Computers & Operations Research ∎ (∎∎∎∎) ∎∎∎–∎∎∎

Contents lists available at SciVerse ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/caor

A GRASP with path-relinking heuristic for the survivable IP/MPLS-over-WSON multi-layer network optimization problem

Oscar Pedrola [a,*], Marc Ruiz [a], Luis Velasco [a], Davide Careglio [a], Oscar González de Dios [b], Jaume Comellas [a]

[a] Advanced Broadband Communications Center (CCABA), Universitat Politècnica de Catalunya (UPC), 08034 Barcelona, Spain
[b] Telefónica I+D, Don Ramón de la Cruz 82-84, 28006 Madrid, Spain

ARTICLE INFO

Keywords:
Multi-layer optimization
Survivability
Greedy randomized adaptive search procedure (GRASP)
Path-relinking (PR)
Biased random-key genetic algorithm (BRKGA)

ABSTRACT

In this paper we deal with the survivable *internet protocol* (IP)/*multi-protocol label switching* (MPLS)-over-*wavelength switched optical network* (WSON) multi-layer network optimization problem (SIMNO). This problem entails planning an IP/MPLS network layer over a photonic mesh infrastructure whilst, at the same time, ensuring the highest availability of services and minimizing the *capital expenditures* (CAPEX) investments. Such a problem is currently identified as an open issue among network operators, and hence, its solution is of great interest. To tackle SIMNO, we first provide an *integer linear programming* (ILP) formulation which provides an insight into the complexity of its managing. Then, a *greedy randomized adaptive search procedure* (GRASP) with *path-relinking* (PR) together with a *biased random-key genetic algorithm* (BRKGA) are specifically developed to help solve the problem. The performance of both heuristics is exhaustively tested and compared making use of various network and traffic instances. Numerical experiments show the benefits of using GRASP instead of BRKGA when dealing with highly complex network scenarios. Moreover, we verified that the use of GRASP with PR remarkably improves the basic GRASP algorithm, particularly in real-sized, complex scenarios such as those proposed in this paper.

1. Introduction

With the advance in optics and the commercialization of enhanced devices like wavelength selective switches and tunable lasers, nowadays it is possible to remotely configure *optical cross-connects* (OXCs), and thus, to deploy *wavelength switched optical networks* (WSON). Strictly speaking, WSON extends the concept of *automatically switched optical network* (ASON) [1] by applying an intelligent control plane based on *generalized multi-protocol label switching* (GMPLS) [2]. In fact, WSONs standardization activities are currently in progress in the *internet engineering task force* (IETF) within the *common control and measurement plane* (CCAMP) working group [3]. WSONs enable to dynamically reconfigure networks, i.e., enable the automatization of the setup and tear-down of end-to-end optical connections (known as lightpaths) and the recovery of such lightpaths in case of failure. Thus, WSONs allow for an efficient network operation which implies significant savings in the core transport network. Today, the optical layer (managed by a network operator) is an already deployed photonic infrastructure that provides, at the same time, different client networks with transport services such as leased lines, packet-switched networks (e.g., Internet), *virtual private networks* (VPNs), *synchronous digital hierarchy* (SDH) networks, etc. Our goal in this paper is to further improve its benefits by applying an intelligent interworking strategy between the packet and WSON layers based on a multi-layer optimization process. Indeed, a multi-layer network can perform an optimal load balancing between these two layers optimizing both the cost of the packet layer and the utilization of the WSON layer. Without loss of generality, we assume in this work a multi-layer network which consists of an *internet protocol* (IP)/*multi-protocol label switching*(MPLS) packet layer over a photonic WSON transport layer, but the study herein presented is applicable to other packet technologies such as the emerging *multi-protocol label switching transport profile* (MPLS-TP) and *provider backbone bridges traffic engineering* (PBB-TE) transport alternatives.

Hence, in this paper we tackle, for the first time to the best of our knowledge, the problem of a joint optimization of survivable non-symmetrical network layers so as to provide network operators with a competitive multi-layer network planning tool which aims at minimizing the *capital expenditures* (CAPEX) (i.e., those costs related with purchasing and installing fixed infrastructures, such as equipments).

[*] Corresponding author. Tel.: +34 93 401 7182; fax: +34 93 401 7055.
E-mail address: opedrola@ac.upc.edu (O. Pedrola).

0305-0548/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2011.10.026

This multi-layer network is specifically designed to provide companies with premium *layer* 1(L1) and L2 VPN services. These services have stringent availability requirements, and therefore, ensuring network recovery in front of any kind of network component failure becomes crucial to the services' success. Indeed, in such high-capacity multi-layer network scenario, any single link or node failure would lead to tremendous losses for both network operators and clients. Thus, the concept of survivability, which allows a network to quickly recover from any kind of outage and restore the affected traffic, becomes a critical objective in the design and planning of next-generation high-speed multi-layer networks. Another advantage of the multi-layer approach is the fact that it allows the application of specifically designed multi-layer recovery mechanisms. These procedures are able to trigger coordinated actions across both layers, thereby substantially reducing the over-dimensioning of IP/MPLS nodes when compared to the single-layer approach (i.e., separate optimization of layers) [4].

Therefore, and strictly speaking, in this work we deal with the so-called *survivable IP/MPLS-over-WSON multi-layer network optimization* (SIMNO) problem. To this end, and given the operator-dependent input parameters, that is, the WSON network deployed and the traffic demands to be satisfied, we design the IP/MPLS layer. It consists in the dimensioning of its nodes with the required *opto-electronic* (OE) interfaces and in the establishment of the virtual link connectivity at the IP/MPLS level through the given WSON layer so that every traffic demand can be successfully accommodated. Note that in the SIMNO problem, the over-dimensioning of IP/MPLS nodes required to guarantee recovery in front of any kind of network component outage is minimized thanks to the application of multi-layer optimization techniques. Therefore, we provide a solution to a real problem which is of great interest to network operators. Indeed, following the SIMNO approach, operators will be able to deploy a survivable IP/MPLS layer on top of an already deployed WSON infrastructure while minimizing their CAPEX investments. In this work, CAPEX involve the costs of both IP/MPLS nodes and OE ports installed on them, as well as the cost of using both optical ports and kilometers of optical fiber from an existing WSON network.

In order to deal with SIMNO, we present and evaluate a formal model of the problem by means of an *integer linear programming* (ILP) formulation. Since the resultant model is computationally impractical, we make use of two well-known and powerful meta-heuristic models to help solve the problem, these are, a *greedy randomized adaptive search procedure* (GRASP) together with a *path-relinking* (PR) intensification method, and a *biased random-key genetic algorithm* (BRKGA). To evaluate both heuristics, we carry out a set of experiments using both methodologies and assess their respective performances. Furthermore, we evaluate the impact of introducing the PR intensification strategy into GRASP in the so-called GRASP *with path-relinking* (GRASP+PR) meta-heuristic. To conduct such experiments, we consider a set of network traffic models which are consistent with the traffic profiles foreseen in the years to come and evaluate them in three different IP/MPLS network configurations of a realistic Spanish telecommunications network.

The remainder of this paper is organized as follows. In Section 2, we briefly survey previous works on the design and evaluation of survivable multi-layer networks. Section 3 describes the SIMNO problem in detail. First, the multi-layer network architecture characteristics and survivability restoration schemes are presented. Then, a mathematical formulation of the SIMNO problem is provided. Afterwards, in Section 4, both the GRASP+PR and BRKGA meta-heuristics considered to solve the SIMNO problem are described. Illustrative computational experiments are provided in Section 5 and finally concluding remarks are made in Section 6.

2. Related work and contributions

Survivable multi-layer networks have traditionally been designed following the classical overlay approach where two redundant IP/MPLS networks are deployed over the photonic infrastructure. However, operators are now facing the challenge of dimensioning networks able to cope with the expected huge IP traffic volumes, and at the same time, keeping constant or even reducing connectivity prices. Hence, operators look for technologies providing the lowest possible network costs.

In protection and restoration schemes developed for legacy technologies, only optical links and electronic ports/interfaces have been considered as points of failure. For this reason, networks implement protection or restoration mechanisms to survive to such kind of failures. IP/MPLS nodes are not, nevertheless, as trusty as legacy telecommunication equipments. This is mainly due to the constant software and hardware upgrades they undergo [4,5]. To tackle this issue, backbone nodes redundancy-based schemes have been proposed for operators willing to protect their networks against IP/MPLS nodes failures [4]. However, this approach entails a substantial increase in network CAPEX, thereby clearly demonstrating that the duplicate network scheme is far away from being the optimal solution, and that the design and evaluation of novel survivable multi-layer network optimization methods such as SIMNO has gained great momentum.

In the literature, multiple recovery schemes have been specifically designed and tailored for multi-layer networks. For example, a comprehensive survey of them can be found in [5]. Another interesting study involving the evaluation of a coordinated link restoration scheme to be used in packet-over-optical networks can be found in [6]. In that work, authors illustrate a novel scheme which is cost effective compared to duplicating nodes, though it has the disadvantage of requiring the IP/MPLS and optical topologies to be symmetrical (i.e., every node has both packet and optical switching capabilities). It is worth noticing that the underlying WSON, which supports a number of heterogenous client networks and provides a range of services to residential and business customers, needs to provide different availability degrees. Hence, if symmetrical topologies are considered, the IP/MPLS layer should be designed to cope with the requirements of the most constraining service, thereby highly and unnecessarily increasing network CAPEX.

Accordingly, the SIMNO approach is aimed at defining orchestrated interworking recovery actions to avoid the duplication of IP/MPLS backbone nodes. However, in this case, no symmetrical topologies are required, and hence, a number of client networks with different availability degrees can be allocated on top of the WSON. In addition, we rely on lightpath restoration, a technique which provides a finer granularity to recover selected lightpaths in very short times (e.g., on the order of hundreds of ms [7]), and on a novel connectivity restoration scheme to deal, not only with IP/MPLS node failures, but also with the rest of failures.

In the literature, we find a few interesting works addressing the IP/MPLS-over-WSON multi-layer network planning problem. In [8], the authors present an ILP formulation aimed at maximizing a utility function for the network operator, that is, the difference between revenues and costs, considering a scenario without failures. To this end, authors propose a Lagrangian relaxation-based method. A similar approach is not, nonetheless, applicable to the SIMNO problem owing to both its size and structure. Indeed, SIMNO includes a huge set of single failure scenarios (i.e., every IP/MPLS node, OE port and optical link in the network). For this very reason, in this work we develop and evaluate two different meta-heuristic methods to solve the SIMNO problem. Strictly speaking, an heuristic based on GRASP and PR [9,10] and another on BRKGA [11] are proposed to find cost-effective solutions for the SIMNO problem within practical running times. As a matter of fact, previous works have already considered evolutionary genetic algorithms (GA) for the planning of optical networks. For instance, in [12] a GA-based heuristic for the single layer survivable optical network planning is presented, and in [13], a GA is applied to dimension single layer dynamic optical networks. In this paper, by contrast, we consider the GRASP methodology to solve the SIMNO problem and compare its performance to that of the novel BRKGA meta-heuristic. Moreover, we evaluate the impact of the PR intensification strategy on the results obtained by GRASP, thereby illustrating one more time a successful application of this combined meta-heuristic.

