Traffic Grooming in Multi-Layer WDM Networks: Meta-Heuristics versus Sequential Algorithms

Elias A. DOUMITH and Maurice GAGNAIRE
Computer Science & Networks
GET/ENST
46, rue Barrault, 75013 Paris - FRANCE
Email: {elias.doumith, maurice.gagnaire}@enst.fr

Abstract— **This paper deals with electrical traffic grooming into optical lightpaths. In the context of a multi-layer EXC/OXC network, electrical traffic grooming aims to optimize resources utilization in terms of required optical channels. It also enables to relax the wavelength continuity constraint. Whereas most of the existing studies are applied either to permanent traffic demands or to dynamic traffic demands, we consider dynamic scheduled traffic demands for which the date of activation and life duration are pre-known. Our objective is to compare two algorithmic approaches: Meta-heuristics and sequential algorithms. The former approach deals with the scheduled demands as a set. It performs global optimization by considering all the possible grooming combinations. The latter approach tries to route the demands iteratively at their instant of arrival. We compare our algorithmic approaches with two other grooming strategies proposed in the literature.**

I. INTRODUCTION

WDM optical networks have been widely deployed as a transport network for long-distance high-speed networks. Nowadays, an optical communication path referred as lightpath can carry about 10 to 40 Gbps of data traffic while the capacity requirement of a traffic request may be far less than that. Thus an essential functionality of WDM networks, referred to as traffic grooming, is to aggregate low speed traffic connections onto high speed wavelength channels in a resource-efficient way. Traffic grooming aims to maximize the network throughput for given network resources or to minimize the resource consumption when satisfying a given set of traffic requests. A key component that enables traffic grooming in mesh networks is an Electrical Cross-Connect (EXC) coupled with an Optical Cross-Connect (OXC). Such hybrid node architecture is able to perform data switching and grooming at the electrical level while it is still capable of multiplexing/demultiplexing and switching traffic streams at the optical level.

In this paper, we mainly consider dynamic traffic demands for which the arrival time as well as the holding time are known. Each request needs to be setup by: 1) determining a route across the network connecting its source node to its destination node; 2) determining at which intermediate nodes the request will pass through the EXC level in order to be aggregated with other requests. As the traffic matrix is known in advance, this routing and grooming problem is resolved into a network dimensioning problem. From the dimensioning point of view, the network must be able to carry the whole data traffic at each instant. The primary design objective is to minimize the network deployment cost.

The paper is organized as follows. We begin with a brief description of the previous works reported for the grooming problem. Section III describes the adopted traffic model as well as the architecture of the hybrid OXC/EXC network node considered in our study. Section IV describes how the grooming process can be physically implemented in the network nodes. In Section V, we describe both the proposed meta-heuristic algorithm as well as the sequential algorithm. These algorithms are used in order to dimension a network able to handle a given set of requests. Extensive simulations have been carried out in order to evaluate the performance of both algorithms. The obtained numerical results are shown in Section VI. In Section VII we provide a summary of our major observations and some directions for future work.

II. PREVIOUS WORK

Traffic grooming started to become a relevant research topic since 1998. The early research efforts on traffic grooming have mainly focused on SONET/WDM ring networks. The major cost of such networks is considered to be dominated by SONET add/drop multiplexers (ADMs). Several optimal and near-optimal algorithms have been proposed to solve the traffic grooming and wavelength assignment problem for ring networks and single-hub ring networks. The objective of such algorithms is to minimize the number of required wavelengths and SONET ADMs. These algorithms focus on static traffic only (permanent circuits). For example, the algorithm in [1] formulated the grooming optimization problem as an ILP for ring networks considering static traffic. In this paper, the authors compare single-hop grooming and multihop grooming for such networks. In this same paper, heuristic approaches are proposed. These heuristics divide the traffic grooming problem into two steps. In the first step, the traffic demands are assigned to circles. In the second step, a traffic grooming algorithm is employed to reorganize the circles or connections

---

A part of this work is associated with the research activities of the NoE e-Photon ONe European Network of Excellence

on wavelengths. The authors propose a Simulated Annealing algorithm for single-hop connections and a greedy algorithm for multihop connections.

Later on, researches began to consider mesh networks instead of ring networks. In this context, the objective of traffic grooming algorithms becomes the reduction of the number of transceivers (electrical ports of the EXC). The work in [2] formulates the static traffic grooming problem as an ILP and also proposes a heuristic approach for this problem. This proposed heuristic is a sequential approach that solves the traffic grooming problem for one connection request at a time. For each request, it tries to simultaneously route and groom the considered request on a suitable path. Recent works [3] begin to consider dynamic traffic patterns in WDM mesh networks. These studies consider mainly sequential algorithm and compute the blocking probability/rejection ratio for WDM networks with constrained grooming capability.

In our approach, we consider the routing and grooming problem for WDM mesh networks under dynamic but deterministic traffic demands. For this task, we have developed two grooming algorithms. Our first algorithm is based on a meta-heuristic approach. It deals with the whole set of demands at once and tries to find the best grooming solution based on the beforehand knowledge of the arrival times, the holding times and the required rates. Meanwhile, our second algorithm is a sequential approach that solves the traffic grooming problem for one connection request at a time. For each request, it tries to simultaneously route and groom the considered request on the available network resources. If this is not possible, the algorithm adds additional resources and routes the request on the least expensive route.

