BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 63, No. 3, 2015
DOI: 10.1515/bpasts-2015-0085

Towards the boundary between easy and hard control problems in multicast Clos networks

P. OBSZARSKI, A. JASTRZĘBSKI, and M. KUBALE*

Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology,
11/12 Gabiela Narutowicza St., 80-233 Gdańsk, Poland

Abstract. In this article we study 3-stage Clos networks with multicast calls in general and 2-cast calls, in particular. We investigate various sizes of input and output switches and discuss some routing problems involved in blocking states. To express our results in a formal way we introduce a model of hypergraph edge-coloring. A new class of bipartite hypergraphs corresponding to Clos networks is studied. We identify some polynomially solvable instances as well as a number of NP-complete cases. Our results warn of possible troubles arising in the control of Clos networks even if they are composed of small-size switches in outer stages. This is in sharp contrast to classical unicast Clos networks for which all the control problems are polynomially solvable.

Key words: Clos network, 2-cast call, hypergraph edge-coloring, rearrageable network, nonblocking network, NP-completeness, 3-uniform hypergraph.

1. Introduction

Clos networks were introduced by Charles Clos in his seminal paper [1]. However, our basic notation and terminology follows that of Hwang [2]. In this article we discuss 3-*stage* Clos networks which consist of several switches arranged in 3 stages. Interconnection networks of this type are characterized as follows:

- The first (input) stage consists of r_1 switches (crossbars) each with n_1 inputs and m outputs. We say that such a switch is of size $n_1 \times m$.
- The second (middle) stage consists of m switches each of size $r_1 \times r_2$.
- The third (output) stage consists of r_2 switches each of size $m \times n_2$.
- There exists exactly one link between each middle switch and each input and output switch.

Clos network with these parameters is denoted $C(n_1, r_1, m, n_2, r_2)$. An example of such a network is presented in Fig. 1.

Clos networks are theoretical idealization of practical multistage switching systems, which means that each connection from an inlet on input switch to an outlet on output switch consists of links that are not to be shared with any other connections.

We recall that in multicast calls each request is a pair of one idle (unused) inlet and a set of idle outlets. A multicast call is said to be *f-cast* if f outlets can be requested in one call. To allow multicast connections, the switches with fan-out capability are required. A switch with *fan-out* property is capable of distributing a signal from one input to a few outputs. In our model we assume that only switches in the second and the third stage have such a property (so-called model 1 in [3]). Accordingly, a Clos network having switches with fan-out property in the middle and output switches which are capable of implementing 2-cast calls is said to be a 2-*cast Clos network*. Furthermore, such a network is called *rearrangeable* if an idle inlet can always be connected to a couple of idle outlets but for this to take place, existing calls may have to be moved by assigning them to different middle stage switches in the network. If, however, an idle inlet can be connected to a couple of idle outlets without having to rearrange existing calls then such a 2-cast Clos network is called *nonblocking*. Some sufficient conditions for a 2-cast Clos network to be rearrangeable and/or nonblocking are given in Sec. 3.

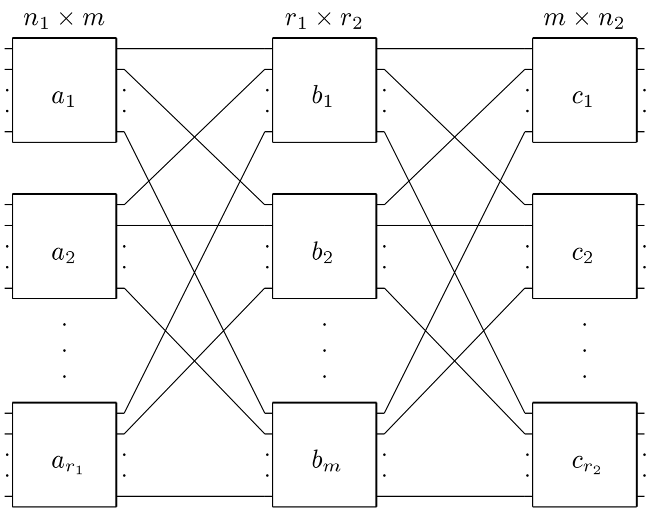

Fig. 1. Example of Clos network $C(n_1, r_1, m, n_2, r_2)$

In general, considerations devoted to combinatorial properties of 2-cast Clos networks may be headed into two di-