3. SIMNO problem formulation

3.1. Multi-layer network architecture

The multi-layer network architecture considered in this work is depicted in Fig. 1. In this reference scenario, three types of IP/MPLS nodes can be distinguished at the packet layer (IP/MPLS), these are, *metro* nodes performing client flow aggregation, *transit* nodes providing routing flexibility, and *interconnection* nodes supporting inter-operator connection. Additionally, transport nodes (OXCs) connected by fiber links create an WSON layer. In order to minimize the overall number of OE ports in the network, metro-to-metro connections are avoided being every metro node connected to one or more transit nodes. Moreover, while it is typical that a transit node is collocated with a transport node, metro nodes are usually closer to clients, and thus, some ad hoc connectivity is used to connect metro to transport nodes. Fig. 1 illustrates an exemplary end-to-end MPLS *label switched path* (LSP) established between two metro nodes (orange line). Note that in this example, the LSP makes use of interconnection nodes to pass from a network operated by one particular carrier to another network operated by another different carrier.

Fig. 2 depicts an example illustrating how a multi-layer network can be designed. To be precise, Fig. 2a, shows a portion of the multi-layer network where each IP/MPLS metro node is connected to a transit node through virtual links, and hence, a virtual topology is created. Each virtual link is supported by a lightpath in the WSON layer. This lightpath is routed through the minimum cost path over the WSON layer. In the example, metro router M1 is connected to transit router T1 by means of only one lightpath. However, and in order to guarantee the survivability of the network, extra-capacity has already been added to every node.

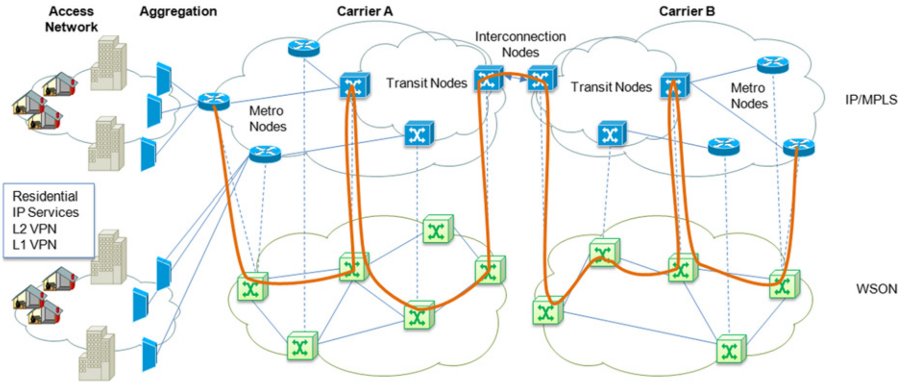

Fig. 1. Metro and multi-layer network architecture.

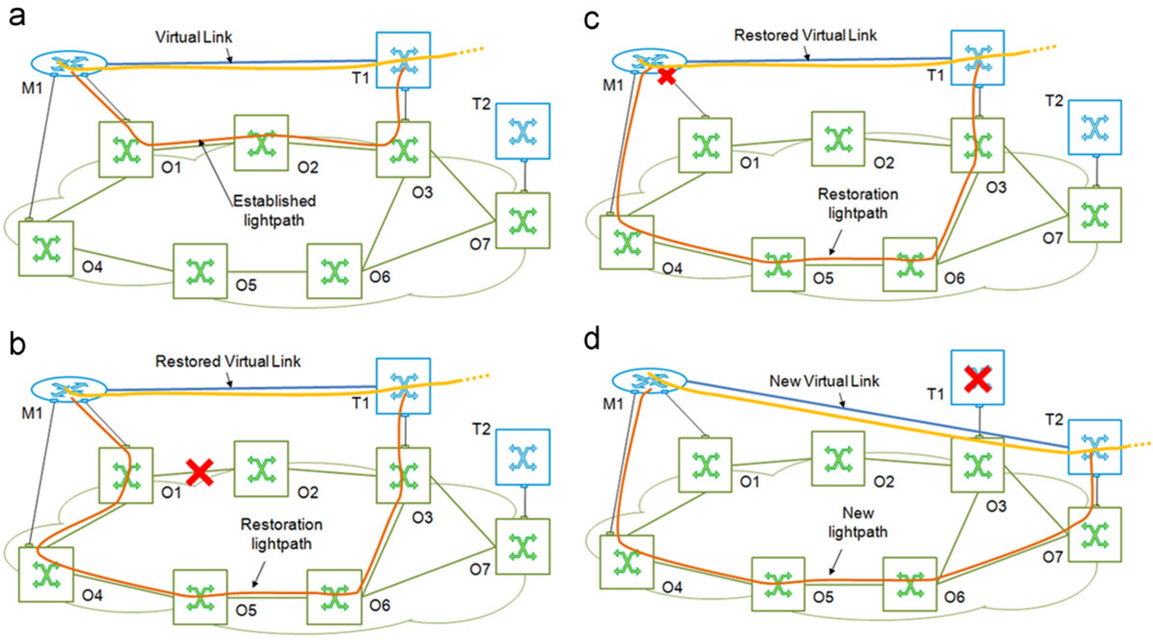

Fig. 2. (a) Design of a multi-layer planned network portion; (b) recovery from a link failure; (c) from a port failure; and (d) from a node failure.

In multi-layer problems, the components that may fail are optical links, OE ports and both optical and IP/MPLS nodes. We consider every component in the network as being mutually failure-independent, and thus, multiple failure scenarios are not considered in this work since their probability to happen is extremely low. Moreover, complete optical node failures are also highly unlikely and thus are also neglected in this work. This is not, however, the case with IP/MPLS nodes whose failures, mainly caused by software crashes, are a great deal more frequent.

On the one hand, in the event of an optical link failure, the multi-layer network can apply joint recovery schemes to restore the affected traffic demands. For example, when the optical link O1–O2 fails (Fig. 2b) recovery actions are triggered to restore the metro-to-transit (M1–T1) connectivity. Note that if a lightpath is restored at the optical layer, the connectivity at the IP/MPLS layer remains unaltered (with the corresponding CAPEX savings implications). In contrast, if no restoration is possible, a new lightpath has to be established to connect the IP/MPLS metro node to a different transit node (e.g., M1–T2), thus restoring the metro-to-transit connectivity. Note, however, that in this case transit node T2 must be over-dimensioned with additional OE ports to be able to cope with the requirements of this newly created lightpath. Once the connectivity is restored, the MPLS LSP can be eventually rerouted over the reconfigured virtual topology. The same actions are taken in the event of a port failure (Fig. 2c).

On the other hand, in the event of an IP/MPLS node failure (Fig. 2d), new lightpaths are established between every metro node connected to the failed node and a different transit node in order to properly restore the metro-to-transit connectivity. Therefore, in this failure scenario, setting up new virtual links is required. In the example, virtual link M1–T2 is created. After reconfiguring the virtual topology, the affected MPLS LSPs are rerouted.

3.2. Problem statement

For the sake of clarity, the following information defines the problem input data:

- The WSON network topology consisting of both OXC nodes and fiber links.
- The correspondences between IP/MPLS nodes and OXC nodes are established beforehand.
- Each IP/MPLS node can establish a connection to each other so that all possible virtual links needed to establish a mesh virtual connectivity are predefined.
- The origin/destination (O/D) matrix and the bandwidth of each demand.

A solution to the problem must specify the configuration of each IP/MPLS node in terms of switching capability and number and bitrate of OE ports. For each virtual link used in the optimal solution, a supporting lightpath must be established in the WSON network. Moreover, the route of the MPLS LSP over the virtual topology must be determined for every demand.

Additionally to the aforementioned, the following assumptions are considered:

1. Given a bandwidth threshold, the set of demands is divided into two subsets: one with the demands whose bandwidth is lower than the threshold (subset 1), and another one with those demands whose bandwidth is higher or equal than the threshold (subset 2).
2. The route of an MPLS LSP consists of two metro nodes (source and destination) and a number of intermediate transit nodes. While the demands in subset 1 are routed by, at least, one transit node, those in subset 2 can use an optical bypass which connects both end nodes directly (i.e., no intermediate IP/MPLS node is traversed). Note that although optical bypassing can generally reduce network costs since it leads to a reduction in the number of ports and switching capability of transit nodes, its use has been restricted to just highly loaded virtual links to avoid MAC address table explosion [6].
3. For the sake of simplicity, we define a virtual metro node for those demands whose source or destination is a node outside the network. Such a node represents any external network and is connected to every interconnection node of the IP/MPLS network being planned. Hence, neither its requirements (i.e., number of ports and switching capability) nor its cost are taken into consideration to evaluate the feasibility of network solutions.
4. When a failure occurs, all affected MPLS LSPs must be re-routed. Complementary, the non-affected LSPs must remain in their current routes. However, WSON route and/or OE port assignment may change.

For the forthcoming ILP, we have considered a *node-link* formulation for the IP/MPLS routing and network planning constraints and an *arc-path* approach for the assignment of virtual links to lightpaths. A set of WSON routes is pre-computed and available for each virtual link.

Note that as a result of the proposed routing strategy, one virtual link can be supported by a number of parallel lightpaths, thus each virtual link has been divided into several entities called channels. In such a way, the aggregation of demands is facilitated, and hence, an optimal exploitation of the network capacity is guaranteed. Each channel of a virtual link carrying an MPLS LSP is associated with one lightpath in the WSON network. Then, four ports with the same bitrate must be installed in order to establish the required MPLS-to-MPLS virtual connection (i.e., two ports are installed in the IP/MPLS nodes and two more in the associated OXCs).

It is worth noting that failures affecting fiber links and IP/MPLS nodes can be identified before the optimization begins owing to the fact that the WSON network topology and the location of IP/MPLS nodes are known. In contrast, the number and location of OE ports is unknown until the optimization ends. Hence, the consideration of port failures drastically increases the complexity of the problem (note that even non-linear constraints would appear). Aiming at including port failures while keeping the linearity of the problem, we have attached a number of slots (i.e., a virtual port location which might or might not have a port installed on it) to each IP/MPLS node. This data structure allows us to define beforehand the number of failures since failures are associated to the pre-defined slots. Thus, considering failures in slots is equivalent to consider failures in OE ports.

Every single failure represents a specific failure scenario, which is characterized by the IP/MPLS nodes, slots, virtual links, and WSON routes that can be used when the failure occurs. The network dimensioning is unique and must ensure that every demand is transported under any failure scenario guaranteeing network survivability. For this very reason, the model obtains one channel-to-slot assignment and another channel-to-lightpath assignment for each failure scenario. This fact complicates the formulation but provides flexibility to perform the network planning, and hence, to reduce network CAPEX.