III. TRAFFIC CHARACTERIZATION AND NODE ARCHITECTURE

A. *Traffic Characterization*

The traffic offered to the network is generally decomposed into traffic demands characterized by their bit rate, arrival time, holding time and source/destination nodes. In our case study, we mainly consider requests that are said to be *Scheduled* because their characteristics are known before their instant of arrival and to be *Electrical* because they do not require the whole capacity of an optical channel. Such traffic requests are referred to as Scheduled Electrical Demands (SEDs) opposed to Lightpath Demands (LDs) where the required data rate is equal to the capacity of an optical channel. The traffic request characteristics are generated as follows:

- Each network node i is assigned a weight $p(i)$ that is proportional to the number of citizens surrounding its geographical location. These weights are integer numbers ranging between 1 and 10 and represent the capacity of a node to send/receive data. Consequently, the source and destination nodes of the requests are chosen among the network nodes according to this weight distribution.
- The set-up and tear-down dates of the requests are quantized with a constant quantization step of 30 minutes. Thus the set-up and tear-down dates are chosen uniformly in the set $[0\,..\,48]$ where 48 is the number of half hours in a day.
- The required rate of the requests is chosen uniformly in the interval $]0, 0.9 \times C_\omega]$, C_ω being the capacity of the full optical channel.

Let N_{SED} be the number of SEDs to be generated during the simulation period. The data traffic flow (in bps) entering the network is measured at each instant. The average of this traffic flow is noted ϕ_{SED} while the overall peak of this traffic flow is noted π_{SED}.

B. *Node Architecture*

We have considered a multi-layer node architecture that is able not only to handle sub-wavelength traffic requests such as SEDs but that can add/drop and switch full wavelength traffic requests such as LDs. This multi-layer node (Figure 1) comprises a non-blocking optical cross-connect (OXC) with switching capabilities for wavelength channels coupled with a non-blocking electrical cross-connect (EXC) with switching capabilities for smaller granularities. The OXC and the EXC are connected by a limited number of transponders. This multi-layer node architecture as well as its equivalent auxiliary graph model were largely detailed in [4], [5]. In the following, we briefly describe the basic elements of such architecture.

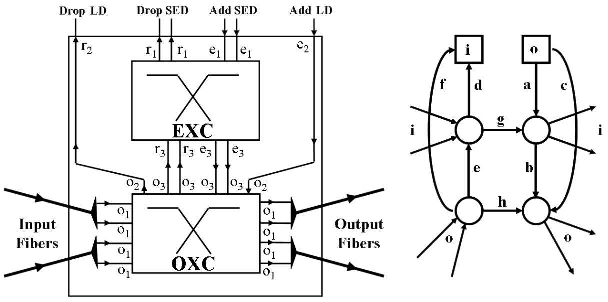

Fig. 1. EXC/OXC node architecture and its equivalent auxiliary graph

In general, SED traffic requests with different granularities are generated at the EXC level. This is achieved by using an emitting electrical e_1-port and is represented by the use of the ***a*** edge in the auxiliary graph. Several SED connections may be aggregated into the same connection and transmitted through the optical layer. Such aggregated connections are referred to as *Grooming Lightpath* (GL). A GL is a direct connection between nonadjacent nodes acting as a logical one-hop link, where all intermediate nodes are passed through at the OXC level. An existing GL is represented by an ***i*** edge in the auxiliary graph. At its source node, the GL is switched from the EXC to the OXC using an emitting electrical e_3-port and an optical o_3-port. This is represented by the use of the ***b*** edge in the auxiliary graph. At the intermediate nodes, the GL remains at the OXC level and it passes through the OXC using two optical o_1-ports. This is represented by the use of the ***h*** edge in the auxiliary graph. When at least one SED belonging to a GL must be extracted, this GL is switched back from the OXC to the EXC using an optical o_3-port and a receiving electrical r_3-port. This is represented by the use of the ***e*** edge in the auxiliary graph. Multihop grooming can be achieved using the ***g*** edge of the auxiliary graph. Finally, an SED request is dropped from the network using a receiving electrical r_1-port. This is represented by the use of the ***d*** edge in the auxiliary graph.

As a result, each node is composed of a set of optical o_i-ports $(i = 1, 2, 3)$, a set of emitting electrical e_i-ports $(i = 1, 2, 3)$ and another set of receiving electrical r_i-ports $(i = 1, 2, 3)$. Due to the need for large buffers and fast electronics in order to process the signals at the EXC level, the cost of an electrical port is higher than the cost of an optical port. Let κ be the ratio of the cost of an electrical port to the cost of an optical port. Nowadays, $\kappa \approx 5$.

IV. TRAFFIC GROOMING PROCESS

The characteristic of SEDs lies on the fact that their flow is smaller than the capacity of a wavelength. This particularity is taken into account by the use of electrical aggregation. The grooming of multiple SEDs consists in multiplexing their electrical flows on a same GL within an EXC. The grooming process requires that all the considered SEDs share at least a common fiber link and that all of them are active during a common period of time [6].

One can show that grooming multiple traffic requests on the same GL can be achieved by a series of consecutive grooming processes where the requests are considered two by two. Consequently, we will consider only the case where we groom together only two requests at a time. The iterative nature of the used algorithms ensures that multiple requests are groomed together.

In order to illustrate the grooming process, let us consider two traffic requests δ_1 and δ_2 given by:

- Source node : S_1 & S_2 resp.
- Destination node : D_1 & D_2 resp.
- Set-up time : α_1 & α_2 resp.
- Tear-down time : β_1 & β_2 resp.
- Rate : ω_1 & ω_2 resp.

In addition, we suppose that these two demands can be aggregated together since they satisfy the grooming constraints (common links, common period of time) as shown in Figure 2. Theoretically, grooming these two demands must yield the aggregated demand $\delta_a\ \big(G_1, G_2, \alpha_2, \beta_1, \omega_1 + \omega_2\big)$.

