European Journal of Operational Research 224 (2013) 614–624

Contents lists available at SciVerse ScienceDirect

European Journal of Operational Research

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

Innovative Applications of O.R.

Metaheuristic hybridizations for the regenerator placement and dimensioning problem in sub-wavelength switching optical networks

Oscar Pedrola [a,b,*], Davide Careglio [a], Miroslaw Klinkowski [c], Luis Velasco [a], Keren Bergman [b], Josep Solé-Pareta [a]

[a] Department of Computer Architecture, Universitat Politècnica de Catalunya (UPC), 08034 Barcelona, Spain
[b] Department of Electrical Engineering, Columbia University, New York, NY 10027, USA
[c] National Institute of Telecommunications, 04-894 Warsaw, Poland

ARTICLE INFO

Article history:
Received 28 November 2011
Accepted 17 August 2012
Available online 10 September 2012

Keywords:
OR in telecommunications
Metaheuristics
Sub-wavelength
Regenerator

ABSTRACT

Physical layer impairments severely limit the reach and capacity of optical systems, thereby hampering the deployment of transparent optical networks (i.e., no electrical signal regenerators are required). Besides, the high cost and power-consumption of regeneration devices makes it unaffordable for network operators to consider the opaque architecture (i.e., regeneration is available at every network node). In this context, translucent architectures (i.e., regeneration is only available at selected nodes) have emerged as the most promising short term solution to decrease costs and energy consumption in optical backbone networks. Concurrently, the coarse granularity and inflexibility of legacy optical technologies have re-fostered great interest in sub-wavelength switching optical networks, which introduce optical switching in the time domain so as to further improve resources utilization. In these networks, the complex regenerator placement and dimensioning problem emerges. In short, this problem aims at minimizing the number of electrical regenerators deployed in the network. To tackle it, in this paper both a greedy randomized adaptive search procedure and a biased random-key genetic algorithm are developed. Further, we enhance their performance by introducing both path-relinking and variable neighborhood descent as effective intensification procedures. The resulting hybridizations are compared among each other as well as against results from optimal and heuristic mixed integer linear programming formulations. Illustrative results over a broad range of network scenarios show that the biased random-key genetic algorithm working in conjunction with these two intensification mechanisms represents a compelling network planning algorithm for the design of future sub-wavelength optical networks.

1. Introduction

The advent of new disruptive bandwidth-intensive services and applications such as HDTV, VoIP, VoD, interactive teleconferencing, and storage area networks (SANs), has led to a huge surge of IP traffic which, ultimately, has enabled optical networks to become an unrivaled contender for the transmission in access, metro and core areas. In order to be able to better cope with current traffic demands, legacy optical transport networks (OTNs) have evolved over the last years from SONET/SDH (synchronous optical network/synchronous digital hierarchy) over static point-to-point dense wavelength division multiplexing (DWDM) links towards wavelength-switched optical networks (WSONs) (Lee et al., 2011). By cleverly dealing with the wavelength and space domains, WSONs enable dynamical reconfiguration of networks and allow for efficient network operation. However, based on recent measurements, network operators now foresee a highly dynamic data traffic scenario in which traffic flows occupying small fractions of a wavelength and short-lived connections will predominate (González de Dios et al., 2011). In this context, and due to their coarse granularity (a whole wavelength), WSONs will not be able to provide the degree of both flexibility and efficiency required. Consequently, nowadays sub-wavelength switching paradigms like the well-known optical packet switching and optical burst switching (OPS/OBS) (Ben Yoo, 2006), have become potential candidates to cope with the needs of next-generation OTNs. By leveraging recent advances in nanosecond-range photonic devices such as fast tunable lasers and fast switching elements, sub-wavelength technologies exploit the time domain to further improve the utilization of network resources, and consequently, the use of any network resource (e.g., optical–electrical–optical (O/E/O) regenerators in a node) is subject to the so-called statistical multiplexing concept, whereby resources are accessible according to their timely availability (i.e., there is a fair competition for the use of resources

* Corresponding author at: Department of Computer Architecture, Universitat Politècnica de Catalunya (UPC), 08034 Barcelona, Spain.
E-mail address: opedrola@ac.upc.edu (O. Pedrola).

0377-2217/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ejor.2012.08.011

among all packets/bursts/flows in the network). Without loss of generality, we assume in this work a layer 2 (L2) optical transport technology based on OBS, but the study herein presented is applicable to any of the aforementioned sub-wavelength technologies. The OBS consideration will however be made explicit only when strictly necessary. Moreover, hereinafter in this paper, we use the terms *packet* and *regenerator* generically to refer to the optical data unit of the sub-wavelength optical network (i.e., packets/bursts/flows) and to the O/E/O device.

So far, OTNs have been classified into three major network architectures based on the amount of regenerators they require (Ramamurthy et al., 1999): (i) Transparent networks, where the data signal remains in the optical domain for the entire end-to-end path. In this approach, there is no need for regenerators as it is assumed either an ideal physical layer or availability of all-optical 3R (re-amplifying, re-shaping, re-timing of the signal) regenerators; (ii) opaque networks, where the data signal undergoes regeneration at every node along its path; and (iii) translucent networks, where regenerations are only allowed at selected points in the network. Although the ultimate objective is to deploy transparent optical networks, it has precisely been the evolution from traditional opaque towards transparent network architectures that has brought to light the serious impact that physical layer impairments (PLIs) have on the optical end-to-end signal quality of transmission (QoT) (Ramamurthy et al., 1999). PLIs severely limit the reach and capacity of optical systems, and consequently, hamper the deployment of transparent optical networks, at least until all-optical 3R regeneration devices become mature enough to be considered as a viable solution (Rochette et al., 2006). Moreover, due to the fact that regenerators are both expensive and power-consuming, the opaque concept is not scalable to next-generation OTNs which strive for cost-effective power-efficient network architectures. For these very reasons, the deployment of translucent optical networks is currently considered the most promising short term solution to decrease costs and energy consumption in optical backbone networks. Note that in this paper we use QoT as a generic term to indicate the quality of a signal. To guarantee negligible bit error rate (BER) at reception, the received signal QoT must be above a given threshold. A series of technology-dependent QoT estimators exist in literature such as the computation of either the optical signal to noise ratio (OSNR), the Q-factor, or the BER (Agrawal, 2002).

