Traffic Grooming in WDM Networks with Multi-Layer Switches

G. Huiban, S. Pérennes and M. Syska
MASCOTTE I3S-INRIA 2004, route des Lucioles
B.P. 93 - F-06902 Sophia Antipolis Cedex, FRANCE

Abstract— **We develop traffic grooming algorithms for WDM networks with multi-layer switches. We consider a node as an N-layer switch, in which a given layer k is an aggregated set of elements of layer $k-1$. Typical examples of layers are wavelengths, bands and fibers. The cost of a given node depends on the number of input and output ports of each layer. Assuming this model and a traffic matrix - with unity elements in layer 0 - minimizing the cost of the network will consist in grooming traffic in such a way that as much traffic as possible will be switched in the higher possible layer (fibers in our example).**

When some traffic is switched along a path in the network within the same layer, we represent it with a pipe. Each pipe has an associated linear cost depending on the current layer and on the number of nodes crossed in that pipe. In the case of a two layers model the problem was considered in [1] for rings or in [2] for general topologies.

We present an integer linear programming formulation for this model that aims to minimize the overall cost of the network for a given input traffic matrix. We ran experiments using the CPLEX optimization package on various topologies such as actual networks like the Pan-European all optical network [3] as well as rings and meshes of various sizes.

I. INTRODUCTION

Optical networks using WDM are now widely deployed in carrier networks. One important problem is the design and planning of such networks. Given a traffic matrix of demands between all node pairs, and given capacity constraints on the links, the problem is to find the best traffic routing and traffic grooming solution that minimizes the cost of the equipment required in the nodes. We have developed a software tool dedicated to that problem within the cadre of the French national research program in telecommunications (project RNRT PORTO). In this article we present the model of network defined in that project and the grooming algorithms we have developed. The physical paths assigned to the demands are computed in the first step of the optimization process. Hence the set of paths used as an input of the grooming problem may not lead to an optimal solution of the general problem (routing and grooming). However, the optimization process may be done in several steps over subsets of demands (for instance by sorting them according to their sizes) and the physical paths could have been fixed by the manager of the network on some geographical or protection purposes.

A lot of work has been done on traffic grooming in SONET/SDH WDM rings. The general traffic grooming problem being NP-complete [4], recent works focus on specific issues. Most of the algorithms aim to groom traffic in such a way that all the traffic between any given pair of nodes is carried on the minimum number of wavelengths (efficient use of the fiber). However, a large part of the cost depends on the size of multiplexing equipment required at each node. Hence, in order to minimize the overall network cost, algorithms have to take into account a tradeoff between the number of wavelengths used and the number of required ADM's [1], [5].

In this article we consider a model of network in which the topology is arbitrary and the data is transmitted into a set of hierarchical optical containers. For example wavelengths (W) are included in bands (B) that are included in fibers (F). Each level in the hierarchy is called a *layer* and is mapped to a specific switching cost.

This model has been defined with our industrial partners in the PORTO project as an abstraction of the existing devices. This includes systems made of Optical Cross Connect operating in different layers F-OXC, B-OXC, W-OXC or Photonic Switches (PXC [6]). We believe that it is general enough so that the results may be useful for more specific networks by adjusting the cost function and the number of layers.

In section II we introduce the network model with a 3 layers example. In section III we define the grooming problem associated with that network model. In section III-A we present the integer linear program, the algorithms and heuristics we have implemented. Then we discuss complexity and efficiency issues in section IV. Results of tests and experiments are provided in section VI.

II. OPTICAL NETWORK MODEL

We state the model for WDM optical network by describing the switching capabilities and by defining pipes.

A. Multi-layer switches

For sake of simplicity we first introduce switching capabilities in the 3 layers case. The corresponding switch is pictured on Fig. 1.

Let us consider the example of traffic on Fig. 2. The network is a tree made of 7 nodes, $N_0, .., N_6$, connected with one fiber per link. A fiber is a container of size 2 bands and each band is a container of size 2 wavelengths. Three demands of size 1 wavelength each ($N_0 \rightarrow N_5$, $N_0 \rightarrow N_6$, $N_1 \rightarrow N_6$) are routed using shortest paths. In the wavelength layer this corresponds to the 3 arrow paths.