At node G_1, the two demands δ_1 and δ_2 must be passed to the EXC in order to be groomed together, hence two r_3-ports and two o_3-ports need to be added at this node. Similarly, in order to restitute these two demands, two e_3-ports and two o_3-ports need to be added at node G_2. Also, the aggregated demand δ_a requires electrical (e_3/r_3)-ports and optical (o_3)-ports at both ends. In brief, in order to groom these two demands, six additional electrical ports and six additional optical ports are necessary. However, the number of optical o_1-ports along the common path is reduced. In addition, the number of additional ports can be reduced due to space-time reutilization between the different SED requests.

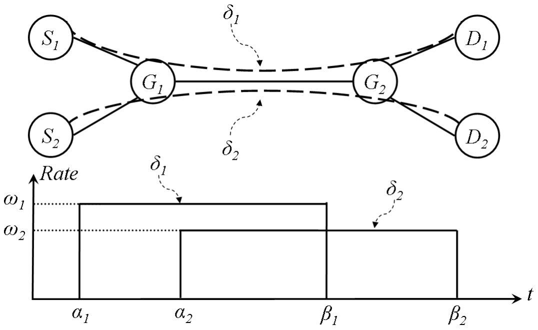

Fig. 2. A simple network for grooming example

Grooming the two requests δ_1 and δ_2 yields to the aggregated demand δ_a and to a set of additional demands. Some of these demands take into account the non common portions of the paths of δ_1 and δ_2. The characteristics of such demands are:

- $\delta_b\ \big(S_1, G_1, \alpha_1, \beta_1, \omega_1\big)$
- $\delta_c\ \big(G_2, D_1, \alpha_1, \beta_1, \omega_1\big)$
- $\delta_d\ \big(S_2, G_1, \alpha_2, \beta_2, \omega_2\big)$
- $\delta_e\ \big(G_2, D_2, \alpha_2, \beta_2, \omega_2\big)$

The rest of these demands take into account the non common periods of time. The characteristics of such demands are:

- $\delta_f\ \big(G_1, G_2, \alpha_1, \alpha_2, \omega_1\big)$
- $\delta_g\ \big(G_1, G_2, \beta_1, \beta_2, \omega_2\big)$

According to the characteristics of the initial demands δ_1 and δ_2, some of these demands may not exist. However, the existing requests among these last six demands form the set of marginal demands. As a result of the grooming process, the set of all the SEDs must be modified by adding the aggregated request δ_a and the set of marginal demands and by removing the original demands δ_1 and δ_2.

V. TRAFFIC GROOMING ALGORITHMS

As just shown, the grooming process implies the use of additional e_3, r_3 and o_3-ports within the network but it reduces the number of required o_1-ports. Our optimization problem is then based on a trade-off between the cost penalty due to additional ports and the benefit due to the reduction in the number of other ports. The overall cost of the network can be reduced thanks to the resource reutilization between different requests. This network cost ζ is expressed as a function of the number ϑ of optical ports and the number φ of electrical ports. A coefficient κ (in front of φ) represents the fact that an electrical port is κ times more expensive than an optical port.

$$\zeta = \vartheta + \kappa.\varphi$$

A. Meta-Heuristics Algorithm

1) Notations: Before detailing the proposed algorithm, let us define some useful expressions and concepts:

- *Common Path Length* (CPL): It is a function that indicates if two demands satisfy the grooming constraints or not. CPL is equal to 0 if the considered requests are not simultaneous, otherwise CPL is equal to the maximum number of consecutive common links/hops used by these two requests.

- *Successful Grooming Pair* (SGP): It is a pair of demands with a positive CPL for which, once the grooming process is applied, the overall network cost is reduced compared to the original overall cost.
- *Unsuccessful Grooming Pair* (UGP): The pair of requests that does not satisfy the previous statement.

2) Proposed Iterative Greedy Algorithm: Once a path is assigned to each request between its source and its destination nodes, one can choose to use an exhaustive algorithm to groom each possible pair of demands. This method is time consuming and requires considerable memory resources. In order to reduce the computation time, one can choose to limit the number of iterations of the algorithm. This can be implemented by using a list T$_list$ with a limited capacity which keeps in memory the pairs of requests marked as UGPs. The algorithm stops once all the CPL are null or the T$_list$ has reached its maximum capacity.

By grooming demands with large CPL, the reduction in the number of required o_1-ports can be important. We hope that this reduction can compensate the increase in the number of required e_3, r_3 and o_3-ports. An intuitive idea is to start trying to groom pair of demands with large CPL before trying to groom pair of demands with smaller one. In order to test a sufficient number of pairs of demands, the capacity of T$_list$ must be set to a large value. This capacity can be reduced significantly if the algorithm is modified as follows:

- The algorithm stops when all the CPLs are null or when we try to groom a consecutive number of pairs of demands without cost improvement. This is implemented by resetting the T$_list$ each time we find an SGP.
- **Step 1:** In order not to neglect the pairs of demands with small CPL, we try to groom pairs of demands with a constant CPL starting from the largest CPL. When all the pairs of demands with this constant CPL are tested and founded as UGP or when the T$_list$ has reached its maximum capacity, we decrease the value of the CPL, reset the T$_list$ and retry to groom additional pairs of demands with the new value of CPL. This is repeated until all the possible value of the CPL are considered. Let L_1 be the maximum capacity of the T$_list$.
- **Step 2:** Because, in this case, pairs of demands with large CPLs cannot take advantage of the resources added by pairs of demands with small CPLs carried later, we try a last attempt to groom pairs of demands without fixing the value of the CPL and starting from its largest value. Let L_2 be the maximum capacity of the T$_list$ for this last attempt.

