Design of Fault-Tolerant Logical Topologies in Wavelength-Routed Optical IP Networks

A. Nucci¹, B. Sansò², T.G. Crainic², E. Leonardi³, M. Ajmone Marsan³
¹ – Dipartimento di Elettronica - Politecnico di Torino
² – Department of Electrical and Software Engineering - École Polytechnique de Montréal
³ – Centre de Recherche sur les Transports (CRT) - École Polytechnique de Montréal
email: nucci@mail.tlc.polito.it, bruni@ crt.umontreal.ca, theo@crt.umontreal.ca, leonardi@mail.tlc.polito.it, ajmone@mail.tlc.polito.it

Abstract — In this paper we illustrate a new methodology for the design of fault-tolerant logical topologies in wavelength-routed optical networks exploiting wavelength division multiplexing, and supporting IP datagram flows. Our design approach generalizes the “design protection” concepts, and relies on the dynamic capabilities of IP to re-route datagrams when faults occur, thus achieving protection and restoration, and leading to high-performance cost-effective fault-tolerant logical topologies. In this paper for the first time we consider the resilience properties of the topology during the logical topology optimization process, thus extending the optimization of the network resilience also to the space of logical topologies. Numerical results clearly show that our approach outperforms previous ones, being able to obtain very effective survivable logical topologies with limited complexity.

Keywords — WDM; IP; Logical Topology Design; Survivability; Fault-Tolerance; Resilience; Protection; Restoration

I. INTRODUCTION

For the short-term implementation of a high-capacity IP infrastructure, network designers are considering the use of optical networks exploiting Wavelength Division Multiplexing (WDM) and Wavelength Routing (WR). Indeed, such networks permit the exploitation of the huge fiber capacity, with no need for complex processing functionalities in the optical domain. In WR IP networks, nodes comprise an optical section, and an electronic section; the former is an optical cross-connect (OXC), while the latter is a high-capacity IP router. Nodes are connected by optical fibers over which a WDM scheme is implemented. At each node, incoming WDM channels can either be transparently connected to outgoing channels through the OXC, possibly after wavelength conversion (but with no processing of in-transit information), or converted to the electronic domain, so that packets can be passed to the IP router, processed, and possibly retransmitted after IP routing. This setup allows the definition within the optical domain of semi-permanent optical pipes called “lightpaths” that may extend over several physical links. Thus, lightpaths can be seen as chains of physical channels through which packets are moved from a router to another toward their destinations. OXCs transparently connect the incoming WDM channels corresponding to in-transit lightpaths, and convert to the electronic domain the incoming WDM channels corresponding to terminating lightpaths. The set of lightpaths and routers defines a logical topology, overlayed to the physical topology made of optical fibers and OXCs.

In order to best exploit the capacity of a WDM infrastructure, a crucial task is the identification of the best feasible logical topology for the transport of a given traffic pattern.

In recent years, the Logical Topology Design (LTD) problem in WDM networks was extensively studied, considering a number of different setups. It was shown that the identification of the optimal logical topology is computationally intractable for large size networks [1], [2], and several heuristic approaches were proposed for the identification of suboptimal solutions in several different conditions [3], [4], [5].

The search for good solutions of the LTD problem can consider different objectives; in this paper we concentrate on an important characteristic of the logical topology that is produced by the LTD algorithm: its resistance to physical faults. Note that, since several lightpaths may traverse the same fiber on different wavelengths, the fault of a single physical link may cause the disruption of a number of lightpaths.

Many physical level protection and/or restoration schemes were proposed for WR networks (see [6], [7] for a survey). Such schemes always rely on the presence of spare resources in the network, that are used to restore disrupted lightpaths. In general, two disjoint physical paths $p_1$ and $p_2$ are associated with each existing lightpath $l$. In normal conditions, only path $p_1$ is used to support lightpath
l; however, enough capacity is reserved on \( p_2 \) to be able to re-route lightpath \( l \) when one of the physical links belonging to \( p_1 \) fails. Protection and/or restoration schemes normally guarantee the resilience to the failure of a single physical link; thus, the capacity to be reserved in the network must be sufficient to successfully re-route all the lightpaths disrupted by the failure of one link.

The main advantage of physical protection and/or restoration schemes, is that they entail no complex management procedure, and thus guarantee fast restorations to link failures. Their main drawback is that a large amount of network resources, in terms of both processing and bandwidth is devoted to restoration. As a consequence, physical protection and/or restoration schemes, in many situations, are not cost-effective.

In [8], [9], [10], a different fault-tolerance methodology called design protection was presented. In this case, restoration is obtained by exploiting the dynamic capabilities of IP routing. When a physical link fails in the optical network, the IP routing algorithm is able to update its tables, and restore disrupted paths, if the set of non-disrupted lightpaths still forms a connected topology. In order to achieve a good degree of fault resilience, it is fundamental to map (i.e., to route) the lightpaths onto the physical topology in such a way that, given any single physical link failure, the set of non-disrupted lightpaths still forms a connected network. Thus, an optimization of the physical mapping of lightpaths is necessary [9], [10]. In this paper we generalize the design protection approach, providing a powerful framework for the design of logical topologies with a good degree of fault resilience. Our approach differs from the approach proposed in [8], [9], [10], since it considers the resilience properties of the topology directly during the LTD optimization process, thus extending the optimization of the network resilience properties also to the space of the logical topologies. In addition, we further extend the design protection approach, also considering multicast traffic.

First, we formalize our approach aimed at the research of logical topologies with good degrees of fault resilience, providing an integer linear programming formulation which, however, is computationally intractable even for networks of moderate size; then we propose a methodology based on the application of the Tabu Search metaheuristic that produces suboptimal solutions, while limiting the computational effort.

Note that, according to the design protection approach, restoration requires the execution of signalling and management procedures, and thus entails significant delays; however, design protection is much more cost-effective with respect to physical protection schemes. Thus, design protection should not be necessarily considered as an alternative to physical protection, but as a complementary technique that may be successfully employed in order to reduce the level of physical protection required to achieve a desired level of fault-tolerance.

II. PROBLEM STATEMENT

The Fault-tolerant Logical Topology Design Problem (FLTDP) under a given unicast and multicast traffic pattern can be stated as follows:

**GIVEN:**

i) an existing physical topology (which must be at least 2-connected), comprising nodes equipped with a limited integer number of tunable transmitters and receivers, connected by optical fibers that support a limited number of wavelengths;

ii) a description of the average traffic exchanged by sources and sets of destinations;

iii) a multi-hop routing strategy, defined both for unicast and multicast flows;

**FIND**

a logical topology and a “mapping” that optimize (i.e. maximize or minimize) an objective function.

The objective function must be carefully selected, in order to obtain the best trade-off between network performance in normal conditions and fault resilience properties. Since the network performance depends on the network failure state, the objective function must combine the network performance levels under different network failure states.