*e-mail: kubale@eti.pg.gda.pl

rections. First, we may try to estimate a sufficient number of middle-stage switches that need to be installed in order to guarantee that the network is rearrangeable or nonblocking. This problem is briefly discussed in Sec. 3. Secondly, given a Clos network, we may try to design a routing algorithm that attempts to minimize the number of middle switches used. In this article we mainly point out in Secs. 4 and 5 several reasons of intractability involved in the second approach. The fact that rearranging multicast Clos networks is NP-complete was first noticed in [4]. In this paper, however, we go a step further by approaching the boundary between easy and hard problems arising in the control of such networks and showing that almost all blocking states of $C(2, r_1, m, 2, r_2)$, $m \geq 3$ can be unblocked.

The remainder of the paper is organized as follows. In the next section we introduce a mathematical model (hypergraph edge-coloring) for analyzing the states of 2-cast Clos networks. Section 3 is devoted to some bounds on the number of middle switches for a Clos network to be rearrangeable. The main results of this paper are given in the last two sections. In Sec. 4 we point out several reasons of intractability involved in the control of 2-cast Clos networks. In Sec. 5 we attempt to elaborate the border between polynomially solvable and NP-hard control problems and point out special cases that can be solved efficiently in linear time.

2. Edge-coloring of hypergraphs

The problem of routing multicast Clos networks can be modeled by a hypergraph edge-coloring. In particular, 2-cast Clos network can be modeled by an edge-coloring of 3-uniform hypergraph, which is introduced in this section.

Let $H = (V, E)$ be a *hypergraph*, where $V = V(H)$ is a set of *vertices* and $E = E(H)$ is a multiset of nonempty subsets of V called *hyperedges* or simply *edges*. We say that a hyperedge e and a vertex v are *incident* if $v \in e$. Also, two edges that have a vertex in common are said to be *adjacent*. A *d-edge* is a hyperedge that contains exactly d vertices. We say that $d = |e|$ is the *dimension* of hyperedge e. If all edges of a hypergraph are of the same dimension d then the hypergraph is said to be *d-uniform*. A hypergraph is *simple* if each edge is unique (in the sense of vertices it contains). A hypergraph H is *linear* if each pair of edges share at most one vertex. Linearity is stronger than simplicity in the sense that each linear hypergraph is also simple. Notice that simple (linear) 2-uniform hypergraphs are just graphs.

The *degree* $\deg(v)$ of a vertex $v \in V$ is the number of edges in which v occurs. $\Delta(H) = \max_{v \in V} \deg(v)$ is the *degree* of H.

A *proper edge-coloring* (*k-edge-coloring*) of a hypergraph H with k colors is a function $c : E(H) \rightarrow \{1, ..., k\}$ such that no two adjacent hyperedges get the same color (number). A coloring that uses the minimum number of colors is called *optimal*. The *chromatic index* $\chi'(H)$ of H is defined to be the number of colors in an optimal coloring of H.

For a hypergraph H, the *line graph* $L(H)$ is a simple graph representing adjacency between hyperedges in H. More precisely, each hyperedge of H is assigned a vertex in $L(H)$ and two vertices in $L(H)$ are adjacent if and only if their corresponding hyperedges in H have a vertex in common. It is easy to notice that an edge-coloring of a hypergraph H is equivalent to vertex-coloring of its line graph $L(H)$. See Fig. 2a,b for an example of a hypergraph and its line graph.