It is clear then that the development of techniques to perform a sparse placement of regenerators is crucial to the success of translucent architectures (Shen and Tucker, 2007). Indeed, in this planning problem, the identification of the optimal trade-off between network construction costs (i.e., regenerators are expensive) and service provisioning performance (i.e., proper optical end-to-end QoT must be ensured) is of great importance. Hence, both the regenerator placement (RP), and routing and RP (RRP) if routing constraints are included, have recently received close attention from the research community. Although these problems have been thoroughly studied in the context of translucent WSONs (Manousakis et al., 2010), the resulting algorithms are not applicable to sub-wavelength switching networks since due to their statistical multiplexing nature there exists competition for accessing regenerator resources. Indeed, in sub-wavelength networks, RRP extends to the so-called routing and RP and dimensioning (RRPD) problem (Pedrola et al., 2011). Being the locations of regenerators selected (RP), the dimensioning phase (D) is responsible for obtaining the minimum amount of such regenerators so as to meet a predefined target QoT network performance. In the context of OBS networks, in Pedrola et al. (2011), we show that the joint RRPD problem leads to a very complex formulation, and consequently, we propose to solve both the routing and RPD subproblems separately. To be precise, we provide a mixed integer linear programming (MILP) model for the routing problem (minimizing congestion in network bottleneck links), and an optimal MILP formulation to solve RPD. It is worth pointing out that the routing formulations proposed are optimal and can be efficiently solved, and hence, the reader is referred to Pedrola et al. (2011) for further details. In contrast, RPD results in a complex formulation for which only fairly small problem instances can be solved exactly. Hence, we also provide both MILP-based and heuristic RPD algorithms and assess their performance (Pedrola et al., 2011).

1.1. Related work and contributions

In the context of communication networks, operations research (OR) methodologies provide a means of efficiently solving real life problems which are currently identified as open issues among network operators, and consequently, their solution is of great interest. Indeed, by applying powerful metaheuristic techniques one can gain a valuable insight into the problem in hand, as they allow for the consideration of real-sized, complex network and traffic scenarios such as the ones used in this paper. Successful applications of OR in this field are, among others, efficient heuristics for routing and wavelength assignment (RWA) in optical networks (Skorin-Kapov, 2007), optimization of network design/planning problems (Höller et al., 2008), and multicast routing algorithms for IP networks (Liang et al., 2010).

In this paper, we restrict our attention to the modeling of efficient metaheuristic hybridizations to solve the complex RPD problem found in sub-wavelength switching networks. To this end, a greedy randomized adaptive search procedure (GRASP) (Feo and Resende, 1995) and a biased random-key genetic algorithm (BRKGA) (Gonalves and Resende, 2010) are proposed. GRASP-based heuristics have been used to solve a wide range of problems with many and varied applications in the real life such as the design of communication networks (Palmieri et al., 2010), and collection and delivery operations (Villegasa et al., 2012). Similarly, BRKGAs, have also been recently used to solve complex communication network problems such as routing in IP and optical networks (RWA) (Reis et al., 2010; Noronha et al., 2010).

In order to further enhance these methodologies, we introduce an adaptation of the variable neighborhood descent (VND), and the path-relinking (PR) intensification procedures. Hansen et al. (2010) proposed VND as a search heuristic within the framework of variable neighborhood search methods. PR, by contrast, was first applied in the context of GRASP by Laguna and Martí (1999), thereby developing the powerful and widely used GRASP + PR algorithm. For a wide variety of examples and applications of GRASP + PR, the reader is referred to Resende and Ribeiro (2005). Both VND and PR have proven to be efficient in solving real life problems. For instance, in the field of optical networks, Martins et al. (2012) recently applied VND to solve the RWA problem, and Pedrola et al. (2012) implemented a GRASP + PR algorithm to tackle the complex multilayer IP/MPLS-over-WSON optimization problem.

Through extensive experiments, we show that BRKGA-based hybridizations outperform those based on GRASP and that the introduction of both VND and PR results in significant performance improvement for both algorithms. Further, by comparing the results of the metaheuristic hybridizations with that of MILP optimal and heuristic algorithms, this work reports yet another successful application of OR methods in the field of optical networks.

The rest of this paper is organized as follows: Section 2 provides a detailed description of the problem at hand and refers to the MILP optimal and heuristic RPD formulations. Sections 3 and 4 provide detailed descriptions of the metaheuristic methods developed to solve RPD. Exhaustive computational experiments are provided in Section 5, and finally concluding remarks are made in Section 6.

2. RPD problem description and MILP models

In order to make this paper self-contained, in this section we provide the MILP-based optimal and heuristic formulations proposed in Pedrola et al. (2011) to solve the RPD problem in sub-wavelength switching optical networks.

The network infrastructure consists of a set of nodes $\mathcal{V}$, and a set of bidirectional links $\mathcal{E}$. Given a set of traffic demands $\mathcal{D}$, which contains the average traffic load $h_d \in \mathbb{R}_+$ offered to every single source-termination $(s-t)$ pair of nodes, the single-path routing model proposed in Pedrola et al. (2011) obtains the valid path p_d to be followed by all packets belonging to demand $d \in \mathcal{D}$. Thus, we denote $\mathcal{Q} = \{p_d, d \in \mathcal{D}\}$ as the set of selected paths to be used to route packets through the network. Consequently, $\rho_p = h_d$ denotes the average traffic load offered to path $p \in \mathcal{Q}$; Eventually, $\mathcal{V}_p$ denotes the set of intermediate nodes on path p (i.e., all nodes in p except s_p and t_p).

Then, the objective of the RPD problem is to find: (a) the location of regenerator sites at selected nodes on those paths not meeting the QoT system specifications; and (b) the number of such regenerators in each node in order to guarantee a given target burst loss probability (B^{QoT}).

Let $\mathcal{P}^o \subseteq \mathcal{Q}$ denote the subset of paths $p \in \mathcal{Q}$ which, due to their QoT at the receiving end, require regeneration at some node $v \in \mathcal{V}_p$. For each $p \in \mathcal{P}^o$, let $\mathcal{S}_p = \{s_1, \ldots, s_{|\mathcal{S}_p|}\}$ denote the set of different options to establish a QoT compliant path, where $s_i \subseteq \mathcal{V},\ i = 1 \ldots |\mathcal{S}_p|$ and size $|\mathcal{S}_p|$ depends on the length of the transparent segments in path p (i.e., the distance an optical signal can travel without the need to undergo regeneration). In order to obtain $\mathcal{S}_p$, $p \in \mathcal{P}^o$ (i.e., all possible regeneration options) we can make use of any valid (sub-wavelength technology-dependent) QoT estimator. The selection of the regeneration option s from set $\mathcal{S}_p, p \in \mathcal{P}^o$ is performed according to a regenerator placement decision variable z_{ps} such that the following constraints are fulfilled:

$$\sum_{s \in \mathcal{S}_p} z_{ps} = 1, \quad \forall p \in \mathcal{P}^o, \tag{1a}$$

$$z_{ps} \in \{0,1\}, \quad \forall s \in \mathcal{S}_p,\ \forall p \in \mathcal{P}^o. \tag{1b}$$

Moreover, let ρ_v^o denote the offered traffic load requiring regeneration at node v. To estimate ρ_v^o (approximately) we add up the traffic load ρ_p offered to each path $p \in \mathcal{P}^o$ that both crosses and undergoes regeneration at node v:

$$\rho_v^o = \sum_{p \in \mathcal{P}^o : \mathcal{V}_p \ni v} \sum_{s \in \mathcal{S}_p : s \ni v} z_{ps} \rho_p. \tag{2}$$

Similarly,

$$\rho_v = \sum_{p \in \mathcal{P}^o : \mathcal{V}_p \ni v} \rho_p, \tag{3}$$

denotes an estimation of the maximal traffic load that is subject to regeneration at node $v \in \mathcal{V}$.