Finally, for what concerns the multi-hop routing strategy adopted on the logical topology, we selected shortest path routing for unicast traffic, and the SCTF algorithm proposed in [11] for multicast traffic, and reported below:

**SCTF algorithm**

Let \( o \) be the source of multicast traffic, \( O \) be a temporary set of nodes used for the building of the steiner tree, and \( D \) the set of destinations to be reached. Let \( T \) be a dynamic structure in which the Steiner Tree is gathered:

**Step 0:** set \( O = \{ o \} \); \( D = D \); \( T = NULL \);

**Step 1:** select the node \( d \in D \) closest to nodes in \( O \);

**Step 2:** select \( d' \), the node in \( O \) closest to \( d \);

**Step 3:** extract \( d \) from \( D \);

**Step 4:** select shortest path route \( r \) from \( d' \) to \( d \);

**Step 5:** add the link of \( r \) to \( T \);

**Step 6:** insert all the nodes crossed by \( r \) in \( O \);
Step 7: if \( \mathcal{D} \neq \emptyset \) GOTO Step 1, else STOP.

The algorithm requires at each step the evaluation of all the shortest paths from nodes in \( O \) to nodes in \( D \). To reduce the algorithm complexity, it is possible to associate a weight \( w \) with each node in \( O \), and consider only the \( k \) nodes with highest weight when selecting the closest destination \( d \) to \( O \).

A. Problem Formulation

In this section we report an ILP formulation of the FLTD problem. In order to limit the complexity of the notation, we restrict our formulation to networks supporting unicast traffic only. However, an ILP formulation for networks supporting multicast as well as unicast traffic can be developed in a very similar way.

A.1 Notation

We adopt the notational typology for multi-layered networks presented in [12]. The supra-index indicates the layer, starting by the lowest layer, zero, that represents the physical network. Let \( G^0 = (V, E^0) \) be the unidirectional graph representing the physical topology. It is composed by OXC nodes \( V \) interconnected by optical fibers \( E^0 \). Let \( |V| = N \) be the cardinality of set \( V \) and \( |E^0| = M \) that of set \( E^0 \). Let \( R_t \) and \( T_t \) be the numbers of receivers and transmitters at physical node \( i \in V \). Let \( S_k \) be the network state, where \( S_0 \) represents the no-failure state, while \( S_v \) for \( v \geq 1 \) is the state in which the optical fiber \( v \in E^0 \) is broken. Let \( \mathcal{S} \) be the set of all operational states, whose cardinality is \( |\mathcal{S}| = M + 1 \). Let \( \mathcal{E} \) be the set of all possible lightpaths in any logical topology. Let \( G^1(S_0) = (V, E^1(S_0)) \) be the directed graph representing the logical topology in the no-failure state. It is composed of IP routers \( V \) interconnected by lightpaths \( E^1(S_0) \subseteq \mathcal{E} \). Let \( G^1(S_v) = (V, E^1(S_v)) \) denote the logical topology in the network state \( S_v \), obtained from \( G^1(S_0) \) by dropping all the lightpaths \( u \in E^1(S_0) \) crossing over the optical fiber \( v \in E^0 \). Let \( \Lambda = (\lambda^{sd}) \) indicate the average traffic matrix where each entry \( \lambda^{sd} \) represents the average traffic flow between source \( s \) and destination \( d \).

A.2 Decision Variables

Three types of binary variables are introduced into the formulation: \( X_u \), \( t^{sd}_u(S_k) \), \( Y_{uv} \), that correspond, respectively, to logical topology, routing, and mapping.

The logical topology variables \( X_u \in \{0, 1\} \) describe the lightpaths comprised in the logical topology \( G^1(S_0) \):

\[
X_u = \begin{cases} 
1, & \text{if lightpath } u \in \mathcal{E} \text{ belongs to the logical topology } G^1(S_0) \\
0, & \text{otherwise} 
\end{cases}
\]

Then we can state that logical topology \( G^1(S_0) = (V, E^1(S_0)) \) comprises the lightpaths \( E^1(S_0) = \{u : X_u = 1, \ u \in \mathcal{E}\} \).

The routing variables \( t^{sd}_u(S_v) \in \{0, 1\} \) contain the routing information for the logical topology \( G^1(S_v) \):

\[
t^{sd}_u(S_v) = \begin{cases} 
1, & \text{if traffic } s \rightarrow d \text{ crosses lightpath } u \in \mathcal{E} \text{ in state } S_v \\
0, & \text{otherwise} 
\end{cases}
\]

The mapping variables \( Y_{uv} \in \{0, 1\} \), finally, contain the routing information of lightpaths belonging to the logical topology \( G^1(S_0) \) over the physical topology \( G^0 \):

\[
Y_{uv} = \begin{cases} 
1, & \text{if lightpath } u \in \mathcal{E} \text{ crosses the optical fiber } v \in E^0 \\
0, & \text{otherwise} 
\end{cases}
\]

A.3 Mathematical Model

Let \( \Gamma^+(i) \) be the set of lightpaths outgoing from node \( i \in V \) and \( \Theta^-(i) \) be the set of lightpaths incoming to node \( i \in V \). Let \( \Theta^+(i) \) be the set of physical links outgoing from node \( i \in V \) and \( \Theta^-(i) \) be the set of physical links incoming to node \( i \in V \). Let, finally, \( O(u) \) be the origin node and \( D(u) \) the destination node of lightpath \( u \in \mathcal{E} \). We can then write the model constraints.

A.4 Constraints

- Connectivity:

\[
\sum_{u \in \Gamma^+(i)} X_u \leq T_i \quad \forall i \in V \quad (1)
\]

\[
\sum_{u \in \Theta^-(i)} X_u \leq R_i \quad \forall i \in V \quad (2)
\]

where equations (1) indicate that the number of lightpaths outgoing from each node cannot be larger than the number of transmitters in the node, for each logical topology in the no failure state \( (G^1(S_0)) \); equations (2) indicate that the number of lightpaths incoming to each node cannot be larger than the number of receivers in the node, for each logical topology in the no failure state \( (G^1(S_0)) \);

- Routing:

\[
\sum_{u \in \Gamma^+(i)} t^{sd}_u(S_v) - \sum_{u \in \Theta^-(i)} t^{sd}_u(S_v) = \begin{cases} 
1, & \text{if } s = i \\
-1, & \text{if } d = i \\
0, & \text{otherwise} 
\end{cases} \forall s, d, i \in V, \forall S_v \in \mathcal{S} \quad (3)
\]

\[
t^{sd}_u(S_v) \leq X_u \quad \forall s, d \in V, \forall S_v \in \mathcal{S}, \forall u \in \mathcal{E} \quad (4)
\]
where equations (3) represent the routing continuity constraints for packet routes on the logical topology $G^1(S_v)$. They state that for each network operational state, an available (working) path on the logical topology must exist for each source-destination pair; equations (4), instead, state that traffic can be routed only on lightpaths belonging to the logical topology.