0-7803-7400-2/02/$17.00 (C) 2002 IEEE

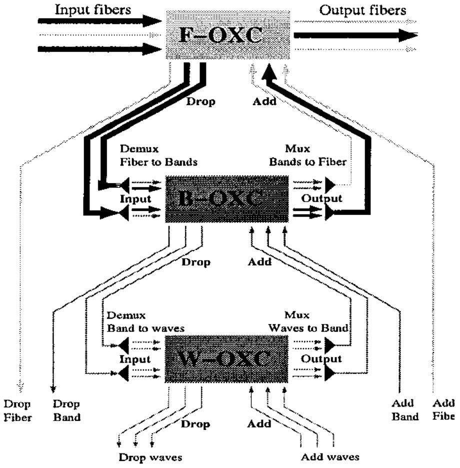

Fig. 1. A 3-layer switch

Now let us focus on the processing of these paths by the 3-layer switches of nodes N_2, N_3 and N_4. Switch N_2 processes the incoming traffic at the band layer as depicted on Fig. 1 using black bold arrows. N_2 has to select the included bands of the two incoming fibers from N_0 and N_1 and also has to multiplex them into the same outgoing fiber to N_3. This processing step requires one F-OXC equipped with at least two input ports and two output port connected to a B-OXC so we could look inside these two fibers. Then, the B-OXC will use 4 input ports in order to process the 4 bands included in the 2 input fibers and 2 output ports for the resulting output dedicated to N_3. The two bands will be fitted into one input port of the F-OXC box and switched through one additional output port. The case of node N_3 is different: all the bands included in the input fiber have to be switched onto the same output fiber connected to N_4. In this case we only need to provide N_3 with an F-OXC with one input port and one output port. This is the best grooming one can expect since switching occurs with the simplest OXC, that is the F-OXC one. Last, N_4 has to use a W-OXC to separate $N_0 \rightarrow N_5$ and $N_0 \rightarrow N_6$.

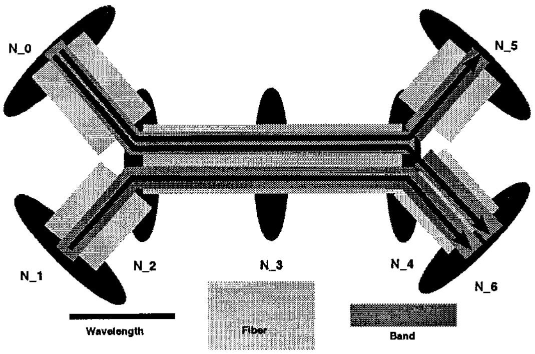

Fig. 2. A grooming example

B. Pipe definition

When some traffic is routed along a path in the network and switched within the same layer on some sub-path, we represent it with a pipe.

In the example of Fig. 2 we observe that some traffic is carried on the network within the same layer on certain sub-paths. This is the case of demand $N_1 \rightarrow N_6$ that crosses the network in the band layer and never has to be processed by a W-OXC. We will call that sequence of bands from N_1 to N_6 a band-pipe. Another band-pipe exists from N_0 to N_4. Note also a fiber-pipe from N_2 to N_4. The grooming algorithm decides the way pipes of lower layers are included in pipes of upper layers. A different grooming could have been done by multiplexing $N_0 \rightarrow N_6$ and $N_1 \rightarrow N_6$ at node N_2 and would have lead to a different set of pipes.

Usually one considers that F-OXC's are much less expensive than B-OXC's and that B-OXC's are much less expensive than W-OXC's. Hence the grooming algorithm has a strong influence on the cost of nodes.

Note that we do not consider here the wavelength-continuity constraint along a pipe but usually a good grooming algorithm leads to a simpler wavelength assignment problem. The general wavelength assignment problem was deeply treated in [7].

III. The grooming problem

The topology of the physical layer, the links capacity, the traffic demand of each node-pair and the routing of demands are fixed as input of the grooming problem. The output is a set of pipes that could transport the traffic under capacity constraints and that minimize the cost of pipes defined below. In example 2 fiber-pipes are layer 3 pipes with capacity 2, bands-pipes are layer 2 pipes with capacity 2 and wavelengths-pipes are layer 1 pipes.

Demands of size s are considered as s demands of size 1 that have each an associated path computed in the routing phase of the optimization process.

