of H_n, is it possible to realize these source-target pairs by arc-disjoint paths ? That is, is it possible to find for each $[s_i, t_i] \in R$ a directed path P_i from s_i to t_i such that the paths P_i are pairwise arc-disjoint ? Or, in the terminology of interconnection networks : is the hypercube *rearrangeable* ? Szymanski [8] conjectured that the answer is yes for any $n \geq 1$; he proved it for $n \leq 3$, with the stronger property that all the requests are satisfied by shortest paths. In the following, we will refer to Szymanski's conjecture with the shortest paths property as the *strong Szymanski's conjecture*. Lubiw [6] gave a counterexample to the strong Szymanski's conjecture : she gave an example of a permutation in H_5 which cannot be realized by arc-disjoint and shortest paths. Moreover, Darmet [4] gave a counterexample to the strong Szymanski's conjecture for H_4. It is presented in Figure 2, which gives a 1-1 routing request π on the vertices of H_4. Note that π is a partial permutation, but, still, π cannot be routed by arc-disjoint and shortest paths. Any permutation realizing at least the requests from π would fail to be routed by arc-disjoint and shortest paths as well.

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\pi(x)$	10	2	14		8	3	15	11					9	1	13	

Fig. 2. π cannot be routed in H_4 by arc-disjoint and shortest paths

Note that all the source-target pairs which are given in the table are such that $dist(x, \pi(x)) = 2$ in H_4. This is the base of the proof : suppose we want to find a shortest path for the source-target pair $[6, 15]$. Then we can either route via 14, or via 7. Suppose we route via 14, that is $6 \rightarrow 15 : 6, 14, 15$. Hence the only solution to route the pair $[2, 14]$ is $2 \rightarrow 14 : 2, 10, 14$, which means that $0 \rightarrow 10 : 0, 8, 10$, hence $4 \rightarrow 8 : 4, 12, 8$, and consequently $12 \rightarrow 9 : 12, 13, 9$. In that case, the arcs $(14, 15)$ and $(12, 13)$ have both been used. Hence it is impossible to route from 14 to 13 by a shortest path in an arc-disjoint fashion. Similarly, if we decide to route from 6 to 15 via 7, we end up with a contradiction of the same sort.

Observe that we will see in Theorem 2 that all the permutations on H_4 that realize at least π can be routed by non shortest paths.

Since we managed to find a counterexample for $n = 4$ to the strong Szymanski's conjecture, it is easy to see that for every $n \geq 4$, we can find a permutation for which it is not possible to route by shortest paths, since for a hypercube H_n with $n \geq 4$, H_4 is a subgraph of H_n. To this end, we take any subgraph isomorphic to H_4 in H_n and apply π on this subgraph ; then we "complete" π to get a permutation on H_4.

However, until now, it is still an open problem whether Szymanski's conjecture (i.e. the rearrangeability of H_n, with $n \geq 4$, without the shortest paths condition) holds. Note that many authors [5,3,7,2] have been solving the problem partially, whether by proving that some families of permutations could be routed in any H_n, or by proving that by doubling the arcs of H_n in one or several of its dimensions, any permutation could be routed in H_n. For a good survey of the results concerning the latter, we refer to [5].

In this paper, we first show in Section 2 that Szymanski's conjecture holds for $n = 4$; for this, we use a computer program. We will see that this proof implies the use of 2-1 routing requests. Hence, in Section 3, we will focus on the (2-1)-rearrangeability of H_n, that is the rearrangeability of H_n with respect to the 2-1 routing requests. In H_3, we show that two such (2-1) routing requests cannot be routed by arc-disjoint paths ; this can be generalized, and we show in Section 3.2 that for any $n \geq 3$, H_n is not (2-1)-rearrangeable.

2 On the rearrangeability of H_4

Here, we prove Szymanski's conjecture for $n = 4$. This is done using a computer program, whose steps are detailed below. First, we observe that deleting all the arcs in dimension i ($0 \leq i \leq n-1$) in H_n results in a disconnected graph, each of the two connected components being a copy of H_{n-1}. Hence, we will call a *cut* in dimension i ($0 \leq i \leq n-1$) the deletion of all the directed egdes of H_n in dimension i. The two copies of H_{n-1} that we get that way are called *subcubes*. For example, the *cut in dimension 0* of H_4 gives us two subcubes of dimension 3, say $H_{3,0}$ and $H_{3,1}$, where $H_{3,0}$ (resp. $H_{3,1}$) is the subgraph of H_4 induced by the vertices p (resp. q) with $0 \leq p \leq 7$ (resp. with $8 \leq q \leq 15$).

H_4 has 16 vertices, and therefore there are 16! permutations on the vertices of H_4. Hence, by brute force method, we could have tried to route each and every permutation on the vertices of H_4. However, a deeper study of H_3 will save us many unnecessary computations.

2.1 Converting the problem from H_4 to H_3

The main idea here is to answer the following question : can any 2-1 routing request be routed by arc-disjoint paths in H_3 ? Indeed, if we manage to prove this, then it is not difficult to see that any permutation can be routed by arc-disjoint paths in H_4 : suppose we have a permutation π in H_4, and let us cut H_4 in dimension 0. Let $V_{3,0} = \{v \in V_4 \mid 0 \leq v \leq 7\}$ and $V_{3,1} = \{v \in V_4 \mid 8 \leq v \leq 15\}$. Now let us route π using the following "cross first strategy". For each $v \in V_{3,i}$ ($0 \leq i \leq 1$) :

- If $\pi(v) \in V_{3,j}$ with $i \neq j$, then route using the arc (v, v') such that $v' \in V_{3,j}$, and route $v' \to \pi(v)$ using only the arcs of $V_{3,j}$;
- If $\pi(v) \in V_{3,i}$, then route using only the arcs of $V_{3,i}$.