- Mapping:

$$Y_{uv} \leq X_u \quad \forall u \in \mathcal{E}, v \in E^0$$  \hspace{1cm} (5)

$$\sum_{v \in \Theta^+(i)} Y_{uv} - \sum_{v \in \Theta^-(i)} Y_{uv} = \begin{cases} 
1 & \text{if } O(u) = i \\
-1 & \text{if } D(u) = i \\
0 & \text{otherwise}
\end{cases} \quad \forall u \in \mathcal{E}, \forall i \in V$$  \hspace{1cm} (6)

$$\sum_{s,d \in V} t_{sd}^d(S_v) \leq \left( \sum_{s,d \in V} \lambda_{sd} \right) (1 - Y_{uv}) \quad \forall u \in \mathcal{E}, \forall v \in E^0, \forall s, d \in V$$  \hspace{1cm} (7)

where equations (5) ensure that only the lightpaths in the considered logical topology are mapped; equations (6) represent routing continuity constraints for lightpaths on the physical topology $G^0$; equations (7) impose that all the lightpaths that cross the physical link $v$ are not available in state $S_v$.

In order to extend the formulation to the case in which a limited number of wavelengths is available, an additional constraint must be introduced. Let $W_v$ be the number of wavelengths supported on each fiber. The set of lightpaths $v \in E^1(S_0)$ must satisfy the following constraint:

- Mapping

$$\sum_{u \in \mathcal{E}} Y_{uv} \leq W_v \quad \forall v \in E^0$$  \hspace{1cm} (8)

which indicates that the number of lightpaths that cross over each optical fiber has to be smaller than the wavelength number.

A.5 Objective Function

As objective function we selected the minimization of the maximum congestion level over all failure states:

$$\min_H$$

with:

$$H \geq \left[ \sum_{s,d \in V} t_{sd}^d(S_v) \lambda_{sd} \right] \quad \forall S_v \in \mathcal{S}, \forall u \in \mathcal{E}$$

A.6 Observations

Note, first, that in the above formulation the routing of packets on the logical topology is unspecified, thus the minimization of the maximum congestion level over all failure states is jointly performed on all admissible logical topologies and routings.

Note, also, that the above ILP model provides a logical topology that can tolerate any single link failure. Indeed, the obtained logical topology is surely connected under any single link failure (equations 3).

Lightpath capacity constraints are ignored in the above formulation, for the sake of the model simplicity; we notice, however, that the minimization of the maximum congestion level corresponds to a minimization of the lightpath capacity needed to guarantee an efficient transport of the offered traffic. Thus, a minimization of the maximum congestion level leads to the minimization of the capacity needed to guarantee good performance.

The FLTDP is NP-Hard, since it is a generalization of the traditional LTD problem that was proved to be NP-hard. Even for moderate size networks, an optimal solution of the FLTDP problem appears to be quite problematic due to the large number of variables and constraints involved in the formulation. Thus, the development of heuristic solution methodologies is required. The approach to the FLTDP solution problem in [8] and [14] consists in decomposing the whole problem in two independent subproblems: the Logical Topology Design (LTD) problem, in which the logical topology optimization is performed on the basis of the congestion level in the full operational state ($S_0$), thus ignoring the resilience property of the solution; and the Fault-tolerant Mapping (FM) problem, according to which the mapping of the logical topology lightpaths on the physical topology is aimed at the achievement of good resilience properties.

While the LTD problem has been widely investigated in the literature, and many algorithms have been proposed ([2], [3], [4], [5]), the FM problem has been considered only recently. In [8] a heuristic approach based on the application of the Tabu Search optimization algorithm has been proposed, while in [14] an ILP formulation of the FM problem is provided, and the problem is optimally solved for moderate size networks by applying the branch and cut algorithm.

B. FLTDP Under Shortest-Path Routing

In order to restrict the optimization to act only on the set of the admissible logical topologies with shortest path routing, we need to introduce some extra variables and constraints.
Let us introduce an extra set of variables \( \tau^u_{sd}(S_v) \in \{-1, 0, 1\} \), that represent a possible alternative routing with respect to the routing specified by \( t^u_{sd} \) on the logical topology:

\[
\tau^u_{sd}(S_v) = \begin{cases} 
1, & \text{if traffic } s \to d \text{ is re-routed on } \text{lighpath } u \in \mathcal{E} \text{ in state } S_v \\
-1, & \text{if traffic } s \to d \text{ is no longer routed on } \text{lighpath } u \in \mathcal{E} \text{ in state } S_v \\
0, & \text{otherwise}
\end{cases}
\]

\( \tau^u_{sd}(S_v) \) must satisfy the following constraints:

\[
\sum_{u \in \Gamma^+(i)} \tau^u_{sd}(S_v) - \sum_{u \in \Gamma^-(i)} \tau^u_{sd}(S_v) = 0 \tag{9}
\]

\[
\tau^u_{sd}(S_v) \leq X_u - Y_{sv} - t^u_{sd}(S_v) \quad \forall s, d, i \in V, \forall S_v \in S \tag{10}
\]

\[
\tau^u_{sd}(S_v) \leq X_u - t^u_{sd}(S_v) \quad \forall s, d, i \in V, \forall u \in \mathcal{E} \forall v \in \mathcal{B} \tag{11}
\]

\[
-\tau^u_{sd}(S_v) \leq t^u_{sd} \quad \forall s, d, i \in V, \forall u \in \mathcal{E}, \forall S_v \in S \tag{12}
\]

where equations (9) represent the routing continuity constraints for the re-routing on logical topology \( G^1(S_v) \). Equations (10) and (11), instead, state that traffic can be re-routed only on a path consisting of working lightpaths belonging to the logical topology; equations (12), finally, state that, after re-routing, routes defined by \( t^u_{sd} \) may be no longer valid.

Finally, we define \( G \) as:

\[
G = - \sum_{\substack{s_u \in S \\forall u \in \mathcal{E} \\forall s_i \in V}} \tau^u_{sd}(S_v)
\]

Note that \( G \) represents the difference between the total path length before re-routing and after re-routing. It is possible to find a set of \( \tau^u_{sd} \) such that \( G \) assumes negative values whenever the set of \( t^u_{sd} \) does not describe a shortest path routing. In conclusion, if and only if the set \( t^u_{sd} \) defines a shortest path routing we find \( \min G = 0 \); thus, selecting:

\[
\min [\epsilon H + \max G]
\]

as objective function, where \( \epsilon < 1/\sum_{\substack{s_u \in S \\forall u \in \mathcal{E} \\forall s_i \in V}} \lambda_{sd} \), we obtain the result of restricting the optimization to the set of logical topologies implementing a shortest path routing. Indeed, we observe that, by construction, \( 0 < \epsilon H < 1 \), while \( \max G \) can assume only integer non positive values. Thus, the minimum cost solution (i.e., the solution that minimizes \( \epsilon H + \max G \)) is by contraction the solution that minimizes \( H \) among those for which \( \max G = 0 \), i.e., it results the logical topology implementing a shortest path routing that minimizes the maximum congestion level.