Eventually, we define a regenerator pool dimensioning function $F_v(\cdot)$, which for a given traffic load ρ_v^o, determines the minimum number of regenerators to be allocated in node v. This number must ensure that a given B^{QoT} is met. Assuming Poisson arrivals and fairness in the access to regenerator pools among packets, such a function is given by the following discontinuous, step-increasing function,

$$F_v(\rho_v^o) = \left\lceil B^{-1}\left(\rho_v^o, B^{QoT}\right) \right\rceil, \tag{4}$$

where B corresponds to the Erlang B-loss formula, which for a given number of regenerators $r \in \mathbb{N}$ available at node v can be calculated as,

$$B(\rho_v^o, r) = \frac{(\rho_v^o)^r / r!}{\sum_{k=0}^{r} (\rho_v^o)^k / k!}, \tag{5}$$

and where $B^{-1}\left(\rho_v^o, B^{QoT}\right)$ is the inverse function of (5) extended to the real domain, and $\lceil \cdot \rceil$ is the ceiling function. For the purpose of problem formulation, it is convenient to define a_r as the maximal load supported by r regenerators given a B^{QoT}, that is, $a_r = B^{-1}(r, B^{QoT})$. Finally, let R denote the number of regenerators required in the most loaded node, that is,

$$R = \max\{F_v(\rho_v) : v \in \mathcal{V}\}. \tag{6}$$

Note that we can make use of vector $\mathbf{a} = (a_1, \ldots, a_R)$ to obtain the piecewise linear approximation of $F_v(\cdot)$, which for a single node $v \in \mathcal{V}$, can be expressed as $F_v(\rho_v^o) = \min\{r : a_r > \rho_v^o\}$. Vector $\mathbf{a}$ is also essential for the dimensioning phase in the heuristic RPD methods proposed, as it helps determine $F_v(\rho_v^o)$ according to Procedure 1, which has a polynomial time complexity $O(R)$.

Procedure 1. Regenerator pool dimensioning

```
1: r ← 0
2: while ρ_v^o > a_r do
3:     r ← r + 1
4: end while
5: F_v ← r
```

2.1. Optimal RPD problem formulation (MP1)

Taking into consideration the aforementioned network modeling assumptions, the RPD problem can be formulated as the following MILP problem:

$$\underset{\mathbf{u},\boldsymbol{\rho}^o,\mathbf{z}}{\text{minimize}} \quad F = \sum_v \sum_r u_v^r r \tag{MP1}$$

subject to

$$\sum_r u_v^r a_r - \rho_v^o \geqslant 0, \quad \forall v \in \mathcal{V}, \tag{7a}$$

$$\sum_r u_v^r = 1, \quad \forall v \in \mathcal{V}, \tag{7b}$$

$$\sum_{s \in \mathcal{S}_p} z_{ps} = 1, \quad \forall p \in \mathcal{P}^o, \tag{7c}$$

$$\sum_{p \in \mathcal{P}^o : \mathcal{V}_p \ni v} \sum_{s \in \mathcal{S}_p : s \ni v} z_{ps} \rho_p - \rho_v^o = 0, \quad \forall v \in \mathcal{V}, \tag{7d}$$

$$u_v^r \in \{0,1\}, \quad \forall r \in [1,R],\ \forall v \in \mathcal{V}, \tag{7e}$$

$$z_{ps} \in \{0,1\}, \quad \forall p \in \mathcal{P}^o,\ \forall s \in \mathcal{S}_p, \tag{7f}$$

$$\rho_v^o \in \mathbb{R}^+, \quad \forall v \in \mathcal{V}. \tag{7g}$$

The objective of optimization Problem (MP1) is to minimize the total number of regenerators that have to be placed in the network. In (MP1), decision variables u_v^r have been introduced in order to represent the number of regenerators required in node v. Here, we consider ρ_v^o to be an auxiliary variable representing the traffic load requiring regeneration offered to node $v \in \mathcal{V}$. Constraints (7a) and (7b) result from the 0–1 representation of the piecewise linear approximation of $F_v(\cdot)$. Constraints (7c) are the QoT compliant path selection constraints. Constraints (7d) are the traffic load offered to a regenerator node calculation constraints. Eventually, 7e, 7f, and 7g are the variable range constraints. In MP1, the total amount of variables can be approximated by $|\mathcal{V}| \cdot R + |\mathcal{P}^o| \cdot \Theta$, where Θ represents an upper bound on the maximum size of set $\mathcal{S}_p, p \in \mathcal{P}^o$. Besides, the size of the constraint set is $3 \cdot |\mathcal{V}| + |\mathcal{P}^o|$. For example, assuming a value of $R = 200$ and the German network (see Section 5), the problem size increases up to approximately 5×10^4 variables and 1×10^3 constraints, a fact which makes it highly difficult to find an exact solution within a reasonable amount of computational time.

2.2. A MILP-based RPD heuristic

In an attempt to lower the complexity of (MP1), the Reduced MP1 (R-MP1) algorithm introduces two additional constraints to the problem, thereby narrowing the search space. To be precise, the new constraints are the sequentially obtained solutions of the load-based MILP formulation as detailed in Pedrola et al. (2011). Although these new constraints may exclude the optimal solution of (MP1), we expect them to both speed up computation times and lower optimality gaps while still providing good near-optimal solutions. Therefore, let us denote, respectively, g^* and k^* as such load-based sequentially obtained solutions. Then, we reformulate (MP1) as follows:

$$\underset{\mathbf{u},\rho^o,\mathbf{z}}{minimize} \quad F = \sum_v \sum_r u_v^r r \tag{R-MP1}$$