A flowchart detailing the grooming algorithm is drawn in Figure 3. The section of the flowchart representing step 1 is enclosed by dashed line. One can choose to repeat step 1 several times before going on to step 2. Let N_1 be the number of times that step 1 will be repeated.

3) Other Greedy Algorithms: In this section, we will introduce two basic greedy algorithms. This will allow us to evaluate the performance of our proposed greedy algorithms in terms of computation time as well as in terms of network's cost obtained at the last iteration.

Theoretically, by grooming two demands, only the number of o_1-ports is reduced. An intuitive idea is to always try to groom the pair of demands with the largest CPL. Thus the reduction in the number of o_1-ports can be more important. The Greedy1 algorithm analyses all the pairs of demands and marks them as SGPs or UGPs. Within the set of pairs of demands marked as SGPs, we choose the pair with the largest CPL. As a result of grooming this pair of demands, the set of all the SEDs must be modified by removing the original demands and by adding the aggregated demand and the set of marginal demands.

Due to the fact that at each iteration we analyse all the pairs of demands, the Greedy1 algorithm is too time consuming. An alternative solution is to groom any pair of demands marked as SGP regardless of the value of their CPL. At each iteration, the Greedy2 algorithm searches the set of demands in a random order and tries to groom any pair of demands with a positive CPL. Then the set of all the SEDs must be modified accordingly.

B. Sequential Algorithm

Using the auxiliary graph model of an EXC/OXC node (Section III-B), the network can be represented by an equivalent graph. The routing and grooming of SEDs is based on a shortest path algorithm such as Dijkstra Algorithm applied to this network graph. The network operator may decide of the shortest path criteria by suitably choosing the link cost/weight parameters. The cost of a link depends on the current network state as well as on the characteristic of the request itself.

1) Weight assignment: SED routing assumes the assignment of an infinite weight to ***c*** and ***f*** edges while the remaining edges are assigned a finite weight unless they have reached their maximum capacity. The finite weight of an ***a***, ***b***, ***d***, ***e***, ***g*** or ***h*** edge is set to zero while the finite weight assigned to an ***o*** or ***i*** edge depends on the required rate of the request to be routed and on its life duration. Let us consider a new SED request characterized by its rate x and its active period $[t_a, t_b]$. The finite weight assigned to an ***o*** edge is the ratio of its maximum capacity C to its free capacity F augmented by $(1 - x)$. This additional cost represents the fact that the SED does not use the whole capacity of the optical channel.

The weight assigned to an ***i*** edge representing a new GL is equal to the sum of the weights of the different edges forming this GL. However, the weight assigned to an existing GL composed of k WDM channels (k hops) is computed as follows:

- Let C_t be the maximum capacity of the considered GL and y_t its carried data rate evaluated at instant t. C_t and y_t are functions of the already routed SEDs.
- The difference $F_t = C_t - y_t$ represents the free capacity of the GL at time instant t.