Note, however, that the objective function, in this case, is non-linear, thus, since the formulation falls in the class of integer non linear programming problems, no general methodologies and tools are available for an optimal solution of the problem. For this reason, the development of on-purpose powerful heuristic (sub-optimal) approaches is needed. In this paper we propose a sub-optimal solution methodology based on the application of Tabu Search.

We will show in the results section that our approach significantly outperforms the previously proposed approaches, since the resilience properties of the topology are considerent during the logical topology optimization.

### III. Mapping Between Physical and Logical Topology: HDAP

The definition of algorithms that optimally map the lightpaths on the physical topology is an important sub-problem of FLTP. This problem is related to inequalities (5) and (6) in Section II. The mapping problem can be formalized as follows: given a logical topology, find a routing for each lightpath of the logical topology over the physical topology, such that a single optical link failure leaves the virtual topology connected.

This problem was recently studied in [8], [14]. In [8] this problem was found NP-complete, and Tabu Search was proposed to find a sub-optimal solution. An ILP formulation of the problem was, instead, provided and solved in [14] for networks of moderate size, applying the CPLEX [15] optimization tool. However, since the mapping problem is only a part of FLTP, the utilization of a computationally expensive algorithm to solve the mapping problem could have a disruptive impact on the CPU time necessary for the solution of FLTP. Thus, for the solution of the mapping problem, in this section we present a simple greedy algorithm, the Heuristic Disjoint Alternate Path (HDAP), whose computational complexity is small. A brief description of the HDAP algorithm follows.

**HDAP algorithm**

Let \( OR(i) \) and \( IR(i) \) be the sets of already routed lightpaths, respectively outgoing from and incoming to node \( i \), and let \( ON(i) \) and \( IN(i) \) be the sets of outgoing and incoming lightpaths not yet routed. Let \( V_{ij} \) be the lightpath belonging to the logical topology, with endpoints \( i \) and \( j \). Let \( \mathcal{O} \) denote a set of nodes. Initialize \( \mathcal{O} \) to the set of all nodes in the network:
Step 0: route all lightpaths $V_{ij}$ whose endpoints are adjacent in the logical topology. Insert $V_{ij}$ in $(OR(i), IR(j))$ and extract $V_{ij}$ from $(ON(i), IN(j))$;

Step 1: if $O = \emptyset$ STOP, otherwise select $i \in O$ and extract $i$ from $O$; 

Step 2: if $ON(i) = \emptyset$ GOTO Step 3, otherwise select each $V_{ik} \in ON(i)$ and try to find a route for $V_{ik}$ which is physically disjoint from the routes on which the $V_{ij} \in OR(i)$ and the $V_{jk} \in IR(k)$ have already been routed. If a physically disjoint route for $V_{ik}$ has not been found, $V_{ik}$ is routed on the shortest path. If also the shortest path is not available, due to the lack of free wavelengths, lightpath $V_{ij}$ is not mapped. 

Step 3: if $IN(i) = \emptyset$ GOTO Step 1, select each $V_{ki} \in IN(i)$ and try to find a route for $V_{ki}$ which is physically disjoint from the routes on which the $V_{ij} \in IR(i)$ and the $V_{kj} \in OR(k)$ have already been routed. If a physically disjoint route for $V_{ki}$ has not been found, $V_{ki}$ is routed on the shortest path. If also the shortest path is not available, due to the lack of free wavelengths, lightpath $V_{ji}$ is not mapped.

An example of mapping produced by HDAP is shown in Fig. 1. The lightpaths that are mapped first over the physical topology are those whose end points are two adjacent physical nodes (see for example the lightpaths 1 $\rightarrow$ 2, 2 $\rightarrow$ 3, 3 $\rightarrow$ 2 etc.). Then, starting from node 1, the steps 2 and 3 of HDAP are iteratively and sequentially applied to all nodes of the network. Focusing on node 1, HDAP maps the outgoing remaining lightpath 1 $\rightarrow$ 6 over the optical fibers (1, 3) and (3, 6). Note that all the possible routes for lightpath 1 $\rightarrow$ 6 must comprise fiber (1, 3), since lightpath 1 $\rightarrow$ 2 is already routed on fiber (1, 2). Concerning the incoming lightpaths of node 1, HDAP maps lightpath 4 $\rightarrow$ 1 over the optical fibers (4, 2) and (2, 1). It is interesting to look at the mapping for lightpath 6 $\rightarrow$ 1. All the paths outgoing from node 6 must cross optical fiber (6, 3), since (6, 5) has been already used by lightpath 6 $\rightarrow$ 5. The paths incoming to node 1 must use optical fiber (3, 1) since (2, 1) has been already used by lightpath 4 $\rightarrow$ 1. Then the only possible physical path for lightpath 6 $\rightarrow$ 1 is represented by the sequence of optical fibers (6, 3) and (3, 1). Note that if no disjoint path for 6 $\rightarrow$ 1 were possible, the algorithm would have selected one shortest path. It is worth to notice that if the maximum number of lightpaths that crosses a fiber is limited, some lightpath could be not mapped over the physical topology.

IV. Tabu Search for FLTDP: TABUFLTDP

A. General Description of Tabu Search

The heuristic we propose for the solution of FLTDP relies on the application of the Tabu Search (TS) methodology. The TS algorithm can be seen as an evolution of the classical local optimum solution search algorithm called Steepest Descent (SD); however, thanks to the TS mechanism that allows worsening solutions to be also accepted, contrary to SD, TS is not subject to local minima entrapments. TS is based on a partial exploration of the space of admissible solutions, finalized to the discovery of a good solution. The exploration starts from an initial solution that is generally obtained with a greedy algorithm, and when a stop criterion is satisfied, the algorithm returns the best visited solution. For each admissible solution, a class of neighbor solutions is defined. A neighbor solution is defined as a solution that can be obtained from the current solution by applying an appropriate transformation, and is also called a move. The set of all admissible moves uniquely defines the neighborhood of each solution.

At each iteration of the TS algorithm, all solutions in the neighborhood of the current one are evaluated, and the best is selected as the new current solution. Note that, in order to efficiently explore the solution space, the definition of neighborhood may change during the solution space exploration; in this way it is possible to achieve an intensification or a diversification of the search in different solution regions.