$$subject\ to \sum_v y_v \leqslant k^*, \tag{8a}$$

$$\sum_v \rho_v^o \leqslant g^*, \tag{8b}$$

and subject to constraints (7a)–(7f) and (7g). In constraint (8a), y_v values denote a vector of binary decision variables, that is, $\mathbf{y} = (y_1, \ldots, y_{|\mathcal{V}|})$, where each value corresponds to one node and determines if it is used as regeneration point by some path $p \in \mathcal{P}^o$ ($y_v = 1$) or not ($y_v = 0$). This constraint ensures that the solution will have at most k^* regeneration nodes. Then, constraint (8b) is formulated so as to find among the remaining solutions the one that minimizes the total network load requiring regeneration. Here it is worth pointing out that as long as the scenario considered does not involve optical paths requiring a large number of regenerations, constraint (8a) is very unlikely to exclude the optimal solution of (MP1). Basically, it is due to the fact that the dimensioning function of our problem is (4), which favors, to some degree, the grouping of regenerators. Constraint (8b), by contrast, is just an heuristic approach to help solve the problem. Notice that (8b) does not deal with the distribution of the load but with its minimization, and thus, the optimal solution in terms of the number of regenerators is generally excluded. Given the complexity of (MP1) and in order to provide an improved alternative to (R-MP1), in the next sections we propose and describe efficient heuristic RPD algorithms.

3. A GRASP-based RPD heuristic

The multi-start GRASP procedure basically consists of two phases. In the first phase, 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.

3.1. Construction procedure

In order to construct a solution, our problem consists in selecting, for each path $p_i \in \mathcal{P}^o = \{p_1, \ldots, p_{|\mathcal{P}^o|}\}$, a regeneration option $s_j \in \mathcal{S}_{p_i} = \{s_1, \ldots, s_{|\mathcal{S}_{p_i}|}\}$. For the sake of clarity, let us define a path instantiation u_i^j as the assignment of regeneration option $s_j \in \mathcal{S}_{p_i}$ to path $p_i \in \mathcal{P}^o$, that is, $u_i^j = < p_i, s_j >$. Moreover, let $\mathcal{U} = \bigcup_{p_i \in \mathcal{P}^o} u_i^j$ denote the complete set of path instantiations. Note that we are dealing with an unconstrained problem, and thus, any path $p \in \mathcal{P}^o$, can take any $s \in \mathcal{S}_p$ independently of the decision taken by other paths (i.e., no path instantiation can lead to an unfeasible solution). Hence, once $\mathcal{U}$ is generated, a feasible solution to the RPD problem can be obtained. Let us denote with $g(\cdot)$ the cost function which aims at minimizing the total amount of regenerators to be deployed given a B^{QoT} target performance. Function $g(\cdot)$ makes use of Procedure 1 as defined in Section 2 to compute the number of regenerators. Note that set $\mathcal{U}$ helps determine load ρ_v^o, $v \in \mathcal{V}$, and thus, a solution to the problem can be obtained by applying $g(\mathcal{U})$. However, it must be noted that such an operation is required in every phase of the GRASP heuristic and performed a very high number of times. In addition, parameter R (see Eq. (6)) can be extremely high in some network scenarios. For this very reason, we use a binary search algorithm to reduce the complexity of Procedure 1 to $O(logR)$. Our approach to construct solutions takes into consideration the order in which path instantiations are performed, as in fact, due to the already deployed regenerators, such order has influence on subsequent path instantiations. Thus, we consider the order $\mathcal{O}_x = \{p_1, \ldots, p_{|\mathcal{P}^o|}\}$ as a guide to iteratively perform, for each path $p \in \mathcal{P}^o$, the required path instantiation and generate $\mathcal{U}$. From this point on, it will be up to the subsequent intensification strategies to improve set $\mathcal{U}$.

Specifically, the construction procedure that we consider to solve the RPD problem is the greedy randomized construction (GRC) as described in the next subsection.

3.1.1. Greedy randomized construction (GRC)

GRC relies on a restricted candidate list (RCL) which is made up of the paths $p \in \mathcal{P}^o$ with the best (smallest) incremental costs $c(p)$. Paths are iteratively processed, and at each step, costs $c(p), p \in \mathcal{P}^o$ are recomputed to account for the paths already processed (i.e., path instantiations previously performed). All regenerative options $s \in \mathcal{S}_p$ are considered, and $c(p)$ is given the value of the option s with the lowest incremental cost (i.e., $c(p) = \min_{s \in \mathcal{S}_p}\{c(s)\}$). It is worth mentioning that costs $c(s)$ account for the increment in regenerators caused by the selection of s as regeneration option for path p. Therefore, RCL is dynamically built with all paths $p \in \mathcal{P}^o$ whose cost $c(p)$ falls within the interval defined by the real parameter $\alpha \in [0,1]$ (see lines 5–7 in Procedure 2). Then, one of the paths in the RCL (p_i) is randomly chosen and the regeneration option $s_j \in \mathcal{S}_{p_i}$ with the lowest incremental cost is selected to perform the required path instantiation u_i^j. Once $\mathcal{Q}$ becomes an empty set, all paths in $\mathcal{P}^o$ have been processed, and hence, we can finally obtain the total number of regenerators deployed by applying $g(\mathcal{U})$. Note that costs $c(s_j), s_j \in \mathcal{S}_{p_i}$, are again recomputed at each iteration in order to take into account previous path instantiations (i.e., regenerator sites already distributed in the network). In this algorithm, parameter α needs to be adjusted as shown in Section 5.

Procedure 2. Greedy randomized construction (GRC)

INPUT: $\mathcal{P}^o, \mathcal{S}_p\ \forall p \in \mathcal{P}^o, \alpha$
OUTPUT: $\mathcal{O}_x, \mathcal{U}_x, g(\mathcal{U}_x)$
1: $\mathcal{U}_x \leftarrow \emptyset, \mathcal{O}_x \leftarrow \emptyset$;
2: Initialize the candidate set: $\mathcal{Q} \leftarrow \mathcal{P}^o$;
3: Evaluate the incremental cost $c(p)$ for all $p \in \mathcal{Q}$;
4: **while** $\mathcal{Q} \neq \emptyset$ **do**
5: $c^{min} \leftarrow min\{c(p) \mid p \in \mathcal{Q}\}$;
6: $c^{max} \leftarrow max\{c(p) \mid p \in \mathcal{Q}\}$;
7: $RCL \leftarrow \{p \in \mathcal{Q} \mid c(p) \leqslant c^{min} + \alpha(c^{max} - c^{min})\}$;
8: Select an element p_i from RCL at random;
9: $\mathcal{O}_x \leftarrow \mathcal{O}_x \cup \{p_i\}$;
10: Take element $s_j \in \mathcal{S}_{p_i}$ such that $c(s_j) = c(p_i)$;
11: Perform path instantiation u_i^j
12: $\mathcal{U}_x \leftarrow \mathcal{U}_x \cup \left\{u_i^j\right\}$;
13: Update candidate set $\mathcal{Q}$;
14: Reevaluate the incremental cost $c(p)$ for all $p \in \mathcal{Q}$;
15: **end while**

3.2. Local search

In this section, we provide the details concerning neighbor generation as well as the pseudo-code and operation of the local search algorithm adopted, namely the VND technique.