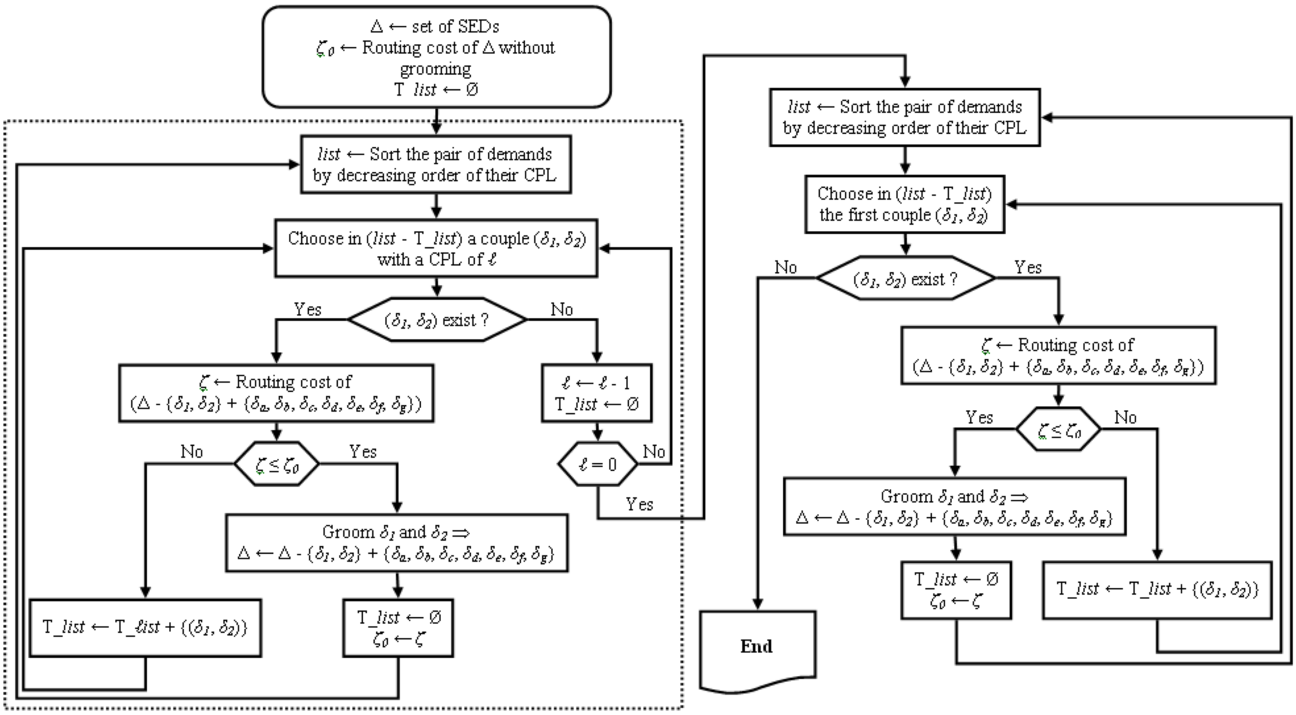

Fig. 3. Flowchart of the grooming algorithm

- It may happen that the considered GL can carry the new SED during a period $T_H \subset [t_a, t_b]$ ($x \leqslant F_t \; \forall t \in T_H$ and $x > F_t$ elsewhere). Let F be the minimum value of F_t for any t during T_H. The weight ω_H for holding the new request during T_H is equal to $(F - x) \times k + k$.
- If sufficient resources exist in the network, the duration and/or the free capacity of the considered GL can be extended by reserving additional resources in order to carry the new SED during its whole life period $[t_a, t_b]$. The weight ω_E for extending the considered GL is equal to the sum of the weights assigned to the various edges of the GL. These weights are the same as the weights described above for creating a new GL but are evaluated during $T_E = [t_a, t_b] - T_H$.
- Let $\overline{T_t} = t_b - t_a$ and let $\overline{T_H}$ be the length of the time interval T_H. The weight ω assigned to the edge ***i*** representing the considered GL is equal to:

$$\omega = \frac{\overline{T_H}}{\overline{T_t}} \times \omega_H + \frac{\overline{T_t} - \overline{T_H}}{\overline{T_t}} \times \omega_E$$

For instance, let us consider the GL given by Figure 4 and let us consider a new SED to be routed given by : $x = 0.25$, $t_0 < t_a < t_1$ and $t_b = t_6$. The considered GL can carry the new SED during $T_H = [t_a, t_1] \bigcup [t_2, t_5]$. The free capacity F is equal to 0.35 during T_H, thus $\omega_H = 1.1 \times k$. The cost ω_E of extending the considered GL during $T_E = [t_1, t_2] \bigcup [t_5, t_6]$ is equal to the sum of the weights assigned to the various edges of the GL. For an edge ***o*** of the GL given by Figure 4, its maximum capacity C is equal to 4 and its free capacity F during T_E is equal to 1. Thus, the weight assigned to this ***o*** edge is equal to $4/1 + (1 - 0.25) = 4.75$.

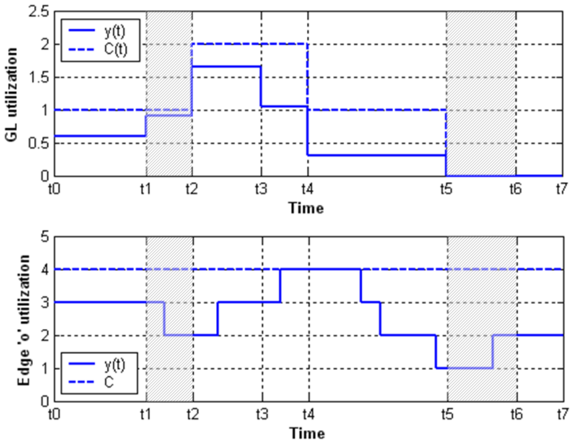

Fig. 4. An illustration of auxiliary graph edges' weight updating in the case of SED

In order to implement the ratio κ of the cost of an electrical port to the cost of an optical port in the routing algorithm, the weight assigned to ***b*** and ***e*** edges is fixed to a positive value.

2) Proposed Sequential Algorithm: The Dijkstra algorithm is initially designed to route dynamic traffic requests in a blocking scenario where we try to achieve some objectives depending on the adopted policy. This algorithm can be slightly modified in order to be used as a tool for network dimensioning. In this latter case, the objective of the algorithm is to minimize the congestion observed in the network. Given a set of requests, the modified algorithm fixes a value to the congestion and tries to route as much requests as possible. If some requests cannot be routed, the algorithm increases the value of the congestion and tries to route additional requests. When all the requests are routed, we evaluate the global network cost expressed as the number of optical and electrical ports.

VI. NUMERICAL RESULTS

In this section, we experimentally evaluate the performance of the grooming algorithms described in Section V. For this purpose, we have considered the 29 node and 44 link NSF backbone network shown in Figure 5. A set of $N_{SED} = 5000$ SED requests is generated as stated in Section III-A. The average traffic flow entering the network at each instant is equal to $\phi_{SED} = 1.52\ Tbps$ while the peak of such traffic flow is equal to $\pi_{SED} = 2.18\ Tbps$.

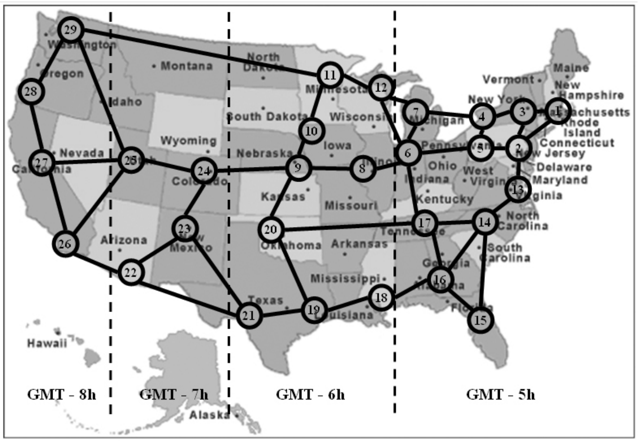

Fig. 5. 29 node and 44 link NSF backbone network

It is to be noted that all the simulations have been carried out on a 3 GHz PC with 512 MB of RAM.

