Approximable Row-Column Routing Problems in All-Optical Mesh Networks

Olivier Cogis, Jérôme Palaysi

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

Unité Mixte CNRS - Université Montpellier II 5506

RAPPORT DE RECHERCHE

Approximable Row-column Routing Problems in All-Optical Mesh Networks

Olivier Cogis **Jérôme Palaysi**

juillet 2004 R.R.LIRMM 04011

161, Rue Ada - 34392 Montpellier Cedex 5 - France
Tél: +33 (0)4 67 41 85 85 - Fax: +33 (0)4 67 41 85 00
<nom de l'auteur>@lirmm.fr

Abstract

In all-optical networks, several communications can be transmitted through the same fiber link provided that they use different wavelengths. When given a list of pairs of nodes standing for as many point to point communication requests, the objective is, according to this rule, to assign to each request both a path through the network and a single wavelength to convey the information.

The ALL-OPTICAL ROUTING problem (minimizing the overall number of assigned wavelengths) has been paid a lot of attention and is known to be $\mathcal{NP}$–*hard*. Thus rings, trees and meshes have been investigated as specific networks, but leading to yet as many $\mathcal{NP}$–*hard* problems.

This paper investigates row-column routings in meshes (paths are allowed one turn only). We show the ROW-COLUMN MINIMUM LOAD ROUTING and the ROW-COLUMN ALL-OPTICAL ROUTING problems to be $\mathcal{NP}$–*hard*, as well as the k-CHOICES MINIMUM LOAD ROUTING problem. The latter we prove to be k–APX, yielding the ROW-COLUMN MINIMUM LOAD ROUTING problem to be 2–APX, while no approximation ratio can be less than $\frac{3}{2}$. From there, we prove the ROW-COLUMN ALL-OPTICAL ROUTING problem to be APX.

These results can be extended to tori.

keywords: minimum load routing, all-optical networks, mesh, torus, 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 to each communication request a physical path into the network, as for usual circuit switching; each router being set up, messages can stay in their optical state from start to end. The all-optical network commutation nodes we are interested in are Wavelength Routing Optical Cross-connect (WR-OXC) with Optical Add/Drop Multiplexer (OADM) (see for instance [Bea00]). An example of such a router is depicted in figure 1.

Wavelength Division Multiplexing (WDM) is a technique (see for instance [BBG$^+$97]) that proposes to take advantage of the huge bandwidth of optical fiber 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*, wether 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 not be unique 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

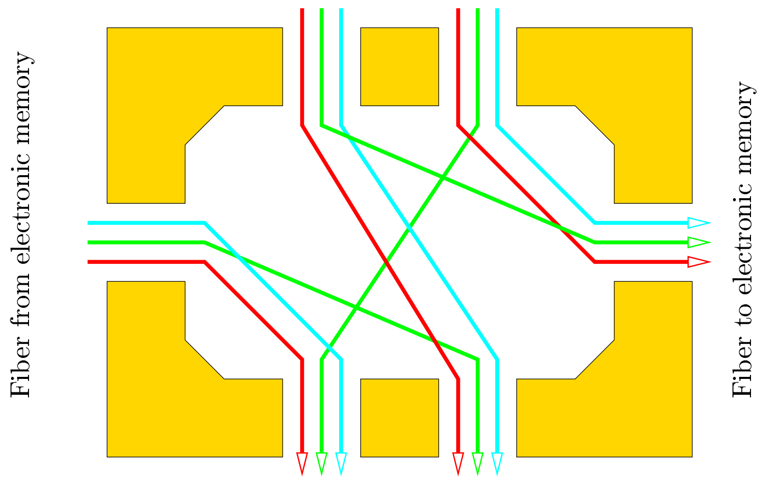

Figure 1: As an example, a router WR-OXC with OADM dedicated to directed communications.

of each request by a path in the graph[1], and an ***all-optical routing*** for this instance 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 commun edge bear the same colour.

As wavelengths are usually a critical ressource, the ***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 an all-optical routing problem will be called an ***optimal all-optical routing*** (see figure 2 for an example).