3.3. Notation

The following notation has been defined for sets and parameters:

Optical topology

$\mathcal{L}$ set of fiber links, index l
$\mathcal{K}(e)$ set of WSON routes for virtual link e, index k

p_l^k binary, equal to 1 if route k contains fiber link l

Len_l integer, with the length of fiber link l in km

w_l integer, with the number of wavelengths of fiber link l

Virtual topology

$\mathcal{N}$ set of IP/MPLS nodes, index n

$\mathcal{N}_m$ subset of $\mathcal{N}$ containing the metro nodes

$\mathcal{N}_t$ subset of $\mathcal{N}$ containing the transit nodes

$\mathcal{N}_v$ subset of $\mathcal{N}$ containing the interconnection nodes

$\mathcal{S}(n)$ set of slots of node n, index s

$\mathcal{E}$ set of virtual links, index e

$\mathcal{E}(n)$ set of virtual links incident to node n, index e

$\mathcal{E}_h(n)$ subset of $\mathcal{E}(n)$ containing the links reserved to demands belonging to subset 2

$\mathcal{E}_t(n)$ subset of $\mathcal{E}(n)$ defined by: $\mathcal{E}(n)-\mathcal{E}_h(n)$

$\mathcal{I}(e)$ end nodes of virtual link e, index n

$\mathcal{C}(e)$ set of channels of virtual link e, index c

Demands

$\mathcal{D}$ set of demands, index d

$\mathcal{SD}(d)$ source and destination nodes of demand d

b_d integer, with the bandwidth of demand d in Gbps

h_d binary, equal to 1 if demand d belongs to subset 2

Failures

$\mathcal{F}$ set of failure scenarios, index f. *Note:* Scenario 0 represents the non-failure scenario

a^{fk} binary, equal to 1 if WSON route k is available under failure scenario f

a^{fns} binary, equal to 1 if slot s of node n is available under failure scenario f

Equipment costs and others

c_{fo} real, with the cost per kilometer of restorable lightpath

$\mathcal{PT}$ set of OE port bitrates

pc_i real, with the cost of one port with bitrate i. *Note:* this value includes the cost of the associated OXC port

pk_i integer, with the capacity of one OE port with bitrate i in Gbps

$\mathcal{RT}$ set of router classes

rc_j real, with the cost of one router of class j

rk_j integer, with the switching capability of one router of class j in Gbps

rpk_j integer, with the number of slots available in a router of class j

M a large positive constant

The decision variables are

x_{dec}^f binary, equal to 1 if demand d is routed through channel c of virtual link e, under failure scenario f. 0 otherwise

x_d^f binary, equal to 1 if the route of demand d under failure scenario f must be the same than that in the basic scenario. 0 otherwise

y_{ec}^{fk} binary, equal to 1 if channel c of virtual link e is assigned to WSON route k, under failure scenario f. 0 otherwise

y_{ec}^{fns} binary, equal to 1 if channel c of virtual link e is assigned to slot s of node n, under failure scenario f. 0 otherwise

z_i^{ns} binary, equal to 1 if slot s of node n is equipped with a port with bitrate i. 0 otherwise

z_j^n binary, equal to 1 if node n is equipped with a router of class j. 0 otherwise

t^{fns} positive integer, with the total amount of traffic (in Gbps) in slot s of node n under failure scenario f

3.4. Mathematical formulation

The cost of the network can be computed as the sum of two parts: the cost of equipping nodes and installing ports ($cost_{Equip}$) and the cost of the lightpaths established to support the virtual links ($cost_{Lightpath}$). Both costs can be computed as follows:

$$cost_{Equip} = \sum_{n \in \mathcal{N}_m \cup \mathcal{N}_t} \left(\sum_{s \in \mathcal{S}(n)} \sum_{i \in \mathcal{PT}} pc_i \cdot z_i^{ns} + \sum_{j \in \mathcal{RT}} rc_j \cdot z_j^n \right), \tag{1}$$

$$cost_{Lightpath} = c_{fo} \cdot \sum_{e \in \mathcal{E}} \sum_{c \in \mathcal{C}(e)} \sum_{k \in \mathcal{K}(e)} y_{ec}^{0k} \cdot \sum_{l \in \mathcal{L}} len_l \cdot p_l^k. \tag{2}$$

Finally, the formulation of the problem is as follows:

$$min\ CAPEX = cost_{Equip} + cost_{Lightpath} \tag{3}$$

$$s.t. \sum_{e \in \mathcal{E}_t(n)} \sum_{c \in \mathcal{C}(e)} x_{dec}^f + h_d \cdot \sum_{e \in \mathcal{E}_k(n)} \sum_{c \in \mathcal{C}(e)} x_{dec}^f = 1, \quad \forall d \in \mathcal{D},\ f \in \mathcal{F},\ n \in \mathcal{SD}(d), \tag{4}$$

$$\sum_{e \in \mathcal{E}(n)} \sum_{c \in \mathcal{C}(e)} x_{dec}^f \le 2, \quad \forall d \in \mathcal{D},\ f \in \mathcal{F},\ n \in \overline{\mathcal{SD}(d)} \cap \mathcal{N}_t, \tag{5}$$

$$\sum_{e \in \mathcal{E}(n)} \sum_{c \in \mathcal{C}(e)} x_{dec}^f \le 0, \quad \forall d \in \mathcal{D},\ f \in \mathcal{F},\ n \in \overline{\mathcal{SD}(d)} \cap (\mathcal{N}_m \cup \mathcal{N}_v), \tag{6}$$

$$\sum_{e' \in \mathcal{E}(n)} \sum_{c' \in \mathcal{C}(e')} x_{de'c'}^f \ge \sum_{c \in \mathcal{C}(e)} x_{dec}^f, \quad \forall d \in \mathcal{D},\ f \in \mathcal{F}, \quad n \in \overline{\mathcal{SD}(d)} \cap \mathcal{N}_t,\ e \in \mathcal{E}(n), \tag{7}$$

$$\sum_{d \in \mathcal{D}} x_{dec}^f \le M \cdot \sum_{k \in \mathcal{K}(e)} a^{fk} \cdot y_{ec}^{fk}, \quad \forall f \in \mathcal{F},\ e \in \mathcal{E},\ c \in \mathcal{C}(e), \tag{8}$$

$$\sum_{k \in \mathcal{K}(e)} y_{ec}^{fk} \le 1, \quad \forall f \in \mathcal{F},\ e \in \mathcal{E},\ c \in \mathcal{C}(e), \tag{9}$$

$$\sum_{e \in \mathcal{E}} \sum_{c \in \mathcal{C}(e)} \sum_{k \in \mathcal{K}(e)} p_l^k \cdot y_{ec}^{fk} \le w_l, \quad \forall f \in \mathcal{F},\ l \in \mathcal{L}, \tag{10}$$

$$\sum_{d \in \mathcal{D}} x_{dec}^f \le M \cdot \sum_{s \in \mathcal{S}(n)} a^{fns} \cdot y_{ec}^{fns}, \quad \forall f \in \mathcal{F},\ e \in \mathcal{E},\ c \in \mathcal{C}(e),\ n \in \mathcal{I}(e), \tag{11}$$

$$\sum_{s \in \mathcal{S}(n)} y_{ec}^{fns} \le 1, \quad \forall f \in \mathcal{F},\ e \in \mathcal{E},\ c \in \mathcal{C}(e),\ n \in \mathcal{I}(e), \tag{12}$$

$$\sum_{d \in \mathcal{D}} b_d \cdot x_{dec}^f - M \cdot (1 - y_{ec}^{fns}) \le t^{fns}, \quad \forall n \in \mathcal{N}_m \cup \mathcal{N}_t,\ s \in \mathcal{S}(n), \quad e \in \mathcal{E}(n),\ c \in \mathcal{C}(e),\ f \in \mathcal{F}, \tag{13}$$

$$t^{fns} \le \sum_{i \in \mathcal{PT}} pk_i \cdot z_i^{ns}, \quad \forall n \in \mathcal{N}_m \cup \mathcal{N}_t,\ s \in \mathcal{S}(n),\ f \in \mathcal{F}, \tag{14}$$

$$\sum_{i \in \mathcal{PT}} z_i^{ns} \le 1, \quad \forall n \in \mathcal{N}_m \cup \mathcal{N}_t,\ s \in \mathcal{S}(n), \tag{15}$$

$$\sum_{s \in \mathcal{S}(n)} t^{fns} \le \sum_{j \in \mathcal{RT}} rk_j \cdot z_j^n, \quad \forall n \in \mathcal{N}_m \cup \mathcal{N}_t,\ f \in \mathcal{F}, \tag{16}$$

$$\sum_{s \in \mathcal{S}(n)} \sum_{i \in \mathcal{PT}} z_i^{ns} \le \sum_{j \in \mathcal{RT}} rpk_j \cdot z_j^n, \quad \forall n \in \mathcal{N}_m \cup \mathcal{N}_t,\ f \in \mathcal{F}, \tag{17}$$

Please cite this article as: Pedrola O, et al. A GRASP with path-relinking heuristic for the survivable IP/MPLS-over-WSON multi-layer network optimization problem. Computers and Operations Research (2011), doi:10.1016/j.cor.2011.10.026

$$\sum_{i \in \mathcal{RT}} z_i^n \le 1, \quad \forall n \in \mathcal{N}_m \cup \mathcal{N}_t, \tag{18}$$

$$\sum_{n \in \mathcal{I}(e)} \sum_{s \in \mathcal{S}(n)} (1-a^{fns}) \cdot y_{ec}^{0ns} + M \cdot (1-x_{dec}^f) \ge x_d^f,$$
$$\forall d \in \mathcal{D}, \quad f \in \mathcal{F}-\{0\}, \; e \in \mathcal{E}, \; c \in \mathcal{C}(e), \tag{19}$$

$$\sum_{c \in \mathcal{C}(e)} x_{dec}^0 - \sum_{c \in \mathcal{C}(e)} x_{dec}^f \le x_d^f, \quad \forall d \in \mathcal{D}, \; f \in \mathcal{F}, \; e \in \mathcal{E}, \tag{20}$$

$$\sum_{c \in \mathcal{C}(e)} x_{dec}^0 - \sum_{c \in \mathcal{C}(e)} x_{dec}^f \ge -x_d^f, \quad \forall d \in \mathcal{D}, \; f \in \mathcal{F}, \; e \in \mathcal{E}, \tag{21}$$

$$x_d^f, x_{dec}^f, y_{ec}^{fk}, y_{ec}^{fns}, z_i^{ns}, z_j^n \in \{0,1\}, \quad f \in \mathcal{F}, \; d \in \mathcal{D}, \; n \in \mathcal{N}, \; e \in \mathcal{E},$$
$$s \in \mathcal{S}(n), \quad c \in \mathcal{C}(e), \; k \in \mathcal{K}(e), \; i \in \mathcal{PT}, \; j \in \mathcal{RT}, \tag{22}$$

$$t^{fns} \in \mathbb{Z}^+, \quad f \in \mathcal{F}, \; n \in \mathcal{N}, \; s \in \mathcal{S}(n). \tag{23}$$

The objective function (3) minimizes the total cost of the network.

Constraints (4)–(7) are responsible for routing and aggregating the demands through the virtual topology. Constraint (4) ensures that every demand is routed under any failure scenario. Constraints (5)–(7) make sure the continuity of each MPLS route through the virtual topology.

Constraints (8)–(12) connect the virtual topology with the WSON. Constraints (8)–(10) assign one WSON route to each used channel in a virtual link. Additionally, constraint (10) ensures that the reserved WSON capacity is not violated. Constraints (11) and (12) connect both ends of each channel with two ports.

Constraints (13)–(18) dimension the IP/MPLS network. In constraint (13), the maximum amount of traffic routed through a slot is computed. This variable is used in constraints (14) and (15) to dimension the port that must be placed in that slot. The dimension of each node is determined in constraints (16)–(18).

Constraint (19) fixes those demands that must remain in their routes under every failure scenario, and constraints (20) and (21) prevent that the route of those demands changes. Finally, constraints (22) and (23) define the variables either as binary or integer.

With respect to the complexity of the problem, it is worth mentioning that even simpler versions of the survivable network planning model have been shown to be NP-hard [14]. Indeed, considering the problem in hand, the total amount of variables can be approximated by $|\mathcal{F}| \cdot |\mathcal{E}| \cdot \text{max}C \cdot (|\mathcal{D}|+|\mathcal{K}|+|\mathcal{N}| \cdot \text{max}S)$, where maxC and maxS are the maximum number of channels in a virtual link and slots in a node, respectively. The size of the constraint set can be approximated by $|\mathcal{F}| \cdot |\mathcal{E}| \cdot |\mathcal{D}| \cdot \text{max}C$. For example, taking into account the instances presented in Section 5, the problem size raises to 10^{10} variables and 10^9 constraints, thereby making impractical its exact solution. Owing to this fact, in the next section, heuristic methods are proposed to provide near-optimal solutions with reasonable computational effort.

