Approximable Row-Column Routing Problems in All-Optical Mesh Networks (revisited)

Jérôme Palaysi, Olivier Cogis, Guillaume Bagan

Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier

LIRMM, Univ. Montpellier 2, CNRS

RAPPORT DE RECHERCHE

Approximable Row-Column Routing Problems in All-Optical Mesh Networks (revisited)

Guillaume Bagan, Olivier Cogis and Jérôme Palaysi

may 2007

R.R.LIRMM RR-07012

161, Rue Ada - 34392 Montpellier Cedex 5 - France
Tél: +33 (0)4 67 41 85 85 - Fax: +33 (0)4 67 41 85 00

Abstract

In all-optical networks, several communications can be transmitted through the same fiber link provided that they use different wavelengths. The MINIMUM ALL-OPTICAL ROUTING problem (given a list of pairs of nodes standing for as many point to point communication requests, assign to each request a route along with a wavelength so as to minimize the overall number of assigned wavelengths) has been paid a lot of attention and is known to be $\mathcal{NP}$*–hard*. Rings, trees and meshes have thus been investigated as specific networks, but leading to just as many $\mathcal{NP}$*–hard* problems.

This paper investigates row-column routings in meshes (paths are allowed one turn only). We first show the MINIMUM LOAD ROW-COLUMN ROUTING problem to be $\mathcal{NP}$*–hard* but $2\text{–}APX$ (more generally, the MINIMUM LOAD k-CHOICES ROUTING problem is $\mathcal{NP}$*–hard* but $k\text{–}APX$), then that the MINIMUM ROW-COLUMN PATHS COLOURING problem is $4\text{–}APX$ (more generally, any d-segmentable routing of load L in a hypermesh of dimension d can be coloured with $2d(L-1)+1$ colours at most). From there, we prove the MINIMUM ALL-OPTICAL ROW-COLUMN ROUTING problem to be APX.

keywords: minimum load routing, minimum path colouring, all-optical networks, mesh, row-column routing, approximation algorithms

1 Introduction

In optical networks, links are optical fibers. Each time a message reaches a router, it is converted from optical to electronic state and back again to optical state. These electronic switchings are considered as bottlenecks for the network.

Contrary to optical networks which use expensive optoelectronic conversions, all-optical networks allocate a physical path in the network to each communication request, as for usual circuit switching; when each router is set up, messages can stay in their optical state from start to end. The all-optical network communication nodes we are interested in are Wavelength Routing Optical Cross-connect (WR-OXC) with Optical Add/Drop Multiplexer (OADM) (see for instance [1]). An example of such a router is depicted in figure 1.

Wavelength Division Multiplexing (WDM) is a technique (see for instance [2]) that proposes to take advantage of the huge optical fiber bandwidth by allocating a unique frequency to each communication. Several communications can simultaneously use the same fiber as long as their wavelengths are different.

In this context, *networks* can be viewed as *graphs*, whether directed or not, and ***communication requests*** in the network as *pairs of nodes* of the graph. A ***communication instance*** can then be defined as a graph together with a family of pairs of nodes (pairs may be repeated in the given family of requests). Given some communication instance, a ***routing*** for this instance can be defined as a family of paths in the graph yielded by linking the two nodes of each request by a path in the graph[1], and an ***all-optical routing*** for this instance

[1] When two different requests are made of the same pair of nodes, they may be assigned

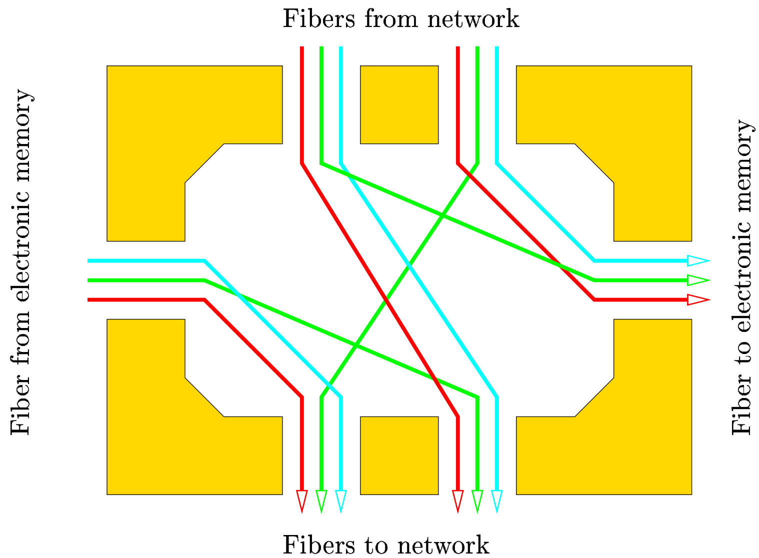

Figure 1: As an example, a router WR-OXC with OADM dedicated to directed communications.

is a routing for this instance where each routing path is assigned a *colour*[2] in such a way that no two paths using a common edge bear the same colour.

As wavelengths are usually a critical resource, the ***minimum all-optical routing problem*** is the *optimization problem* defined as: given some communication instance, compute an all-optical routing for this instance which minimizes the overall number of colours used to label the routing paths. An optimal solution to a minimum all-optical routing problem will be called a ***minimal all-optical routing*** (see figure 2 for an example).

The minimum all-optical routing problem is $\mathcal{NP}$*–hard* in general, whether graphs are directed [3] or not [4, 5, 3]. Moreover, restricted to directed graphs, the problem is known to be *No-APX*[3] [1, corollary 3.1.5]. Therefore some topologies have been selected to be paid specific attention.

When networks are linear (i.e. the graph is a path), the problem is equiv-

different paths in the graph.