Clearly, following this strategy, any arc of the form (v, v') with $v \in V_{3,i}$ and $v' \in V_{3,j}$ with $i \neq j$ will be used at most once. Moreover, this strategy induces in each of the subcubes $H_{3,0}$ and $H_{3,1}$ a 2-1 routing request. Hence, if any 2-1 routing request can be routed by arc-disjoint paths in H_3, then H_4 is rearrangeable.

To know whether any 2-1 routing request on H_3 can be routed by arc-disjoint paths, we use the computer. For a better understanding, we give an overview of the algorithm used : first, for each request $[s_i, t_i]$, we are allowed certain paths depending on the distance from s_i to t_i in H_3. These paths are the following :

- if $dist(s_i, t_i)$=1, we can only use the shortest path, that is the arc (s_i, t_i) ;
- if $dist(s_i, t_i)$=2, we can use the 2 shortest paths or the 6 paths of length 4 in H_3 ;
- if $dist(s_i, t_i)$=3, we can use the 6 shortest paths or the 6 paths of length 5 in H_3.

Note that, for each request, we order the possible paths by priority (in that case, the shortest paths will be placed first, then the non shortest ones).
The algorithm is the following : for each request, take the path with higher priority. If at least one arc of this path is already used by a previous request, then try the second path, etc., till one path is such that no arc has been used before. If it is not possible, then backtrack to the previous request, and do the same thing recursively till we can find a path P with no arc already used. In that case, use the path P, and try to route the next request. If no path P is found, the routing request is said to be failing. If a path is found for each of the requests, then the given 2-1 routing request can be routed in H_3 by arc-disjoint paths.
Thanks to the computer, we are able to show that "most" of the 2-1 routings on H_3 can be routed by arc-disjoint paths. In fact, only 72 of them did not get through our algorithm (cf. Section 5). Let us call them the *72 failing routing requests*. Thanks to the numerous automorphisms of H_3, we can show that only two of them are non equivalent by automorphism : these are the 2-1 routing requests f_3 and g_3 defined in Section 3.1.
Note that our algorithm does not try each and every possible path for a given request. However, we show in Section 3.1 that f_3 and g_3 cannot indeed be routed in H_3 by arc-disjoint paths.

2.2 Getting back to H_4

Now let us consider one of those 72 2-1 failing routing requests, say ρ. The aim is to consider ρ as the "projection" on H_3 of a permutation π of the vertices of H_4 and to retrieve all the possible corresponding permutations π. Depending on which of the 4 dimensions we decide to cut H_4, and, having done so, on which of the 2 subcubes of dimension 3 we consider, there are $4 \cdot 2 = 8$ possibilities. Once we have decided this, we have to "rebuild" π from the informations given by ρ. In each ρ among the 72 failing routing requests, three of the eight vertices are used twice as a source (this can be verified in Section 5) ; hence, only five distinct vertices are used as sources. Consider the following example (Figure 3), where we consider ρ in the subgraph $H_{3,0}$ induced by a cut in dimension 0.

x	0	1	2	3	4	5	6	7
$\rho(x)$			5 7	4	3 2	6	1 0	

Fig. 3. A 2-1 routing request ρ in H_3

In that case, we see that 2 is taken twice as a source. Hence the corresponding permutations in H_4 will either have $[2, 5]$ and $[10, 7]$, or $[2, 7]$ and $[10, 5]$, as source-target pairs. The same goes for a vertex which is only taken once as a source. Take, for instance, vertex 3. The corresponding permutations in H_4 either could have $[3, 4]$ as a source-target pair, or $[11, 4]$. As we have five vertices which are sources at least once, this gives us $2^5 = 32$ possible different sets of 8 requests in H_4. This fixes only 8 requests ; hence, there are $8! = 40320$ possibilities for the 8 remaining requests in H_4.
Consequently, for each of the 8 considered subcubes of dimension 3, and for each 2-1 routing request ρ in this subcube, we need to test $32 \cdot 40320 = 1290240$ permutations π in H_4. Thanks to the computer, it is very easy and fast to verify that those permutations in H_4 can be routed by arc-disjoint paths. Indeed, suppose we have cut H_4 in dimension 0, and that we are looking at $H_{3,0}$, i.e. the hypercube of dimension 3 induced by the vertices $0 \leq p \leq 7$. In that case, for each of the 72 2-1 failing routing requests ρ, we have to test the rearrangeability of H_4 on the permutations $\pi_{i,\rho}$ $(1 \leq i \leq 32 \cdot 40320)$. For a given $\pi_{i,\rho}$, let us cut H_4 in a different dimension (say 1) and see, in each of the two subcubes induced by the cutting, if the 2-1 routing requests induced by this cut is among the 72 failing ones. If this is not the case, then we know it is possible to route $\pi_{i,\rho}$ in an arc-disjoint fashion thanks to this new cutting. If this is the case, let us try by cutting in another dimension (say 2), etc.
It appears that, for each of the 72 2-1 routing requests ρ, no $\pi_{i,\rho}$ is such that, by cutting H_4 in one of the three other dimensions, the new 2-1 routing requests given in each of the two subcubes are among the 72 failing ones. Consequently, if a permutation π is such that a cut in dimension $0 \leq d \leq 3$ induces one of the 72 failing routing requests in at least one of its two subcubes of dimension 3, then there exists $0 \leq d' \neq d \leq 3$ such that a cut in dimension d' does not imply this situation. Hence the following Theorem, which answers and confirms Szymanski's conjecture for the hypercube of dimension 4.