4. SIMNO meta-heuristic resolution methods

4.1. A GRASP with PR heuristic

The GRASP procedure is an iterative two phase meta-heuristic method based on a multi-start randomized search technique with a proven effectiveness in solving hard combinatorial optimization problems. It was first presented in [15,16], by Feo and Resende, and later formalized and given its acronym in [17], by Feo et al. Since then, it has been used to solve a wide range of problems (see e.g., [18–20]) with many and varied applications in the real life such as the design of communication networks, collection and delivery operations and computational biology. For recent and comprehensive surveys of GRASP we refer the reader to [21–24].

In the first phase of the multi-start GRASP procedure, a greedy randomized feasible solution of the problem is built by means of a *construction* procedure. Then, in the second phase, a *local search* technique to explore an appropriately defined neighborhood is applied in an attempt to improve the current solution. These two phases are repeated until a stopping criterion is met, and once the procedure finishes the best solution found over all GRASP iterations is returned. Note that with the basic GRASP methodology, iterations are independent from each other as previous solutions of the algorithm do not have any influence on the current iteration. One approach to include memory in the GRASP procedure is with PR, a method which was first introduced by Glover in [25], as an strategy to integrate both intensification and diversification in the context of tabu search [26]. This approach generates new solutions by exploring the trajectories connecting high-quality solutions. The path evaluated starts at a so-called *initiating* solution and moves towards a so-called *guiding* solution which is usually taken from an stored set of good quality solutions called the *elite set*.

PR was first applied in the context of GRASP by Laguna and Martí in [9], and widely applied ever since. Resende and Ribeiro present a wide variety of examples and applications of GRASP+PR in [10]. After a solution is output from the multi-start phase (i.e., construction plus local search), PR is applied between the current solution and a selected solution from the elite set. Then, the best solution found in this iteration is candidate for inclusion in the elite set and it is only added if a certain quality and diversity criteria is met. In this work, we make use of a GRASP+PR heuristic to solve the SIMNO problem. In the next subsections, the fundamental blocks and considerations of our heuristic are presented.

4.1.1. Construction procedure

Given the fact that our problem primarily consists in routing, one-by-one, a set of demands over a virtual topology, the value of the cost function, $g(\cdot)$, for any constructed solution, strictly depends on the selected set of virtual MPLS routes, $\mathcal{R} = \{r_{d_1}, \ldots, r_{d_i}, \ldots, r_{d_j}, \ldots, r_{d_{|\mathcal{D}|}}\}$, to be followed by each demand $d \in \mathcal{D}$. Note, however, that the selection of these routes is, for its part, strongly dependent on the ordering in which these demands are processed (i.e., ordering $\mathcal{O}_x = \{d_1, \ldots, d_i, \ldots, d_j, \ldots, d_{|\mathcal{D}|}\}$). Indeed, such ordering does have strong influence on resources utilization.

Let us first denote $\mathcal{C}_d$ as a set of pre-computed virtual routes available for every demand $d \in \mathcal{D}$. Then, in order to build a solution, we rely on a *restricted candidate list* (RCL) containing the demands $d \in \mathcal{D}$ with the best (i.e., smallest) incremental costs $(c(d))$, that is, RCL$_d$. To compute the incremental cost $c(d)$ for each demand $d \in \mathcal{D}$, we first evaluate the incremental cost of the virtual routes available in $\mathcal{C}_d, d \in \mathcal{D}$, and then, $c(d)$ is given the cost of the less expensive route (i.e., $c(d) = \min_{r \in \mathcal{C}_d}\{c(r)\}$). RCL$_d$ is associated with a threshold parameter in the real interval [0,1]: α. Hence, RCL$_d$ is dynamically formed by all elements (i.e., demands) which can be inserted into the partial solution ensuring its feasibility and whose incremental cost falls within the interval defined by the threshold parameter (see Procedure 1). However, after carrying out a number of tests, we realized that Procedure 1 becomes a really time-consuming process if real-sized, complex problem instances are considered (see Section 5.1). Note that to generate RCL$_d$, the cost $c(r)$ for all routes in $\mathcal{C}_d, d \in \mathcal{D}$ must be recomputed at each iteration of the *while* loop (see lines 5–16 in Procedure 1). In order to minimize this problem, we include an additional parameter (τ) which determines the maximum number of demands that can be evaluated. Hence, at each iteration, a maximum of

$\tau \cdot |\mathcal{Q}|$ candidate demands are randomly selected from set $\mathcal{Q}$. As shown in Procedure 1, once the demand to be served is obtained (and added to the ordering vector), we select the route r_d with the minimum incremental cost to fill the set of selected routes $\mathcal{R}$. Here it is worth noting that the selection of r_d could also have been made by means of a second RCL, in this case, however, containing the routes with the smallest incremental costs, and controlled by another threshold parameter β. In fact, in our preliminary experimentations we found that values of $\beta > 0$ always led to worst performance results (see Section 5.2 for further details), and hence, we do not consider this second RCL in our construction algorithm. Eventually, once the *while* loop ends, both the ordering $\mathcal{O}_x$ and the set of routes $\mathcal{R}$ for all demands are obtained. Note that to calculate $c(d)$ and build RCL_d we take into account the current state of the network (i.e., the resources already reserved by previous demands). Moreover, if a route $r_d \in \mathcal{C}_d$ results in an unfeasible solution, its cost $c(r)$ is set to ∞, thereby avoiding its selection. Hence, at this point, a feasible solution for the network dimensioning without considering failures is obtained. The above-mentioned, is shown between lines 1 and 17 in the pseudo-code of our greedy randomized construction (GRC) algorithm in Procedure 1. The routing of demands is mainly performed over a virtual topology which is precomputed beforehand over the given optical network topology. Virtual links are created between every pair of metro and transit, transit and transit, and transit to interconnection IP/MPLS nodes satisfying that its distance is lower than a given threshold. For each virtual link, a set of routes over the optical network are computed: the shortest one and a number of restoration routes. In order to obtain $\mathcal{C}_d$ for each demand $d \in \mathcal{D}$, we consider a k-shortest path algorithm. In fact, two subsets of routes are pre-computed, one over the virtual topology and another one over the optical topology, thus enabling optical by-passing. Route pre-computation is performed just once at the heuristic startup.

Procedure 1. Greedy randomized construction heuristic.

```
INPUT: $\mathcal{D},\mathcal{C}_d \forall d \in \mathcal{D},\alpha,\tau$
OUTPUT: $\mathcal{O}_x,\mathcal{R},g(\mathcal{O}_x,\mathcal{R})$
1: $\mathcal{R} \leftarrow \emptyset, \mathcal{O}_x \leftarrow \emptyset$
2: Initialize the candidate set: $\mathcal{Q} \leftarrow \mathcal{D}$
3: Initialize the restricted candidate set: $\mathcal{Y}$ with $\tau \cdot |\mathcal{Q}|$ demands randomly selected from $\mathcal{Q}$
4: Evaluate the incremental cost $c(d)$ for all $d \in \mathcal{Y}$
5: While $\mathcal{Q} \neq \emptyset$ do
6:    $c^{min} \leftarrow min\{c(d) | d \in \mathcal{Y}\}$
7:    $c^{max} \leftarrow max\{c(d) | d \in \mathcal{Y}\}$
8:    $RCL_d \leftarrow \{d \in \mathcal{Y} | c(d) \leq c^{min} + \alpha(c^{max} - c^{min})\}$
9:    Select an element $d$ from $RCL_d$ at random
10:     $\mathcal{O}_x \leftarrow \mathcal{O}_x \cup \{d\}$
11:     Take route $r_d \in \mathcal{C}_d$ such that $c(r_d) = c(d)$, and route $d$ through $r_d$
12:     $\mathcal{R} \leftarrow \mathcal{R} \cup \{r_d\}$
13:    Update the candidate set $\mathcal{Q}$
14:    $\mathcal{Y} \leftarrow$ a maximum of $\tau \cdot |\mathcal{Q}|$ demands randomly selected from $\mathcal{Q}$
15:    Reevaluate the incremental cost $c(d)$ for all $d \in \mathcal{Y}$
16: end while
17: Dimension the network
18: Let $\mathcal{A}_{pf}$ denote the set of affected paths under failure scenario $f$
19: for all failure scenario $f \in \mathcal{F}$ do
20:    $\mathcal{A}_{pf} \leftarrow \emptyset$
21:    $\mathcal{A}_{pf} \leftarrow$ GenerateFailure($f$)
22:    if $\mathcal{A}_{pf} == \emptyset$ then
23:       Recover from failure $f$
24:    else
25:       Reroute ($\mathcal{A}_{pf}$)
26:       Increment IP/MPLS nodes capacity
27:       Recover from failure $f$
28:    end if
29: end for
```

Due to the fact that network components such as optical links, OE ports, and IP/MPLS nodes are subject to failures, we build a set of simple failure scenarios where one component fails in each one. Then, for each failure scenario, we remove the element in failure from the network and compute the list of affected MPLS LSPs being each path subsequently rerouted. If additional OE ports need to be installed in the IP/MPLS nodes (i.e., over-dimensioning), checks are performed to ensure the feasibility of the solution. This process is illustrated between lines 18 and 29 in Procedure 1.

As it has been previously explained, in the event of an optical link failure, lightpath restoration is tried as a first option by means of the predefined set of restoration routes. If this restoration succeeds, the associated virtual link (and thus every MPLS LSP using it) is automatically restored. On the contrary, MPLS LSPs are rerouted over the new virtual topology, thereby likely increasing both IP/MPLS nodes switching capabilities and installed OE ports.

Therefore, a feasible solution must provide us with the set of virtual routes that are to be used to carry the amount of traffic $b_d, \forall d \in \mathcal{D}$ as well as with the required over-dimensioning of IP/MPLS nodes so that network survivability is guaranteed. Hence, once a set of routes $\mathcal{R}$ is obtained, cost function $g(\mathcal{O}_x,\mathcal{R})$ accounts for the CAPEX investments required to serve all traffic demands and to guarantee network recovery in front of any of the considered failures. Finally, and for the sake of clarity, hereinafter in this paper we skip the set of routes $\mathcal{R}$ from the parameters in cost function $g(\cdot)$. Note that once the order $\mathcal{O}_x$ for serving the demands is specified, the selection of routes is a pure greedy process.

4.1.2. Local search

Recalling that a solution to our problem can be defined by $\mathcal{O}_x$ (i.e., the ordering in which the demands are to be served), and for the purpose of neighborhood creation, we refer to a feasible solution obtained by Procedure 1 as $\mathcal{O}_x$. Due to the fact that a feasible solution $\mathcal{O}_x$ has no guarantee of being locally optimal, GRASP heuristics apply a local search procedure starting at $\mathcal{O}_x$ in the hope of finding a better solution in its neighborhood. Then, let us denote with $N_q(\mathcal{O}_x)$, the set of solutions in the *qth* neighborhood structure of $\mathcal{O}_x$. Thus, assuming an ordering of $|\mathcal{D}|$ traffic demands, $\mathcal{O}_x = \{d_1, \ldots, d_i, \ldots, d_j, \ldots, d_{|\mathcal{D}|}\}$, we define the neighbor of this ordering as an ordering in which d_i is interchanged with d_j. Let us denote such interchange operation in $\mathcal{O}_x$ as $I(d_i,d_j)_{\mathcal{O}_x}$. In order to generate a random neighbor in the first neighborhood (i.e., a 1-move neighbor) of $\mathcal{O}_x$ (i.e., $N_1(\mathcal{O}_x)$), we choose pivots d_i and d_j uniformly among the $|\mathcal{D}|$ demands. Hence, creating a q-move neighbor implies that this random interchange of demands is performed q times, though always ensuring that an interchange of the randomly selected pivots will bring the solution a neighborhood further.