[2] When k colours are used to label the routing paths, it is not uncommon to use integers 1 to k as colours, though basically the set of colours is not an ordered set (on the other hand, referring to the i^{th} colour becomes handy when expressing some algorithm making use of colours).

[3] For more about approximation theory, the reader can be referred to [6]. For short, given some $\mathcal{NP}$*–minimization problem* and some real number d, a polynomial algorithm A is said to be a *d–approximation algorithm* for the problem, and the problem is then said to be d–APX (or simply APX if the exact value of d is not under consideration), when, given any instance I of the problem, one has $\frac{A(I)}{OPT(I)} \leq d$, where $A(I)$ is the cost of the solution computed by algorithm A, and $OPT(I)$ is the cost of an optimal solution ($OPT(I)$ is always assumed to be strictly positive). If no such d exists, then the problem is said to be *No-APX*.

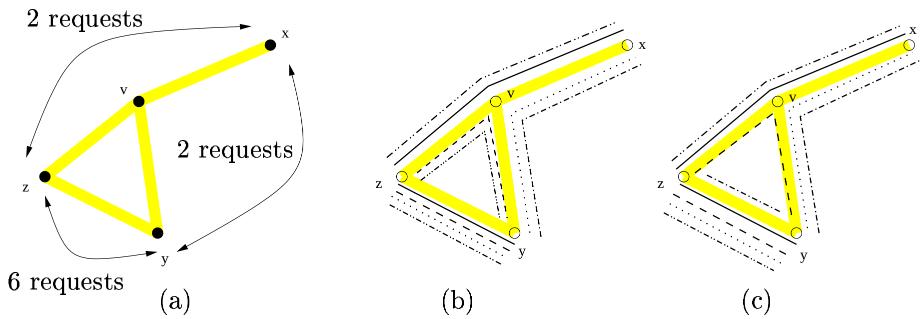

Figure 2: Figure (a) shows a communication instance I. Figure (b) and (c) show all-optical routing R_b and R_c resp. which are solutions to I. R_c is a minimal all-optical routing for I, but R_b is not (R_b, resp. R_c, makes use of 6 colours, resp. 5). On the other hand, R_b is a minimum load routing for I while R_c is not (R_b makes every link support 4 colours, R_c makes link zy support 5 colours).

alent to the interval graph vertex colouring problem, known to be in $\mathcal{P}$ (see for instance [7, p. 176]). It is again $\mathcal{NP}$*–hard* when networks are rings (i.e. when graphs are cycles), whether directed or not [8], but is shown to be 2–APX ([9, 10], see also [8]).

Restricted to undirected stars (i.e. graphs made of edges which all together share a common end-point), the minimum all-optical routing problem is $\mathcal{NP}$–*hard* but $4/3$–APX [11]. If restricted to directed stars, the problem is in $\mathcal{P}$ and the same holds for spiders (i.e. graphs made of paths which together share a common end-point) [1, 12].

For trees of rings (Trees of rings can be defined inductively. A cycle is a tree of rings. If A and B are two trees of rings, identifying some vertex x of A with some vertex y of B yields a tree of rings.), whether directed or not, the problem is $\mathcal{NP}$*–hard* but APX in the undirected case [9] as well as in the directed case [13].

Indeed, when all-optical networks are concerned, *meshes* (graphs with a grid pattern, see figure 3 and definition below) have been considered as real competitive solutions among current metropolitan topologies [1, 14, 15]. Good results corroborate this idea for deflecting routing methods [14], while trees can be disconnected by a single link failure, meshes need up to four links to fail (in most cases) at an expense of no more than twice as many links. Furthermore, meshes have already been used in the past to build parallel computers : 2D meshes for Intel Paragon, Intel Delta, Symult 2010 or IBM Victor multiprocessors, and 3D meshes for Wavetracer computer Zaphir or J-Machine (MIT).

Restricted to meshes, the minimum all-optical routing problem is still $\mathcal{NP}$–*hard* [4]. To our knowledge, it is not known whether it is APX (at least, if it is d–APX, then one must have $d \geq 2$ [4]), and the best result is a $poly(\ln\ln N)$ approximation algorithm on meshes of $N \times N$ nodes. Turning to particular routings commonly used in meshes therefore seems worthwhile (see for example [16, 17]),

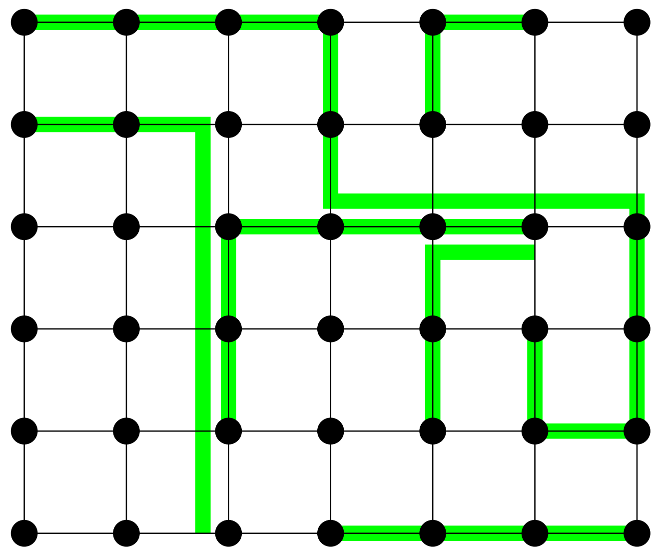

Figure 3: A routing in $M_{[5\times 6]}$ mesh made of 6 paths, from which 5 are RC-paths, from which 1 is a row-path.

and this paper is devoted to the all-optical routing problem in meshes when restricted to "row-column" routings ("RC-routings" for short, also known as "XY routings" or "E-cube routings"), which we now define in a formal way.