3.2.1. Neighbor generation

Once the path instantiation set $\mathcal{U}_x$ is obtained by means of GRC, local search aims at improving such a solution by exploring its neighborhood. Note that, due to the fact that path instantiations are performed taking into account the current location and number of regenerators deployed, only the last path to be instantiated takes its decision with a whole view of the problem. Changing one regenerator selection may therefore impact on subsequent decisions and eventually provide a different solution. Neighbor generation tries to exploit this issue. To this end, random neighbors in the first neighborhood (i.e., 1-*move* neighbor) of $\mathcal{U}_x$, that is, $N_1(\mathcal{U}_x)$, are generated by uniformly selecting a pair of pivots p_i, p_j among those in set $\mathcal{P}^o$. Then, we take their respective path instantiations $(u_i^m,\ u_i^n)$ and try to improve the selection of s_m and s_n as regeneration options for paths p_i and p_j respectively. We evaluate the incremental costs of all $s_k \in \mathcal{S}_{p_i}, s_l \in \mathcal{S}_{p_j}$ so that both path instantiations are recomputed and inserted in $\mathcal{U}_x$ again. Note that a *q-move* neighbor is generated by performing such a random pair selection and re-computation operation for q consecutive times over the same set $\mathcal{U}_x$.

3.2.2. Variable neighborhood descent (VND)

The pseudo-code for the VND algorithm is illustrated in Procedure 3. Starting at $\mathcal{U}_x$, VND begins the search by constructing a set of *MaxSearch* neighbors in N_1, and if among them all, an improving $\mathcal{U}_N$ solution is found, the algorithm moves to $\mathcal{U}_N$ and continues the search in N_1. If no improvement is found, by contrast, VND switches to N_2 and so on. Due to the fact that VND switches back to N_1 every time an improvement is found, this algorithm is able to perform an exhaustive search until the last allowed neighborhood N_{MAX} is reached. Note that the intensity of the search in each neighborhood structure depends on the number of neighbors sampled (*MaxSearch*). The best solution found ($\mathcal{U}_{BEST}$) is returned as output when neighborhood N_{MAX} is reached and no improvement is found.

Procedure 3. Variable neighborhood descent (VND)

INPUT: $\mathcal{U}_x, MaxSearch, N_{MAX}$;
OUTPUT: $\mathcal{U}_{BEST}$;
1: $\mathcal{U}_B \leftarrow \mathcal{U}_x, k \leftarrow 1, \mathcal{U}_{BEST} \leftarrow \mathcal{U}_x$;
2: **repeat**
3: $i \leftarrow 0, \mathcal{U}_N \leftarrow \mathcal{U}_B$;
4: **repeat**
5: $\mathcal{U}_{x'} \leftarrow$ Create-N_k-neighbor($\mathcal{U}_B$);
6: **if** $g(\mathcal{U}_{x'}) < g(\mathcal{U}_N)$ **then**
7: $\mathcal{U}_N \leftarrow \mathcal{U}_{x'}$;
8: **end if**
9: $i \leftarrow i + 1$;
10: **until** $i \geqslant MaxSearch$
11: **if** $g(\mathcal{U}_N) < g(\mathcal{U}_B)$
12: $\mathcal{U}_B \leftarrow \mathcal{U}_N$;
13: $k \leftarrow 1$;
14: **else**
15: $k \leftarrow k + 1$;
16: **end if**
17: **if** $g(\mathcal{U}_B) < g(\mathcal{U}_{BEST})$
18: $\mathcal{U}_{BEST} \leftarrow \mathcal{U}_B$;
19: **end if**
20: **until** $k \geqslant N_{MAX}$

3.3. Path relinking

PR (Glover, 1996) generates new solutions by exploring the trajectories connecting pairs of high-quality solutions. To ensure a proper PR operation, the management of the *elite set* (*ES*) has to balance between quality and diversity attributes (Resende and Werneck, 2004).

3.3.1. Greedy PR (GPR)

To perform PR, we make use of the path instantiation set $\mathcal{U}$ so as to easily detect solution differences. Hence, given an initiating ($\mathcal{U}_i$) and a guiding ($\mathcal{U}_g$) solution, we obtain the set of divergences $\Psi_{i,g}$ by identifying those path instantiations in $\mathcal{U}_g$ which differ from those selected in $\mathcal{U}_i$, that is, $\Psi_{i,g} = \mathcal{U}_g \setminus \{\mathcal{U}_g \cap \mathcal{U}_i\}$. In this work, we consider the greedy PR (GPR) approach (Resende et al., 2010) to build the path from the initiating towards the guiding solution. Therefore, at each single movement we evaluate the impact that all path instantiations in $\Psi_{i,g}$ have when introduced in $\mathcal{U}_i$. Among them, we select the one minimizing $g(\cdot)$, that is, u^*, and replace the corresponding path instantiation in $\mathcal{U}_i$. Ties in this case are broken randomly. Finally, u^* is removed from set $\Psi_{i,g}$. In this way, we progressively move towards $\mathcal{U}_g$ and until $\Psi_{i,g}$ becomes an empty set (i.e., the guiding solution has been reached). PR is implemented using the *back-and-forward* (PR_{bf}) strategy, which explores the path in both directions (Resende and Werneck, 2004).

Moreover, the selection of a solution from *ES* depends on both a distance measure and a selection policy. We consider the approach presented in Resende and Werneck (2004), where the authors propose to select the elite solution with probabilities proportional to their distance to the solution on which to perform PR ($\mathcal{U}_x$). Note that the maximum distance between two solutions $\mathcal{U}_x$ and $\mathcal{U}_y$, that is, $d_{x,y}$, is equal to $|\mathcal{P}^o|$. As to *ES* management (i.e., which solution can be inserted and which has to be removed in order to keep $|ES|$ constant), it is worth noticing that a solution whose quality is lower than the best stored in *ES* and higher than the worst in *ES*, will be added iff its distance to *ES* (i.e., $d_{x,ES} = \min_{\mathcal{U}_i \in ES}\{d_{x,i}\}$) is larger than a pre-established threshold d_{th}, that is, $d_{x,ES} > d_{th}$, where $d_{x,ES} = \min_{\mathcal{U}_i \in ES}\{d_{x,i}\}$.