Theorem 2 *Any permutation π on the vertices of H_4 can be routed by arc-disjoint paths, that is H_4 is rearrangeable.*

3 2-1 routing requests in H_n

In Section 3.1, we study two examples of 2-1 routing requests, f_3 and g_3 (cf. Section 2.1), and prove that they cannot be routed in H_3 by arc-disjoint paths. Starting from g_3, we show in Section 3.2 a recursively constructed 2-1 routing request on H_n, g_n, for which no arc-disjoint routing can be found ; this proves Theorem 14.

3.1 H_3 is not (2-1)-rearrangeable

We have seen in Section 2.1 that among the 72 failing 2-1 routing requests, only two of them are non equivalent by automorphism of H_3. We denote those two routing requests f_3 and g_3, which are as follows :

x	0	1	2	3	4	5	6	7	x	0	1	2	3	4	5	6	7
$f_3(x)$	3		5	4		0	1		$g_3(x)$			5	4	7	2		0
	6		7			2							6		3		1

Fig. 4. f_3 (left) and g_3 (right)

We are going to show that none of them can be routed in H_3 by arc-disjoint paths. First, we introduce some auxiliary notions. We call an arc (x, y) of H_n a d–arc (downwards going arc) if $x > y$, otherwise, i.e. if $x < y$, we call it a u–arc. Note that if H_n is drawn using a "level" representation in such a way that 0 is the lowest and $2^n - 1$ the highest vertex in the drawing, then d–arcs are really directed downwards and u–arcs upwards (cf. for instance Figure 5) .

We easily verify that for any $s, t \in V_n$ all shortest directed paths from s to t in H_n use the same number of d–arcs (resp. u–arcs) ; let us denote it $d(s, t)$ (resp. $u(s, t)$). Also, any directed path P from s to t uses at least $d(s, t)$ d–arcs and $u(s, t)$ u–arcs, whereas the difference of the number of d–arcs and that of u–arcs P actually uses equals $d(s, t) - u(s, t)$.
For a routing request $R = [s_1, t_1], \ldots, [s_r, t_r]$ in H_n, we define $d(R)$ and $u(R)$ as follows : $d(R) = \Sigma_{i=1}^{r} d(s_i, t_i)$ and $u(R) = \Sigma_{i=1}^{r} u(s_i, t_i)$.

Finally we define, for $x \in V_n$: $x^{in} = \{(y, x); y \in V_n \text{ and } (y, x) \in A_n\}$ and $x^{out} = \{(x, y); y \in V_n \text{ and } (x, y) \in A_n\}$. Now we are ready to prove the following.

Proposition 3 *Neither f_3 nor g_3 can be routed by arc-disjoint paths in H_3.*

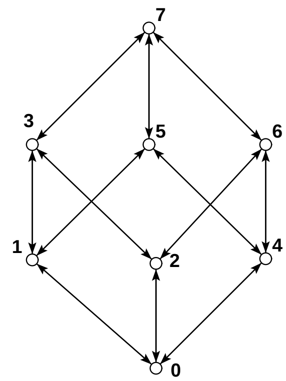

Fig. 5. The hypercube H_3 drawn in "level representation"

PROOF. First, observe that $d(f_3) = 9$, $u(f_3) = 11$, $d(g_3) = 12$, and $u(g_3) = 8$.

Assume that there is a routing by arc-disjoint paths in H_3 for f_3, let us denote it by ρ. Analyzing the sources and targets of f_3 we conclude that ρ must use the arcs $(0,4)$, $(1,5)$, $(2,6)$ and $(3,7)$, i.e. all the arcs leading from the subcube induced in H_3 by the vertices $\{0,1,2,3\}$ to the subcube induced by the vertices $\{4,5,6,7\}$ (the reason is that 4 pairs of f_3 have their sources in the first subcube and targets in the second one). It follows quite similarly that ρ must also use the arcs $(0,1)$, $(2,3)$, $(4,5)$ and $(6,7)$. Further, we conclude that ρ uses not more than 2 arcs from each of the following sets : 1^{out}, 4^{out} and 5^{in} (since, e.g. 1 is a target but not a source, hence exactly one path from ρ ends in 1 and no path from ρ begins there). Since ρ uses $(1,5)$ and $(4,5)$, it does not use $(7,5)$.

Let us look at the set 1^{out} : at least one of its arcs is not used by ρ ; since we showed above that $(1,5)$ must be used, there are 2 cases to be considered :

(1) $(1,3)$ is not used by ρ : then all the remaining 11 u–arcs have to be used by ρ and ρ is necessarily a shortest path routing. It follows that $0 \to 3 : 0,2,3$, $0 \to 6 : 0,4,6$, $2 \to 7 : 2,6,7$, $2 \to 5 : 2,0,1,5$. Now consider $6 \to 1$: it can start neither with $(6,7)$ (used already) nor with $(6,2)$ (there is no way out from 2), hence $6 \to 1 : 6,4,5,1$; finally, $5 \to 0 : 5,4,0$. This is a contradiction, since all the 3 arcs from 4^{in} are already used and therefore $3 \to 4$ cannot be done.