The all-optical routing problem is $\mathcal{NP}$*–hard* in general, wether graphs are directed [EJ97b] or not [KL84, Rab96, EJ97b]. Moreover, restricted to directed graphs, the problem is known to be *No-APX*[3] [Bea00, corollary 3.1.5]. Therefore

[1] When two different requests are made of the same pair of nodes, they may not be assigned the same path in the graph.

[2] When k colours are used to label the routing paths, it is not uncommun to use integers 1 to k as colours, though basically the set of colours is not an ordered set (on the other hand, refering 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 [Vaz01]. 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

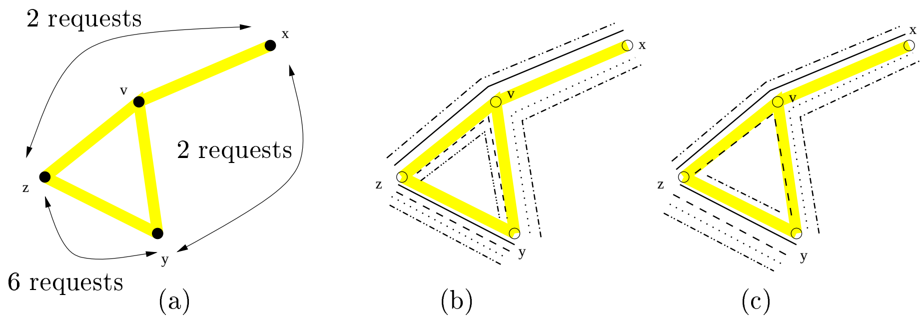

Figure 2: Figure (a) shows a communication instance I. Figure (b) and (c) show all-optical routing R_b and R_c resp. which are solution to I. R_c is an optimal all-optical routing for I, but R_b is not (R_b, resp. R_c, makes use of 6 colours, resp. 5). On the other hand, R_b is a minimum load routing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

some topologies have been selected to be paid specific attention.

When networks are linear (i.e. the graph is a path), the problem is equivalent to the interval graphs vertex colouring problem, known to be in $\mathcal{P}$(see for instance [Wes96, p. 176]). It is again $\mathcal{NP}$*–hard* when networks are rings (i.e. when graphs are cycles), wether directed or not [EJ01], but is shown to be 2–APX ([RU94, MKR95], see also [EJ01]).

Restricted to undirected stars (i.e. graphs made of edges which all together share a common end-point), the all-optical routing problem is $\mathcal{NP}$*–hard* but 4/3–APX [Erl99]. If restricted to directed stars, the problem is in $\mathcal{P}$ and the same holds for spiders (i.e. graphs made of paths which all together share a common end-point) [Bea00, WW98].

And for trees of rings (i.e. the graph is the result of expanding each node of a tree into a cycle in such a way that when to nodes are adjacent in the tree, the corresponding cycles must have one and one only vertex in common), wether directed or not, the problem is $\mathcal{NP}$*–hard* but APX in the undirected case [RU94] as well as in the directed case [KP96].

As a matter of fact, 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 [Bea00, Chi97, SSV97]. For deflecting routing methods [Chi97], good results corroborate this idea. 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 much 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).

---

assumed to be strictly positive). If no such d exists, then the problem is said to be *No-APX*.

Restricted to meshes, the all-optical problem is still $\mathcal{NP}$*–hard* [KL84]. To our knowledge it is not known wether it is APX (at least, if it is d–APX, then one must have $d \geq 2$ [KL84]), and the best result is a $poly(\ln \ln N)$ approximation algorithm on mesh of $N \times N$ nodes, while computing the number of colors of an optimal all-optical routing is APX [Rab96]. Therefore, turning to particular routings commonly used in meshes seems worthwile (see for example [BBP+96, Pal02a]), and this paper is devoted to the all-optical routing problem in meshes when restricting to "row-column" routings (also known as "XY routings" or "E-cube routings"), which we now define in a formal way.

From now on, 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 grah*** is any of its subgraphs which is a path.