A. Iterative Greedy Algorithm

1) Comparison between our iterative greedy algorithm with other greedy algorithms: First, we assign to each request the shortest path between its source node and its destination node. When no grooming is performed, a network able to handle the set of 5000 SEDs is composed of 19811 optical ports and 8106 electrical ports. Given that $\kappa = 5$, the overall cost of the network is equal to 60341. The congestion, defined as the number of wavelengths used on the most loaded link, is equal to 246.

Without changing the path assigned to the requests and by only applying our iterative greedy algorithm, the number of optical ports required to handle this set of SEDs is reduced to 14559 ports while the number of electrical ports is reduced to 7308 ports. This result is achieved when the parameters L_1, L_2 and N_1 of our proposed iterative greedy algorithm are fixed to 100, 1000 and 1 respectively. As a result, the overall cost of the network has decreased to 51099 which represents a gain of 15.31%. The congestion has decreased also to 152. The processing time to obtain this result is roughly one hour.

Similarly, we have optimized the grooming of the 5000 SEDs according to the Greedy1 and the Greedy2 algorithms. For both algorithms, we have limited the computation time to two hours. The least expensive network obtained by means of the Greedy1 algorithm is composed of 18152 optical ports and of 7927 electrical ports. Subsequently, the overall cost of the network is equal to 57787. Compared with the case where no grooming is performed, this solution represents a gain of 4.23%. In this case, the congestion was evaluated to 199. For the Greedy2 algorithm, the least expensive network is composed of 18704 optical ports and of 8027 electrical ports. Subsequently, the overall cost of the network is equal to 58839. Compared with the case where no grooming is performed, this solution represents a gain of 2.49%. In this case, the congestion was evaluated to 231. Figure 6 illustrates the overall network cost versus the computing time for the three algorithms.

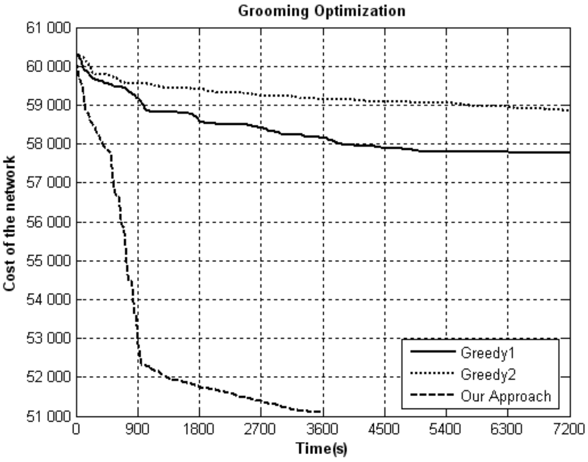

Fig. 6. Evolution of the network's cost versus computation time for different grooming algorithms

Note that the number of required optical o_1-ports reaches a minimum when each SED request passes through the EXC at each intermediate node that it traverses. When routing according to the shortest path algorithm, this minimum is equal to 7238 o_1-ports. Moreover, the number of required electrical ports reaches a minimum when the SEDs are groomed together without any need to additional ports at intermediate nodes. As a result, one can compute a lower bound on the overall network cost. This lower bound depends only on the routes assigned to each request. For the shortest path routing algorithm, the lower bound on the network cost is given by 9393 optical ports and 6118 electrical ports. Subsequently, this lower bound is equal to 39983. It is to be noted that this lower bound is rarely achievable because it is almost impossible to groom all the SEDs without the need to additional ports at the intermediate nodes. Indeed, when the grooming is performed at each intermediate node, the number of required optical ports is equal to 14656 ports while the number of required electrical ports is equal to 11381. The cost of such a network is equal to 71561 which is higher than the estimated lower bound.

2) Impact of the chosen routing solution on the grooming optimization: In the previous section we have shown that when no grooming is performed, the overall cost of the network is equal to 60341. If all the SEDs pass through the EXC at each intermediate node they traverse, the overall cost of the network is found equal to 71561. By grooming the SEDs according to the solution obtained by means of our iterative greedy algorithm, the overall cost of the network is equal to 51099. Note that all the above results are obtained when the requests are routed along the shortest path between their source and their destination nodes.

In this section, we evaluate the impact of different routing solutions on the network cost. For this task, we have generated randomly 50 different routing solutions. For each routing solution, we have applied our iterative greedy algorithm with the parameters L_1, L_2 and N_1 being fixed to 100, 1000 and 1 respectively. In average over these 50 routing solutions, a network able to handle the set of SEDs is composed of 14520 optical ports and 7323 electrical ports. As a result, the overall cost of the network is equal to 51135 which represents a gain of 15.25%. The cheapest network observed during our simulations is composed of 14123 optical ports and of 7294 electrical ports which represent a global cost of 50593. Table I shows the resource required for the different routing solutions of the set of 5000 SEDs.

TABLE I

DIFFERENT SED ROUTING SOLUTIONS

SEDs' Routing and Grooming Cost								
Shortest Path without Grooming	o_1 ports	15758	e_1 ports	2022	r_1 ports	2031	Optical ports	19811
	o_2 ports	-	e_2 ports	-	r_2 ports	-	Electrical ports	8106
	o_3 ports	4053	e_3 ports	2022	r_3 ports	2031	Congestion	246
Shortest Path with Grooming	o_1 ports	11304	e_1 ports	2022	r_1 ports	2031	Optical ports	14559
	o_2 ports	-	e_2 ports	-	r_2 ports	-	Electrical ports	7308
	o_3 ports	3255	e_3 ports	1626	r_3 ports	1629	Congestion	152
Average Cost with Grooming	o_1 ports	11250	e_1 ports	2022	r_1 ports	2031	Optical ports	14520
	o_2 ports	-	e_2 ports	-	r_2 ports	-	Electrical ports	7323
	o_3 ports	3270	e_3 ports	1632	r_3 ports	1638	Congestion	154
Best Cost with Grooming	o_1 ports	10882	e_1 ports	2022	r_1 ports	2031	Optical ports	14123
	o_2 ports	-	e_2 ports	-	r_2 ports	-	Electrical ports	7294
	o_3 ports	3241	e_3 ports	1621	r_3 ports	1620	Congestion	152

In average over the 50 routing solutions, the overall network cost varies from 60341 at the beginning of the simulation to about 51135 after one hour and a half of computation time. Figure 7 plots the evolution of this network cost versus the computation time. The continuous curve refers to the average evolution of the network cost observed over the 50 SED routing solutions. The vertical lines correspond to the mean square variation of the network cost over the 50 SED routing solutions.