3.4. GRASP + PR algorithm

The GRASP + PR algorithm design considered in this paper is the evolutionary GRASP + PR (EPR) implementation (Resende et al., 2010). Specifically, the authors propose three different hybridizations of GRASP + PR, namely static, dynamic and evolutionary GRASP + PR. Among them, the evolutionary variant displayed better performances. For this reason, in order to tackle RPD, we consider the evolutionary scheme. EPR is based on an evolutionary post-processing phase for GRASP + PR algorithms introduced by Resende and Werneck (2004). The pseudo-code for EPR is shown in Procedure 4. After set *ES* becomes full, the so-called dynamic GRASP + PR (DPR) algorithm is executed for *LocalItr* iterations (between lines 4 and 12 in Procedure 4). Then, the set of solutions in *ES* is evolved. This process is repeated for a maximum of *GlobItr* iterations and eventually the best solution in *ES* is returned as output.

It is worth mentioning that we use a parallel implementation of the algorithm in order to better exploit the capacity of our evaluation platform (see Section 5) and to both speed up the algorithm execution time and enhance its performance. A pool of k threads is generated, each of them running in parallel the inner loop of the EPR algorithm, that is, DPR. During *LocalItr* iterations all threads share a common *ES* which is accessed following a mutual-exclusion policy. Once all threads have finished their task, *ES* is evolved as dictated by the EPR method.

Procedure 4. Evolutionary GRASP + PR (EPR)

INPUT: *GlobItr*, *LocalItr*, $|ES|$;
OUTPUT: $\mathcal{U}_{BEST}$;
1: $i \leftarrow 0, j \leftarrow 0, ES \leftarrow \emptyset$;
2: Execute multi-start GRASP phase until *ES* is full;
3: **repeat**
4: **repeat**
5: $\mathcal{U}_x \leftarrow$ ConstructionProcedure;
6: $\mathcal{U}_{x'} \leftarrow$ Local Search starting at $\mathcal{U}_x$;
7: Randomly select $\mathcal{U}_e$ from *ES*;
8: $\mathcal{U}_y \leftarrow \mathrm{PR}_{bf}(\mathcal{U}'_x, \mathcal{U}_e)$;
9: $\mathcal{U}_{y'} \leftarrow$ Local Search starting at $\mathcal{U}_y$;
10: Try to insert $\mathcal{U}_{y'}$ in *ES*;
11: $j \leftarrow j + 1$;
12: **until** $j \geqslant$ *LocalItr*
13: Improvement $\leftarrow$1;
14: **while** Improvement **do**
15: Improvement $\leftarrow$0;
16: Apply $\mathrm{PR}_{bf}(\mathcal{U}_x, \mathcal{U}_{x'})$ for every pair $(\mathcal{U}_x, \mathcal{U}_{x'}) \in$ *ES* and let $\mathcal{U}_y$ be the best solution found;
17: $\mathcal{U}_{y'} \leftarrow$ Local Search starting at $\mathcal{U}_y$;
18: **if** $\mathcal{U}_{y'}$ can be inserted in *ES* **then**
19: Improvement $\leftarrow$1;
20: **end if**
21: **end while**
22: $i \leftarrow i + 1$;
23: **until** $i \geqslant$ *GlobalItr*
24: $\mathcal{U}_{BEST} = \min_{\mathcal{U}_k \in ES}\{g(\mathcal{U}_k)\}$;

4. A BRKGA-based RPD heuristic

In BRKGA, a population of p individuals is evolved over a number of generations. Each individual is represented by an array of n genes (called a chromosome), and where each gene can take any value in the real interval [0,1]. Thus, each chromosome encodes a solution of the problem and a fitness value, that is, the value of the objective function. In BRKGA, individuals of the population are divided into the elite set p_e (those individuals with the best fitness values), and a non-elite set. Whilst the majority of new individuals are generated by crossover combining two elements, one elite and another non-elite, elite individuals are copied unchanged from one generation to the next so as to keep track of good solutions. With the very same objective, in the crossover operation an inheritance probability (ρ_e) is defined as the probability that an offspring inherits the gene of its elite parent. Finally, a small number of mutant individuals are introduced 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.

The decoder algorithm for our BRKGA was first presented in Pedrola et al. (2011), where our aim was to develop a simple and straightforward decoding algorithm, as fast cost function evaluations are crucial to BRKGAs. For completeness, the decoder pseudo-code is illustrated in Procedure 5. Each chromosome contains as many genes as nodes in the network, and each gene assigns its random value to the corresponding node (see line 2 in Procedure 5). Then, the regeneration option minimizing the sum of such node metric is selected. Hence, a very fast fitness computation can be obtained with this decoding algorithm, even if complex problem instances are considered. However, such simplicity may also hinder the possibility of approaching optimality. This is indeed an important issue to be tackled since the reduction of just one regenerator unit implies significant cost and energy savings in the network, and thus, it cannot be neglected. For this very reason, in the next section, we re-consider the implementation of BRKGA for the RPD problem, and propose an enhanced method including VND and PR as intensification strategies, namely the BVR algorithm.

Procedure 5. BRKGA decoder

INPUT: $\mathcal{N}$, *chromo*, $\mathcal{P}^o, \mathcal{S}_p \forall p \in \mathcal{P}^o$;
OUTPUT: $\mathcal{U}_x, g(\mathcal{U}_x)$;
1: **for all** node $n \in \mathcal{N}$
2: *n.metric* $\leftarrow$ *chromo.getGene*(*n*);
3: **end for**
4: $\mathcal{U}_x \leftarrow \emptyset$;
5: **for all** path $p_i \in \mathcal{P}^o$ **do**
6: **for all** option $s_j \in \mathcal{S}_{p_i}$ **do**
7: **for all** node $n \in s_j$ **do**
8: $c(s_j) \leftarrow c(s_j) + n.metric$;
9: **end for**
10: **end for**
11: $s^* = \min_{s_j \in \mathcal{S}_{\sqrt{}}}\{c(s_j)\}$;
12: $\mathcal{U}_x \leftarrow \mathcal{U}_x \cup \{s^*\}$;
13: **end for**

4.1. BRKGA with VND and PR (BVR) algorithm

Although in the literature several works report successful implementations of genetic algorithms working in conjunction with PR (see e.g., Reeves and Yamada (1998) and Zhang and Lai (2006)), BVR proposes a novel algorithm implementation in which both a local search (VND) and a PR strategy are inserted into the basic BRKGA methodology. The pseudo-code for BVR is illustrated in Procedure 6. The input parameters *gens* and *GlobItr* define the maximum number of generations over which the initial population is evolved. After BRKGA is run for *gens* generations, the chromosomes belonging to the elite set (p_e) in the resulting population $\mathcal{P}_{op}$ are all candidate to be inserted in *ES*. Then, following the same probabilistic approach as in the GRASP + PR algorithm described in Section 3, a solution from *ES* is selected to perform both PR with $\mathcal{U}_x$ and the subsequent local search intensification. Then, the evolution of the current population is resumed for another *gens* generations. Finally, the best solution stored in *ES* is returned as output.