In order to generate "solvable" problems, we define the grooming problem over only two layers. However, solving this problem on k layers can be approximated by solving the problem recursively. In the case of example 2 that means a two steps algorithm: first step is grooming of wavelengths into bands, and second step is grooming of bands into fibers.

Each pipe of layer l has a capacity (number of pipes of layer $l-1$ it can carry) and an associated price depending both on thc pipe length and on l.

A pipe belonging to layer l could be defined as a virtual - sized and priced - edge of the virtual graph G_l that represents the connections available in layer l. The initial set of demands defines the set of arcs of G_0.

The design of the network consists in choosing which pipes are used in each layer such that the pipes of layer l are carried in the pipes of layer $l+1$ and the number of pipes of layer k

going through an edge e of the physical network do not exceed the capacity of that edge.

Example 1: in the case of the problem of Fig. 2, one solution is to use the 4 band-pipes $N_0 \rightarrow N_4$, $N_1 \rightarrow N_6$, $N_4 \rightarrow N_5$, $N_4 \rightarrow N_6$. Another solution would consist in using pipes $N_0 \rightarrow N_2$, $N_1 \rightarrow N_2$, $N_2 \rightarrow N_6$, $N_4 \rightarrow N_5$.

Then the problem is to find the min-cost set of pipes among all possible pipes that can be selected at each layer. The set of all elementary dipaths of the network has an exponential size but, hopefully, pipes that are not a sub-path of a traffic demand dipath are not to be considered. Indeed it is impossible to use them while a traffic demand entering a pipe is carried till the other pipe extremity. An example is shown on Fig. 3. Hence, we restrict the set of pipes to the set of all sub-dipaths of every demand dipath. Let $\mathcal{D}$ be the set of demands, and for $d \in \mathcal{D}$ let $|d|$ be the length of the physical path assigned to d. Then the size of the pipe set is less than $\sum_{d \in \mathcal{D}} (|d| \times (|d|+1))/2$. This is suitable for practical computations as the average length of demands dipaths is usually small (less than 3).

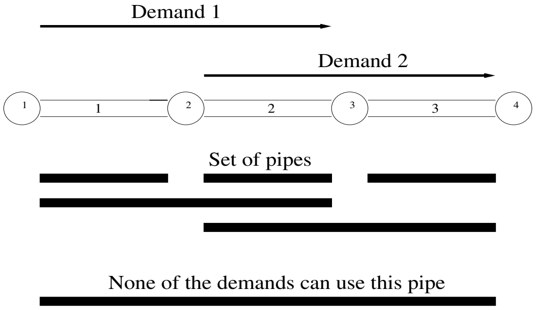

Fig. 3. Useless pipes

As we said before we restrict our study to the two layers sub-problem : pipes of layer l are considered as traffic demands that must be carried by the layer $l+1$ that we aim to design. Therefore, when considering a pair of consecutive layers, we are facing an assignment problem: a set of pipes is available, and a set of demands (lower layer pipes) have to be groomed inside them.

This model has some advantages: it prevents from modeling a switch (which would increase the complexity of the model, as in [2]); and it gives a more generic approach to the problem: it introduces a pure grooming problem in terms of an intuitive combinatorial problem .

This bottom-up approach is not necessarily optimal, since an optimum cost for layer 1 could turn out to be poorly groomable in layer 2. This question will not be addressed here, since we focus on establishing a framework and giving results for a two layer network which can be used recursively for several layer networks.

A. Integer linear problem formulation

Here we present the integer linear formulation of grooming layer l pipes into layer $l+1$ pipes. Recall that the traffic demands can be formally considered as layer 0 pipes.

1) Model for the physical network: The network is modeled by a directed graph $G = (V, E)$.

- $V = \{v_1, \ldots, v_n\}$ the set of the n nodes of the network.
- $E = \{e_1, \ldots, e_m\}$ the set of m edges of the physical network $e_i = (x_i, y_i)$, $(x_i, y_i) \in V^2$.
- $\mathcal{D} = \{d_1, \ldots, d_q\}$ the set of q unit traffic demands. Formally d_i is the dipath assigned to the demand (i.e. a path on the physical graph).

As the edges have capacity constraints, let define the capacity function by: $C_a : E{\rightarrow}N; e{\mapsto}C_a(e)$.