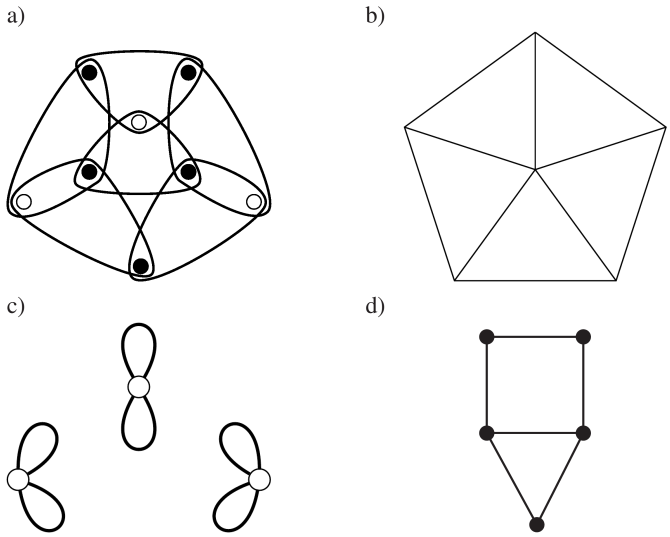

Fig. 2. Example of bipartite hypergraph and some related graphs: a) hypergraph H. Black vertices belong to out-partition, white vertices belong to in-partition, $\Delta(H) = 3$; b) line graph $L(H)$, $\Delta(L(H)) = 5$; c) graph induced by set V_i of H, $H(V_i)$. $\Delta_i = 2$; d) Graph induced by set V_o of H, $H(V_o)$. $\Delta_o = 3$

In this paper a special class of 3-uniform hypergraphs is considered. For this reason we need some additional notions. In particular, we say that a hypergraph is *bipartite* if its vertex set can be split into two partitions in such a way that each edge has exactly one vertex in the first partition and two vertices in the second one. In connection with Clos networks terminology we call the first partition the *in-partition* and the second one the *out-partition*. We denote these partitions V_i and V_o, respectively. An example of a bipartite hypergraph is shown in Fig. 2a. The maximum degree of a vertex in the in-partition is called the *in-degree* and denoted Δ_i, while the maximum degree of a vertex in the out-partition is called the *out-degree* and denoted Δ_o. Given a bipartite hypergraph H, by $H(V_i)$ ($H(V_o)$) we mean the graph induced by set V_i (V_o), respectively. More precisely, $H(V_o)$ is a graph or multigraph on $|V_o|$ vertices and $|E(H)|$ edges in which each 3-edge $\{u, v, w\}$ of H, $u \in V_i$, $v, w \in V_o$ generates one edge $\{v, w\}$ in $H(V_o)$, while $H(V_i)$ is a graph consisting of $|V_i|$ isolated vertices with bunches of $|E(H)|$ 1-edges (loops) on them, one loop $\{u\}$ for each 3-edge $\{u, v, w\}$ of H, $u \in V_i$. An example of such graphs is depicted in Fig. 2c,d.

Edge-coloring of hypergraphs can be applied to model the operation of multicast Clos networks in the following manner. Input and output switches correspond to vertices of a hypergraph, connecting paths and connection requests are represented by hyperedges. The middle switch participating in a connection determines the color of the corresponding hyperedge. An example of reduction from a 2-cast Clos network to edge-coloring of a 3-uniform hypergraph is presented in Fig. 3a,b.