Thereafter, all graphs we consider are undirected graphs: a ***graph*** G is an ordered pair (V, E) where E, the set of *edges* of G, is a set of pairs of elements of V, the set of *vertices* of G. When needed, $V(G)$ (resp. $E(G)$) denotes the set of vertices (resp. the set of edges) of G.

Given integer i, $P_{[i]}$ denotes the graph such that $V(P_{[i]}) = \{0, 1, ..., i-1, i\}$ and $E(P_{[i]}) = \big\{\{0,1\},\{1,2\},...,\{i-1,i\}\big\}$. A ***path*** is a graph isomorphic[4] to $P_{[i]}$ for some integer i.

A ***subgraph*** of a graph G is a graph H such that $V(H) \subseteq V(G)$ and $E(H)$ $\subseteq E(G)$. A ***path of a graph*** is any of its subgraphs which is a path.

The ***cartesian product*** of two graphs G and G' is the graph whose vertices are the ordered pairs (x, x') where x is a vertex of G and x' a vertex of G' and such that there is an edge from (x, x') to (y, y') if and only if $x = y$ and $\{x', y'\}$ is an edge of G', or $x' = y'$ and $\{x, y\}$ is an edge of G.

Given integers i and j, $M_{[i\times j]}$ denotes the cartesian product of $P_{[i]}$ and $P_{[j]}$. A ***mesh*** is a graph isomorphic to $M_{[i\times j]}$ for some integers i and j. See figure 3 where $M_{[5\times 6]}$ is given a planar representation which suggests the following definitions.

In a mesh, a ***row path*** (resp. a ***column path***) is a path whose every edge is of the form $\{(p,\ q),\ (p, q{+}1)\}$ (resp. $\{(p,\ q),\ (p{+}1, q)\}$) for some integers p

[4]Two graphs are isomorphic when renaming their vertices can yield the same graph.

and q, and a ***row-column path*** (RC-path, for short) is a path which is the union of a row path and a column path of the mesh (see figure 3). Note that row paths and column paths are considered as special instances of RC-paths (formally, a path of length 0 can be viewed both as a row path and a column path). Given some communication instance whose network is a mesh, a ***row-column routing*** (RC-routing, for short) for this instance is a routing made of RC-paths only.

Given some integer k, let the ***k-all-optical RC-routing problem*** be the decision problem defined as: given some communication instance in a mesh, is there an all-optical RC-routing for this instance (i.e. an all-optical routing which is a RC-routing) and which uses k colours at most? For any integer k, we prove that the k-all-optical RC-routing problem is *$\mathcal{NP}$–complete*. This implies that the ***minimum all-optical RC-routing problem*** (the restriction of the minimum all-optical routing problem to meshes and RC-routings) is *$\mathcal{NP}$–hard*. It turns out that this last result must have been known (for instance a proof can be derived from [18] where communication instances on rings are mapped on meshes), though it seems not to have been published as such.

None the less, we give a genuine proof and, as the minimum all-optical RC-routing problem is therefore *$\mathcal{NP}$–hard*, we prove it to be APX by providing an 8–APX algorithm which, given some instance I of the minimum all-optical RC-routing problem, works through two main steps:

- step 1: compute some RC-routing R for I
- step 2: assign colours to the paths of R to make it an all-optical routing for I

The APX result stems from APX results with regard to each of these two steps which we now look at in detail, as step 1 computes a so-called *minimum load RC-routing* and step 2 addresses the so-called *minimum RC-path colouring problem* when taking into account the routing load.

Given some communication instance and some routing for this instance, the ***load of an edge*** is the number of routing paths which this edge belongs to, and the ***load of the routing*** is the maximum load of an edge with regard to this routing (see figure 2 for an example).

The L–***load routing problem*** is then the *decision problem* consisting of, given some communication instance and some positive integer L, answering the question: is there a routing for the instance whose load is at most L? The ***minimum load routing problem*** is then associated *optimization problem* defined as: given some communication instance, compute a routing for this instance which minimizes the routing load. A solution to a minimum load routing problem will be called a ***minimum load routing***.

The minimum load routing problem has already been paid some attention. On the one hand, the load of a minimum load routing is clearly at most the number of colours used in a minimal all-optical routing, while their difference cannot be bounded by a constant in general [19, 20], and on the other hand, one can see that if network nodes are converters, that is if any path can change its colour at any node, minimizing the overall number of colours used in a routing for this instance reduces to computing a minimum load routing.

The 1-load routing problem is known to be $\mathcal{NP}$*–complete* in meshes [4], yielding that the minimum load routing problem is $\mathcal{NP}$*–hard*, and thus, if APX, no better than 2–APX. Still it is not known, to our knowledge, whether the minimum load routing problem is APX or not (though, in the directed case there is no $c\log(\log(n))$-approximation algorithm for this problem unless $\mathcal{NP}\subseteq D-\ TIME(n^{O(\log(\log(\log(n))))})$ [21]). As we are interested in RC-routings, we specialize these problems to meshes in the ***L-load RC-routing problem*** and in the ***minimum load RC-routing problem*** respectively, namely by restricting routings to RC-routings. We first show that the 1-load RC-routing problem is in $\mathcal{P}$, due to the proof that the so-called *2-choice 1-load problem* (see section 2) is in $\mathcal{P}$. We then prove the row-column L–load routing problems to be $\mathcal{NP}$*–complete* for $L \geq 2$, yielding the minimum load RC-routing problem to be $\mathcal{NP}$*–hard* and ensuring that the so-called *k-choice minimum load routing problem* also is (where $k \geq 2$ is some positive integer, see section 2). The latter is then proved to be k–APX, yielding that the minimum load RC-routing problem is in turn 2–APX (while not d–APX for any $d < \frac{3}{2}$). This 2–APX result is used by step 1 (mentioned above) to compute R as an approximate minimum load RC-routing for I.