2) Model for the logical network: We suppose that the set of potential pipes of the current layer, their capacity and their cost are known. Let be $P = \{p_1, \ldots, p_k, \ldots, p_p\}$ be the set of p pipes. Formally we can consider p_k as an elementary dipath of the physical network. To each potential pipe p are associated a capacity $Cap(p)$ and a price $cost(p)$: $P{\rightarrow}N$ $, p{\mapsto}Cap(p)$; $P{\rightarrow}N$ $, p{\mapsto}cost(p)$.

Note that for practical application at layer l, $Cap(p)$ will be usually a constant C_l equal to the number of layer $l-1$ pipes that a pipe of layer l can carry.

We also define the following sets :

- $\forall e_i \in E,\ \Gamma_i$ is the set of pipes using the edge e_i: $\forall p_k \in P \quad e_i \subset p_k \Rightarrow p_k \in \Gamma_i$
- $\forall d_i \in \mathcal{D},\ \Omega_i$ is the set of pipes that the traffic demand d_i can use: $\forall p_k \in P \quad p_k \subset d_i \Rightarrow p_k \in \Omega_i$

Last we define the set $P_{i,j} = \Omega_i \cap \Gamma_j$. It represents the set of pipes that the traffic demand d_i can use on the edge e_j.

3) Variables of the model:

- $X_{i,j} \in \{0, 1\}$, indicating if the traffic demand d_i uses the pipe p_j or not.
- $P_j \in N$ gives the number of used pipes p_j.

4) Constraints:

$$\forall d_i \in \mathcal{D}, \forall e_j \in E : \sum_{p_k \in P_{i,j}} X_{i,k} = 1, \tag{1}$$

$$\forall p_k \in P, \sum_{d_i \in \mathcal{D}} X_{i,k} \leqslant P_k Cap(p_k) \tag{2}$$

$$\forall e_i \in E, \sum_{p_k \in \Gamma_i} Cap(p_j) \times P_j \leqslant\ C_a(a_i) \tag{3}$$

- Equation 1 ensures that the grooming is consistent, that is a traffic demand uses exactly one pipe on each edge of the network, thus if a demand uses more than one pipe, the pipes follow each other.
- Equation 2 states the capacity constraint for the layer $i+1$ pipes : it ensures that the number of layer i pipes using one layer $i+1$ pipe of kind p_j is less than the capacity allowed by this pipe which is $P_j Cap(p_j)$.

- Equation 3 is the capacity constraint on an edge of the physical network. It ensures that the next stage of the grooming will always be feasible.

5) Objective function: The main purpose of this study is to minimize the cost of the pipes. The objective function is naturally: Min $\sum_{p_k \in P} \; cost(p_j)P_j$

B. Potential pipes set and pipes cost

We choose as a set of potential pipes in layer $l+1$ the set of all the subpaths of existing pipes in layer l.

The switching equipment cost and the bandwidth cost are taking into account by pricing the layer l pipes. A simple cost function for that is a linear one which prices a layer l pipe of length n with $cost = \alpha_l + \beta_l n$ where α_l represents the cost induced by having at each pipe extremity switching equipment suitable for layer $l-1$ pipes, $\beta_l n$ represents the cost of the bandwidth used and also the cost of the simpler intermediary switches able to process containers in layer l.

Note that choosing $\alpha_l = 0$ means that no economy is made when grooming. Then the grooming problem becomes trivial since choosing pipes of length 1 (i.e. edges) in such a way that the capacity constraint is satisfied is an optimal solution. On the opposite, taking $\beta_l = 0$ means that our aim is to minimize the number of pipes, even if this induces a large bandwidth waste. In practical applications the cost function provides a tradeoff between the bandwidth and switches costs.

IV. Complexity and efficiency

Let L be the maximum length of a path on the physical network, r the number of traffic demands, p the number of pipes, and m the number of edges. The number of equations is $O(pL^2)$; and the number of variables is $O((m+p)p)$. In order to reduce this size, we perform some additional operations.

A. Multiple demands and split

A slight modification of the initial linear problem allows to treat M multiple unit demands almost as a single one. The idea is to replace unit demands by weighted ones. Note that this cannot be done trivially since to obtain an efficient solution one demand with weight M may be carried by several sequences of pipes; which means that the demand may have to be *split*.