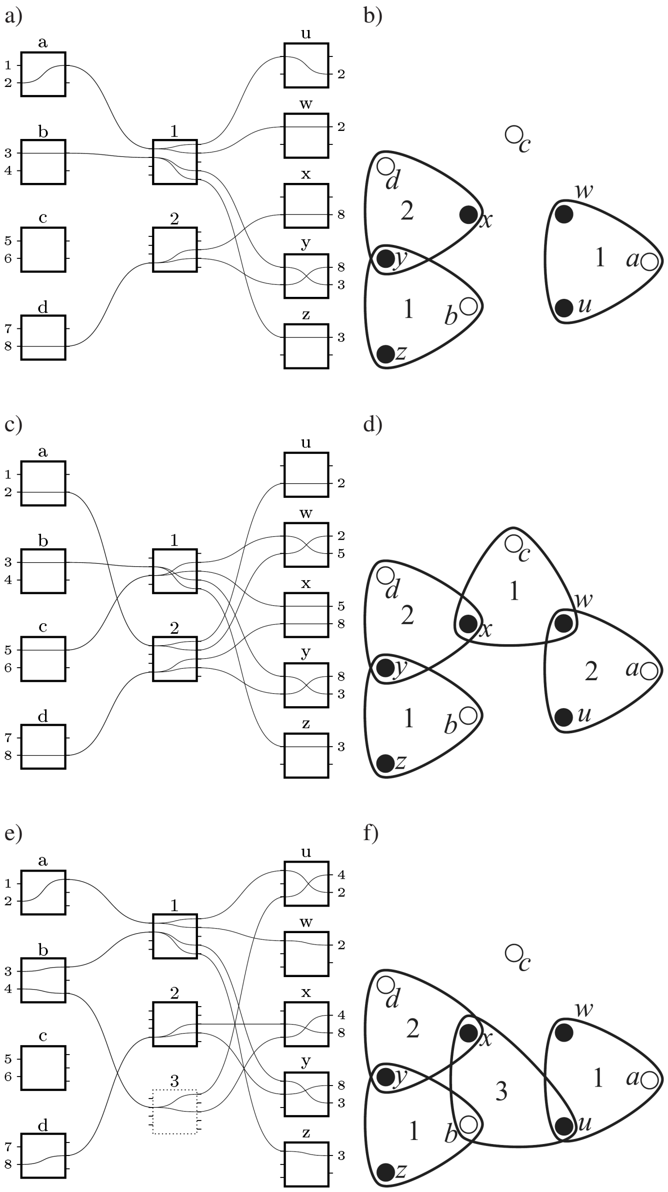

Fig. 3. Example of 2-cast Clos network and corresponding hypergraphs: a) Clos 2-cast network $C(2,4,2,2,5)$ with 3 connecting paths; b) hypergraph corresponding to the Clos network (a) (vertices from V_o are black and from V_i are white) and its coloring (numbers inside hyperedges); c) Clos network (a) that needed to be rerouted after one call was added (from c to x and w); d) hypergraph corresponding to the Clos network (c) and its coloring; e) Clos network (a) with one additional call (from b to u and x) that cannot be rerouted and needs the 3rd middle switch; f) hypergraph corresponding to the Clos network (e) and its coloring

In the following we mainly discuss 3-uniform hypergraphs. Henceforth, by the term hypergraph we mean a 3-uniform hypergraph, unless otherwise stated. The only exception is made for propositions, conjectures and theorems, where we use the full specification of a hypergraph at hand. Observe that although we simplify the model by omitting 2-edges (that correspond to 2-cast calls with outlets on the same output switch) we can take them into account, since each 2-edge can be extended to 3-edge by introducing one dummy vertex into it.

Although our considerations are focused on 2-cast Clos networks, they can be easily generalized to multicast Clos networks. In fact, given a set of 2-cast calls modeled by 3-edges of a hypergraph H, we can introduce to each hyperedge of H one dummy vertex. The new hypergraph I with 4-edges has the line graph $L(I)$ isomorphic to $L(H)$. Therefore, edge-coloring of H is equivalent to edge-coloring of I. Consequently, a larger $(k+1)$-cast Clos network can mimic the operation of smaller k-cast network for any $k \geq 2$.

3. Upper bounds on the chromatic index

Proper estimation of the number of switches in the central stage is essential in the phase of designing a Clos network structure. In this section we present some relevant conjectures and bounds on that number. Note that the minimal number of switches in the middle stage of a rearrangeable multicast network cannot be less than $\max\{\chi'(H)\}$, where maximum is among all possible hypergraphs realized by the network.

We start with a basic bound that applies to all 3-uniform hypergraphs.

Proposition 1. Each 3-uniform bipartite hypergraph H can be edge-colored with $3\Delta(H)-2$ colors.

To justify this fact it is enough to notice that each vertex of a 3-edge e is of degree at most $\Delta(H)$. Therefore, edge e has at most $3\Delta(H)-3$ neighboring edges and at worst a color $3\Delta(H)-2$ is free for e. This bound guarantees that a new request can never be blocked regardless of the current state of Clos network $C(n_1,r_1,m,n_2,r_2)$ provided that $m \geq 3\max\{n_1,n_2\}-2$.

There is a much stronger conjecture formulated in [5]. Unfortunately, the problem has been open for more than two decades.