Now we turn to the question of colouring paths of a given routing. More precisely, for any interger k, let the ***path k-colouring problem*** be the decision problem defined as: given some path family in a graph, is there an all-optical routing whose paths are the given paths and which uses k colours at most? Is known that, for $k \geq 3$, the path k-colouring problem is $\mathcal{NP}$*–complete*, and it remains so when restricted to routings of load 2 in meshes [17]. Therefore, the ***minimum path colouring problem*** (the optimization problem associated to the path k-colouring problem) is $\mathcal{NP}$*–hard*, and it has been shown to be *No-APX*, even when restricted to routings of load 2 in meshes [17]. Restricting in turn these two problems into the ***RC-path k-colouring problem*** and the ***minimum RC-path colouring problem***, respectively, by forcing the routings to be RC-routings, it turns out that the minimum RC-path colouring problem is still $\mathcal{NP}$*–hard* (see, for instance, [17]) but it is known that any RC-routing can be coloured into an all-optical RC-routing using $8 \times L$ colours at most, where L is the routing load [22, 17, 23, 16].

Let d, n_1, n_2, ... n_d be non-negative integers and let $M_{[n_1 \times n_2 \times \ldots \times n_d]}$ denote the hypermesh where, with $0 \leq i_k, j_k \leq n_k$ for all $k \in [1, d]$, nodes $x = (i_1, \ldots i_d)$ and $y = (j_1, \ldots j_d)$ are adjacent iff $i_k = j_k$ for all $k \in [1, d]$ but one, say $k*$, for which $|i_{k*} - j_{k*}| = 1$, the edge xy being called an **edge of direction** $k*$. A **mesh of dimension** d is a graph M isomorphic to such a $M_{[n_1 \times n_2 \times \ldots \times n_d]}$, and, for $k \in [1, d]$, $E_k(M)$ denotes the set of edges of M which are of direction k.

Let P be a path in some hypermesh M of dimension d. If for all $i \in [1, d]$ the set $E_i(G) \cap E(P)$ induces a path in G, then P is said to be a **direction-segmentable path**. A routing in a hypermesh whose every path is direction-segmentable is a **direction-segmentable routing**.

Proving that any direction-segmentable routing in a hypermesh of dimension

d can be coloured in polynomial time using at most $2d(L-1)+1$ colours, where L is the routing load, we show that any RC-routing can be coloured into an all-optical RC-routing using $4 \times L$ colours at most, i.e. the best upper bound involving the routing load to our knowledge. This result is used in step 2 (mentioned above) to colour the paths of R and make it an all-optical RC-routing for I.

The sequel is organized as follows.

- Section 2 is devoted to load routing problems, where the minimum load RC-routing problem is proved to be 2–*APX*.
- Section 3 is devoted to the minimum path colouring problem in d-dimensional meshes when restricted to some special paths, yielding the minimum RC-path colouring problem to be 4–*APX*.
- Section 4 is devoted to the row-column all-optical routing problem, where the minimum all-optical RC-routing problem is then proved to be 8–*APX*.

We conclude in section 5.

2 Load RC-routing problems

We investigate both decision and minimization load RC-routing problems.

2.1 The L-load RC-routing problem $\mathcal{NP}$-*completeness*

It turns out that the L-load RC-routing problem is in $\mathcal{P}$ when $L = 1$ and otherwise $\mathcal{NP}$–*complete*.

Our proof refers to the celebrated SATISFIABILITY problem whose restriction as 3–*SAT* is $\mathcal{NP}$–*complete* (for instance, see [24, p. 39, p. 48]) while its 2–*SAT* restriction is in $\mathcal{P}$ (for instance, see [25, p. 185]). Hereafter, we use sets of clauses, sets of literals and boolean variables as in [24] rather than conjunctive normal forms of boolean expressions as in [25].

2.1.1 $L = 1$

We first enlarge the problem to all kinds of networks.

The ***1-load 2-choice routing problem*** is the decision problem defined as follows:

instance: a communication instance I **and** to each request $\{a, b\}$ in I, the assignment of two not necessarily distinct paths P_0^{ab} and P_1^{ab} joining a and b in the I network

question: is there a routing of load 1 for I such that, for each request $\{a, b\}$ of I, the corresponding routing path is P_0^{ab} or P_1^{ab} ?

We first prove:

Theorem 1 *The 1-load 2-choices routing problem is in $\mathcal{P}$.*

Proof. We reduce the 1-load 2-choice routing problem to $2\text{–}SAT$.

Assume $R = \{r_i | 1 \leq i \leq n\}$ is the set of requests of some instance I of a 1-load 2-choice routing problem such that P_0^i and P_1^i are the two paths assigned to the request r_i for $1 \leq i \leq n$. Using R as a set of boolean variables, we define C as the set of 2-clauses which, in turn, are defined for each pair $\{i, j\}$ with $1 \leq i, j \leq n$, according to three possible events:

- $\{\neg r_i, \neg r_j\}$ when P_1^i and P_1^j share a common edge
- $\{r_i, r_j\}$ when P_0^i and P_0^j share a common edge
- $\{\neg r_i, r_j\}$ when P_1^i and P_0^j share a common edge

Assume that S is a routing satisfying the set of requests R and let ϕ be a truth assignment of R such that, for each request r for which $P_0^{ab} \neq P_1^{ab}$, $\phi(r) = true$, resp. $\phi(r) = false$, if r is satisfied in S by path P_1^{ab}, resp. by path P_0^{ab} (values of $\phi(r)$ are indifferent for other requests r, if any). It can be checked that ϕ satisfies C.