Example 2: Let consider a 4 nodes path $A \rightarrow B \rightarrow C \rightarrow D$ and assume that every pipe of level 1 has capacity 8. Suppose the 4 demands of size 4: $R_1 = A \rightarrow B$, $R_2 = A \rightarrow C$, $R_3 = C \rightarrow D$, $R_4 = B \rightarrow D$, and demand $R_5 = A \rightarrow D$ of size 8. Then a good solution would be to use the 4 pipes AB, AC, CD, BD; the traffic would then be carried as follows : R_1 uses AB, R_2 uses AC, R_3 uses CD and R_4 uses BD. Note that once this is done each pipe still can carry 4 more units of traffic. Hence we split R_5 into two parts, each carrying 4 units of traffic. One part is sent along AB and BD, the other one along AC and CD.

In order to take this splitting technique into account we allow each demand of weight M to be split into s_{max} similar virtual demands with some non fixed weights $w_1, w_2, \dots w_{s_{max}}$ satisfying $\sum w_i = M$. Then, each virtual demand must be packed into a unique sequence of pipes. The linear program is then written by considering these virtual variable weighted demands as being the initial demands of the problem.

The main advantage is that this reduces notably the size of the problem, especially when the weight of some demand is large. Note that the choice of the parameter s_{max} is quite important, since choosing $s_{max} = 1$ is equivalent to forbid the splitting of demand which can lead to poor quality solutions. On the opposite, taking s_{max} equal to the maximum weight of a demand always leads to an exact formulation but is equivalent to the unitary model and increases the size of the linear problem. In order to be able to use a small value for s_{max} the idea is to perform a preliminary phase allowing to decrease the demand weight.

B. A preliminary phase to reduce demand maximum weight

In order to decrease the weight of the demands, we perform first some trivial grooming. By trivial we means that if a demand weight is larger than the capacity of a pipe, we decide to use a direct pipe following exactly the demand dipath to carry C_i units of the demand. Note that during this phase no capacity is wasted.

Once this phase has been performed, the demand weight is smaller than the pipe capacity C_i, which allows us to use a small number of splits. This has a dramatic impact on the number of variables (and especially of decision variables) of the linear program generated.

C. Filtering the set of potential pipes

From initial experiments it turned out that taking all the subpaths of existing demands as the set of potential pipes was still leading to hard integral linear program, indeed this set was still too large.

Hence the idea was to decrease its size by removing pipes which are a priori useless. The main goal was to discard any long pipe that would in any case be almost empty. We focused of those pipes for several reasons: first, the vast majority of the pipes belong to this set, since there is only a few short pipes, and most long pipes are likely to be used by only a few units of traffic. Secondly such pipes are not likely to be used, since they “waste” a lot of capacity. Last, the existence of such pipes hardens the problem a lot since it induces very large gaps between the integral and the real solution of the linear system.

Example 3: Assume the path of example 2 and demands $A \rightarrow D$ with weight 1 and $B \rightarrow D$ with weight 7. A good solution would be to choose two pipes AB and BD in such a way that the pipe BD will be completely filled, and the pipe AB with length 1 would be only $1/8$ filled. If the pipe AD is taken as a member of the potential pipes set, the real solution of the linear program would be to take pipe BD and $1/8$ of the pipe AD.

In this situation the real solution is quite far from the integral one, our idea is then to forbid pipe AD since it is a long pipe which is such that only one unit of traffic can use it.

In order to eliminate such pathological pipes we use an evaluation function f that given a pipe, its length l and the maximum amount of traffic t that it can use, assigns a pipe grade $f(l, t)$. All pipes receiving a grade lower than $mingrade$ are discarded. Note that by tuning the value $mingrade$ the selectivity of this process can be adjusted (as an example we can keep 10 percents of the best pipes).

For experiments we choose $f(l, t) = lt$. Note that other kinds of criteria $p \mapsto grade(p)$ can be used.

D. Conclusion on the LP section

The brute LP problem was quite long to solve, even on small networks, but once the preliminary greedy allocation and the pipe filtering were performed, we obtained fast running times (with a split value s_{max} taken small (3 for instance). Moreover, the quality of the solution did not suffered too much from the 3 simplifications (greedy allocation, pipe filtering and controlled split) that we performed.

V. "Greedy" algorithms

In this part we outline greedy algorithms that we have implemented, they are simple, naive and very fast. They all rely on well known principles for covering problems (selection of minimal cost, or best payoff, element for covering, see [8]).