Conjecture 1 [5]. Every bipartite 3-uniform hypergraph is (2Δ)-edge-colorable.

In terms of Clos network the truth of Conjecture 1 implies that $C(n_1,r_1,m,n_2,r_2)$ is rearrangeable if $m \geq 2\max\{n_1,n_2\}$. For some special cases the conjecture has been proven to be true. For example:

- $\Delta_i \geq \Delta_o$ and $\Delta_o \leq 3$ [6]
- $\Delta_i > \Delta_o$ and $|V_o| \leq 4$ [7]
- $|V_i| \leq 4$ [8]

A generalization of Hwang-Lin's conjecture has been proposed in [6].

Conjecture 2. If $\Delta_i \geq \Delta_o$ then bipartite 3-uniform hypergraph is $(\Delta_o+\Delta_i)$-edge-colorable.

Based on some computational experiments, we propose even stronger conjecture.

Conjecture 3. If $\Delta_i \geq 2\Delta_o$ then bipartite 3-uniform hypergraph is Δ_i-edge-colorable.

If our conjecture is true then so are Conjectures 1 and 2.

4. Complexity of edge-coloring in special cases

In this section we discuss the cases in which $H(V_o)$ forms some highly structured graphs. We give a detailed proof of NP-completeness of deciding the 3-edge-colorability of H when the $H(V_o)$ graph is composed of cycles. However, we also consider forests, trees, collections of cliques and collections of multipaths. Finally we indicate a polynomial-time algorithm for the case in which $H(V_o)$ is a collection of stars.

Theorem 1. It is NP-complete to decide whether a bipartite 3-uniform hypergraph H with $\Delta_i = 3$ has a 3-edge-coloring in the following cases:

a) H is linear and $H(V_o)$ is a collection of cycles C_4,
b) H is linear, $H(V_o)$ is a forest and $\Delta_o = 3$,
c) H is linear, $H(V_o)$ is a caterpillar and $\Delta_o = 3$,
d) H is simple and $H(V_o)$ is a collection of complete graphs K_4,
e) H is simple, $H(V_o)$ is a collection of multipaths on three vertices and $\Delta_o = 3$,
f) H is simple, $H(V_o)$ is a multipath and $\Delta_o = 3$.

Proof. Obviously, the problem is in NP, since checking whether a coloring is proper requires only verifying if adjacent edges are of different color. Below we consider 6 cases. In each case the proof is based on the fact that 3-edge-coloring of a cubic graph is an NP-complete problem [9].

Case (a) Let us consider a cubic graph G. We transform G to a 3-uniform and bipartite hypergraph H by replacing each edge $\{u, v\} \in E(G)$ with a gadget shown in Fig. 4. Dashed line marks a graph C_4 in $H(V_o)$. The gadget is joint to the graph via vertices u and v. White vertices constitute the in-partition. Note, that $\Delta_i = 3$ as the initial graph is cubic. The black vertices stand for the out-partition. Together with the hyperedges of gadget they form cycles C_4, one for each gadget. Finally, we have to explain why graph G is 3-colorable if and only if edges of the corresponding hypergraph H can be colored with three colors. First, let us assume that we know a 3-edge-coloring of G. Then, consider each gadget separately, and color the hypergraph H in the following manner.

- the far left hyperedge and the far right hyperedge get the same color as that of the edge $\{u, v\}$ of G replaced by the gadget,
- two hyperedges inside the gadget get the remaining two colors.

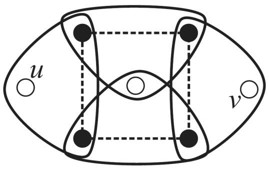

Fig. 4. Gadget for cycle. The dashed lines mark underlying cycle C_4

One can easy see that each gadget of hypergraph is properly colored with three colors. Hence the whole hypergraph can also be colored with three colors.

Now let us assume that we have a 3-edge-coloring of the hypergraph H. The crucial point is that hyperedges on the both sides of a gadget have to be colored with the same color in each proper 3-edge-coloring. Otherwise, since the inner side hyperedges are adjacent to both outer side hyperedges and to each other, the fourth color would have to be used. On the basis of this fact, all we need to do is to color the edges of graph G with the colors of gadgets' outer side hyperedges.