3) Impact of the parameters L_1 and L_2 on the grooming optimization: In this section, we evaluate the impact of the size of the T_$lists$ on the performance of our iterative greedy algorithm. Table II shows the number of required optical and electrical ports for different values of the parameters L_1 and L_2 at different time instance. Note that those results are obtained when the SEDs are routed according to the shortest path algorithm. Figure 8 plots the evolution of the overall network cost versus the computation time.

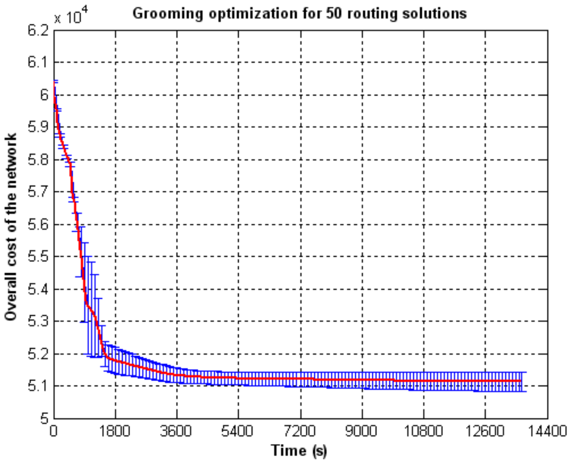

Fig. 7. Evolution of the overall network cost versus computation time (set of 50 SED routing solutions)

TABLE II

IMPACT OF THE PARAMETERS L_1 AND L_2

		Step 1: after 500 s	**End of Step 1**	**Step 2 : $L_2 = L_1$**	**Step 2 : L_2 = 1000**
L_1 = 10	Elapsed time	500	780	823	10508
	Optical ports	17715	16143	16051	14410
	Electrical ports	7836	7436	7414	7263
	Congestion	204	188	187	152
L_1 = 50	Elapsed time	500	1384	1426	5642
	Optical ports	18462	15433	15407	14573
	Electrical ports	7955	7386	7388	7310
	Congestion	208	169	169	152
L_1 = 100	Elapsed time	500	977	1009	3583
	Optical ports	18152	15413	15401	14559
	Electrical ports	7927	7378	7380	7308
	Congestion	199	169	169	152
L_1 = 250	Elapsed time	500	2826	2997	3274
	Optical ports	18407	14547	14511	14482
	Electrical ports	7950	7312	7312	7313
	Congestion	208	154	152	151
L_1 = 500	Elapsed time	500	5366	5689	6952
	Optical ports	18471	13509	13499	13476
	Electrical ports	7954	7254	7254	7253
	Congestion	208	146	146	146
L_1 = 1000	Elapsed time	500	7302	8325	8325
	Optical ports	18454	13223	13186	13186
	Electrical ports	7955	7242	7243	7243
	Congestion	208	143	143	143

It is to be noted that, at the beginning of the simulation, the slope of the network cost evolution is steeper for small value of L_1 than for large values. Thus we can reach better network cost in shorter time. However, the network cost obtained at the end of the greedy algorithm is smaller for larger value of L_1 but it requires extensive time computation. Finally, the impact of the L_2 parameter becomes negligible for large values of the L_1 parameter. To conclude, L_1 = 100 and L_2 = 1000 seems a good trade-off between the size of the T_$lists$, the computation time and the cost obtained at the end of the grooming optimization.

4) Impact of the parameter N_1 on the grooming optimization: We have stated that the step 1 of our proposed greedy algorithm can be repeated several times. In this section, we evaluate the impact of the number of times this step is repeated N_1 on the performance of the grooming algorithm. Table III shows the overall cost of the network for different values of the parameter N_1 as well as the time needed to reach this solution. Figure 9 plots the evolution of the overall network cost versus the computation time.

From these results, we can conclude that the performance of our algorithm increases as the number of iteration increases.

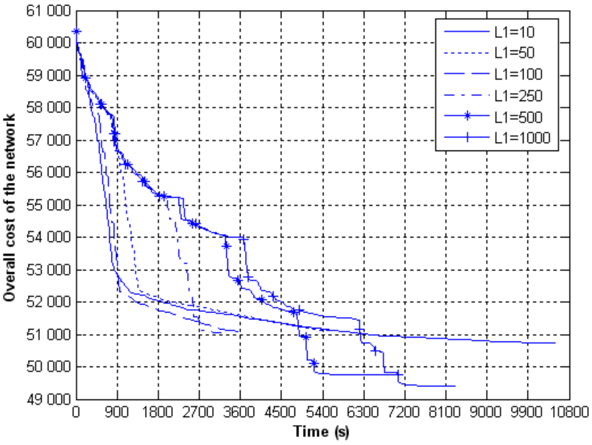

Fig. 8. Evolution of the overall network cost versus computation time for different values of L_1