In order to use such an algorithm we need a way to evaluate how good is a pipe. For this we use a function $grade(p)$ which assigns some pertinence factor to each pipe. The greedy algorithm then follows :

```
While it exists demands and while the capacity constraints are
respected
  1) For each pipe $p \in P$, compute $grade(p)$.
  2) Let $p_0 \in P$ s.t. $grade(p_0) = max_{p \in P}\{grade(p)\}$.
  3) Check if the problem is still feasible if $p_0$ is chosen.
  4) If YES, add $p_0$ in the set of selected pipes, and update
     the traffic considering the use of $p_0$.
  5) If NO, discard $p_0$.
```

Note that the algorithm depends mainly on how we grade the pipes. Indeed the algorithm performing first the preliminary phase of the LP algorithm and then using pipes of length 1 belongs to the greedy family.

As we wish for full and long pipes, natural grade functions are lt (where l is the pipe length, and t the potential traffic crossing), but several others can be used (one could for example require a pipe to have a potential use of at least 50 percents of its capacity to receive a good grade).

VI. Numerical results

The algorithms have been implemented in C++ and use the ILOG-CPLEX optimizer for the linear programming part. Experiments have been done on real networks like COST239 [3] or the French backbone (Long distance) as well as for random instances on rings and meshes.

A. Tradeoff

Due to the linear cost of pipes in layer l, $\alpha_l + \beta_l n$, the nature of the problem depends on the ratio β_l / α_l. When $\alpha_l << \beta_l$, grooming is useless while when $\beta_l >> \alpha_l$ all the pipes have the same cost (which means that one must minimize the number of pipes). In Fig. 4 the impact of this ratio is depicted, note the continuous evolution from a situation where grooming is high and profitable (when $\beta_l << \alpha_l$, to a situation in which grooming do not yield any profit when $\beta_l >> \alpha_l$). For the same reason, the total number of pipes will increase with β_l, while the total bandwidth in use should mainly decrease (since a high grooming induces bandwidth wasting). This is depicted in Fig. 5. In both figures, the results are given in doted line for the Long distance network and in solid line for a 20 nodes ring with 80 demands.

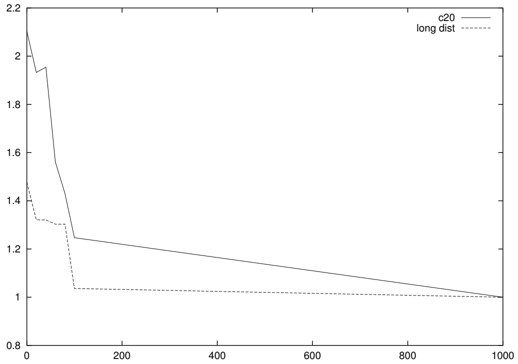

Fig. 4. Average length of pipes when the value of β_i grows (α_i is 100)

B. Number of splits

In this section we give some practical results on the impact of the number of splits. As already claimed, a small number of splits induces smaller ILP and considerably shorter running times. We observed that allowing 3 splits was enough to reach optimality in all the examples we have tested. Note that our linear programming formulation allows the number of splits to depend on the request weight which would probably turn out to be a better choice than a uniform number of splits.

C. Pipe filtering

In this section, we provide some practical results on how the filtering of pipes according grades impacts on the solution

0-7803-7400-2/02/$17.00 (C) 2002 IEEE

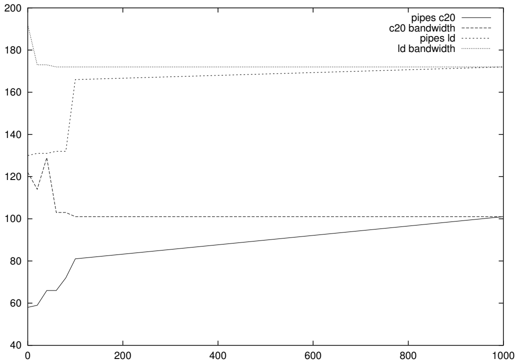

Fig. 5. Total number of pipes when the value of β_i grows, (α_i is 100), and total bandwidth used

quality. We take as an example the French optical backbone network with different pipe grade thresholds. The number of pipes decreases with higher thresholds, and this reduces highly the average solving time. Note that a high selectivity seems to be the good approach (see Fig. 6). We even noticed that, in some cases, the solution found with a high threshold was slightly better than the solution found with the whole pipe set (threshold equals 0). This phenomena was due to the time limits we gave to CPLEX for finding a solution.