Case (b) Given any cubic graph G, replace each edge $\{u, v\}$ of it with the gadget presented in Fig. 5. The dashed lines mark a tree in $H(V_o)$. We connect the gadget via vertices u and v. Bearing in mind the proof of case (a), to make this one valid we need to show that the gadget can be edge-colored with three colors and each proper 3-coloring requires edges e_1 and e_2 to be of the same color. Edges e_1, e_3 and e_4 are adjacent to each other, so they should be colored with three distinct colors. Let us assume that this colors are c_1, c_2 and c_3, respectively. Observe that e_5 needs to be colored with the same color as e_1 in each 3-coloring, so it gets c_1. This and the fact that e_6 is colored with c_2, implies that e_7 is colored with c_3. So edge e_2 in each proper 3-edge-coloring must be colored with c_1.

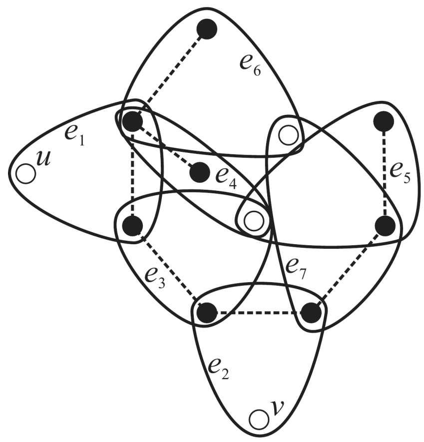

Fig. 5. Gadget for forest. The dashed lines mark an underlying tree

Case (c) We have proven that the problem is NP-hard for a set of trees (forest). It occurs that it is enough to connect gadgets together via a vertex of degree 1 in edges e_6 or e_5 using some additional edges in order to get a caterpillar (in $H(V_o)$), i.e. a tree such that deleting vertices of degree 1 results in a path.

Case (d) The proof remains very similar to that for cycles C_4. Obviously, the gadget is different (see Fig. 6) but argument remains the same. Hence, we leave detailed considerations to the reader.

Case (e) Yet another proof similar to that of case (a). The gadget that replaces edges of G is presented in Fig. 7. Observe that $H(V_o)$ is a collection of multipaths, where a multipath is a path in which parallel edges are allowed.

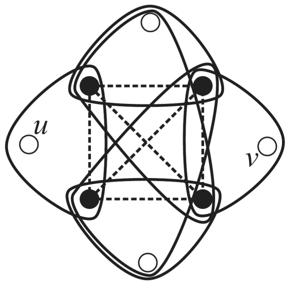

Fig. 6. Gadget for complete graphs. The dashed lines mark underlying K_4

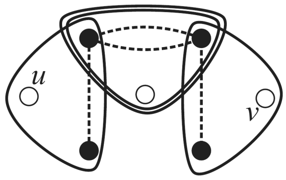

Fig. 7. Gadget for multipath. The dashed lines mark an underlying multipath

Case (f) Notice that if we connect gadgets from Fig. 7 together via vertices of degree 1 and some additional hyperedges we get a single multipath in $H(V_o)$.

It has been shown in the proof of Theorem $1(a)$ that in considered hypergraphs $\Delta_i = 3$. Also note that $\Delta_o = 2$, since $H(V_o)$ is a collection of cycles in that case. Hence we have:

Corollary 1. Edge-coloring of bipartite 3-uniform hypergraphs is NP-complete even for hypergraphs with $\Delta_i = 3$ and $\Delta_o = 2$.

In practice this means that there exists no polynomial-time algorithm for rearranging general multicast Clos networks unless P = NP. This is in sharp contrast to unicast Clos networks for which we have rearranging algorithms that run in polynomial time [10, 11].

However, the problem becomes polynomial if both the in-degree and the out-degree are bounded by 2. Then we have the degree configuration $\Delta_i = \Delta_o = 2$, which is claimed to be polynomial in Sec. 5. The edge-coloring problem also becomes polynomial if we decrease the size of cycles and consider a collection of triangles C_3 in the out-partition (for any in-degree). This situation is similar to that described in Theorem 2.