Conversely, let ϕ be a truth assignment of R which satisfies C, and define the routing S in such a way that if $\phi(r) = true$, resp. $\phi(r) = false$, r is satisfied in S by path P_1^{ab}, resp. by path P_0^{ab}. It can be checked that S is a routing solution to the 2-choice 1-load routing instance.

Thus, there exists a solution to the 1-load 2-choice routing problem instance if and only if there exists a solution to the $2\text{–}SAT$ problem instance associated with C.

As $2\text{–}SAT$ is in $\mathcal{P}$, we conclude from the fact that the set of clauses C can be computed in polynomial time. $\diamondsuit$

Noticing that there are at most two possible RC-paths joining any two vertices in a mesh, the following straightforwardly stems from theorem 1:

Theorem 2 *The 1-load RC-routing problem is in $\mathcal{P}$.*

2.1.2 $L \geq 2$

Reducing $3\text{–}SAT$ to the L-load RC-routing problem, we now solve the general case.

Theorem 3 *The L-load RC-routing problem is $\mathcal{NP}$–complete for $L \geq 2$.*

Proof. We assume $L = 2$ (the proof is easily extended for $L > 2$ by solely adding a convenient number of so-called "blocking requests" as defined below).

Clearly the problem is in $\mathcal{NP}$. Using a reduction of $3\text{–}SAT$, we prove it to be $\mathcal{NP}$*–complete*. Let C be some instance of $3\text{–}SAT$ with $C = \{c_1, c_2, ...c_m\}$, a set of 3-clauses over the set of boolean variables $X = \{x_1, x_2, ...x_n\}$. We now define an instance I of the 2-load RC-routing problem using the $M_{[(2n)\times(2m+1)]}$ mesh as the problem network (see fig. 4 for an example):

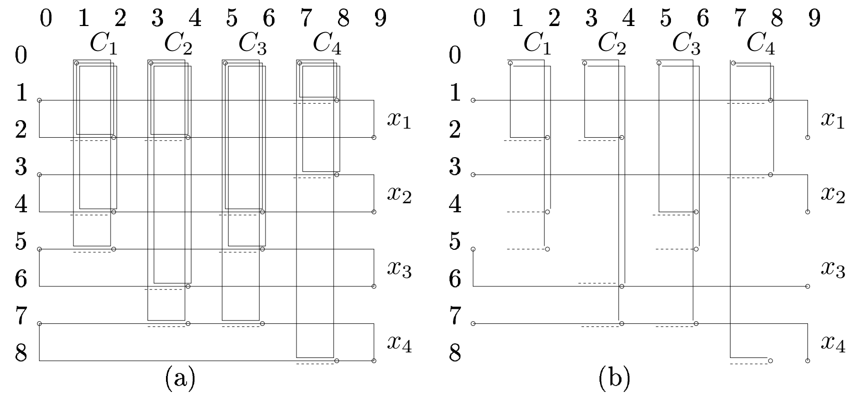

Figure 4: Let $C = \{C_1, C_2, C_3, C_4\}$ with $C_1 = \{x_1, x_2, \neg x_3\}$, $C_2 = \{x_1, x_3, \neg x_4\}$, $C_3 = \{x_2, \neg x_3, \neg x_4\}$ and $C_4 = \{\neg x_1, \neg x_2, x_4\}$. Figure (a) shows the communication instance I associated with C and figure (b) shows a row-column 2–load routing solution to I. The network in instance I is the mesh $M_{[8\times 9]}$. In figure (a) each "horizontal" (resp. "vertical") rectangle bears the two possible RC-paths satisfying the communication request associated with one of the variables x_1, x_2, x_3 and x_4 (resp. to one of the literals of clauses C_1, C_2, C_3 and C_4, with vertical rectangles being grouped according to the clause to which the literal they stand for belongs). "Blocking requests" are depicted with dotted lines.

- to each variable x_i, we assign the request $r_i = \{(2i-1, 0), (2i, 2m+1)\}$
- to each positive literal $l \in c_j$, with $l = x_i$, we assign the request $r_{i,j} = \{(0, 2j-1), (2i, 2j)\}$ together with a so-called "blocking request" $blk_{i,j} = \{(2i, 2j-1), (2i, 2j)\}$
- to each negative literal $l \in c_j$, with $l = \neg x_i$, we assign the request $r'_{i,j} = \{(0, 2j-1), (2i-1, 2j)\}$ together with a so-called "blocking request" $blk'_{i,j} = \{(2i-1, 2j-1), (2i-1, 2j)\}$

Fact 1 If there exists some truth assignment φ satisfying C, then there exists a 2-load RC-routing solution to I.

Assume that φ satisfies C, and to each request r of I, choose the path that joins the two end-nodes of r in $M_{[(2n)\times(2m+1)]}$ according to the following:

- for any i, $1 \leq i \leq n$, if $\varphi(x_i) = true$ (resp. $\varphi(x_i) = false$), the path selected for r_i uses column $2m+1$ (resp. column 0);
- for any j, $1 \leq j \leq m$, there exists at least one literal $l \in c_j$ such that $\varphi(l) = true$; choose one such literal l and, in order to join the two end-nodes of its corresponding request, select the row-column path using column $2j-1$, while paths selected with regard to the requests which are associated with the two other literals of c_j use column $2j$;

- for any blocking request, the selected path is the only row-column path joining its two nodes in the network (actually a row path).

It can be checked that the thus computed routing is indeed a 2-load RC-routing solution to I.

Fact 2 If there exists a 2-load RC-routing solution R to I, then there exists some truth assignment φ satisfying C.

Assume that R is a 2-load RC-routing solution R to I, we construct a truth assignment φ of C as follows: for any i, $1 \leq i \leq n$, if the path selected for r_i uses column $2m+1$ (resp. column 0), $\varphi(x_i) = true$ (resp. $\varphi(x_i) = false$). We now prove that φ satisfies C.