The ***cartesian sum*** 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 sum 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 $M_{[4 \times 5]}$ 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 and q, and a ***row-column path*** 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 row-column 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*** for this instance is a routing made of row-column paths only.

We can now specialize the all-optical routing problem : the ***row-column all-optical routing problem*** is the all-optical routing problem restricted to networks being meshes and whose solutions are to be row-column routings.

To our knowledge, the row-column all-optical routing problem has been known to be $\mathcal{NP}$*–hard* (for instance a proof can be derived from [EJ97a] where communication instances on rings are mapped on meshes) though the result seems not to have been published as such. In any case we give it a genuine proof and we then prove the optimization problem to be APX.

In order to do so, we first turn our attention to another optimization problem related to communication requests which we will eventually specialize to meshes.

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

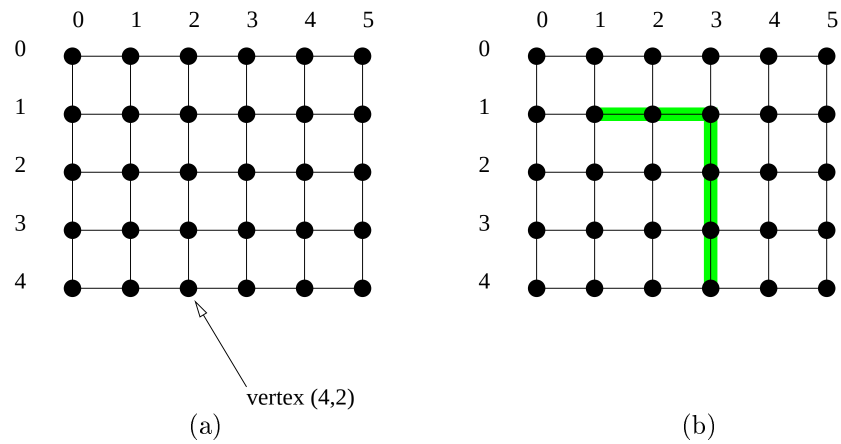

Figure 3: Figure (a) gives a planar row-column representation of the mesh $M_{[4\times 5]}$. Figure (b) shows the row-column path with end-points $(1,1)$ and $(4,3)$, and, therefore, with node $(1,3)$ as "turning-point".

Namely, given somme 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 regards to this routing (see figure 2 for an example).

The L*−**load routing problem*** is then the *decision problem* consisting in, 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 ? And the ***minimum load routing problem*** is the associated *optimization problem* defined as : given some communication instance, compute a routing for this instance which minimizes the routing load[5]. A solution to a minimum load routing problem will be called a ***minimum load routing***.

Clearly, the load of a minimum load routing is at most the number of colours used in an optimal all-optical routing, but their difference cannot be bounded by a constant in general [ABNC$^+$94, ABNC$^+$96].

As the 1-load routing problem is known to be $\mathcal{NP}$*−complete* in meshes [KL84], yielding the minimum load routing problem to be $\mathcal{NP}$*−hard*, we specialize these problems to meshes into the ***row-column*** L*−**load routing problem*** and to the ***row-column minimum load routing problem*** respectively, namely by restricting networks to be meshes and routings to be row-column ones.

Section 2 is devoted to load routing problems. We first show that the 1-load row-column routing problem is in $\mathcal{P}$, due the proof that the so-called *2-choices 1-load problem* (see section 2) is in $\mathcal{P}$. We then prove the row-column

[5]One can easily see that if networks 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.

L–load routing problem to be $\mathcal{NP}$*–complete* for $L \geq 2$, yieldind the row-column minimum load routing problem to be $\mathcal{NP}$*–hard* and ensuring that the so-called *k-choices minimum load routing problem* is too (where $k \geq 2$ is some positive integer, see section 2). The latter is then proved to be k–APX, yielding that the row-column minimum load routing problem is in turn 2–APX (while not $d - APX$ for any $d < \frac{3}{2}$).

Section 3 is devoted to the row-column all-optical routing problem. Due to a routing paths coloration result [Pal02a, BBP+96][6] and making use of results from section 2, we prove the row-column all-optical routing problem to be d–APX for some constant d (namely we show that $d \leq 16$).