A special rule, the Tabu list, is introduced in order to prevent the algorithm to deterministically cycle among already visited solutions. The Tabu list stores the last accepted moves; until a move is stored in the Tabu list, it cannot be used to generate a new move. The choice of the Tabu list size is very important in the optimization procedure: too small a size could cause the cyclic repetition of the same solutions, while too large a size can severely limit the number of applicable moves, thus preventing a good exploration of the solution space.

B. Fundamental Aspects of TabuFLTDP

Four fundamental aspects that must be defined in TS concern:

- the choice of an initial solution
- the definition of the topology perturbation that generates the neighborhood
- the evaluation of the visited solutions
- the stop criterion

As initial solution we select the result of the Route and Remove (R&R) [2] heuristic, which initially considers a fully-connected optical topology, and sequentially re-
moves a set of least-loaded lightpaths from the logical topology, until the degree constraints are satisfied. To describe this algorithm, we use a bipartite graph associated with the current logical topology according to the following rules: two vertices $i_n$ and $o_n$ in the bipartite graph correspond to each node $n$ in the (logical) topology; in the bipartite graph, an edge exists between $i_s$ and $o_d$, whose weight is initialized to the traffic flow value between nodes $s$ and $d$; a boolean variable is associated with each edge, which can assume the values Removable or Unremovable.

The algorithm can be described as follows:

**Step 0:** select the fully-connected logical topology and mark all lightpaths as Removable;

**Step 1:** if all the in/out-degree constraints are satisfied GO TO Step 2, else STOP;

**Step 2:** select $o'$, the node in $O$ closest to $d$;

**Step 3:** solve both the unicast and the multicast routing problems on the current topology and compute traffic flows on lightpaths;

**Step 4:** assign to each edge of the bipartite graph a weight equal to the flow traversing the associated lightpath;

**Step 5:** find a set of edges that can be removed from the graph by solving a 1-minimal Weight Matching\(^1\) (1-$mWM$) on the bipartite graph. Only the edges that are marked as Removable can be chosen in the matching;

**Step 6:** remove all edges in the 1-minimal Weight Matching, together with the corresponding lightpath in the logical topology, only if the resulting logical topology remains connected and GOTO Step 1. If the removal of a matched lightpath would disconnect the logical topology, mark the lightpath as Unremovable.

The perturbation is defined according to the following algorithm:

- within the current solution, $2L'$ nodes, $n_1, n_2, \ldots, n_{2L'}$, are selected, such that lightpaths $n_{2i-1} \rightarrow n_{2i}$, $i = 1, 2, \ldots, L'$ are present in the logical topology, and lightpaths $n_{2i+1} \rightarrow n_{2i}$, $i = 1, 2, \ldots, L' - 1$, and $n_1 \rightarrow n_{L'}$ are absent;

- in the perturbed solution, lightpaths $n_{2i-1} \rightarrow n_{2i}$ are replaced with lightpaths $n_{2i+1} \rightarrow n_{2i}$, $i = 1, 2, \ldots, L' - 1$, and $n_1 \rightarrow n_{L'}$.

This is equivalent to identifying a cycle of $2L'$ lightpaths, $L'$ of which are present in the topology, while the other $L'$ are missing; in the identified cycle, a present lightpath is followed by a missing one, as shown in Figure 2. To obtain the neighbor topology, we have to follow the lightpaths in the reverse direction, and remove the existing lightpaths while adding the missing ones. If $L' = 2$, the resulting perturbation is equivalent to a well known “branch exchange” operation.

This perturbation guarantees that degree constraints are not violated, thus generating a valid move. Note that with this perturbation it is very easy and fast to implement a diversification and/or intensification criterion; we can carefully explore a region of the solution space with small cycles, and move to another region of the solution space with large cycles.

The evaluation of solutions is performed by routing the traffic on the topology for all the single link failure states.
mapping algorithm.

ing algorithm on the logical topology and of the HD AP cycle used for the
tiv ely the
evalution is based on
that the ev aluation of
sion found by FL TDP;
mum netw ork congestion le v el o v er all f ailure states. Note
applying procedure
C. T abuFL TDP Pseudocode
The pseudo-code for TabuFLTDP is gi v en in Fig. 3.

First, let us define some useful notation:
• $F_{eval}(T)$ is the evaluation function to compute the merit coefficient of logical topology $T$. It returns $M$, the maximum network congestion level over all failure states. Note that the evaluation of $M$ requires the execution of the routing algorithm on the logical topology and of the HDAP mapping algorithm.
• $BuildInitialSolution$ is used to build an initial logical topology applying R&R;
• $BuildCycle(T, l)$ is used to build the neighbor solution of the current solution $T$, using cycles of length $l$. When the diversification criterion has to be used, the cycle is longer than in normal TabuFLTDP. We denote with $q$ the length of the normal cycle and with $p$ the length of the cycle used for the diversification criterion, where $p \gg q$;
• $BuildNeighborhood(T)$ is a procedure to build the neighborhood of the current solution $T$, by applying iteratively the $BuildCycle(T, l)$ procedure;
• $BestNeighSol(T)$ is a procedure that evaluates each solution in the neighborhood of $T$, and returns the best solution. The evaluation is based on $F_{eval}(T)$;
• $TabuList$ is a fixed size Tabu list to store the last moves;
• $N(T)$ is the neighborhood of logical topology $T$, built applying procedure $BuildNeighborhood(T)$ and using only lightpaths not belonging to $TabuList$;
• $T, T^*$ and $T^{**}$ represent respectively the current logical topology, the best logical topology in $N(T)$, and the best solution found by FLTDP;
• $M, M^*$ and $M^{**}$ represent respectively the merit associated with the logical topologies $T, T^*$ and $T^{**}$;
• $IterationsNumber$ is the number of iterations;
• $LimitDiv$ is the number of consecutive iteration without improvements after which the diversification criterion is applied;
• $IterBest$ is the iteration at which $T^{**}$ is found.

$T := BuildInitialSolution$
$IterBest := 1$
$T^{**} := T$
$M^{**} := M$

For $counter = 1$ to $IterationsNumber$
do
  if $counter - IterBest = LimitDiv$ then
    $T := BuildCycle(T, p)$
    $N(T) := BuildNeighborhood(T)$
    $T^{*} := BestNeighSol$ and update TabuList
  if $M^{*} \leq M^{**}$ then
    $T^{**} := T^{*}$
    $M^{**} := M^{*}$
  $T := T^{*}$
endfor

Fig. 3. TabuFLTDP pseudocode
V. C o m p l e x i t y

We now discuss the complexity of the proposed heuristics. Let $F$ be the number of multicast flows that must be transported in the network. Let $\Delta = T_i = R_i \ \forall i \in V$ denote the identical in/out degrees for each node in the logical topology. For each analysed logical topology, we route i) its lightpaths over the physical topology with the HDAP algorithm, and ii) the unicast and multicast traffic over the logical topology.