Let us slightly simplify the problem from Theorem $1(e)$ to the one with $H(V_o)$ being a collection of multipaths P_3. It is not difficult to see that the problem becomes polynomial. First observe, that the dependencies in terms of edge-coloring are the same for multipaths on three vertices, multicycles based on C_3 and stars. Simply each edge is then adjacent to each other. In the following we look closer to the cases with $H(V_o)$ composed of a number of stars.

Theorem 2. There exists an $O(|E(H)| \log \Delta(H))$-time algorithm for optimal edge-coloring of a bipartite 3-uniform hypergraph H in which $H(V_o)$ is a collection of stars.

Proof. Consider a bipartite hypergraph H in which a set of stars constitutes the out-partition. Notice that in such a configuration each hyperedge has at least one vertex of degree 1. Furthermore, such a vertex must be in out-partition and it is a leaf of a star in $H(V_o)$. Since such vertices do not influence edge-coloring so we get rid of them and leave 2-edges only (see Fig. 8). Obviously, the remaining graph is a bipartite multigraph. Clearly, the edge-coloring of H is equivalent to edge-coloring of such a multigraph, which can be obtained in time $O(|E| \log \Delta)$ [12].

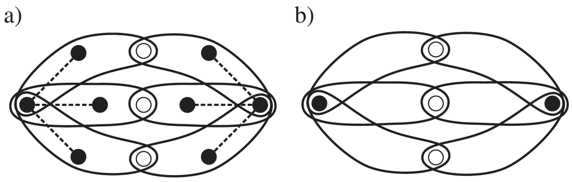

Fig. 8. Illustration for the proof of Theorem 2: a) example of a bipartite hypergraph, b) bipartite multigraph corresponding to (a)

5. Complexity status for bounded degree bipartite hypergraphs

In relation with Clos networks the restriction on Δ_i may stand for the maximal number of calls from a single switch of the first stage. Similarly Δ_o restricts the number of calls leaving any switch form the third stage. We may write that considered networks are of type $C(\Delta_i, r_1, m, \Delta_o, r_2)$.

Table 1 collects the information about the complexity status for various combinations of Δ_i and Δ_o. Below we explain the crucial points of this table.

Table 1
The complexity of edge-coloring depending on the in- and out-degrees for 3-uniform bipartite hypergraphs

	$\Delta_i = 1$	$\Delta_i = 2$	$\Delta_i \geq 3$
$\Delta_o = 1$	trivial	linear	linear
$\Delta_o = 2$	linear	linear[a]	NP-hard
$\Delta_o \geq 3$	NP-hard	NP-hard	NP-hard

[a] [14]

In the cases with $\Delta_o = 1$, the $H(V_o)$ graph is simply a matching. Then the model describes bunches of hyperedges sharing at most one vertex from V_i. In this case all hyperedges sharing the same vertex must get various colors.

If $\Delta_i = 1$, the in-partition consists of r_1 vertices, each of which is either isolated or incident with one hyperedge. As noticed in the proof of Theorem 2, such vertices do not influence the edge-coloring. This implies that this is the structure of $H(V_o)$ that determines the hypergraph colorability. Although the classical edge-coloring problem is linearly solvable for graphs with $\Delta = 2$ and NP-hard if $\Delta \geq 3$ [9], the edge-coloring of bipartite hypergraphs is linearly solvable for $\Delta_i = 1$ and $\Delta_o \leq 2$, and it is NP-hard if $\Delta_i = 1$ and $\Delta_o \geq 3$. If $\Delta_i = \Delta_o = 1$, the edge-coloring problem becomes trivial, since H consists of isolated hyperedges. Of course, the $C(1, r_1, m, 1, r_2)$ networks have no practical significance.