We conclude in section 4 where extensions to tori are mentionned.

2 Row-column load routing problems

We investigate both decision and minimization load routing problems.

2.1 The row-column L-load routing problem NP–completeness

As stated before, the ***row-column L-load routing problem*** is a decision problem :

instance: a communication instance where the network is a mesh **and** a positive integer L

question: is there a row-column routing for the communication instance whose load is at most L ?

It turns out that this 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 [GJ79, p. 39, p. 48]) while its 2–SAT restriction is in $\mathcal{P}$ (for instance, see [Pap94, p. 185]). Hereafter, we use sets of clauses, sets of literals and boolean variables as in [GJ79] rather than conjunctive normal form of boolean expression as in [Pap94].

2.1.1 $L = 1$

We fisrt enlarge the problem to all kinds of networks.

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

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

[6]Though the algorithm given in [BBP+96], as we understand it, seems not to prove the result. See appendix for a possible counterexample.

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

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

Proof We reduce the 2-choices 1-load routing problem to 2–SAT.

Assume $R = \{r_i | 1 \leq i \leq n\}$ is the set of requests of some instance I of a 2-choices 1-load 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 an interpretation 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 an interpretation 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-choices 1-load routing instance.

Thus, there exists a solution to the 2-choices 1-load routing instance if and onnly if there exists a solution to the 2–SAT problem associated to C.

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

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

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

2.1.2 $L \geq 2$

Reducing 3–SAT to the row-column L–load routing problem, we now solve the general case.

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

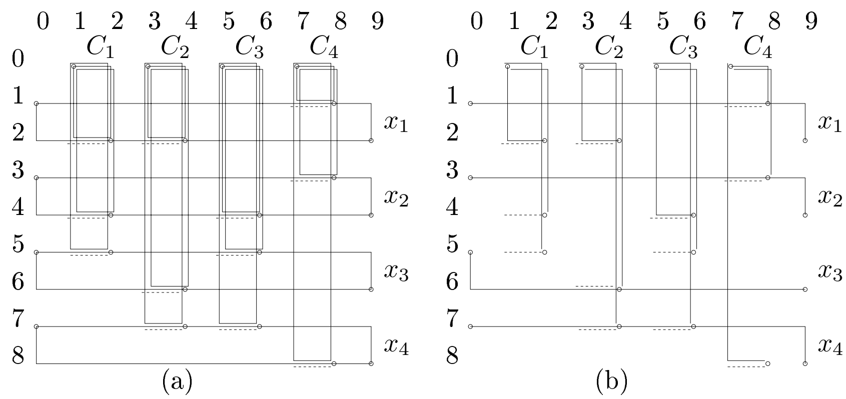

Figure 4: Let $C = \{C_1, C_2, C_3, C_4\}$ with $C_1 = \{x_1, x_2, \neg x_3\}$, $C_2 = \{x_1, x_3, \neg x_4\}$, $C_3 = \{x_2, \neg x_3, \neg x_4\}$ and $C_4 = \{\neg x_1, \neg x_2, x_4\}$. Figure (a) shows the communication instance I associated to C and figure (b) shows a row-column 2–load routing solution 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 row-column paths satisfying the communication request associated to one of variables x_1, x_2, x_3 and x_4 (resp. to one of the litterals of clauses C_1, C_2, C_3 and C_4, vertical rectangles being grouped according to the clause to which belongs the litteral they stand for). “Blocking request” are depicted in dotted lines.

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–SAT, we prove it to be $\mathcal{NP}$*–complete*. Let C be some instance of 3–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 row-column 2-load routing problem using the $M_{[(2n)\times(2m+1)]}$ mesh as the problem network (see fig. 4 for an example) :

- 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 interpretation φ satisfying C, then there exists a row-column 2-load routing solution to I.

Assume 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 litteral $l \in c_j$ such that $\varphi(l) = true$; chose one such litteral 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 regards 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 routing so computed is indeed a row-column 2-load routing solution to I.