Several approaches have been proposed in the literature to perform local search. Among them, we find techniques such as the *variable neighborhood search* (VNS) and *variable neighborhood descent* (VND), and the *approximate local search* (ALS) procedures (see [27,28]). In this work, we make use the ALS procedure to implement the local search in the GRASP multi-start phase. ALS

Please cite this article as: Pedrola O, et al. A GRASP with path-relinking heuristic for the survivable IP/MPLS-over-WSON multi-layer network optimization problem. Computers and Operations Research (2011), doi:10.1016/j.cor.2011.10.026

was first proposed in [28] as a trade-off between the *first-fit* and *best-fit* approaches within the N_1 and N_2 neighborhoods of a solution. As shown in the pseudo-code of Procedure 2, this technique randomly samples the 1-move and 2-move neighborhoods of $\mathcal{O}_x$. This exploration is stopped when either the set of improving solutions CS is full or a maximum of $MaxSearch$ neighbors have been explored. Then, the algorithm selects either in a greedy or a probabilistic fashion one of the solutions in CS to continue the exploration. In [28], the greedy selection outperformed the probabilistic one, and thus, in this paper we consider the greedy choice to select a solution from CS as well as an equal probability to generate a 1-move or a 2-move neighbor. The algorithm finishes when set CS is empty and returns as output the best solution found $\mathcal{O}_B$.

Procedure 2. Approximate local search (ALS) heuristic.

```
INPUT: $\mathcal{O}_x$,$MaxCS$,$MaxSearch$
OUTPUT: $\mathcal{O}_B$
$\mathcal{O}_B \leftarrow \mathcal{O}_x$;
repeat
    $i \leftarrow 0, CS \leftarrow \emptyset$;
    repeat
        $\mathcal{O}_{x'} \leftarrow$ Generate-1-or-2-move-neighbor ($\mathcal{O}_B$);
        if $g(\mathcal{O}_{x'}) < g(\mathcal{O}_B)$
            $CS \leftarrow CS \cup \{\mathcal{O}_{x'}\}$;
        end if
        $i \leftarrow i+1$;
    until $|CS| \geq MaxCS$ or $i \geq MaxSearch$
    if $CS \neq \emptyset$
        Select $\mathcal{O}_x = \min_{\mathcal{O}_k \in CS}\{g(\mathcal{O}_k)\}$;
        $\mathcal{O}_B \leftarrow \mathcal{O}_x$;
    end if
until $CS = \emptyset$
```

4.1.3. Path-relinking

As mentioned before, PR is an intensification strategy which generates new solutions by exploring the trajectories linking two high-quality solutions (starting at an initiating solution towards the guiding one). The path connecting both solutions is generated by sequentially introducing attributes of the guiding solution into the initiating one. To ensure that PR is only applied among high-quality solutions, a set of elite solutions (ES) must be both maintained and cleverly managed during all GRASP iterations. Note that with the attribute high-quality we are not only referring to their cost function value but also to the diversity they add to ES.

PR implementation: Several approaches on how to perform PR have been proposed and evaluated (see e.g., [29]). These techniques mainly deal with the process that is in charge of creating the path towards the guiding solution. The most usual approach consists in building the path in a greedy fashion (i.e., the most profitable or least costly move is selected). However, in this work, we have developed a specific strategy to perform PR. Two main reasons support this modeling decision. First, evaluating the cost of each possible move towards the guiding solution would entail extremely long computation times, and second, and most compelling, is the fact that in our problem instances, hundreds of demands are to be served (see Section 5.1), and therefore, the path connecting two high-quality solutions may easily have hundreds of moves. Thus, the use of PR would be inadvisable since it would require most of the time available, thereby drastically reducing the number of iterations performed.

Let $\mathcal{O}_1 = \{d_1, \ldots, d_{|\mathcal{D}|}\}, \mathcal{O}_2 = \{d'_1, \ldots, d_{|\mathcal{D}|}{}'\}$ be two feasible solutions interpreted as vectors (i.e., $\mathcal{O}_1(1) = d_1$, and $\mathcal{O}_2(1) = d'_1$). For the sake of this example, let us define $\mathcal{O}_1$ as the initiating solution ($\mathcal{O}_{INIT}$), and $\mathcal{O}_2$ as the guiding one ($\mathcal{O}_{GUID}$). Then, let us also denote a move from $\mathcal{O}_{INIT}$ to $\mathcal{O}_{GUID}$ as,

$$move(i)_{\mathcal{O}_{INIT}} = I(\mathcal{O}_{INIT}(i), \mathcal{O}_{GUID}(i))_{\mathcal{O}_{INIT}},$$

that is, an interchange of demand positions applied to ordering $\mathcal{O}_{INIT}$. Note that in the case that $\mathcal{O}_{INIT}(i) = \mathcal{O}_{GUID}(i)$ no move is performed. Thus, given $\mathcal{O}_{INIT}$ and $\mathcal{O}_{GUID}$, we build the path by progressively transforming $\mathcal{O}_{INIT}$ into $\mathcal{O}_{GUID}$ (i.e., by iteratively applying $move(i), i = 1, \ldots |\mathcal{D}|$). However, as aforementioned, the size of our problem instances is really high, thus making impractical the evaluation of each solution found along the path created by PR. Hence, we propose to sample the path every T moves in the search for an improving solution, and if found, a thorough evaluation of the nearby solutions is carried out. The value of T is defined by an input parameter N_{SAMPLE} that decides into how many regions the path between both solutions must be divided. Fig. 3 illustrates this method by showing the path being evaluated between two high quality solutions $\mathcal{O}_{INIT}$ and $\mathcal{O}_{GUID}$. We uniformly sample the path built and when $\mathcal{O}_{GUID}$ is reached the best solution found during the sampling process ($\mathcal{O}_{BS}$) is selected.

If $g(\mathcal{O}_{BS}) < \min(g(\mathcal{O}_{INIT}), g(\mathcal{O}_{GUID}))$ a move to the right and to the left of $\mathcal{O}_{BS}$ is assessed (see dotted arrows in Fig. 3). Then, we take the improving direction and iteratively evaluate the subsequent moves until no improvement is found. PR then returns the best solution found during this intensification step ($\mathcal{O}_{BEST}$). In this way, we have a relatively high probability of reaching the best solution in the path connecting $\mathcal{O}_{INIT}$ and $\mathcal{O}_{GUID}$. The pseudo-code for our PR implementation is illustrated in Procedure 3.

Procedure 3. Path-relinking heuristic.

```
INPUT: $\mathcal{O}_{INIT}$,$\mathcal{O}_{GUID}$,$N_{SAMPLE}$
OUTPUT: $\mathcal{O}_{BEST}$
$M \leftarrow$ number of moves from $\mathcal{O}_{INIT}$ to $\mathcal{O}_{GUID}$
$T \leftarrow \left\lfloor \frac{M}{N_{SAMPLE}} \right\rfloor$
$count \leftarrow 1, \mathcal{O}_{BEST} \leftarrow \emptyset, \mathcal{S} \leftarrow \emptyset$
$\mathcal{O}_x \leftarrow \mathcal{O}_{INIT}$
for $i \leftarrow 1, M$ do
    $\mathcal{O}_x \leftarrow move(i)_{\mathcal{O}_x}$
    if $count == T$ then
        if $\mathcal{O}_x$ is feasible then
            $\mathcal{S} \leftarrow \mathcal{S} \cup \{\mathcal{O}_x\}$
        end if
        $count \leftarrow 0$
    end if
    $count \leftarrow count + 1$
end for
Select ordering $\mathcal{O}_{BS} \in \mathcal{S}$ which minimizes cost function $g(\cdot)$
if $g(\mathcal{O}_{BS}) < min(g(\mathcal{O}_{INIT}), g(\mathcal{O}_{GUID}))$ then
    Evaluate a move to the right and to the left of $\mathcal{O}_{BS}$
```

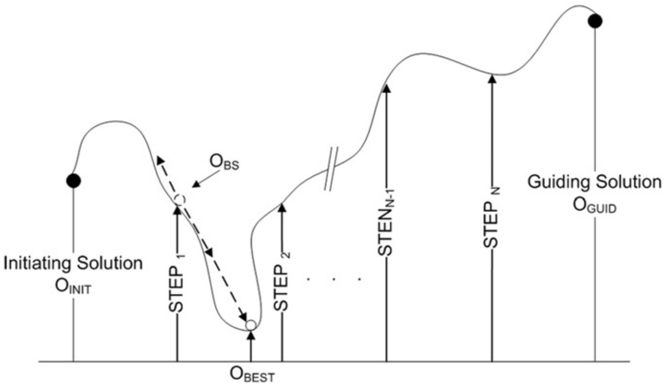

Fig. 3. Path-relinking heuristic implementation.

Please cite this article as: Pedrola O, et al. A GRASP with path-relinking heuristic for the survivable IP/MPLS-over-WSON multi-layer network optimization problem. Computers and Operations Research (2011), doi:10.1016/j.cor.2011.10.026

```
    Take the improving direction and iteratively move until
    no improvement is found
    Return the best feasible solution found $\mathcal{O}_{BEST}$
end if
```

Elite set management and distance measure: Since the PR algorithm operates on *ES*, its management and maintenance is, therefore, crucial to the success of the PR procedure. Previous studies such as [30], have shown that a policy to include solutions in *ES* only based on their individual quality does not lead to the best PR performance. Hence, to include a new solution in *ES*, a trade-off between quality and diversity is usually evaluated (see e.g., [31]).

Initially, *ES* is empty, then, each locally optimal solution obtained and each solution resulting from a PR execution is candidate to be inserted in *ES*. Let us consider $\mathcal{O}_x$ as such candidate solution. If *ES* is not yet full, then, $\mathcal{O}_x$ is simply added to *ES*. Otherwise, if $\mathcal{O}_x$ improves the best solution in *ES*, it replaces an element of the set. In addition, if $\mathcal{O}_x$ improves upon the worst in *ES* and its distance to *ES* is larger than a pre-established threshold δ_{th}, it also replaces an element in *ES*. To this end, let us define $\delta_{x,y}$ as the distance between two solutions $\mathcal{O}_x$ and $\mathcal{O}_y$ (i.e., the number of moves required to reach $\mathcal{O}_y$ from $\mathcal{O}_x$). Then, the distance between a solution $\mathcal{O}_x$ and the whole *ES* can be defined as,

$$\delta_{x,ES} = \min_{O_i \in ES}\{\delta_{x,i}\}.$$

Hence, when *ES* is full, $\mathcal{O}_x$ is inserted in *ES* if its quality is superior to the worst in *ES* and $\delta_{x,ES} \geq \delta_{th}$. This threshold is empirically adjusted in Section 5. With the same diversity objective, and in order to maintain the size of the pool constant, whenever we add a solution to *ES*, another one must be removed. As usual, we remove the closest solution to $\mathcal{O}_x$, which we call $\mathcal{O}_r$, among those with a worse quality. Thus, $\mathcal{O}_r$ can be defined as follows:

$$\mathcal{O}_r = \min_{O_i \in ES: g(\mathcal{O}_i) > g(\mathcal{O}_x)}\{\delta_{x,i}\}.$$

Selection policy: Another important aspect regarding PR is that once a solution $\mathcal{O}_x$ is output from the multi-start phase, another solution $\mathcal{O}_i$ must be selected from *ES* to be path-relinked with $\mathcal{O}_x$. In the literature, a common approach is to select a solution randomly from *ES* [21]. However, this may result in selections that are very close to $\mathcal{O}_x$, thereby reducing the probability of finding better solutions. In an attempt to minimize this issue, we adopt a biased [30] approach in which solutions are selected with probabilities proportional to their distance to $\mathcal{O}_x$. Therefore, the probability p_i of selecting a particular solution $\mathcal{O}_i \in ES$ can be computed as follows:

$$p_i = \frac{\delta_{x,i}}{\sum_{j=1}^{|ES|}\delta_{x,j}}.$$

In order to perform PR, we implement the *back-and-forward* (PR$_{bf}$) strategy, which explores the path in both directions (see e.g., [30]). Once the PR finishes, if no improving solution is found, the best of both extremes is returned as output. Finally, the pseudocode of our GRASP+PR heuristic is shown in Procedure 4, which first executes the GRASP multi-start phase to fill *ES*, and then runs a pre-defined number of GRASP+PR iterations. Procedure 4 returns as output the best solution stored in *ES*. We point out that all input parameters required to call the construction, local search and PR methods will be adjusted in Section 5.