TABLE III

IMPACT OF THE PARAMETER N_1

		$L_1 = 50$; $L_2 = 250$		$L_1 = 100$; $L_2 = 250$		$L_1 = 250$; $L_2 = 250$	
		Step 1	Step 2	Step 1	Step 2	Step 1	Step 2
$N_1 = 1$	Network Cost	52363	51147	52303	51113	51107	51071
	Elapsed Time	1385	5349	1547	5405	2860	3033
$N_1 = 2$	Network Cost	49993	49695	49629	49599	49159	49153
	Elapsed Time	2518	3574	2965	3159	4689	4808
$N_1 = 3$	Network Cost	49627	49591	48917	48899	48411	48399
	Elapsed Time	2776	3052	3745	3922	6104	6306
$N_1 = 4$	Network Cost	49285	49283	48559	48557	47951	47951
	Elapsed Time	2986	3090	4303	4420	7017	7138
$N_1 = 5$	Network Cost	48807	48803	48101	48099	47607	47607
	Elapsed Time	3563	3668	5134	5236	8048	8125

However, $N_1 = 3$ seems a good trade-off between the computation time and the cost obtained at the end of the grooming optimization.

B. *Sequential Algorithm*

In this section, we have implemented the equivalent auxiliary graph representation of a node. Using the Dijkstra algorithm with the proposed dynamic cost assignment functions, a network able to handle the set of 5000 SEDs is composed of 14863 optical ports and 10992 electrical ports. As a result, the overall cost of the network is equal to 69823 and the congestion is equal to 71. This result is achieved when the finite cost assigned to ***b*** and ***e*** edges is null. The overall network cost decreases to 57999 when to finite cost assigned to ***b*** and ***e*** edges is equal to 1 and decreases more to 54457 when this finite cost is fixed to 2. Note that the time needed to obtain these results is less than 100 seconds.

VII. CONCLUSION

In this paper, we have introduced two routing and grooming algorithms which are able to route a large set of SEDs at optimal cost. Our cost function corresponds to the number of required optical and electrical ports to transport all the traffic demands. Our first algorithm is based on a meta-heuristic approach. It deals with the whole set of demands at once and tries to find the least expensive grooming solution. Meanwhile, our second algorithm is a sequential approach that solves the traffic grooming problem for one connection request at a time.

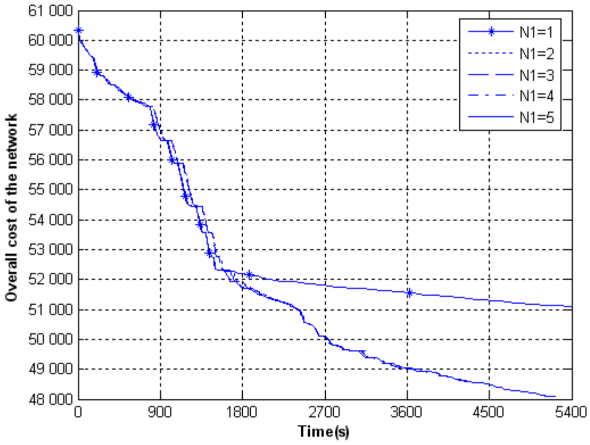

Fig. 9. Evolution of the overall network cost versus computation time for $L_1 = 100$, $L_2 = 250$ and different values of N_1

The meta-heuristic approach is an iterative process based on a greedy algorithm. At each iteration, we try to groom a pair of requests satisfying a given criteria. We have shown in Section VI-A.1 that for specific scenarios, our proposed iterative greedy algorithm enables to obtain smaller network costs than those obtained with basic greedy strategies proposed in the literature in a considerably shorter computation time. Smaller network costs can be obtained by suitably choosing the parameters of the iterative greedy algorithm as shown in Sections VI-A.3 and VI-A.4.

The sequential approach is built upon Dijkstra algorithm combined with dynamic cost assignment functions. By suitably choosing the cost assigned to some specific edges, a network with an acceptable higher overall cost can be obtained in a very short period of time.

REFERENCES

[1] Jian Wang, Wonhong Cho, V. Rao Vemuri and Biswanath Mukherjee, “Improved approaches for cost-effective traffic grooming in WDM ring networks: ILP formulations and single-hop and multihop connections,” *Journal of Lightwave Technology*, vol. 19, no. 11, pp. 1645–1653, November 2001.

[2] M. Brunato and R. Battiti, “A multistart randomized greedy algorithm for traffic grooming on mesh logical topologies,” *In Proceedings of the* 6^{th} *IFIP Conference on Optical Network Design and Modeling (ONDM), Torino, Italy,*, February 2002.

[3] Hongyue Zhu, Hui Zang, Keyao Zhu and Biswanath Mukherjee, “A novel generic graph model for traffic grooming in heterogeneous WDM mesh networks,” *IEEE/ACM Transactions on Networking (TON)*, vol. 11, no. 2, pp. 285–299, April 2003.

[4] E.A. Doumith, M. Gagnaire, O. Audouin and R. Douville, “From Network Planning to Traffic Engineering for Optical VPN and Multi-Granular Random Demands,” 25^{th} *IEEE International Performance Computing and Communications Conference*, April 2006.

[5] E.A. Doumith, M. Gagnaire, O. Audouin and N. Puech, “Traffic engineering for virtual private networks and random traffic demands in WDM optical core networks,” 14^{th} *International Conference on Computer Communications and Networks*, pp. 317–322, October 2005.

[6] E. Salvadori, R. Lo Cigno and Z. Zsóka, “Dynamic Grooming in IP over WDM Networks: A study with Realistic Traffic based on GLANCES Simulation Package,” *In Proceedings of the* 9^{th} *IFIP/IEEE Conference on Optical Network Design and Modeling (ONDM)*, February 2005.