Fact 2 If there exists a row-column 2-load routing solution R to I, then there exists some interpretation φ satisfying C.

Assume R is a row-column 2-load routing solution R to I, we construct an interpretation 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 line 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 row-column L–load routing problem associated with C can be computed in polynomial time. □

Clearly, theorem 3 yields the following :

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

Reminding that it is not known, to our knowledge, wether the minimum load routing problem is APX or not, and as the problem restricted to row-column routings is still $\mathcal{NP}$–*hard*, the question of an approximation algorithm is posed.

2.2 The row-column minimum load routing problem approximation

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

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

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

objective : minimize the load of the routing solution

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

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

We now show this more general problem to be APX.

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

Proof Let I be some instance of the k-choices minimum load routing problem. We restate the problem as a linear programming problem instance as follows. Let $R = \{r_i\}_{1 \leq i \leq n}$ 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 $k_i \leq k$. 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 routings solution to I and solutions to the integer linear programming instance defined as :

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

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

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

$$\text{objective: minimize } \pi = \max_{e \in E(G)} \pi(e)$$

Let π_{IN}^* denote the optimal value of π, and let π_{IR}^* be the optimal value of π when relaxing, for all $i, j, 1 \leq i \leq n, 1 \leq j \leq k_i$, integer condition $x_j^i \in \{0, 1\}$ to real condition $x_j^i \in [0, 1]$. Obviously $\pi_{IR}^* \leq \pi_{IN}^*$.

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

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

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

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

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

We conclude by noticing that the size of the linear programming instance is polynomialy related to the size of the k–choices minimum load routing instance. □

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

Theorem 7 *The row-column minimum load 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 row-column minimum load 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 equel to c is $\mathcal{NP}$*–complete* then, it is known that the optimization problem can't be d-APX for any $d < \frac{c+1}{c}$ [LS95]. We can conclude from the fact that the row-column 2-load routing problem is $\mathcal{NP}$*–complete* (see theorem 3). □

3 The row-column all-optical routing problem

As mentioned in section 1, we first take advantage of the proof of theorem 3.

Theorem 9 *The row-column all-optical routing problem is $\mathcal{NP}$–complete.*

Proof Let C be some instance of $3\text{–}SAT$ and let I be the communication instance associated to C in the proof of theorem 3. One can check that I can be satisfied using 2 colours only if and only if there exists a row-column 2-load routing which satisfies I, that is, due to the proof of theorem 3, if and only if C is satisfaiable. Which leads to conclusion. □

Given a communication instance I and a row-column 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 row-column minimum load routing for I, resp. the number of colours used by an optimal all-optical routing for I. As mentionned before, one has $\pi(S) \leq \omega(S)$, and therefore $\pi(I) \leq w(I)$ as well.

Given a communication instance I in a mesh, any row-column routing S for I can be coloured into an all-optical routing for I using $8\pi(S)$ colours at most ([Pal02a] claimed $9\pi(S)$, [Pal02b, page 70] put it to $8\pi(S)$). Taking advantage of the row-column minimum load routing problem being $2\text{–}APX$, we can show the row-column all-optical routing problem to be APX as well.

Theorem 10 *The row-column all-optical routing problem is 16–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 row-column minimum load routing algorithm whose existence is asserted by theorem 7, and let $c(S)$ be the number of colours used by the paths colouring algorithm from [Pal02a].

We then have $c(S) \leq 8 \times \pi(S) \leq 8 \times 2 \times \pi(I)$, and we conclude with the general inequality $\pi(I) \leq \omega(I)$. □

4 Conclusion

The all-optical routing problem and the minimum load routing problem are both $\mathcal{NP}$*–hard* in general, and it is not known wether they are APX or not.

Restricted to meshes, these two problems are known to be $\mathcal{NP}$*–hard*. In this paper, we proved this still holds even when restricting routings to be row-column routings. Contrary to the general case for which no answer seems to be known, we proved the two problems to be APX.

Speaking of the row-column all-optical routing problem, and due to the indirect proof of the result, we think the constant asserted in the 16-APX result (see theorem 10) should be improved upon.