Procedure 4. GRASP+PR heuristic.

```
INPUT: $GlobalMaxItr, \mathcal{D}, \mathcal{C}_d \forall d \in \mathcal{D}, \alpha, \tau, MaxCS, MaxSearch, N_{SAMPLE}$
OUTPUT: $\mathcal{O}_{BEST}$
$\mathcal{O}_{BEST} \leftarrow \emptyset, ES \leftarrow \emptyset$
Apply GRASP (GRC followed by ALS) for $b = |ES|$ iterations
to populate $ES$
$count \leftarrow 1$
repeat
    $\mathcal{O}_x \leftarrow \mathrm{GRC}(\mathcal{D}, \mathcal{C}_d \forall d \in \mathcal{D}, \alpha, \tau)$
    $\mathcal{O}_{x'} \leftarrow \mathrm{LocalSearch}(\mathcal{O}_x, MaxCS, MaxSearch)$
    Select elite solution $\mathcal{O}_{EL}$ from $ES$
    $\mathcal{O}_B \leftarrow \mathrm{PR}_{bf}(\mathcal{O}_{EL}, \mathcal{O}_{x'}, N_{SAMPLE})$
    Try to insert $\mathcal{O}_B$ in $ES$
      $count \leftarrow count + 1$
until $count > GlobalMaxItr$
$\mathcal{O}_{BEST} \leftarrow \min_{\mathcal{O}_k \in ES}\{g(\mathcal{O}_k)\}$
```

4.2. A BRKGA heuristic

Among meta-heuristics, BRKGAs have recently been proposed to effectively solve optimization problems. For example, BRKGAs have been applied to network related problems such as routing in IP networks and RWA in optical networks [32,33]. Compared with other meta-heuristics, BRKGA is characterized by being able to provide high quality solutions in very short running times. In this Section, we apply the BRKGA meta-heuristic to solve the SIMNO problem.

BRKGA is a class of GA where each individual is represented as an array of n_g *genes*, called *chromosome*, and each gene can take a value, called an *allele*, in the real interval [0,1]. Each chromosome encodes a solution of the problem and a *fitness* level, that is, the objective function value. Identical to GA, a set of *p* individuals, called a *population*, evolves over a number of generations. At each generation, individuals of the current generation are selected to mate and produce offspring, making up the next generation. In BRKGA, individuals of the population are classified into two sets: the elite set p_e, with those individuals with the best fitness values, and the non-elite set. Elite individuals are copied unchanged from one generation to the next, thus keeping track of good solutions. The majority of new individuals are generated by crossover, that is, by combining two elements, one elite and another non-elite, selected at random. An inheritance probability (ρ_e) is defined as the probability that an offspring inherits the allele of its elite parent. Finally, to escape from local optima a small number of mutant individuals (p_m, randomly generated) are introduced at each generation to complete a population. A deterministic algorithm, named *decoder*, transforms any input chromosome into a feasible solution of the optimization problem and computes its fitness value. In the BRKGA framework, the only problem-dependent parts are the chromosome internal structure and the decoder, and thus, one only needs to define them to completely specify a BRKGA heuristic.

Similarly to Section 4.1, the problem primarily consists in routing a set of demands over a virtual topology. In this case, we make use of one gene per virtual link and per IP/MPLS node. These genes are used to compute a metric for each element in order to perform the routing of each demand $d \in \mathcal{D}$. Besides, and recalling that the order in which the demands are served influences the goodness of the solution, additional genes are required to specify it. For this purpose, we use one additional gene per demand $d \in \mathcal{D}$. Therefore, given a virtual network represented by graph $\mathcal{G}(\mathcal{N}, \mathcal{E})$ and the set of demands $\mathcal{D}$, each individual is represented by an array of $|\mathcal{N}| + |\mathcal{E}| + |\mathcal{D}|$ genes.

Here it is worth noticing that both BRKGA and GRASP+PR (see Section 4.1) have the same goal (minimize network CAPEX) and that this is achieved both by minimizing routing costs (i.e., using the cheapest links and nodes), and by grooming the demands so as to minimize the use of resources. On the one hand, BRKGA uses the metrics and the ordering encoded in the chromosome. Metrics are used as a means to stimulate or penalize the use of individual links and nodes so that those resources minimizing the cost of the

Please cite this article as: Pedrola O, et al. A GRASP with path-relinking heuristic for the survivable IP/MPLS-over-WSON multi-layer network optimization problem. Computers and Operations Research (2011), doi:10.1016/j.cor.2011.10.026

network are selected. Ordering, however, is used to improve the grooming of demands, thus making the most of the network resources. On the other hand, GRASP+PR relies on the ordering of demands not only to improve grooming, as in BRKGA, but to minimize the cost of the network too. Since the GRASP construction algorithm deals directly with CAPEX incremental costs, its complexity is greater than that required to decode a chromosome in BRKGA, however, this comes at the benefit of solution quality. Finally, note that fast cost function evaluations are crucial to a BRKGA algorithm, and so the differences among both heuristics when it comes to solution encoding.

To decode chromosomes into feasible solutions, the metric of IP/MPLS nodes and virtual links is initialized using the assigned gene of the input chromosome, and the order in which each demand will be routed is given by the rest of genes. After initializing every element, the network is dimensioned through the routing of the whole set of demands $\mathcal{D}$. A solution to the network dimensioning without considering failures is obtained at this step. To include failures, we use the steps already illustrated in the GRASP construction algorithm (i.e., between lines 18 and 29 in Procedure 1 in Section 4.1.1).

Additionally, in this work, a multi-population strategy where a number of populations are evolved independently has been implemented [34]. The algorithm was designed and implemented as a multi-thread application, where each population runs in a single thread. Populations exchange elite individuals after a pre-determined number of generations. In an initial phase, a data structure representing the network graph is created. At this step, the network graph only contains IP/MPLS and optical nodes and optical links. Afterwards, the virtual topology is generated; virtual links between metro and transit and between transit and transit IP/MPLS nodes are created. Demands pre-routing computation is then performed. To be precise, a set of $k=100$ routes is pre-computed for each demand. During the decoder process, route metric re-computation is performed ensuring that the shortest route (in terms of that metric) is chosen at each step. The parameters considered for the BRKGA algorithm are provided in the next section.

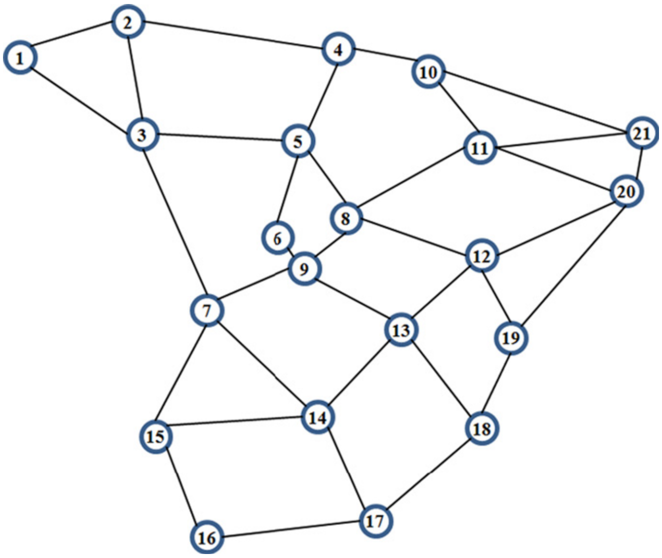

Fig. 4. A realistic Spanish optical core transport network topology.

5. Computational experiments

This section describes the computational experiments carried out to both evaluate and compare the efficiency and performance of the GRASP+PR and BRKGA heuristics proposed in this paper to solve the SIMNO problem. All methodologies have been implemented in Java SE 1.6.0_17 using a sequential approach (though we consider parallel populations in BRKGA), and all experiments have been conducted on Intel Core 2 Quad 2.67 GHz based computers running Windows 7 Professional Edition (64 bits) with 8 GB of RAM.

5.1. Problem instances

The performance of the proposed meta-heuristic algorithms has been compared over the realistic 21-node Spanish national optical network topology shown in Fig. 4. In order to have a representative range of multi-layer networks, we have considered three different IP/MPLS topologies which consist of 40 metro nodes and a different number of transit and interconnection nodes. Table 1 specifies the location of transit and interconnection nodes (identified by the associated OXC location) of each multi-layer network. Moreover, for each multi-layer network, the spatial position of metro nodes is characterized by a uniform coverage degree (CD) based on the p-value of the uniformity Kolmogorov–Smirnov test [35]. Note that whilst values close to 100% indicate that metro nodes are uniformly located on a 2D map, low values denote the presence of areas with high density of metro nodes. Table 1 also contains the CD of the three network instances under study. For the traffic, we assume two types of demands: *national* where both end metro nodes belong to the network, and *interconnection*, where one of the end metro nodes is outside of the network (i.e., either the source or the destination node of the demand is the virtual metro node as defined in Section 3). The mix of national and interconnection traffic is also detailed in Table 1. Therefore, three different multi-layer network scenarios can be identified, from the unbalanced network A, where 70% of the total is interconnection traffic with only three interconnection nodes and several high density metro areas, to the well-balanced network C, with 50% of interconnection traffic, five interconnection nodes and nearly uniform metro areas. Network B is in between of networks A and C. In fact, a brief analysis of the proposed instances identifies differences on the complexity of the problems. For instance, note that the size of virtual topology is 326, 361, and 408 virtual links for networks 1, 2, and 3, respectively. Thus, the mean number of feasible routes for a given demand significantly increases from network A to network C, and consequently differences in the results can be anticipated for each network instance.

Table 1
Network topologies and traffic parameters considered.