Consider clause c_j for $1 \leq j \leq m$. Associated with literals from c_j, there are three requests in I sharing vertex $(0, 2j-1)$ as an end-node. The three of them cannot be assigned a row-column path using row 0, for R is a 2-load routing solution to I. Therefore, at least one of them uses column $2j-1$. Assume, with no loss of generality, that this path is associated with literal x_i (the case $\neg x_i$ would be treated in a similar way). Then, by definition of I, this path uses row $2i$, and, because of the associated blocking request $blk_{i,j} = \{(2i, 2j-1), (2i, 2j)\}$ which also uses row $2i$, request $r_i = \{(2i-1, 0), (2i, 2m+1)\}$ has been assigned a path using a different row, namely row $2i-1$, thus using column $2m+1$, which means $\varphi(x_i) = true$. Thus clause c_j is satisfied, which ultimately leads us to conclude that C itself is satisfied.

We conclude by considering that the instance I of L-load RC-routing problem associated with C can be computed in polynomial time. $\diamondsuit$

Clearly, theorem 3 yields the following:

Theorem 4 *The minimum load RC-routing problem is $\mathcal{NP}$–hard.*

Therefore the question of an approximation algorithm is posed.

2.2 The minimum load RC-routing problem approximation

Again, we first investigate a more general problem, namely, with k being some positive integer, the ***minimum load k-choice-routing problem***, which we define as follows:

instance: a communication instance I **and** to each request $r = \{a, b\}$ in I, the assignment of at most k paths joining a and b in the I network

solution: a routing for I such that each request r from I is satisfied by a path assigned to r

objective: minimize the load of the routing solution

When restricted to RC-paths to join two nodes in a mesh, routing problems become 2-choice-routing problems. This makes the minimum load RC-routing problem a special case of the minimum load k-choice-routing problem, and we clearly may conclude from theorem 4:

Theorem 5 *The minimum load k-choice-routing problem is $\mathcal{NP}$–hard.*

We now show this more general problem to be APX.

Theorem 6 *The minimum load k-choice-routing problem is k–APX.*

Proof. Let I be some instance of the minimum load k-choice-routing problem. We restate the problem as a linear programming problem instance as follows. Let $R = \{r_i\}_{1 \le i \le n}$ be the set of requests from I. To each request r_i is associated a set $P_i = \{p_1^i, p_2^i, ..., p_{k_i}^i\}$ of k_i feasible paths in the network G, with $k_i \le k$. Selecting path p_j^i to join end-nodes of request r_i if and only if $x_j^i = 1$ yields a one-to-one mapping between routing solutions to I and solutions to the integer linear programming instance defined as:

$$x_j^i \in \{0,1\} \text{ for all } i, j, 1 \le i \le n, 1 \le j \le k_i$$

$$\sum_{j=1}^{k_i} x_j^i = 1 \text{ for all } i, 1 \le i \le n$$

$$z \ge \sum_{e \in E(p_j^i)} x_j^i \text{ for every edge } e \text{ of the network } G$$

$$\text{objective: minimize } z$$

For every edge e of the network G, let $\pi(e) = \sum_{e \in E(p_j^i)} x_j^i$, let π_{IN}^* denote the optimal value of π, and let π_{IR}^* be the optimal value of π when relaxing, for all $i, j, 1 \le i \le n, 1 \le j \le k_i$, integer condition $x_j^i \in \{0,1\}$ to real condition $x_j^i \in [0,1]$. Obviously $\pi_{IR}^* \le \pi_{IN}^*$.

For all $i, j, 1 \le i \le n, 1 \le j \le k_i$, assume a_j^i to be the value of x_j^i in an optimal solution to the relaxed linear programming problem and define:

$$b_j^i = \begin{cases} 1 & \text{if } a_j^i = \max_{1 \le h \le k_i} a_h^i \\ 0 & \text{otherwise} \end{cases}$$

(for a given $i, 1 \le i \le n$, if more than one b_j^i is equal to 1, set all of them but one at 0).

Now, as $\max_{1 \le j \le k_i} a_j^i \ge \frac{1}{k}$, letting $\pi_{IN}^{algorithm}$ denote the load associated with the $(b_j^i)_{1 \le i \le n, 1 \le j \le k_i}$ solution yields the following :

$$\frac{\pi_{IN}^{algorithm}}{\pi_{IN}^*} \le \frac{k\pi_{IR}^*}{\pi_{IN}^*} \le \frac{k\pi_{IR}^*}{\pi_{IR}^*} = k$$

We conclude by noting that the size of the linear programming instance is polynomialy related to the size of the k–choice minimum load routing instance. $\diamondsuit$

Get n be a positive integer and let $L = 2n$. Consider the communication instance where the graph is the mesh whose rows and columns are numbered from 0 to 1 and the communication requests are $r_1, ..., r_L$ with $r_i = \{(0,0), (1,1)\}$ for $1 \leq i \leq L$. The algorithm from the theorem above might compute the real solution where $a_j^i = 0.5$ for $1 \leq i \leq L$ and $1 \leq j \leq 2$, possibly yielding an integer solution of load L while the optimum load is $L/2$.

Restricting again k–choice routings to RC-routings in meshes, theorem 6 yields the following.

Theorem 7 *The minimum load RC-routing problem is 2–APX.*

The 2 approximation factor expressed in theorem 7 might be improved upon, but not beyond $\frac{3}{2}$, as stated in the following result.