(2) $(1,0)$ is not used by ρ : because of symmetry and the fact that at most 2 arcs from 4^{out} may be used by ρ, we conclude that $(4,0)$ is not used by ρ either. Since $d(f_3) = 9$ and the d–arcs $(1,0)$, $(4,0)$, $(7,5)$ are not used by ρ, ρ has to be a shortest path routing. This is a contradiction, because, obviously, $5 \to 0$ cannot be routed by a shortest path. This contradiction completes the analysis of the second case and we are done with f_3.

Let us now consider g_3 : Since $d(g_3) = 12$ and in H_3 there are altogether 12 d–arcs, all the d–arcs must be used and we conclude that ρ consists of shortest paths. Consider the subset $S = \{3, 5, 7\}$ of V_3. Observe that, in ρ, there are 5 paths with targets in $V_3 \setminus S$, which start in S. On the other hand, there are 5 arcs leading from vertices in S to vertices in $V_3 \setminus S$. We conclude that $5 \to 3 : 5, 7, 3$. Further, ρ uses not more than 2 arcs from 6^{out}, since 6 is a target and not a source in g_3. However, as we noted above, all d-arcs must be used ; hence ρ does not use $(6, 7)$, and we conclude that $4 \to 7$ cannot be managed by a shortest path. This contradiction accomplishes the whole proof. □

3.2 2-1 routing requests in H_n

In this Section, we are going to define recursively a 2-1 routing request g_n in the hypercube of dimension n, H_n, and show in Theorem 14 that for any $n \geq 3$, g_n cannot be routed in H_n by arc-disjoint paths. This shows that for any $n \geq 3$, H_n is not (2-1)-rearrangeable.

The idea here is to find an "equivalent" of the routing request g_3 of Section 3.1 for any $n \geq 3$, and to generalize the arguments that allowed us to show that g_3 cannot be satisfied by arc-disjoint paths. Because of the remark following Definition 1 we may assume that g_3 is an injection of V_3 into 2^{V_3} fulfilling $|g_3(v)| \leq 2$ for all $v \in V_3$.
We are going to define recursively a 2-1 routing request g_n in H_n for $n \geq 3$; we will do it defining g_n again as an injection of V_n into 2^{V_n}.

Definition 4 (2-1 routing request g_n) *Let g_3 be the injection of V_3 into 2^{V_3} defined in Section 3.1. Let $n \geq 3$ and assume that the injection g_n of V_n into 2^{V_n} is already defined. Define g_{n+1} as follows :*

- *if $0 \leq i \leq 2^n - 1$, put $g_{n+1}(i) = \{j + 2^n; j \in g_n(i)\}$. (Observe that $g_n(i) = \emptyset$ implies $g_{n+1}(i) = \emptyset$ as well.)*
- *if $2^n \leq i \leq 2^{n+1} - 1$, put $g_{n+1}(i) = g_n(i - 2^n)$.*

In what follows we are going to use g_n (for $n \geq 3$) to denote both the 2-1 routing request in H_n and the corresponding injection of V_n into 2^{V_n}. (A misunderstanding will be avoided by the context.)

The Lemmas given below list properties of g_n we will need in the sequel (a proof missing means that the statement follows trivially).

Lemma 5 *For $n \geq 3$, g_n is a 2-1 routing request in H_n.*

Lemma 6 *For $n \geq 3$, every vertex of H_n is used in g_n exactly once as a* *target.*

Lemma 7 *For $n \geq 3$, $d(g_n) = n \cdot 2^{n-1}$.*

PROOF. Observe that for $n \geq 3$, if $0 \leq i, j \leq 2^n - 1$, then $d(i, j + 2^n) = d(i, j)$, and $d(i + 2^n, j) = d(i, j) + 1$. Using Lemma 6, we easily verify that $d(g_{n+1}) = d(g_n) + d(g_n) + 2^n$, $n \geq 3$, and, using the already known equality $d(g_3) = 12$, we accomplish the proof by induction. □

Lemma 8 *For $n \geq 3$, any routing by arc-disjoint paths in H_n, satisfying g_n, must be a shortest paths routing.*

PROOF. Observe that in H_n there are exactly $n \cdot 2^{n-1}$ d–arcs and use Lemma 7. □

Now, let S_n denote the set of those vertices of H_n which have 2 targets in g_n, i.e.

$$S_n = \{i \in V_n; |g_n(i)| = 2\}.$$

The next Lemma characterizes elements of S_n using their binary representation.

Lemma 9 *Let $n \geq 3$ and $i \in V_n$. Then $i \in S_n$ if and only if $B(i)$ ends with 011, 101, or 111, i.e. $B(i) = pp'$, where $p \in \{0, 1\}^{n-3}$ and $p' \in \{011, 101, 111\}$.*

PROOF. Observe that $S_3 = \{3, 5, 7\}$ and use induction. □

Lemma 10 *For $n \geq 3$ there are exactly $5 \cdot 2^{n-3}$ arcs in H_n, leading from a vertex of S_n to a vertex of $V_n \setminus S_n$.*

PROOF. The statement follows directly from Lemma 9. □

Lemma 11 *For $n \geq 3$, $[s, t]$ is a request of g_n with $s, t \in S_n$ if and only if $B(s) = m101$ and $B(t) = \overline{m}011$, where $m \in \{0, 1\}^{n-3}$ and $\overline{m}$ is the complementary string to m (i.e. the string arising from m by changing each bit of m to its opposite : $0 \rightarrow 1$ and $1 \rightarrow 0$).*