Network	**Transit**	**Interconnection**	**Metro** **CD** (%)	**Traffic mix** national/ Interconnection (%)
A	3, 4, 9, 11, 14, 15, 19, 21	6, 8, 20	0.1	30/70
B	1, 9, 10, 12, 14, 16, 20	7, 13, 15, 19	30	40/60
C	3, 4, 5, 8, 9, 14, 19, 21	6, 7, 10, 13, 20	90	50/50

Each multi-layer network has been planned taking into consideration several increasing traffic loads, starting from an initial load of 4 Gbps per metro node and with increments of 45% at each step (roughly representing a year-over-year traffic increase). However, since the complexity of the problems strongly depends on the number of demands to be served, this number should not be increased sharply. Instead, the average requested bandwidth in each demand is increased at each step. Aiming at providing accuracy, each traffic load has been executed three times with randomly generated demands following the above characteristics. This has resulted in a set of 21 traffic instances for each of the networks, that is, $RA_{1\ldots21}$, $RB_{1\ldots21}$, $RC_{1\ldots21}$, for networks A, B and C, respectively. Each of these sets, in blocks of three, represents the same traffic load but with three independent randomly generated representations. Hence, traffic profiles are represented in seven different blocks in increasing order (e.g., $\mathrm{RA}_{1\ddot{3}}$ and $\mathrm{RA}_{4\ddot{6}}$ belong to blocks 1 and 2, respectively). Note that the higher the index of the block, the higher the complexity of the problem. These traffic instances have a minimum number of 120 demands and a maximum of 360. The bandwidth requested per demand can be either 1, 10, 40 or 100 (Gbps), this last being the minimum amount required to perform optical by-passing. Hence, 100 Gbps demands belong to subset 2 and the rest to subset 1 as defined in Section 3.2. We assume the availability of 80 wavelengths at every optical link in the WSON network, a maximum allowed lightpath length of 1000 km, and that each metro node is connected to every interconnection node and a maximum of 4 transit nodes (the nearest 4 transits). Moreover, we fill set $\mathcal{C}_d$ with a maximum of 100 top shortest paths computed over the virtual topology for each demand $d \in \mathcal{D}$. As mentioned in Section 4.1, a set of k routes at the optical level is also pre-computed. In particular, the shortest-path route plus a restoration route per optical hop (note that a number of hops may share the same restoration route). To compute the k-shortest paths we make use of Yen's algorithm implemented as in [36]. Aiming at accurately computing the network CAPEX, we consider an adaptation of the equipment costs proposed in [37] to provide meaningful values for the parameters in Eqs. (1) and (2). The costs of IP/MPLS nodes and OE ports are provided in Tables 2 and 3, respectively. In addition, we consider a cost per kilometer of restorable lightpath equal to 1 cost unit (c.u.).

5.2. Tuning of GRASP+PR and BRKGA parameters

Recently, in [38], an interesting way to solve the problem of parameter tuning for GRASP+PR heuristics has been proposed. This technique makes use of a BRKGA algorithm to explore the GRASP+PR parameter space. In this case, and for each chromosome, a random-key solution vector encodes the set of GRASP+PR parameters that we aim to tune. Then, to obtain the fitness for each chromosome, a set of V independent runs of the GRASP+PR must be executed, each lasting for U iterations. The fitness is calculated as the average objective function $g(\cdot)$ value found in these V executions.

In our problem, however, given the complexity of the real-sized problems studied (i.e., multi-layer network size and traffic instances), we make use of the automatic tuning only for the parameters used in the multi-start phase of GRASP, that is, those parameters required in Procedures 1 and 2 (GRC and ALS). To perform this study, we consider a different set of 10 traffic instances per network. These instances are generated as described in Section 5.1, with increasing load intensities and with the number of demands limited to 40 so as to reduce complexity. The GRASP parameters that are to be tuned and their respective allowed values are: (i) Construction procedure: $\alpha = \{0.0,0.1,0.2,0.3,0.4,0.5\}$, $\tau = \{0.1,0.2,0.3,0.4,0.5\}$, $\beta = \{0.0,0.1,0.2\}$ (recall from Section 4.1.1 that although β is not shown in Procedure 1, it represents the threshold parameter for a hypothetical second RCL used to select the route for each demand). (ii) Local search: $MaxCLS = \{5,10,20\}$, $MaxSearch = \{10,20,40\}$. Our chromosome is therefore defined by these five parameters of the GRASP multi-start phase. In contrast to the BRKGA defined in Section 4.2, here BRKGA does not make use of parallel populations. In Table 4, we provide the set of fixed BRKGA parameters that will be used by both BRKGAs (i.e., automatic tuning of GRASP parameters and the resolution of SIMNO). In addition, to define the BRKGA for the automatic tuning of GRASP parameters, we consider a population size equal to $p=20$. The process is run for 10 generations. To obtain the fitness of each chromosome we perform $V=10$ independent GRASP (GRC+ALS) executions with the time limit set to 2 h. BRKGA tuning is applied to each of the networks (using the 10 different traffic instances), thus resulting in a specific combination of parameters for each network. Table 5 reports, for each parameter and network, the values with higher frequencies of occurrence among the 10 traffic instances. It is worth highlighting that the automatic tuning always reports a value of β equal to 0, thereby eliminating the need for using an additional RCL to manage the selection of routes.

Next, we focus on the tuning of the parameters required to specify the PR method, namely the minimum distance to enter ES (δ_{th}) and the sampling parameter N_{SAMPLE}. In this work, we consider an elite set size ($|ES|$) equal to 6. Hereinafter in this paper, and in order to quantitatively evaluate and compare the results of each experiment, we provide the performance metrics proposed in [29]. Specifically, we provide the number of times (*#Best*) that each method is able to obtain the overall best solution value (*BestVal*) found among all methods being tested. Moreover, for each method, we compute the relative percentage deviation ($Dev(\%)$) between the best solution value obtained by that

Table 2
IP/MPLS nodes features and costs.

Nodes	Class 1	Class 2	Class 3	Class 4	Class 5
Aggregated switching capacity (Gbps)	160	320	640	1280	2560
Max. number of ports	4	8	16	32	64
Cost (c.u.)	3	4.5	6.5	22.5	50.19

Table 3
OE ports features and costs.

OE ports	1 Gb	10 Gb	40 Gb	100 Gb
Cost (c.u.)	0.45	1.5	8.125	24.625

Table 4
Fixed BRKGA parameter values.

p_e	p_m	ρ_e
0.2	0.2	0.7

Table 5
GRASP automatically tuned parameters.

Network	α	τ	β	*MaxCS*	*MaxSearch*
A	0.4	0.1	0.0	5	20
B	0.2	0.5	0.0	5	20
C	0.2	0.2	0.0	5	20

particular method and *BestVal* for that instance. Finally, we report the statistic called *Score* [29,39]. In short, the *Score* parameter counts, for a particular method M_x and for each problem instance, the number of methods that are able to find better solutions than M_x. Hence, the lower the *Score*, the better the method.

In this experiment, we consider four different traffic instances per network, though this time with the number of demands limited to 80. We increase the number of demands so as to obtain more accurate values to execute GRASP+PR with the real-sized traffic instances described in Section 5.1. Since the maximum distance between two solutions depends on the size of the demands set $\mathcal{D}$, we evaluate percentages of this figure as possible δ_{th} values. Moreover, we also test the impact of four different values for N_{SAMPLE}, thus resulting in 16 different parameter combinations for PR. For each traffic instance, we run 10 independent executions with the time limit set to 4 h. The results provided in Table 6 clearly report that the best values for δth and N_{SAMPLE} are $0.1 \cdot |\mathcal{D}|$ and 10, respectively. Indeed, these values lead to results for the three statistics considered which compare favorably with the other values tested.

Finally, to specify the parameters of the BRKGA developed to solve SIMNO, we decided to perform a manual tuning. To this end, we conducted a set of preliminary experiments using several traffic instances for each of the networks evaluated, and took (after testing several combinations) the combination of parameters, that is, (p, p_e, p_m, ρ_e, n_p, i_e), that in average led to the best solutions in all scenarios. The manually tuned parameter values found are those shown in Table 4 as well as a number $n_p = 3$ of parallel populations, an inter-population elite exchange $i_e = 2$ and a chromosome length as described in Section 4.2. Here, it is worth highlighting that we use a reduced population size (p=20). As a consequence of the size of the problems, the length of the chromosome was higher than 300 genes and the decoder algorithm took more than 50 ms to build a single solution from a chromosome, that is, more than 15 s to build one generation when $p = n_g$ was used. Then, the BRKGA heuristic required extremely long times to reach convergence. Reducing the size of the population, the convergence time was reduced to acceptable values. As it has been mentioned, three populations were evolved in parallel and local elite individuals exchange was allowed every 15 generations.

5.3. *BRKGA vs. GRASP vs. GRASP+PR performance comparison*

Having tuned the parameters, we now carry out a performance analysis of the two meta-heuristic models proposed to solve the SIMNO problem, that is, BRKGA and GRASP+PR. Moreover, in order to highlight the benefits of PR, we include in the tests the results obtained by the basic GRASP heuristic (i.e., construction followed by local search). Here it is worth mentioning that the performance of both GRASP+PR and BRKGA was compared against the optimal solution obtained by solving the ILP described in Section 3 over a small multi-layer topology (not shown in this paper). In all the tests conducted, the optimal solution was found within running times of some seconds, in contrast to several hours needed to find the optimal solution using the ILP model.

To evaluate the three different variants, we make use of the 21 traffic instances per network as defined in Section 5.1. For each instance, we run five independent executions with the time limit set to 10 h. The results are reported in Tables 7, 8 and 9, respectively, for networks *A*, *B* and *C*. As it can be observed, basic GRASP outperforms BRKGA in all networks, though more notably in the most complex instances (i.e., networks *B, C*). Note that the performance of BRKGA gradually decreases from network *A* to *C*, with higher complexity resulting in BRKGA finding convergence at very high CAPEX values when compared to both GRASP and GRASP+PR. In fact, in preliminary experiments with smaller problem instances (not shown in this paper), we noticed that BKRGA obtains very good results in very short running times, outperforming GRASP in the trade-off between optimality and complexity. However, in the complex instances considered in this paper, it is very difficult for BRKGA to converge to good quality solutions in short times. To illustrate this behavior, in Fig. 5, we plot the search profile of both BRKGA and GRASP+PR in a 10 h execution using traffic instance RC_{10}. It is easy to observe that due to the complexity of the problem, BRKGA finds it very difficult to converge at good quality CAPEX values, whereas in GRASP+PR early results are already of good quality, thereby showing that the use of GRASP+PR does really pay off when real-sized, complex instances are considered.

Finally, GRASP+PR stands out as the best method providing in all networks the best results for all three metrics considered, a fact which clearly highlights the impact that introducing PR has in the meta-heuristic performance results. In order to graphically

Table 7
Results for traffic instances $RA_{1\ldots21}$.

Method	BRKGA	GRASP	GRASP+PR
Dev (%)	7.79	6.35	2.04
Score	28	25	10
#Best	10	4	15

Table 8
Results for traffic instances $RB_{1\ldots21}$.

Method	BRKGA	GRASP	GRASP+PR
Dev (%)	10.91	4.22	0.88
Score	33	22	8
#Best	6	5	18

Table 9
Results for traffic instances $RC_{1\ldots21}$.

Method	BRKGA	GRASP	GRASP+PR
Dev (%)	22.14	3.12	0.55
Score	42	16	5
#Best	0	9	21

Table 6
PR parameters evaluation.

δ_{th}	$\frac{5}{100}\lvert\mathcal{D}\rvert$				$\frac{10}{100}\lvert\mathcal{D}\rvert$				$\frac{15}{100}\lvert\mathcal{D}\rvert$				$\frac{20}{100}\lvert\mathcal{D}\rvert$			
N_{SAMPLE}	1	10	15	20	1	10	15	20	1	10	15	20	1	10	15	20
Dev (%)	2.6	2.7	2.4	2.7	2.7	**1.8**	2.5	2.7	2.1	2.6	2.8	3	3.4	3.1	2.6	3.1
#Best	0	0	0	0	2	**3**	0	0	3	2	1	1	1	1	0	0
Score	102	86	86	94	77	**55**	96	97	69	91	85	100	105	93	88	101

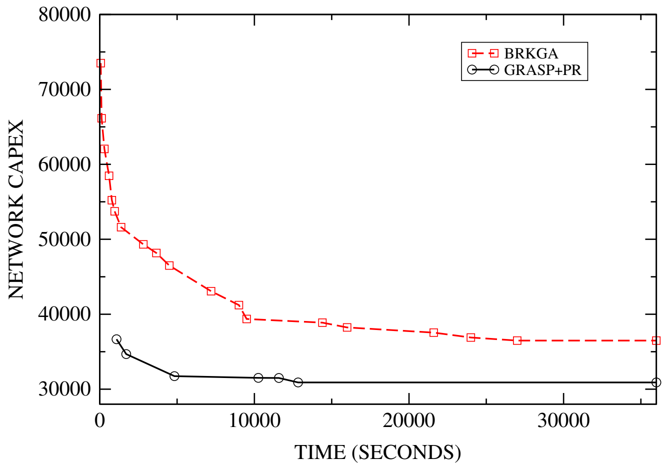

Fig. 5. GRASP+PR vs. BRKGA performance comparison in a 10 h execution (RC_{10}).