The HDAP algorithm has complexity $O(N\Delta(M + \Delta N \log(\Delta N)))$, since at most $O(N\Delta)$ iterations are executed, while at each iteration at most $O(M + \Delta N \log(\Delta N))$ operations are required; $O(M)$ operations, indeed, are necessary to update the cost of the links of the

\[ S_k, \text{ and computing the maximum congestion level over all failure states.} \]

\[ C. \text{ TabuFLTDP Pseudocode} \]

\[ T := BuildInitialSolution \]
\[ IterBest := 1 \]
\[ T^{**} := T \]
\[ M^{**} := M \]

For $counter = 1$ to $IterationsNumber$ do
  if $counter - IterBest = LimitDiv$ then
    $T := BuildCycle(T, p)$
    $N(T) := BuildNeighborhood(T)$
    $T^{*} := BestNeighSol$ and update TabuList
  if $M^{*} \leq M^{**}$ then
    $T^{**} := T^{*}$
    $M^{**} := M^{*}$
  $T := T^{*}$
endfor

\[ \text{Fig. 3. TabuFLTDP pseudocode} \]

\[ \text{V. Complexity} \]

\[ \text{We now discuss the complexity of the proposed heuristics. Let} \ F \ \text{be the number of multicast flows that must be} \]
\[ \text{transported in the network. Let} \ \Delta = T_i = R_i \ \forall i \in V \ \text{denote the identical in/out degrees for each node in} \]
\[ \text{the logical topology. For each analysed logical topology, we route i) its lightpaths over the physical topology with the} \]
\[ \text{HDAP algorithm, and ii) the unicast and multicast traffic over the logical topology.} \]

\[ \text{The HDAP algorithm has complexity} \ O(N\Delta(M + \Delta N \log(\Delta N))) \], since at most \( O(N\Delta) \) iterations are executed, while at each iteration at most \( O(M + \Delta N \log(\Delta N)) \) operations are required; \( O(M) \) operations, indeed, are necessary to update the cost of the links of the
physical topology and $O(N\log(\Delta N))$ operations are necessary to run the Dijkstra algorithm. Since $M \leq N^2$, the HDAP complexity is upper bounded by $O(N^2 \Delta)$.

To evaluate solutions, it is necessary to route the unicast and multicast traffic. The unicast routing algorithm requires $O(N^2 \log(N\Delta))$ operations (Dijkstra algorithm). For the multicast traffic routing, once shortest paths are known, the computation of each Steiner Tree according to algorithm $SCTF$ requires $O(N^2 k)$ operations, where $k$ is an internal parameter of the $SCTF$ algorithm that we set to 5. In conclusion, the computation of the unicast and multicast routing requires $O(N^2 (\log(N\Delta) + Fk))$ operations.

The R&R heuristic to evaluate the initial solution has complexity $O(N^3 (\log(\Delta N) + Fk) + N^4)$, since at most $O(N)$ iterations are needed to complete the algorithm, and at each iteration it is necessary to route traffic and execute a $MW M$ (Maximum Weight Matching) algorithm.

Let us now focus on the complexity of the TS algorithm. At each iteration, the evaluation of all solutions in the neighborhood is necessary; this requires $O(\Delta^2 N^2 (N^3 \Delta + N^2 (\log(N\Delta) + Fk) + MN^2 (\log(N\Delta) + Fk)))$ operations, since $O(\Delta^2 N^2)$ neighbors are evaluated (assuming perturbations are generated using cycles of length 4), and the evaluation of each solution requires the execution of the HDAP algorithm and the execution of the routing algorithms for each failure state (i.e., $M + 1$ times). If the number of iterations is $I$, the resulting complexity is $O(I \Delta^2 N^2 (N^3 \Delta + N^2 (\log(N\Delta) + Fk) + MN^2 (\log(N\Delta) + Fk)))$. Thus, the computational complexity of TS is upper-bounded by $O(I \Delta^2 N^6 (\log(\Delta N) + Fk))$.

VI. NUMERICAL RESULTS

In this section we present numerical results obtained with the proposed approach (called joint optimization), and compare them against those obtained by performing a conventional optimization of the logical topology, and then optimally mapping the lightpaths on the physical topology according to the algorithm proposed in [14], and extended in order to deal with a unidirectional logical topology (this approach is called disjoint optimization).

Most of the reported results refer to the two ten-nodes topologies plotted in Figs. 4 and 5, since we were unable to run the optimal mapping for larger networks. However, results are also reported for larger topologies using our proposed approach. The first network that we consider was obtained by removing some nodes and links by the NSFnet topology, while the second network represents a possible Italian backbone IP network.

We consider randomly generated traffic patterns. Each source-destination traffic flow requires a bandwidth whose value is randomly extracted from an exponential distribution with mean $\mu = 1$.

The Tabu parameters used in our experiments, after an initial calibration were set as follows:

- **Tabu List**: a Tabu list of fixed size equal to 7 is used;
- **Cycles size**: during the exploration of the solution space, cycles of length 4 are used. In some cases, however, different perturbation rules are used to implement the diversification criterion. In particular, to ease the exit from local minima regions, after 50 iterations without improvement, a cycle of size 6 is used;
- **Stop Criterion**: the procedure is stopped after a fixed number of iterations. The number of iterations is set to 300, since this value seems to provide a good trade-off between the conflicting requirements of limiting the CPU time and obtaining good results.

In Tables I, II, III and IV we report results obtained with the disjoint and the joint optimization techniques, for different logical network configurations on the physical topology plotted in Fig. 4. Different columns report results obtained for configuration in which different values of capacity, $LC$, are assigned to each lightpath.

Three important network performance indices are reported for the obtained solutions: the congestion level in the no-failure state ($C_{S_0}$), the maximum value of the congestion level ($C_S$) in all the failure states, and the maximum amount of traffic ($TL$) that is lost in the network, due to a single link failure, expressed as percentage of the total offered traffic. Traffic losses are encountered whenever the flow on a lightpath exceeds the lightpath capacity.

We report results for different values of the nodal connectivity degree ($\Delta$), and maximum number of wavelengths on a fiber ($W$). The average performance indices (Mean) over 10 traffic instances, and the worst case (wrst) indices are reported.

In Tables I, II, III and IV respectively, we report results for four different values of connectivity degree ($\Delta = 2, 3, 4, 5$).

It can be observed that the joint optimization approach in these cases outperforms disjoint optimization, specially for what concerns the maximum traffic lost because of failures. For example, in Table I we see that with disjoint optimization the maximum lost traffic is still non-null when the link capacity $LC$ is 50, while, under joint-optimization, almost null losses are observed when the lightpath capacity is 30.