PROOF. We will prove the lemma by induction on n. Since $S_3 = \{3, 5, 7\}$ and $[5, 3]$ (with $B(5) = 101$ and $B(3) = 011$) is the only request of g_3 with both source and target in S_3, we are done with the case $n = 3$.
Now, let $n \geq 3$, let $s, t \in V_{n+1}$. It follows from Definition 4 that $[s, t]$ is a request in g_{n+1} if and only if one of the following possibilities occurs :

(1) $0 \leq s < 2^n$, $2^n \leq t < 2^{n+1}$ and there is a request $[s', t']$ in g_n such that $s' = s$ and $t' = t - 2^n$;
(2) $2^n \leq s < 2^{n+1}$, $0 \leq t < 2^n$ and there is a request $[s', t']$ in g_n such that $s' = s - 2^n$ and $t' = t$.

Let $[s, t]$ be a request in g_{n+1}, let possibility 1 occur. Obviously, $B(s) = 0B(s')$, $B(t) = 1B(t')$ and therefore (Lemma 9) both s and t belong to S_{n+1} if and only if both s' and t' belong to S_n. Using induction hypothesis we conclude that this happens if and only if $B(s') = m101$ and $B(t') = \overline{m}011$ for some $m \in \{0, 1\}^{n-3}$ and this is equivalent (because of equalities above) to $B(s) = 0m101$, $B(t) = 1\overline{m}011 = \overline{0m}011$. So, in case 1, we are done with the induction step ; we proceed analogously in the case when possibility 2 occurs. □

Lemma 12 *For $n \geq 3$, there are exactly 2^{n-3} requests $[s, t]$ in g_n with $s, t \in S_n$.*

PROOF. Use Lemma 11. □

Lemma 13 *Let $n \geq 3$, let $[s, t]$ be a request of g_n with $s, t \in S_n$. Then any routing by arc-disjoint paths in H_n, satisfying g_n, uses only vertices of S_n in order to route the request $[s, t]$.*

PROOF. First, using Lemma 9, we obtain $|S_n| = 3 \cdot 2^{n-3}$, $n \geq 3$. Let ρ_n be a routing by arc-disjoint paths in H_n, satisfying g_n. Using Lemma 12 and observing that every source vertex of S_n has two targets we conclude that ρ_n contains $5 \cdot 2^{n-3}$ paths leading from S_n to $V_n \setminus S_n$. Since in H_n there are exactly $5 \cdot 2^{n-3}$ arcs leading from a vertex of S_n to a vertex of $V_n \setminus S_n$ (Lemma 10), we conclude that every path of ρ_n leading from S_n to $V_n \setminus S_n$ uses just one of these arcs ; hence, to route a request $[s, t]$ of g_n with $s, t \in S_n$, only the arcs with both end vertices in S_n may be used. □

Property 1 *Let $n \geq 3$ and $m \in \{0, 1\}^{n-3}$, let x and y be vertices of H_n such that $B(x) = m101$ and $B(y) = m111$. Then any routing by arc-disjoint paths in H_n, satisfying g_n, uses the arc (x, y) to route a particular request $[s, t]$ of g_n with $s, t \in S_n$.*

PROOF. Let $n \geq 3$ and let $[s_1, t_1]$ be a request of g_n with $s_1, t_1 \in S_n$. Using Lemma 11 we observe that there is $p \in \{0,1\}^{n-3}$ such that $B(s_1) = p101$ and $B(t_1) = \overline{p}011$. Assume that ρ_n is a routing by arc-disjoint paths in H_n, satisfying g_n and consider the path P of g_n, routing the request $[s_1, t_1]$. Necessarily, there are $x_1, y_1 \in V_n$ such that :

(1) x_1 and y_1 are two consecutive vertices of P and thus (x_1, y_1) is an arc of H_n,
(2) there is $p' \in \{0,1\}^{n-3}$ fulfilling either
 2a. $B(x_1) = p'101$ and $B(y_1) = p'001$,
 or
 2b. $B(x_1) = p'101$ and $B(y_1) = p'111$.

Let us observe that the case 2a cannot occur, because $B(y_1) = p'001$ implies $y_1 \notin S_n$ (Lemma 9) and this is a contradiction to Lemma 13. In this way we found, given a request $[s_1, t_1]$ of g_n with $s_1, t_1 \in S_n$, the arc (x_1, y_1) and $p' \in \{0,1\}^{n-3}$ such that 2b holds. Since the paths of ρ_n are arc-disjoint, we accomplish the proof by observing that $|\{0,1\}^{n-3}| = 2^{n-3}$ and using Lemma 12.

We refer to Figure 6 for an illustration of this Property in the case $n = 4$. In that case, $S_4 = \{3, 5, 7, 11, 13, 15\}$ (Lemma 9), and there are 2 requests of the form $[s_i, t_i]$ with $s_i, t_i \in V_4$, $i = 1, 2$. These requests are $[5, 11]$ and $[13, 3]$ (with $B(5) = 0101$, $B(11) = 1011$, $B(13) = 1101$ and $B(3) = 0011$), cf. Lemma 11. We then see that the two arcs $(5, 7)$ and $(13, 15)$ must be used to route those two requests, otherwise at least one arc (x, y) with $x \in S_4$ and $y \notin S_4$ would be used, which contradicts Lemma 13. □