Again, BVR is implemented following a parallel approach in which k threads run the algorithm separately sharing a common *ES*. Note that the parallel approach not only allows to reduce time complexity but also generates higher quality elite sets, as *ES* is concurrently fed by up to k threads. Finally, all parameters needed to set up both the BVR and EPR algorithms will be adjusted in Section 5.

Procedure 6. BVR algorithm

INPUT: *gens*, *GlobItr*;
OUTPUT: $\mathcal{U}_{BEST}$;
1: $i \leftarrow 0$, $ES \leftarrow \emptyset$;
2: init-BRKGA ();
3: **repeat**
4: $\mathcal{P}_{op} \leftarrow$ run-BRKGA (gens);
5: Take p_e from $\mathcal{P}_{op}$;

(*continued on next page*)

for all chromosome $ch \in p_e$ **do**
  Take $\mathcal{U}_{ch}$ and try to insert it in *ES*;
end for
Randomly select $\mathcal{U}_e$ from *ES*;
$\mathcal{U}_y \leftarrow \mathrm{PR}_{bf}(\mathcal{U}_x, \mathcal{U}_e)$;
$\mathcal{U}_{y'} \leftarrow$ Local Search starting at $\mathcal{U}_y$;
Try to insert $\mathcal{U}_{y'}$ in *ES*;
$i \leftarrow i + 1$;
until $i \geqslant GlobItr$
$\mathcal{U}_{BEST} = \min_{\mathcal{U}_k \in ES}\{g(\mathcal{U}_k)\}$;

5. Computational experiments

This section describes the computational experiments conducted so as to both evaluate and compare the performance of the MILP-based (MP1) (optimal) and (R-MP1) models with that of the EPR and BVR hybridized heuristic models proposed in this paper. The heuristic methodologies have all been implemented in Java SE 1.6.0_17 using a parallel approach. The experiments have been conducted on an Intel (R) Core (TM) i7 CPU 950 at 3.07 Gigahertz with 4 gigabytes RAM under Windows 7 Professional Edition (64 bits). We use CPLEX (version 12.1) as the underlying MILP-solver. Note that with this processor we can make use of up to eight parallel threads, and thus, in all our problems we set $k = 8$.

5.1. Problem instances

The performance of the proposed metaheuristic hybridizations as well as that of the MILP models has been compared over the set of optical core transport networks shown in Fig. 1 (see Pedrola et al., 2011 for more details).

5.2. Sub-wavelength optical network scenario

In this paper, we study the RPD problem under the assumption that OBS is the underlying sub-wavelength switching technology. Optical bursts are generated according to a *Poisson* arrival process and have exponentially distributed lengths. The mean duration of a burst is 100 microseconds (i.e., 1 megabyte). The traffic is uniformly distributed, normalized to the transmission bit-rate and expressed in Erlangs. Here, an Erlang corresponds to the amount of traffic that occupies an entire wavelength.

As to the QoT model, we make use of method proposed in Pedrola et al. (2011) to obtain the set of paths $\mathcal{P}^o$ that do not comply with the OSNR system specifications. Hence, it is possible to identify which bursts will need to undergo a regeneration when sent into the network. This is achieved by defining an OSNR quality threshold (T_{osnr}) below which the optical signal cannot be correctly read, and thus, the burst is discarded. In Pedrola et al. (2011), we show that T_{osnr} is a critical parameter, and that the network configuration must be set in accordance. In our experiments, we assume

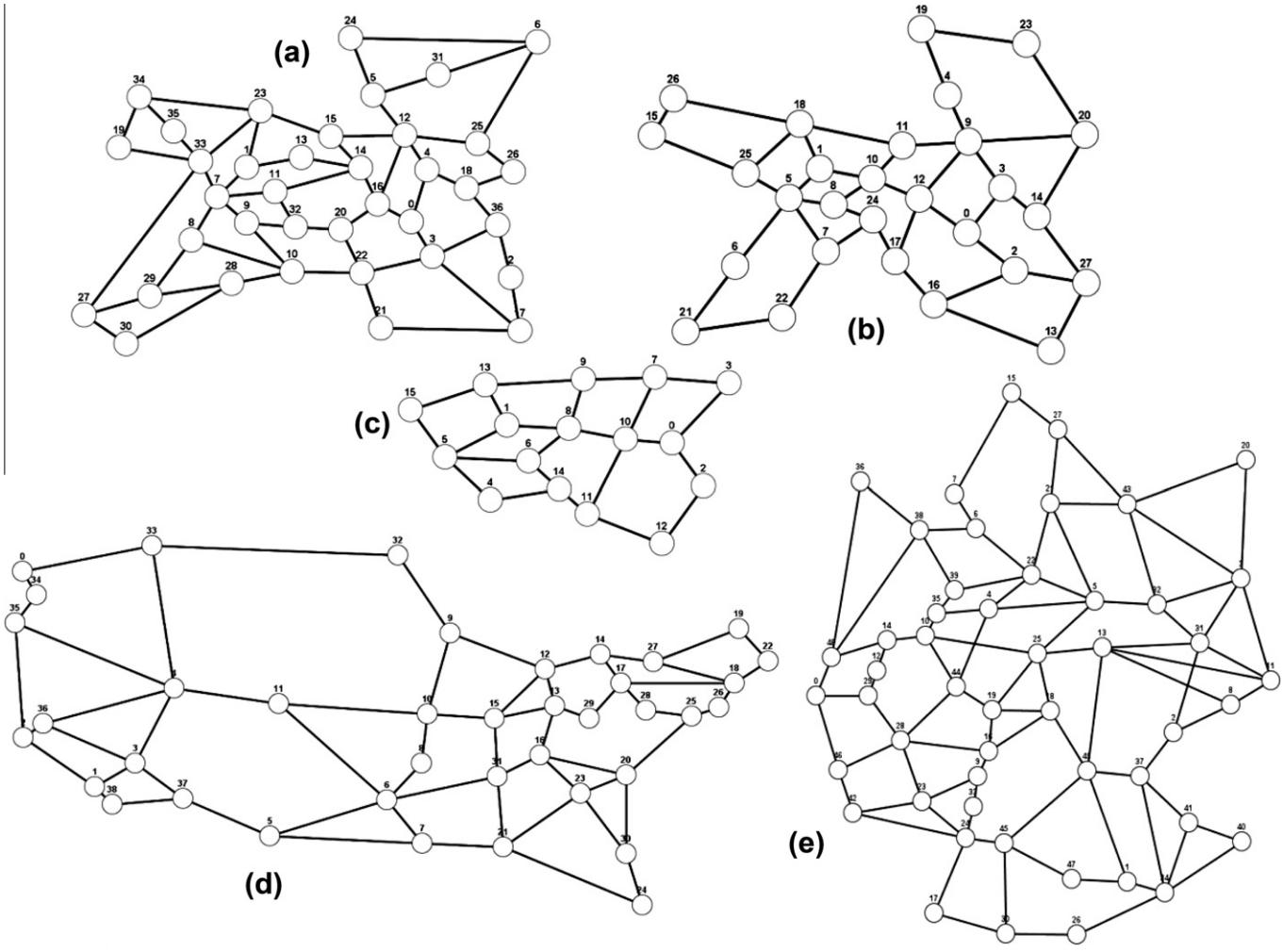

Fig. 1. (a) Large (37 nodes, 57 links), (b) basic (28 nodes, 41 links), (c) core (16 nodes, 23 links), (d) USA–Can (39 nodes, 61 links), and (e) German (50 nodes, 88 links).