The difference between the two optimization procedures increases when the logical topology degree increases. Table IV shows that with joint-optimization no losses are observed for configurations in which the lightpath capacity is...
10, while under disjoint-optimization, losses are still registered when \( LC = 25 \).

Differences become even larger when the physical topology plotted in Fig. 5 is considered. In this case, trying to build a logical topology with nodal degree equal to 2, out of twenty traffic instances, ten times the disjoint optimization algorithm fails, since no mapping exists such that the logical topology remains connected under any single link failure scenario. This means that some source-destination pairs cannot communicate under some failure patterns, whichever capacity is assigned to lightpaths. In those cases, of course, the solution provided by disjoint optimization algorithms leads to unacceptable performance in terms of failure resilience. Table V reports results restricted to the ten cases in which the disjoint optimization does not fail.

In Tables VI, VII and VIII, results are reported for traffic scenarios comprising also multicast flows. Three multicast flows are added to the unicast traffic. For each multicast flow, both the source and the destinations are randomly selected among all nodes in the network. The average number of destinations of each multicast flow is fixed to 6 nodes. Also multicast flows require a bandwidth whose value is randomly extracted from an exponential distribution with mean \( \mu = 1 \).

Tables VI, VII and VIII refer to different values of connectivity degree (\( \Delta = 2, 3, 4 \) respectively). Also in presence of multicast traffic, joint-optimization significantly outperforms disjoint optimization.

Table IX instead reports a comparison between joint optimization and disjoint optimization in terms of the required number of wavelengths. Results refer to four topologies with different number of nodes and links: the network shown in Fig.4 (10 nodes, 14 links), the NSFnet topology (14 nodes, 21 links), the ARPAnet topology (21 nodes, 26 links) and the U.S.A. Long Distance topol-
ogy (28 nodes, 45 links). We observe that while for the 10 nodes topology the logical topology resulting from disjoint-optimization was obtained by applying the optimal mapping algorithm proposed in [14], for larger networks the results were obtained by performing the heuristic HDAP mapping algorithm over the outcome of the logical topology optimization procedure because the algorithm of [14] is too complex for networks of this size. Results show that also in terms of required number of wavelengths, the application of the joint-optimization algorithm appears to be advantageous, since it yields an average saving of about 20%.

Table X reports the CPU time needed to run an iteration of the joint-optimization Tabu Search algorithm. All results were obtained over a 800 MHz Pentium III PC running Linux 6.2.

Table X also reports the iteration number at which the optimum solution was found; the average value over 10 instances (M O-it) and the worst case (wrst O-it) value are reported. In all cases, 300 iterations were run before stopping the algorithm. We notice that only in one instance more than 100 iterations (114) were necessary to find the optimum value.

Finally, Fig. 6 reports the average number of iterations required to find a solution that differs a given percentage from the optimum. It is worth to note that a solution that is few percentages worse then the best can be obtained in a significantly smaller number of iterations than for the best solution.

VII. CONCLUSIONS

In this paper we proposed a new methodology for the design of fault-tolerant logical topologies in wavelength-routed WDM networks supporting unicast and multicast IP datagram flows.

Our approach to protection and restoration generalizes the concepts first proposed in [8], [9], [10], and relies on the exploitation of the intrinsic dynamic capabilities of IP routing, thus leading to cost-effective fault-tolerant logical topologies.

Our approach differs from those proposed in [8], [9], [10], since it considers the resilience properties of the topology during the logical topology optimization process, thus extending the optimization of the network resilience performance also on the space of logical topologies.

Numerical results clearly show that our approach largely outperforms the previous ones, and is able to obtain very good logical topologies with fault-tolerance properties at a limited cost.

REFERENCES

[15] Riferimento CPLEX
**TABLE I**

Comparison between FLTDP and the disjoint optimization of LTPD and Opt-MP with $\Delta = 2$ for network 1.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Disjoint Opt (LTPD+Opt-MP)</th>
<th>Joint Opt (FLTDP)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$LC=15$</td>
<td>$LC=20$</td>
</tr>
<tr>
<td>$C_{S_0}$ Mean</td>
<td>15.96</td>
<td>15.96</td>
</tr>
<tr>
<td>$C_S$ Mean</td>
<td>20.00</td>
<td>23.44</td>
</tr>
<tr>
<td>$TL$ Mean [%]</td>
<td>6.58</td>
<td>1.36</td>
</tr>
<tr>
<td>$C_{S_0}$ wrst</td>
<td>19.84</td>
<td>19.84</td>
</tr>
<tr>
<td>$C_S$ wrst</td>
<td>20.00</td>
<td>25.00</td>
</tr>
<tr>
<td>$TL$ wrst [%]</td>
<td>22.16</td>
<td>15.72</td>
</tr>
</tbody>
</table>

**TABLE II**

Comparison between FLTDP and the disjoint optimization of LTPD and Opt-MP with $\Delta = 3$ for network 1.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Disjoint Opt (LTPD+Opt-MP)</th>
<th>Joint Opt (FLTDP)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$LC=15$</td>
<td>$LC=20$</td>
</tr>
<tr>
<td>$C_{S_0}$ Mean</td>
<td>9.11</td>
<td>9.11</td>
</tr>
<tr>
<td>$C_S$ Mean</td>
<td>12.51</td>
<td>13.11</td>
</tr>
<tr>
<td>$TL$ Mean [%]</td>
<td>0.15</td>
<td>0.00</td>
</tr>
<tr>
<td>$C_{S_0}$ wrst</td>
<td>11.29</td>
<td>11.29</td>
</tr>
<tr>
<td>$C_S$ wrst</td>
<td>15.00</td>
<td>15.66</td>
</tr>
<tr>
<td>$TL$ wrst [%]</td>
<td>0.76</td>
<td>0.00</td>
</tr>
</tbody>
</table>
### TABLE III

Comparison between FLTDP and the disjoint optimization of LTDP and Opt-MP with $\Delta = 4$ for network 1.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Disjoint Opt (LTDP+Opt MP)</th>
<th>Joint Opt (FLTDP)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LC=10</td>
<td>LC=15</td>
</tr>
<tr>
<td></td>
<td>LC=20</td>
<td>LC=25</td>
</tr>
<tr>
<td></td>
<td>LC=30</td>
<td>LC=35</td>
</tr>
<tr>
<td></td>
<td>LC=40</td>
<td></td>
</tr>
<tr>
<td>$C_{S_{h}}$ Mean</td>
<td>6.32</td>
<td>6.32</td>
</tr>
<tr>
<td>$C_{S}$ Mean</td>
<td>10.00</td>
<td>14.42</td>
</tr>
<tr>
<td>$C_{S}$ wrst</td>
<td>7.91</td>
<td>7.91</td>
</tr>
<tr>
<td>$C_{S}$ wrst</td>
<td>10.00</td>
<td>15.00</td>
</tr>
<tr>
<td>$TL$ Mean [%]</td>
<td>17.42</td>
<td>10.36</td>
</tr>
<tr>
<td>$C_{S_{h}}$ wrst</td>
<td>34.11</td>
<td>28.31</td>
</tr>
<tr>
<td>$C_{S}$ wrst</td>
<td>7.91</td>
<td>7.91</td>
</tr>
<tr>
<td>$TL$ wrst [%]</td>
<td>34.11</td>
<td>28.31</td>
</tr>
</tbody>
</table>