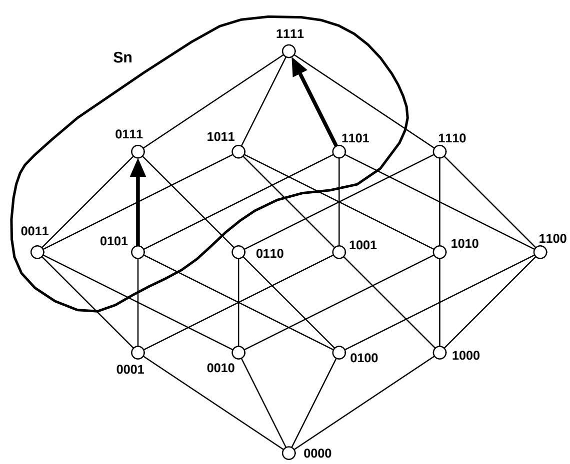

Fig. 6. Property 1 illustrated in H_4

Property 2 *For $n \geq 3$ and $m \in \{0,1\}^{n-3}$, let x and y be vertices of H_n such that $B(x) = m110$ and $B(y) = m111$. Then any routing by arc-disjoint paths in H_n, satisfying g_n, does not use the arc (x, y).*

PROOF. As a first step it is easy to verify by induction that, for $n \geq 3$, if x' is a vertex of H_n fulfilling $B(x') = p110$ for some $p \in \{0,1\}^{n-3}$, then $g_n(x') = \emptyset$ (i.e. x' has no target in g_n).

So, let $n \geq 3$, assume that ρ_n is a routing by arc-disjoint paths in H_n, satisfying g_n. Using Lemma 6 observe that, if $g_n(y') = \emptyset$ for some $y' \in V_n$, then at least one arc from y'^{out} is used by no path of ρ_n. If, moreover, there is an arc in y'^{in} not used by ρ_n, then there are at least two arcs in y'^{out} used by no paths of ρ_n.

Next, using Lemma 7 we observe that ρ_n must use all the d–arcs of H_n.

Let $x \in V_n$ fulfil $B(x) = m110$ for some $m \in \{0,1\}^{n-3}$. If $n = 3$, then $x = 6$, x^{out} consists of two d–arcs and the u–arc $(6,7)$ which is used by no ρ_3. Therefore, in the rest of the proof we may assume $n \geq 4$.

We are going to prove a stronger statement, namely : for $n \geq 4$ and $m \in \{0,1\}^{n-3}$, let x and y be vertices of H_n such that $B(x) = m110$ and $B(y) = m111$. Then any routing by arc-disjoint paths in H_n, satisfying g_n, uses all the arcs from x^{out} with the exception of the arc (x,y).

The proof will be by induction on $|m|_0$, the number of zero bits of the string m.

I. If $|m|_0 = 0$, i.e. if $B(x) = 1\ldots1110$ then the only u–arc contained in x^{out} is (x,y) and all the remaining arcs in x^{out} are d–arcs ; we are done with the basis of induction.

II. Assume $0 < |m|_0 \leq n-3$. We already know that ρ_n does not use at least one arc from x^{out} ; it is necessarily a u–arc. Let the arc not used be (x,y'), where $B(y') = m'110$ for m' arising from m by changing exactly one zero bit of m to 1. Observe that $|m'|_0 = |m|_0 - 1$ and thus, by induction hypothesis, there is exactly one arc from y'^{out} not used by ρ_n (and this is the arc (y',y'') where $B(y'') = m'111$). However, we see that $(x,y') \in y'^{in}$ is not used by ρ_n, and also $g_n(y') = \emptyset$, therefore at least two arcs from y'^{out} are not to be used by ρ_n, which is a contradiction. Hence the arc from x^{out} not to be used by ρ_n must be (x,y) ; we are done also with the induction step, which accomplishes the whole proof.

We refer to Figure 7 for an illustration of Property 2 in the case $n = 4$. □

Theorem 14 *For $n \geq 3$, g_n cannot be routed in H_n by arc-disjoint paths ; in other words, for $n \geq 3$, H_n is not $(2,1)$-rearrangeable.*

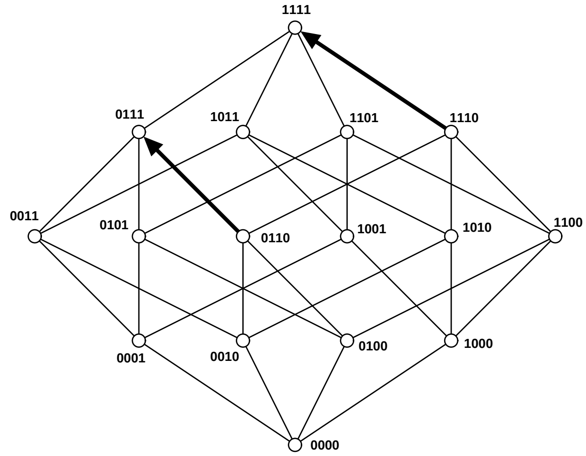

Fig. 7. Property 2 illustrated in H_4. The arcs (1110, 1111) and (0110, 0111) are unused by ρ_4.