Regarding the 2–factor algorithm for the row-column minimum load routing problem, it might also be improved upon, bearing in mind the $\frac{3}{2}$ limit expressed by theorem 8

Last, it is worth noticing that some results can be extended from meshes to tori[7].

[7]Given integer i, $C_{[i]}$ denotes the graph obtained by adding edge $\{i, 0\}$ to path $P_{[i]}$. A

Row-paths and column paths can be defined in tori as they are in meshes, extending straightforwardly row-column paths and row-column routings to tori. On the one hand, as any communication instance on a ring can be mapped on a single row (or a single column) of a torus, the row-column all-optical problem is $\mathcal{NP}$*–complete* in tori as is it in rings. On the other hand, viewing a mesh as a "portion" of some larger torus and using adequate "blocking requests" to confine routings inside such portions, the row-column L–load routing problem can be proved to be $\mathcal{NP}$*–complete* for $L \geq 2$ in tori as in meshes (see appendix for details). Futhermore, the row-column minimum load routing problem in tori, thus $\mathcal{NP}$*–hard*, can be proved to be 8–APX (from theorem 6). Last, using again APX results from [Pal02a] (or [Pal02b]), the row-column all-optical routing problem can be shown to be APX in tori (a priori with rather a large constant, namely 224, but still an APX problem).

References

[ABNC+94] 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.

[ABNC+96] 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.

[BBG+97] 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 IEEE Press, editor, *Proceedings of 2nd IEEE Workshop on Optics and Computer Science*, Geneva, Switzerland, apr 1997. part of IPPS'97.

[BBP+96] Sandeep N. Bhatt, Gianfranco Bilardi, Geppino Pucci, Abhiram Randade, 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.

[Bea00] Bruno Beauquier. *Communications dans les réseaux optiques par multiplexage en longueur d'onde.* PhD thesis, Université de Nice - Sophia Antipolis, jan 2000.

[Chi97] Thierry Chich. *Optimisation du routage par déflexion dans les réseaux de télécommunications métropolitains.* PhD thesis, École Normale Supérieure de Lyon, dec 1997.

cycle is a graph isomorphic to $C_{[i]}$ for some integer i. Given integers i and j, $T_{[i\times j]}$ denotes the cartesian sum of $C_{[i]}$ and $C_{[j]}$. A ***torus*** is a graph isomorphic to $T_{[i\times j]}$ for some integers i and j.

[EJ97a] 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.

[EJ97b] 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.

[EJ01] Thomas Erlebach and Klauss Jansen. The complexity of path coloring and call scheduling. *Theoretical Computer Science*, 255:33–50, 2001.

[Erl99] Thomas Erlebach. *Scheduling Connections in Fast Networks.* PhD thesis, Technische Universität München, 1999.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability : A Guide to the Theory of NP-Completeness.* Freeman, 1979.

[KL84] 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.

[KP96] 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.

[LS95] J. K. Lenstra and D. B. Shmoys. *Scheduling Theory and its Applications*, chapter 1 Computing Near-Optimal Schedules. John Wiley and Sons, 1995.

[MKR95] Milena Mihail, Christos Kaklamanis, and Satish Rao. Efficient access to optical bandwidth - wavelength routing on directed fiber trees, rings, and trees of rings. In IEEE Press, editor, *Proceedings of 36th IEEE Annual Symposium on Foundations of Computer Science (FOCS'95)*, pages 548–557, oct 1995.

[Pal02a] 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.

[Pal02b] Jérôme Palaysi. *Problèmes algorithmiques dans les réseaux tout-optique.* PhD thesis, Université Montpellier II, dec 2002.

[Pap94] Christos H. Papadimitriou. *Computational Complexity.* Addison-Wesley Publishing Company, 1994.

[Rab96] Yuval Rabani. Path coloring in the mesh. In *FOCS*, pages 400–409, 1996.

[RU94] Prabhakar Raghavan and Eli Upfal. Efficient routing in all-optical networks. In ACM Press, editor, *Proceedings of the 26th ACM symposium on Theory of Computing (STOC'94)*, pages 134–143, may 1994.