### TABLE IV

Comparison between FLTDP and the disjoint optimization of LTDP and Opt-MP with $\Delta = 5$ for network 1.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Disjoint Opt (LTDP+Opt MP)</th>
<th>Joint Opt (FLTDP)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LC=10</td>
<td>LC=15</td>
</tr>
<tr>
<td></td>
<td>LC=20</td>
<td>LC=25</td>
</tr>
<tr>
<td></td>
<td>LC=30</td>
<td>LC=35</td>
</tr>
<tr>
<td></td>
<td>LC=40</td>
<td></td>
</tr>
<tr>
<td>$C_{S_{h}}$ Mean</td>
<td>6.27</td>
<td>6.27</td>
</tr>
<tr>
<td>$C_{S}$ Mean</td>
<td>7.62</td>
<td>7.62</td>
</tr>
<tr>
<td>$C_{S}$ wrst</td>
<td>7.48</td>
<td>7.48</td>
</tr>
<tr>
<td>$C_{S}$ wrst</td>
<td>9.74</td>
<td>9.74</td>
</tr>
<tr>
<td>$TL$ Mean [%]</td>
<td>6.27</td>
<td>6.27</td>
</tr>
<tr>
<td>$C_{S_{h}}$ wrst</td>
<td>7.62</td>
<td>7.62</td>
</tr>
<tr>
<td>$C_{S}$ wrst</td>
<td>7.48</td>
<td>7.48</td>
</tr>
<tr>
<td>$TL$ wrst [%]</td>
<td>9.74</td>
<td>9.74</td>
</tr>
</tbody>
</table>


### Table V
Comparison between FLTDP and the disjoint optimization of LTDP and Opt-MP with $\Delta = 5$ for network 2.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Disjoint Opt (LTDP+Opt MP)</th>
<th>Joint Opt (FLTDP)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$LC=20$</td>
<td>$LC=25$</td>
</tr>
<tr>
<td>$C_S$ Mean</td>
<td>13.27</td>
<td>13.27</td>
</tr>
<tr>
<td>$C_q$ Mean</td>
<td>16.81</td>
<td>11.71</td>
</tr>
<tr>
<td>$C_S$ wrst</td>
<td>25.00</td>
<td>30.00</td>
</tr>
<tr>
<td>$C_q$ wrst</td>
<td>29.20</td>
<td>24.43</td>
</tr>
</tbody>
</table>

### Table VI
Comparison between FLTDP and the disjoint optimization of LTDP and Opt-MP with $\Delta = 2$ for network 1 and multicast traffic.
### TABLE VII
Comparison between FLTDP and the disjoint optimization of LTDP and Opt-MP with $\Delta = 3$ for network 1 and multicast traffic.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Disjoint Opt (LTDP+Opt MP)</th>
<th>Joint Opt (FLTDP)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$LC=15$</td>
<td>$LC=20$</td>
</tr>
<tr>
<td>$C_G$ Mean</td>
<td>15.00</td>
<td>20.00</td>
</tr>
<tr>
<td>$TL$ Mean [%]</td>
<td>19.25</td>
<td>13.14</td>
</tr>
<tr>
<td>$C_{S_k}$ wrst</td>
<td>13.03</td>
<td>13.03</td>
</tr>
<tr>
<td>$C_G$ wrst</td>
<td>15.00</td>
<td>20.00</td>
</tr>
<tr>
<td>$TL$ wrst [%]</td>
<td>28.44</td>
<td>23.67</td>
</tr>
</tbody>
</table>

### TABLE VIII
Comparison between FLTDP and the disjoint optimization of LTDP and Opt-MP with $\Delta = 4$ for network 1 and multicast traffic.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Disjoint Opt (LTDP+Opt MP)</th>
<th>Joint Opt (FLTDP)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$LC=15$</td>
<td>$LC=20$</td>
</tr>
<tr>
<td>$C_{S_k}$ Mean</td>
<td>7.02</td>
<td>7.02</td>
</tr>
<tr>
<td>$C_G$ Mean</td>
<td>15.00</td>
<td>18.22</td>
</tr>
<tr>
<td>$TL$ Mean [%]</td>
<td>7.88</td>
<td>3.60</td>
</tr>
<tr>
<td>$C_G$ wrst</td>
<td>15.00</td>
<td>20.00</td>
</tr>
<tr>
<td>$TL$ wrst [%]</td>
<td>23.07</td>
<td>13.68</td>
</tr>
<tr>
<td>Network</td>
<td>Mean W-Number</td>
<td>Disjoint Opt (LTDp+Opt MP)</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------</td>
<td>----------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Δ = 2</td>
</tr>
<tr>
<td>Network 1: 10 nodes 14 links</td>
<td>3.2</td>
<td>5.0</td>
</tr>
<tr>
<td>NSFnet: 14 nodes 21 links</td>
<td>3.9</td>
<td>5.1</td>
</tr>
<tr>
<td>ARPAnet: 21 nodes 26 links</td>
<td>6.4</td>
<td>9.2</td>
</tr>
<tr>
<td>U.S.A. Long Distance: 28 nodes 45 links</td>
<td>7.2</td>
<td>11.4</td>
</tr>
</tbody>
</table>

**TABLE IX**

**NUMBER OF WAVELENGTHS REQUIRED TO MAP THE LOGICAL TOPOLOGY**

<table>
<thead>
<tr>
<th>Network</th>
<th>CPU TIME</th>
<th>Δ = 2</th>
<th>Δ = 3</th>
<th>Δ = 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network 1</td>
<td>0.78</td>
<td>19.88</td>
<td>85</td>
<td>1.38</td>
</tr>
<tr>
<td>NSFnet</td>
<td>6.69</td>
<td>62.22</td>
<td>98</td>
<td>7.12</td>
</tr>
<tr>
<td>ARPAnet</td>
<td>60.46</td>
<td>36.22</td>
<td>67</td>
<td>47.48</td>
</tr>
<tr>
<td>U.S.A. Long Distance</td>
<td>246.78</td>
<td>36.1</td>
<td>60</td>
<td>177.20</td>
</tr>
</tbody>
</table>

**TABLE X**

**CPU TIMES FOR ONE ITERATION OF THE TABU SEARCH ALGORITHM.**
Fig. 6. Average number of iterations versus the percentage distance from the optimal solution.