PROOF. It follows immediately from the definition that, for $n \geq 3$, g_n contains the request $[4, 2^n - 1]$. Assume that there is a routing by arc-disjoint paths in H_n satisfying g_n, let it be ρ_n. Observe that $B(4) = 0 \ldots 0100$, $B(2^n - 1) = 1 \ldots 1111$. Since ρ_n is a shortest path routing (Lemma 8), there are $x, y \in V_n$ and $m \in \{0,1\}^{n-3}$ such that the arc (x, y) is used by ρ_n to route the request $[4, 2^n - 1]$ and either

$$B(x) = m101 \text{ and } B(y) = m111,$$

or

$$B(x) = m110 \text{ and } B(y) = m111.$$

However, the first possibility is excluded by Property 1 (because $4 \notin S_n$), the second by Property 2. The contradiction proves the theorem. □

Remark 15 *We note that if we apply the same construction to the 2-1 routing request f_4, we do not necessarily get a 2-1 routing request in H_n which cannot be routed by arc-disjoint paths.*
Indeed, let f_4 be the 2-1 routing request on H_4 obtained by applying to f_3 the same operation which gave g_4 from g_3. In that case, f_4 can be routed in H_4 by arc-disjoint paths, as shown in Figure 8 below.

x	0	1	2	3	4	5	6	7
$f_4(x)$	11 14		13 15	12		8 10	9	
paths	0,1,3,11 0,2,10,14		2,0,4,12,13 2,3,7,15	3,1,9,13,12		5,4,0,8 5,13,15,14,10	6,14,15,11,9	

x	8	9	10	11	12	13	14	15
$f_4(x)$	3 6		5 7	4		0 2	1	
paths	8,10,11,3 8,12,14,6		10,8,9,1,5 10,2,6,7	11,15,7,6,4		13,9,8,0 13,5,7,3,2	14,12,4,5,1	

Fig. 8. An arc-disjoint routing for the 2-1 routing request f_4

4 Conclusion

In the first part of this paper, we use the computer to prove the rearrangeability of H_4. This proof relies on the "splitting" of any permutation in H_4 into two 2-1 routing requests (one in each of the H_3 obtained by cutting H_4 in dimension 0), which we try to route by arc-disjoint paths in their respective subcube of dimension 3. We note that this method is the one employed by Szymanski [8] to prove the rearrangeability of H_3.
Consequently, our 1-1 routing request problem has turned into a 2-1 routing request problem. Hence, the study of 2-1 routing requests appears to be of high interest as well ; this is the object of the second part of this paper. Indeed, we have derived from our routing algorithm 72 *failing* 2-1 routing requests, among which 2 appear to be non-equivalent by automorphism of H_3. We have shown that these two 2-1 routing requests cannot indeed be routed in H_3 by arc-disjoint paths. Moreover, we have derived from one of them a recursive definition of a 2-1 routing request, and we have shown that it cannot be routed in an arc-disjoint fashion in H_n, for any $n \geq 3$.

Starting from what was a study of 1-1 routing requests in H_n and the rearrangeability of H_n, we have mainly studied and proved the non-rearrangeability of H_n with respect to 2-1 routing requests. Though we have answered the question of the (2-1)-rearrangeability of H_n, the question of the (1-1)-rearrangeability of H_n for any $n \geq 5$ is still open.

References

[1] O. Baudon, G. Fertin, and I. Havel. Routing permutations in the hypercube. *In Proc. of the 25th International Workshop on Graph-Theoretic Concepts in Computer Science (WG'99), Ascona*, 1999.

[2] R. Boppana and C.S. Raghavendra. Optimal self-routing of linear-complement

permutations in hypercubes. In *DISTMEMCC: 5th Distributed Memory Computing Conference*, pages 800–808. IEEE Computer Society Press, 1990.

[3] S.B. Choi and A.K. Somani. Rearrangeable circuit-switched hypercube architectures for routing permutations. *Journal of Parallel and Distributed Computing*, 19:125–130, 1993.

[4] A. Darmet. Private communication, 1992.

[5] Q-P Gu and H. Tamaki. Routing a permutation in the hypercube by two sets of edge disjoint paths. *Journal of Parallel and Distributed Computing*, 44(2):147–152, 1997.

[6] A. Lubiw. Counterexample to a conjecture of Szymanski on hypercube routing. *Information Processing Letters*, 35:57–61, 1990.

[7] A.P. Sprague and H. Tamaki. Routing for involutions of a hypercube. *Discrete Applied Mathematics*, 48:175–186, 1994.

[8] T. Szymanski. On the permutation capability of a circuit-switched hypercube. In *Proc. Internat. Conf. on Parallel Processing*, pages I–103 – I–110, 1989.

[9] K. Zemoudeh and A. Sengupta. Routing frequently used bijections on hypercube. In *DISTMEMCC: 5th Distributed Memory Computing Conference*, pages 824–832. IEEE Computer Society Press, 1990.

5 Appendix : the 72 failing 2-1 routing requests

In this Appendix, the 72 2-1 routing requests which failed to get through our algorithm (cf. Section 2.1) are given. Note that there are three such 2-1 routing requests on one row. For a better understanding, let us explain how a 2-1 routing request is denoted in the following. For this, let us take an example. For instance, take the 2-1 routing request ρ described below :

$$5 \times \times 4\ 3\ 6 \times 1 \times \times \times \times 7\ 2 \times 0$$

In that case, ρ can also be viewed as corresponding to the table given in Figure 9.

x	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
$\rho(x)$	5	×	×	4	3	6	×	1	×	×	×	×	7	2	×	0

Fig. 9. A 2-1 routing request ρ in H_3

If, in the table, $\rho(x) = \times$ twice for some x, then this means that x is not a source in ρ. Otherwise, x is a source in ρ and $\rho(x)$ can be read from the table. This gives the 2-1 routing request ρ below (Figure 10).

x	0	1	2	3	4	5	6	7
$\rho(x)$	5			4	3	2		0
					7	6		1

Fig. 10. Another way to see ρ

The 72 2-1 routing requests that failed to get through our algorithm are then the following.