If $\Delta_i = \Delta_o = 2$, then the line graph of such a bipartite hypergraph is of degree at most 3. If such a line graph is K_4-free then, in virtue of Brooks' theorem [13], it has a vertex-coloring with 3 colors which can be found in linear time [14]. Since such a coloring is equivalent to 3-edge-coloring of H, it follows that there exists an efficient algorithm for rearranging 2-cast networks $C(2, r_1, 3, 2, r_2)$, unless $L(H)$ contains K_4. An example of the smallest network with such a blocking state is shown in Fig. 9. Note that 2-cast networks $C(2, r_1, 4, 2, r_2)$ are already nonblocking.

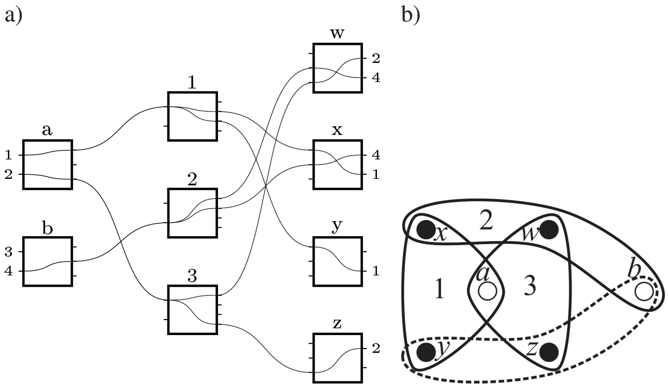

Fig. 9. a) Clos network $C(2, 2, 3, 2, 4)$ with 3 connecting paths. A new call (form b to y and z) would be blocked, b) corresponding hypergraph. The new call is marked with a dashed cycle

The fact that the problem is NP-complete for $\Delta_o = 2$ and $\Delta_i = 3$ is claimed in Corollary 1.

Acknowledgements. The authors wish to thank anonymous referees for their valuable comments. This article has been partially supported by National Science Centre under the grant DEC-2011/02/A/ST6/00201.

REFERENCES

[1] C. Clos, "A study of nonblocking switching networks", *Bell Syst. Tech. J.* 32, 406–424 (1953).

[2] F.K. Hwang, *The Mathematical Theory of Nonblocking Switching Networks*, World Scientific, London, 2004.

[3] F.K. Hwang, "A unifying approach to determine the necessary and sufficient conditions for nonblocking multicast 3-stage Clos networks", *IEEE Trans. on Commun.* 53, 1581–1586 (2005).

[4] A. Jastrzębski and M. Kubale, "Rearrangeability in multicast Clos networks is NP-complete", *Proc. 2-nd Int. Conf. on Information Technology* 1, 183–185 (2010).

[5] F.K. Hwang and C.H. Lin, "Broadcasting in a three-stage point-to-point nonblocking network", *Int. J. Rel. Qual. Safety Eng.* 2, 299–307 (1995).

[6] D.Z. Du and H.Q. Ngo, "An extension of DHH-Erdos conjecture on cycle-plus-triangle graphs", *Taiwan J. Math.* 6, 65–105 (2002).

[7] F.K. Hwang, S.-C. Liaw, and L.D. Tong, "Strictly nonblocking 3-stage Clos networks with some rearrangeable multicast capability", *IEEE Trans. Commun.* 6, 261–267 (2002).

[8] H.-L. Fu and F.K. Hwang, "On 3-stage clos networks with different nonblocking requirements on two types of calls", *J. Comb. Opt.* 9, 263–266 (2005).

[9] I. Holyer, "The NP-completeness of edge-colouring", *SIAM J. Comput.* 10, 718–720 (1981).

[10] M. Kubale, "Average and worst-case performance of Paull's algorithms for rearranging three-stage connection networks", *Annales Des Telecommunications* 40, 270–276 (1985).

[11] M.C. Paull, "Reswitching of connection networks", *Bell Syst. Tech. J.* 41, 833-855 (1962).

[12] R. Cole, K. Ost, and S. Schirra, "Edge-coloring bipartite multigraphs in O(E logD) time", *Combinatorica* 21, 5–12 (2001).

[13] R.L. Brooks, "On colouring the nodes of a network", *Proc. Cambridge Philosophical Society, Math. Phys. Sci.* 37, 194–197 (1941).

[14] S. Skulrattanakulchai, "Δ-list vertex coloring in linear time", *LNCS* 2368, 240–248 (2002).