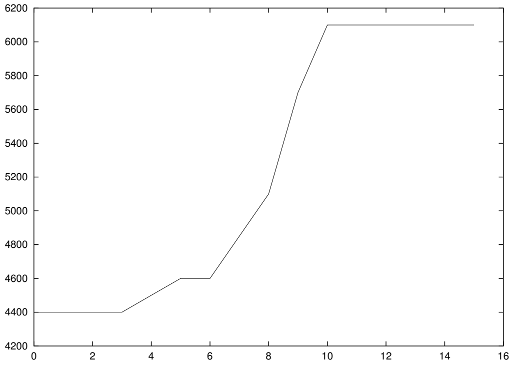

Fig. 6. Cost in function of the threshold

D. Greedy algorithms

Experiments have been made with the greedy algorithms described in section V. Those algorithms are totally independent from any solver and are very fast: the result is almost immediate, whatever may be the networks. However the cost of the solutions found was usually much higher than the one found with the CPLEX solver.

VII. CONCLUSION

In this paper we have presented a framework to study the grooming problem in multi-layer optical networks. We have used a pipe pricing function in order to capture the node equipment and bandwidth costs. Our approach is more general and more accurate than existing ones that use to minimize the number of add-drop nodes.

We also have proposed several problem simplifications (greedy pre-processing, the use of splits, pipe filtering) allowing to find almost optimal solutions for existing networks and some computer generated ones. Last, we proposed fast greedy algorithms.

Our experimental results showed that these simplifications are reasonable, and they highlight an existing tradeoff between grooming the traffic and using the bandwidth efficiently.

From a theoretical point of view, the exact difficulty of the problem is still unknown, indeed the problem is NP-hard but we don't know if polynomial approximation algorithms exist or not. Moreover efficiency of fast algorithms (e.g. like greedy selection of best payoff pipes) is unknown. Note that the problem difficulty depends highly on the pipe pricing: with a pipe cost proportional to its length the problem is trivial while with a constant pipe cost the problem is equivalent to minimizing the number of add-drop nodes which is much harder.

Practically, more experiments should be done, especially we intend to improve greedy algorithms first by finding better grading functions, and secondly by mixing them with ILP approach.

Finally, a very important point that we wish to address is to mix routing and grooming. In this work we assumed that the routing was given and fixed. Such a routing could be the one minimizing the bandwidth used but the suitability for grooming may be improved.

REFERENCES

[1] O. Gerstel, R. Ramaswami, and G.H. Sasaki, "Cost-effective traffic grooming in WDM rings," *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 618–630, October 2000.

[2] T. Cinker, D. Marx, C. P. Larsen, and D. Fogaras, "Heuristic algorithms for joint configuration of the optical and electrical layer in multi-hop wavelength routing networks," in *IEEE INFOCOM 2000*, Tel Aviv, Israel, March 2000, vol. 2, pp. 1000–1009.

[3] P. Batchelor et al, "Ultra high capacity optical transmission networks : Final report of action cost 239," Tech. Rep. ISBN 953-184-013-X, Faculty of Electrical Engineering and Computing, HR, Zagreb, 1999.

[4] A. L. Chiu and E. H. Modiano, "Traffic grooming algorithms for reducing electronic multiplexing costs in WDM ring networks," *Journal of Lightwave Technology*, vol. 18, no. 1, pp. 2–12, January 2000.

[5] X. Zhang and C. Qiao, "An effective and comprehensive approach for traffic grooming and wavelength assignment in SONET/WDM rings," *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 608–617, October 2000.

[6] J. Gruber and R. Ramaswami, "Moving toward all-optical networks," *Lightwave*, December 2000.

[7] D. Banerjee and B. Mukherjee, "Wavelength-routed optical networks: linear formulation, resource budgeting tradeoffs, and a reconfiguration study," *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 598–607, October 2000.

[8] D. S. Hochbaum, Ed., *Approximation algorithms for NP-hard problems*, PWS Publishing Company, 1997.

0-7803-7400-2/02/$17.00 (C) 2002 IEEE