Theorem 8 *If the minimum load RC-routing problem is d–APX for some constant d, then $d \geq 3/2$.*

Proof. Consider an optimization problem to which any solution has a cost which is positive or null, while c is some positive integer. Whenever the problem of the existence of a solution of cost less or equal to c is $\mathcal{NP}$*–complete*, then, it is known that the optimization problem cannot be d*–APX* for any $d < \frac{c+1}{c}$ [26]. We can conclude on the basis of the fact that the 2-load RC-routing problem is $\mathcal{NP}$*–complete* (see theorem 3). $\diamondsuit$

3 The RC-paths colouring problem

Given some routing R solution to a given communication instance I, the **conflict graph** induced by R is the graph G whose nodes are the paths of R, with two paths being adjacent in G when they have at least one edge in common.

Lemma 1 *If G is the conflict graph of some direction-segmentable routing R on a hypermesh of dimension d, then $E(G) \leq d(L-1)(n - \frac{L}{2})$ where n is the order of G and L is the load of R.*

Proof. For every $i \in [1, d]$, let G_i be the subgraph of G induced by conflicts which occur along direction i only, and let L_i be the maximum load on edges of $E_i(G)$. Then G_i is an interval graph, say of order n. On the one hand, with G_i being a triangulated graph, the number of edges of G_i is less or equal to $f_n(k) = (k-1)(n-\frac{k}{2})$ where k is the maximum size of a clique, and, on the other hand, G_i being an interval graph, any clique of maximum size in G_i is of size L_i. Thus $|E(G_i)| \leq (L_i - 1)(n - \frac{L_i}{2})$. As $f_n(k)$ is a non-decreasing function when $k \leq n$ and as $L_i \leq L$ for all $i \in [1, d]$, it follows that $|E(G_i)| \leq (L-1)(n - \frac{L}{2})$. One concludes the proof considering that $|E(G)| \leq \sum_1^d |E_i(G)|$. $\diamondsuit$

Lemma 2 *If G is the conflict graph of a direction-segmentable routing R on a hypermesh of dimension d, one of the nodes of G is of degree at most $2d(L-1)$, where L is the load of R.*

Proof. The average node degree in G is $\frac{2\times E(G)}{n}$, where n is the order of G. One can conclude from lemma 1. $\diamondsuit$

Theorem 9 *Any direction-segmentable routing in a hypermesh of dimension d can be coloured in polynomial time using at most $2d(L-1)+1$ colours, where L is the routing load.*

Proof. By induction on the number n of chains in the routing R. The result is straightforward if $n = 1$. As colouring the routing is equivalent to colouring the nodes of its conflict graph, let $n > 1$ and let G be the conflict graph induced by R. From lemma 2, some node p in G is of degree $2d(L-1)$ at most. Let R' be the routing obtained from R by suppressing the path p, G' be the conflict graph induced by R', and L' be the load of R'. By the induction hypothesis, G' can be coloured using $2d(L'-1)+1$ colours at most, thus $2d(L-1)+1$ colours at most. Considering the degree of p yields the result. $\diamondsuit$

As an interesting special case, theorem 9 yields:

Theorem 10 *Any row-column routing in a mesh can be coloured in polynomial time using at most $4L-3$ colours, where L is the routing load.*

4 The all-optical RC-routing problem

We first take advantage of the proof of theorem 3.

Theorem 11 *For any $k \geq 2$, the k-all-optical RC-routing problem is $\mathcal{NP}$–complete.*

Proof. We assume $k = 2$ (as for theorem 3, the proof is easily extended to $k \geq 2$). Let C be some instance of 3–SAT and let I be the communication instance associated with C in the proof of theorem 3. One can check that I can be satisfied using 2 colours if and only if there exists a 2-load RC-routing which satisfies I, that is, due to the proof of theorem 3, if and only if C is satisfiable. Which leads to the conclusion. $\diamondsuit$

Given a communication instance I and a RC-routing S for this instance, let $\pi(S)$, resp. $\omega(S)$, denote the load, resp. the number of colours, used by S. Similarly, let $\pi(I)$, resp. $\omega(I)$, denote the load of a minimum load RC-routing for I, resp. the number of colours used by a minimal all-optical RC-routing for I. As mentioned before, one has $\pi(S) \leq \omega(S)$, and therefore $\pi(I) \leq w(I)$ as well.

Theorem 12 *The row-column minimum all-optical routing problem is 8–APX.*

Proof. Let I be some communication instance whose network is a mesh, let S be a routing for I computed by a 2-approximation minimum load RC-routing algorithm whose existence is asserted by theorem 7, and let $c(S)$ be the number of colours used by a path colouring algorithm using at most $4 \times \pi(S)$ colours, whose existence is asserted by theorem 10.

We then have $c(S) \leq 4 \times \pi(S) \leq 4 \times 2 \times \pi(I)$, and we conclude with the general inequality $\pi(I) \leq \omega(I)$. $\diamondsuit$

5 Conclusion

In general, the minimum all-optical routing problem and the minimum load routing problem are both $\mathcal{NP}$*–hard*, and it is not known whether they are APX or not, while the minimum path colouring problem is both $\mathcal{NP}$*–hard* and *No-APX*. Restricting these problems to meshes does not change their complexity status.

In this paper, we restricted these three problems to RC-routings in meshes.

Regarding load routing problems, we proved the L-load RC-routing problem to be in $\mathcal{P}$ when $L = 1$ and otherwise $\mathcal{NP}$*–complete*, and we provided a 2–APX algorithm to solve the associated minimizing problem when it is $\mathcal{NP}$*–hard*.

Regarding the minimum RC-path colouring problem, we proved it to be 4-approximable, where L is the load of the path family, which is an improvement over several previous results known to us (namely, 8-approximation algorithms). This result stems from a result expressed for *dimension-segmentable chains* in *meshes of dimension d*.

Regarding the minimum all-optical RC-routing problem, and due to the indirect proof of the result, we think the constant asserted in the 8–APX result (see theorem 12) should be improved upon.