[SSV97] 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.

[Vaz01] Vijay V. Vazirani. *Approximation Algorithms.* Springer, 2001.

[Wes96] Douglas B. West. *Introduction to Graph Theory.* Prentice-Hall, 1996.

[WW98] 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.

Appendix A

In [BBP+96], the following problem is considered:

instance: a communication instance I_c in a mesh and a routing R for I_c;

solution: a positive integer time assignment $\sigma(r)$ to each request r of I_c such that whenever two requests of I_c are assigned the same time, their associated paths in R are edge-disjoints;

objective: minimize $\sigma(R) = \max_{r \in R} \sigma(r)$.

Assuming that communication r lifetime lays in time interval $[\sigma(r), \sigma(r)+1[$, no interference is to be feared, and this objective reduces to minimizing the overall communication duration.

Obviously, it is equivalent to colouring the paths in routing R in such a way that no two paths sharing an edge should be assigned the same colour and so that the least possible number of colours is used.

In their section 5.2, the authors restrict the problem to so-called ESM (Eastward-Southward Mesh) networks: "The $N \times N$ Eastward-Southward mesh network has node-set $V = \{(i,j) : 0 \leq i,j \leq N-1\}$ and arcs connecting each node (i,j) to node $(i+1,j)$ providing that $i < N-1$, and to node $(i,j+1)$, providing that $j < N-1$".

Then their theorem 8 states that a time assignment σ can be computed in polynomial time so that $\sigma(R)$ doest not exceed twice the load of R, leading to the same result as in [Pal02a] that we use in section 3 to prove our theorem 8.

However, we do not feel confident in their proof which, as we understand it, performs a greedy assignment of σ to a family of paths restricted to row-column paths which start going eastward then turn to go southward, and according to the following points (see figure 5):

- To scan node (i,j) means to provide the least "feasible" integer as its time assignment $\sigma(r)$, doing so, one at a time and in any order, to each request r whose associated path has its "turning point" on node (i,j).

- Nodes belonging to a same diagonal of equation $i - j = constant$ are to be scanned consequently and in ascending order of i and j.

- Diagonals are to be scanned one after the other in decreasing order of the *constant* (thus nodes $(N-1,0)$ and $(0,N-1)$ are the firt and the last node to be scanned respectively).

Figure 6 shows a routing R with load 2, while it seems that the last request to be scanned will be assigned time 5, therefore not an assignment fulfilling theorem 8 requirement in [BBP+96].

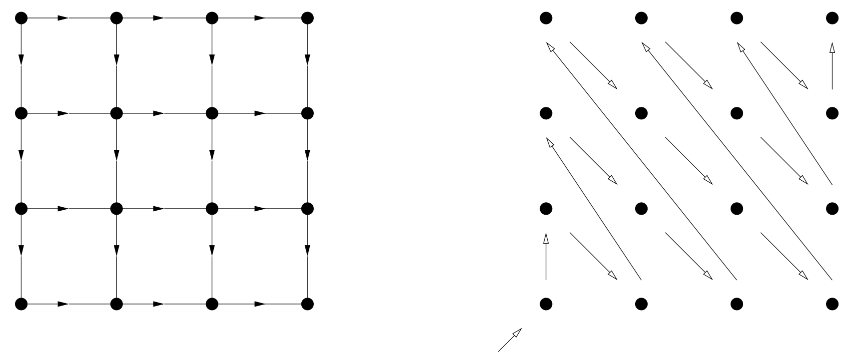

Figure 5: Figure (a) shows a so-called 4×4 ESM. Figure (b) shows the order according to which its nodes are to be scanned.