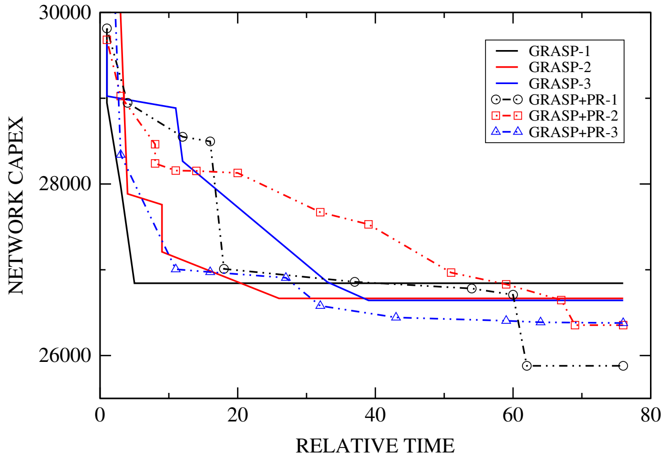

Fig. 6. GRASP vs. GRASP+PR performance comparison using instance RC_5.

illustrate the performance difference between GRASP and GRASP+PR, in Fig. 6, we plot the search profile of three independent runs of both the GRASP and GRASP+PR algorithms considering traffic instance RC_5. Note that in the x-axis times are given in multiples of the average time it takes to perform a basic GRASP iteration (i.e., construction followed by local search), and hence, are shown as relative time units. The results provided claim to show the effectiveness and ability of PR to find regions of the space of solutions that, with the basic GRASP methodology, are highly unlikely to be found. Indeed, in Fig. 6, remarkable differences among the curves displayed by GRASP and GRASP+PR can be observed. Whilst the basic GRASP, after a few initial improvements, presents a rather flat profile, GRASP+PR clearly shows a more successful and thorough exploration of the space of solutions. Therefore, we consider that this study visibly shows to what extent can PR improve the results obtained by a basic GRASP heuristic, and more important, in which problems/scenarios the application of PR is really advisable. In the matter in hand, the application of GRASP+PR will definitely result in significant savings for network operators.

6. Concluding remarks

The objective of this study has been the development of heuristic algorithms aimed at minimizing the CAPEX investments required to plan a survivable IP/MPLS-over-WSON multi-layer network. For this purpose, we proposed a novel multi-layer optimization scheme, and hence, eventually tackled the so-called SIMNO problem. The resolution of this problem is indeed of great interest to network operators. To deal with SIMNO, we have first detailed the multi-layer network architecture under consideration as well as the novel recovery schemes proposed. Then, we have formalized the SIMNO problem by means of an ILP formulation which provided an insight into the complexity of managing the problem in hand. Finally, two powerful meta-heuristic models have been developed to help solve the SIMNO problem within practical running times. To be precise, a BRKGA and a GRASP+PR heuristic have been considered. After performing a set of exhaustive experiments, we have illustrated the difficulty that the BRKGA heuristic has in finding good quality convergence values, particularly when the problem instances are complex. At the same time, we have also shown the efficiency of the GRASP meta-heuristic specifically designed for solving SIMNO, even without the use of PR. However, the main outcome of this study has been the possibility to verify how powerful the PR intensification strategy is. Indeed, GRASP+PR has achieved significant improvements with respect to GRASP, particularly in the more complex network scenarios. In this paper, GRASP+PR has helped to solve a current issue for network operators considering real-sized, complex network and traffic scenarios. Hence, we have illustrated one more time, a successful application of the combined GRASP+PR meta-heuristic.

Acknowledgments

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007–2013 under Grant agreement no. 247674 STRONGEST Project. Moreover, it was supported by the Spanish Ministry of Science through the FPU Program and the DOMINO Project (TEC2010-18522).

References

[1] Architecture for the automatically switched optical network (ASON), ITU-T G.8080 (ITU), 2001.
[2] Mannie E. Generalized multi-protocol label switching (GMPLS) architecture, RFC-3945 (IETF), 2004.
[3] Lee Y, Bernstein G, Imajuku W. Framework for GMPLS and PCE control of wavelength switched optical networks (WSON), IETF draft, draft-ietf-ccamp-rwa-wson-framework-12, February 2011.
[4] Chiu A, Strand J. Joint IP/optical layer restoration after a router failure. In: Proceedings of the IEEE/OSA OFC, 2001.
[5] Chołda P, Jajszczyk A. Recovery and its quality in multilayer networks. IEEE/OSA Journal of Lightwave Technology 2010;28:372–89.
[6] Chigan C, Atkinson GW, Nagarajan R. Cost effectiveness of joint multilayer protection in packet-over-optical networks. IEEE/OSA Journal of Lightwave Technology 2003;21:2694–704.
[7] Velasco L, Agraz F, Martínez R, Casellas R, Spadaro S, Muñoz R, et al. GMPLS-based multi-domain restoration: analysis, strategies, policies and experimental assessment. IEEE/OSA Journal of Optical Communications and Networking 2010;2:427–41.
[8] Zhang X, Shen F, Wang L, Wang S, Li L, Luo H. Two-layer mesh network optimization based on inter-layer decomposition. Photonic Network Communications 2011;21(3):310–20.
[9] Laguna M, Martí R. GRASP and path relinking for 2-layer straight line crossing minimization. INFORMS Journal on Computing 1999;11:44–52.
[10] Resende M, Ribeiro C. GRASP with path-relinking: recent advances and applications. In: Ibaraki T, Nonobe K, Yagiura M, editors. Metaheuristics: progress as real problem solvers. Springer; 2005. p. 29–63.
[11] Gonçalves J, Resende M. Biased random-key genetic algorithms for combinatorial optimization. Journal of Heuristics 2011;17(5):487–525.
[12] Morais R, Pavan C, Pinto A, Requejo C. Genetic algorithm for the topological design of survivable optical transport networks. IEEE/OSA Journal of Optical Communications and Networking 2011;3:17–26.

[13] de Miguel I, Vallejos R, Beghelli A, Durán R. Genetic algorithm for joint routing and dimensioning of dynamic WDM networks. IEEE/OSA Journal of Optical Communications and Networking 2009:608–21.

[14] Pioro M, Medhi D. Routing, flow, and capacity design in communication and computer networks. Morgan Kaufmann Publishers; 2004.

[15] Feo TA, Resende MGC. A probabilistic heuristic for a computationally difficult set covering problem. Operations Research Letters 1989;8:67–71.

[16] Feo TA, Resende MGC. Greedy randomized adaptive search procedures. Journal of Global Optimization 1995;6:109–33.

[17] Feo TA, Resende MGC, Smith SH. A greedy randomized adaptive search procedure for maximum independent set. Operations Research 1994;42: 860–78.

[18] Palmieri F, Fiore U, Ricciardi S. A GRASP-based network re-optimization strategy for improving RWA in multi-constrained optical transport infrastructures. Computer Communications 2010;33(15):1809–22.

[19] Villegasa JG, Prinsa C, Prodhona C, Medaglia AL, Velasco N. A GRASP with evolutionary path relinking for the truck and trailer routing problem. Computers & Operations Research 2011;38(9):1319–34.

[20] Pessoa LS, Resende MGC, Ribeiro CC. A hybrid Lagrangean heuristic with GRASP and path relinking for set k-covering. AT&T Labs Research Technical Report; 2010.

[21] Resende MGC, Ribeiro C. Greedy randomized adaptive search procedures. In: Glover F, Kochenberger G, editors. Handbook of metaheuristics. Kluwer Academic Publishers; 2003. p. 219–49.

[22] Resende MGC, Ribeiro C. Greedy randomized adaptive search procedures: advances and applications. In: Gendreau M, Potvin JY, editors. Handbook of metaheuristics, 2nd ed. Springer Science+Business Media; 2010.

[23] Festa P, Resende MGC. An annotated bibliography of GRASP—part I: algorithms. International Transactions in Operational Research 2009;16:1–24.

[24] Festa P, Resende MGC. An annotated bibliography of GRASP—part II: applications. International Transactions in Operational Research 2009; 16:131–72.

[25] Glover F. Tabu search and adaptive memory programming—advances, applications and challenges. In: Barr RS, Helgason RV, Kennington JL, editors. Interfaces in computer science and operations research. Kluwer Academic Publishers; 1996. p. 1–75.

[26] Glover F, Laguna M. Tabu search. Kluwer Academic Publishers; 1997.

[27] Hansen P, Mladenovic N. Variable neighborhood search: principles and applications. European Journal of Operational Research 2001;130:449–67.

[28] Mateus G, Resende MGC, Silva R. GRASP with path-relinking for the generalized quadratic assignment problem. Journal of Heuristics 2011;17(5):527–65.

[29] Resende MGC, Martí R, Gallego M, Duarte A. GRASP and path relinking for the max–min diversity problem. Computers and Operations Research 2010;37: 498–508.

[30] Resende MGC, Werneck RF. A hybrid heuristic for the p-median problem. Journal of Heuristics 2004;10:59–88.

[31] Duarte A, Martí R, Resende MGC, Silva RMA. GRASP with path relinking heuristics for the antibandwidth problem. Networks 2011;58(3):171–89.

[32] Noronha T, Resende M, Ribeiro C. A biased random-key genetic algorithm for routing and wavelength assignment. Journal of Global Optimization 2011;50(3):503–18.

[33] Reis R, Ritt M, Buriol L, Resende M. A biased random-key genetic algorithm for OSPF and DEFT routing to minimize network congestion. International Transactions in Operational Research 2011;18(3):401–23.

[34] Gonçalves J, Resende MGC. A parallel multi-population genetic algorithm for a constrained two-dimensional orthogonal packing problem. Journal of Combinatorial Optimization 2011;22(2):180–201.

[35] Robert CP, Casella G. Monte Carlo statistical methods. Springer; 2004.

[36] Martins E, Pascoal M. A new implementation of Yen's ranking loopless paths algorithm. 4OR: A Quarterly Journal of Operations Research 2003;1(2):121–33.

[37] Huelsermann R, Gunkel M, Meusburger C, Schupke DA. Cost modelling and evaluation of capital expenditures in optical multilayer networks. Journal of Optical Networking 2008;7(9):814–33.

[38] Festa P, Gonçalves JF, Resende MGC, Silva RMA. Automatic tuning of GRASP with path-relinking heuristics with a biased random-key genetic algorithm. In: Festa P, editor. Experimental algorithms, Lecture notes in computer science, 6049; 2010. p. 338–49.

[39] Ribeiro CC, Uchoa E, Werneck RF. A hybrid GRASP with perturbations for the Steiner problem in graphs. INFORMS Journal on Computing 2002;14:228–46.

Please cite this article as: Pedrola O, et al. A GRASP with path-relinking heuristic for the survivable IP/MPLS-over-WSON multi-layer network optimization problem. Computers and Operations Research (2011), doi:10.1016/j.cor.2011.10.026