× × 5 4 3 6 1 × × × 7 × 2 × 0 ×	× 2 5 4 3 × 1 × × × 7 6 × × 0 ×	× 2 5 4 × 3 1 × × 6 7 0 × × × ×
× × 5 4 7 3 × 1 × × × 6 × 2 × 0	× × 7 4 3 6 × 1 × × × 5 × 2 × 0	× × 4 6 7 2 1 × × × 5 × 3 × 0 ×
× 4 × 5 7 2 1 × × 6 × × 3 0 × ×	× 4 5 × 3 6 1 × × × × × 7 2 0 ×	× 6 × 4 3 2 5 × × 7 × × 1 0 × ×
× 6 1 4 3 2 × × × 7 × 5 × 0 × ×	× 6 1 4 × 2 3 × × 7 5 0 × × × ×	× 6 × 4 7 × 3 2 × × × 5 × × 1 0
× 6 4 × 3 2 5 × × × × × 7 0 1 ×	× 6 5 × 3 2 × 4 × × 7 × × 0 × 1	× 6 5 × 3 × 1 4 × 7 × × × × 0 2
× 6 5 4 1 2 × × × 7 × 0 × 3 × ×	× 6 5 4 2 × 3 × × × 7 0 × × 1 ×	× 6 5 4 3 × × 2 × × × 0 7 × × 1
× 6 7 × × 3 5 4 × × × × × 2 1 0	× 7 × 4 3 × 5 2 × × × 6 × × 1 0	× 7 5 × × 6 3 4 × × × × × 2 1 0
× 2 4 × 3 × 5 0 × 6 7 × × × 1 ×	× 2 5 4 3 × × 0 × 7 × 6 1 × × ×	× 4 5 × 3 6 × 0 × 7 1 × × 2 × ×
× 4 5 × 7 × 3 0 × 6 × × 2 × 1 ×	× 6 4 × 7 3 × 0 × × 5 × 1 2 × ×	× 6 7 4 2 × × 0 × × 1 5 3 × × ×
3 × 5 4 × 2 1 × 6 × 7 × × 0 × ×	3 × 5 4 × 2 × 1 × × 7 6 × × × 0	3 × × 6 × 2 1 4 7 × × 5 × × × 0
3 × 5 4 2 × × 0 7 × 1 6 × × × ×	3 2 × 5 1 × × 0 7 6 × 4 × × × ×	3 2 4 × × 0 1 × 7 6 5 × × × × ×
3 4 1 × × 2 0 × 7 6 5 × × × × ×	3 4 5 × × 2 × 1 7 6 × × × 0 × ×	3 6 × 4 × 2 1 × 5 7 × × × × 0 ×
3 6 × 4 × × 1 2 × 7 × 5 × × × 0	3 6 4 × × × 1 2 7 × 5 × × × 0 ×	3 6 5 × 1 × × 4 7 × × × 2 × × 0
5 × × 4 3 6 × 1 × × × × 7 2 × 0	5 × × 4 7 2 1 × 6 × × 0 3 × × ×	5 × × 4 × 6 1 2 7 × × × × 3 × 0
5 × 4 × 3 6 × 0 7 × × × 1 2 × ×	5 2 × 4 3 × × 1 7 6 × 0 × × × ×	5 2 × 4 3 × 0 × 7 6 × × 1 × × ×
5 4 1 × × 3 × 0 7 6 × × × 2 × ×	5 4 × 0 3 × 1 × 7 6 × × 2 × × ×	5 6 × × 3 × 1 4 7 × × × 2 × 0 ×
5 6 × × × 3 1 4 × 7 × × × 2 × 0	5 6 1 × 3 × × 2 7 × 4 × × × × 0	6 × 1 4 3 × × 2 7 × 5 0 × × × ×
6 × × 4 3 × 5 2 × × × × 7 × 1 0	6 × × 4 × 2 5 1 7 × × × × × 3 0	6 × 5 × 3 2 × 4 7 × × × 1 0 × ×
6 × 5 × × 2 3 4 × × 7 × × × 1 0	6 × 4 0 3 2 × × 7 × 5 × 1 × × ×	6 × 5 4 2 0 × × 7 × 1 × 3 × × ×
6 2 4 × × × 3 0 7 × 5 × × × 1 ×	6 2 5 × 3 × × 1 7 4 × × × × × 0	6 4 × × 3 × 5 0 7 × × × 2 × 1 ×
7 × 1 4 × 6 3 × × × 5 × × 2 0 ×	7 × × 4 3 2 5 × × × × 6 1 × 0 ×	7 × 1 4 × 3 × 2 × × 5 6 × × × 0
7 × × 5 3 2 × 4 × × × × 1 6 × 0	7 × × 6 2 × 5 4 × × × × 3 × 1 0	7 × × 4 3 6 1 × × × × 5 2 0 × ×
7 × 4 × × 6 3 2 × × 5 × × × 1 0	7 × 4 6 × 3 1 × × × 5 0 × 2 × ×	7 2 × 4 × 3 5 × × 6 × × × 0 1 ×
7 2 × 4 × × 3 1 × 6 × 5 × × × 0	7 4 × × × 3 5 1 × 6 × × × 2 × 0	7 4 × 5 × 2 3 × × 6 × 0 × × 1 ×