Last, it is worth noting that, not surprisingly, some results can be extended from meshes to tori.

References

[1] Bruno Beauquier. *Communications dans les réseaux optiques par multiplexage en longueur d'onde.* PhD thesis, Université de Nice - Sophia Antipolis, jan 2000.

[2] Bruno Beauquier, Jean-Claude Bermond, Luisa Gargano, Pavol Hell, Stéphane Pérennes, and Ugo Vaccaro. Graph problems arising from wavelength-routing in all-optical networks. In *Proceedings of 2nd IEEE Workshop on Optics and Computer Science*, Geneva, Switzerland, apr 1997. IEEE Press. part of IPPS'97.

[3] Thomas Erlebach and Klaus Jansen. Scheduling of virtual connections in fast networks. In *Proceedings of the 4th Parallel Systems and Algorithms Workshop (PASA'96)*, pages 13–32. World Scientific Publishing, 1997.

[4] M. R. Kramer and J. Leeuwen. The complexity of wire-routing and finding minimum area layouts for arbitrary VLSI circuits. *Advances in Computing Research*, 2:129–146, 1984.

[5] Yuval Rabani. Path coloring in the mesh. In *FOCS*, pages 400–409, 1996.

[6] Vijay V. Vazirani. *Approximation Algorithms.* Springer, 2001.

[7] Douglas B. West. *Introduction to Graph Theory.* Prentice-Hall, 1996.

[8] Thomas Erlebach and Klaus Jansen. The complexity of path coloring and call scheduling. *Theoretical Computer Science*, 255:33–50, 2001.

[9] Prabhakar Raghavan and Eli Upfal. Efficient routing in all-optical networks. In *Proceedings of the 26th ACM symposium on Theory of Computing (STOC'94)*, pages 134–143. ACM Press, may 1994.

[10] Milena Mihail, Christos Kaklamanis, and Satish Rao. Efficient access to optical bandwidth - wavelength routing on directed fiber trees, rings, and trees of rings. In *Proceedings of 36th IEEE Annual Symposium on Foundations of Computer Science (FOCS'95)*, pages 548–557. IEEE Press, oct 1995.

[11] Thomas Erlebach. *Scheduling Connections in Fast Networks.* PhD thesis, Technische Universität München, 1999.

[12] Gordon Wilfong and Peter Winkler. Ring routing and wavelength translation. In *Annual ACM-SIAM Symposium on Discrete Algorithms*, volume 9, pages 333–341. ACM-SIAM, 1998.

[13] Christos Kaklamanis and Pino Persiano. Efficient wavelength routing on directed fiber trees. In *Lecture Notes in Computer Science*, volume 146, pages 444–473. Springer Verlag, 1996.

[14] Thierry Chich. *Optimisation du routage par déflexion dans les réseaux de télécommunications métropolitains.* PhD thesis, École Normale Supérieure de Lyon, dec 1997.

[15] Heiko Schröder, Ondrej Sýkora, and Imrich Vrt'o. Optical all-to-all communication for some product graphs. In *Proceedings of the 24th Seminar on Current Trends in Theory and Practice of Informatics*, 1997.

[16] Sandeep N. Bhatt, Gianfranco Bilardi, Geppino Pucci, Abhiram Ranade, Arnold L. Rosenberg, and Eric J. Schwabe. On bufferless routing of variable length messages in leveled networks. *Transactions on Computers*, 45(6):714–729, june 1996.

[17] Jérôme Palaysi. Approximability of paths coloring problem in mesh and torus networks. In *Proceedings of the Second Colloquium on Mathematics and Computer Science: Algorithms, Trees, Combinatorics and Probabilities*, pages 213–222, University of Versailles-St-Quentin (France), sep 2002.

[18] Thomas Erlebach and Klaus Jansen. Call scheduling in trees, rings and meshes. Unpublished report, presumably long version to "Call Scheduling in Trees, Rings and Meshes" in Proceedings of the 30th Hawaii International Conference on System Sciences (HICSS-30), IEEE Computer Society Press, pages 221-222, 1997.

[19] Alok Aggarwal, Amotz Bar-Noy, Don Coppersmith, Rajiv Ramaswami, Baruch Schieber, and Sudan Madhu. Efficient routing and scheduling algorithms for optical networks. In ACM-SIAM, editor, *Proceedings of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 412–423, Arlington, Virginia, USA, jan 1994.

[20] Alok Aggarwal, Amotz Bar-Noy, Don Coppersmith, Rajiv Ramaswami, Baruch Schieber, and Sudan Madhu. Efficient routing in optical networks. *Journal of the ACM*, 6(43):973–1001, nov 1996.

[21] Julia Chuzhoy and Joseph Naor. New hardness results for congestion minimisation and machine scheduling. In *36th Annual ACM Symposium on theory of Computing (STOC'O4)*, pages 28–34, Chicago, Illinois, USA, June 2004. ACM press.

[22] Mustapha Kchikech and Olivier Togni. Paths coloring algorithms in mesh networks. In C.S. Calude et al., editor, *DMTCS 2003*, pages 193–202, 2003.

[23] T. Erlebach and K. Jansen. Conversion of coloring algorithms into maximum weight independent set algorithms in communication networks. In Carleton Scientific, editor, *ARACNE 2000*, pages 135–146, 2000.

[24] M. R. Garey and D. S. Johnson. *Computers and Intractability : A Guide to the Theory of NP-Completeness.* Freeman, 1979.

[25] Christos H. Papadimitriou. *Computational Complexity.* Addison-Wesley Publishing Company, 1994.

[26] J. K. Lenstra and D. B. Shmoys. *Scheduling Theory and its Applications*, chapter 1 Computing Near-Optimal Schedules. John Wiley and Sons, 1995.