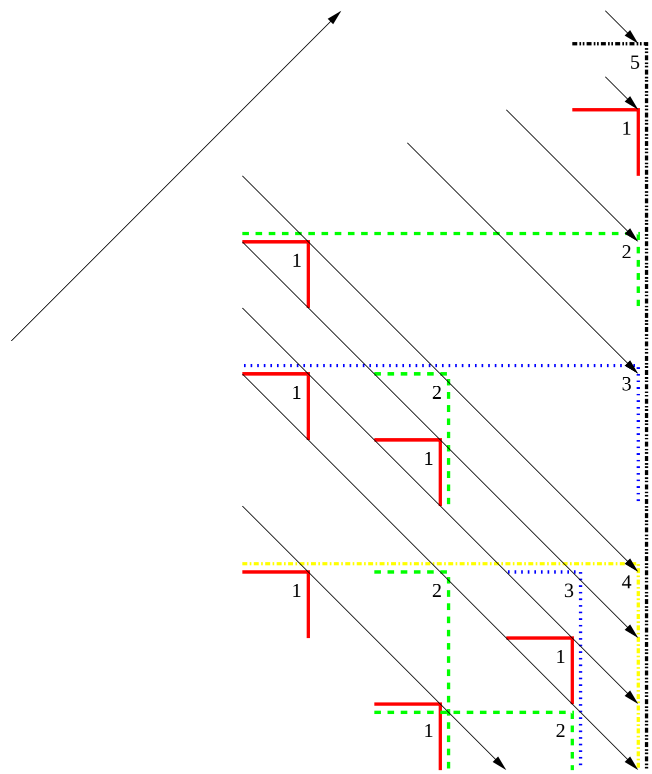

Figure 6: Seemingly a counterexample to the proof of theorem 8 in [BBP[+]96]: the routing load is 2 while the last path to be scanned, according to our understanding of their algorithm, is assigned number 5, thus exceeding twice the load.

Appendix B

We extend the row-column L–load routing problem completeness from meshes to tori.

Theorem 11 *In tori, the row-column L–load routing problem is $\mathcal{NP}$–complete for $L \geq 2$.*

Proof We reduce the row-column L–load routing problem in meshes to the row-column L–load routing problem in tori.

Let M be the mesh $M_{[m \times n]}$ for some integers m and n. Let I_c be some communication requests on M. Let L be some positive integer and I be some row-column L–load routing instance whose network is M and communication instance is I_c.

Let T be the torus $T_{[(m+1)\times(n+1)]}$ and let I' be the row-column L–load routing instance whose network is T and whose communication instance I'_c is defined by adding to I_c the following so-called blocking communication requests:

- for all $j \in [0, n]$, L requests $\{(m+1, j), (m, j)\}$ and L requests $\{(m+1, j), (0, j)\}$
- for all $i \in [0, m]$, L requests $\{(i, n+1), (i, n)\}$ and L requests $\{(i, n+1), (i, 0)\}$

If R_M is a routing of load at most L satisfying I then I' can be satisfied by $R_M \cup S$ where S is a routing satisfying each blocking request by a path of length 1. This routing is of load L.

Conversely, if R_T is a routing of load at most L satisfying I' then the paths of R_T satisfying the blocking requests saturate the edges that where added to the mesh (see figure 7), as deduced from the following facts:

- the number of blocking requests is $2L(n+1)+2L(m+1)$ and each blocking request has one vertex in $V(M)$ and one outside;
- the number of edges with one end point in $V(M)$ and one outside is $2(n+1)+2(m+1)$.

Thus the restriction of R_T to I_c is a routing in M of load at most L and a solution to I.

Which leads to conclusion.

$\square$

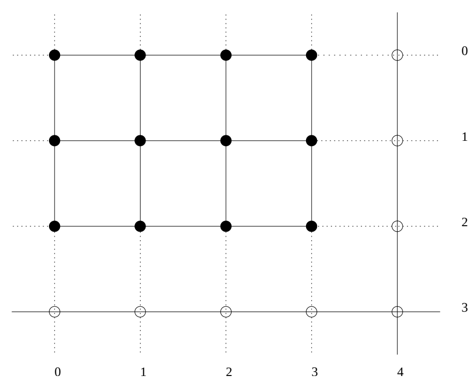

Figure 7: A mesh $M_{[3\times 4]}$ (solid edges). A torus $T_{[4\times 5]}$ (solid and dotted edges). Blocking requests saturate the dotted edges.