Table 1
Network scenarios evaluated.

Scenario	*A*	*B*	*C*	*D*
Erlangs	15	25	15	25
T_{osnr} (decibel)	20	20	21	21

Table 2
T_{osnr} impact on $|\mathcal{P}^o|$ and $|\mathcal{C}^*|$.

	T_{osnr}	USA–Can	German	Core	Basic	Large
$|\mathcal{P}^o|$	20 decibels	421	338	18	349	746
	21 decibels	657	752	55	462	919
$|\mathcal{C}^*|$	20 decibels	21,414	22,987	328	7707	16,438
	21 decibels	23,757	38,615	596	6459	13,616

a realistic scenario with bidirectional network links equipped with 32 wavelengths, each at 10 gigabit/second.

Aiming at performing an exhaustive evaluation of all the methods proposed, we consider a target QoT performance $B^{QoT} = 10^{-3}$ in four different network scenarios (see Table 1). These scenarios represent load values corresponding to both a medium and a highly loaded network, and thresholds corresponding to realistic values. Whilst an increase in load represents an increase in the number of regenerators required to support each independent burst flow, rising T_{osnr} implies increasing the size of set $\mathcal{P}^o$ as shown in Table 2. Further, Table 2 reports the total number of regeneration options $\mathcal{C}^*$ (i.e., $\mathcal{C}^* = \bigcup_{p\in\mathcal{P}^o} \mathcal{S}_p$) under both T_{osnr} values, thus allowing for a fair estimation of the different problem complexities. One can observe that in both the Basic and Large networks such amount of options decreases with the increase of T_{osnr}. This can happen as higher thresholds may limit reachability in the resulting transparent graph, and consequently reduce the number of regeneration options. It is also worth mentioning that in our previous work in Pedrola et al. (2011), the size of set $\mathcal{C}^*$ was limited to a maximum of $|\mathcal{S}_p| = 25,\ \forall p \in \mathcal{P}^o$ so as to reduce complexity. However, such an approach hinders the possibility of approaching optimality, which is our primary objective.

In order to conduct a thorough analysis of all the RPD methods, we first obtain the results for both (MP1) and (R-MP1) using CPLEX as solver. These results provide us with the best known solutions for each of the scenarios evaluated. Afterwards, a set of preliminary experiments is carried out so as to obtain the best set up for both the EPR and the BVR RPD metaheuristics. Finally, we evaluate their performance in terms of solution quality and study the statistical significance of the results obtained through non-parametric tests.

5.3. Experimental analysis

5.3.1. MILP methods results

We use CPLEX to solve (MP1) and (R-MP1) under each network and scenario. Table 3 reports the minimum number of regenerators to be deployed as well as the optimality GAP (%) of the solutions provided by CPLEX. Note that the MILP optimality GAP is defined as: $GAP(\%) = \frac{best-lb}{best}$, where *best* refers to the best current solution, and *lb* to the best current lower bound found by the CPLEX branch-and-bound process.

Each execution is stopped when either the running time reaches 24 hours or the tree size grows up to 4 Gigabyte. One can observe that (MP1) is only able to reach optimality in the most simple problem instance, that is, the CORE network. Indeed, optimality gaps of up to 11.9% are observed in the German network. However, we can observe that with (R-MP1) such gaps are brought to 0 in all cases except for scenarios *C* and *D* in the German network. As aforementioned, however, this is the consequence of introducing in (MP1) two additional heuristic constraints, which definitely reduce the search space, and thus, the complexity of the problem. Whilst (R-MP1) works properly in some scenarios (e.g., Basic (*A*, *C*)), in others, such a cut of the solution space excludes high-quality solutions, thereby hindering the possibility of approaching optimality (USA–Can (*C*, *D*), Large (*B*)). The results in Table 3 are also in line with the different problem complexities given by the values provided in Tables 1 and 2. First, we observe that load variations, which have an effect on parameter *R* (see Eq. (6)), do not have a significant impact on the results obtained, as there are no notable differences among scenarios *A*–*B* and *C*–*D*. Indeed, as shown in Section 2.1, it is both the number of regeneration options and the size of set $\mathcal{P}^o$ what has the greater impact on the number of variables and constraints of the problem. However, rising T_{osnr} does have a clear impact on the problem complexities. As shown in Table 2, in both the Basic and Large topologies, it represents an increase in $\mathcal{P}^o$, but, at the same time, a substantial reduction of $|\mathcal{C}^*|$, thereby lowering the problem complexity. This issue is reflected in the optimality gaps found by (MP1) in these two topologies, as they decrease from scenarios (*A, B*) to (*C, D*). In the USA–Can and German networks, as expected, complexities and gaps increase. We note however that in the German network there is a reduction from scenario *A* to *C*. We attribute this behavior to the very high gap found in scenario *A*, which may be due to a specific particularity of this problem instance.

In the rest of our experiments, we exclude the Core network instance as it does not provide any way of differentiating the performance of the different methods.

5.3.2. EPR and BVR parameter tuning

To perform a comprehensive quantitative analysis of the results, we consider the statistics proposed in Resende et al. (2010). 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 studied. Moreover, we compute the relative percentage deviation (*Dev*) between the best solution value obtained by

Table 3
(MP1) and (R-MP1) Results (total number of regenerators and optimality gaps).

	A		*B*		*C*		*D*	
	MP1	R-MP1	MP1	R-MP1	MP1	R-MP1	MP1	R-MP1
Network								
Core	44	44	62	62	115	115	166	166
Basic	385	384	572	574	580	579	874	873
Large	658	661	998	1139	942	937	1432	1430
USA–Can	269	264	402	400	472	493	710	755
German	193	188	277	275	403	393	601	589
Gap (%)								
Core	0	0	0	0	0	0	0	0
Basic	1.8	0	2.1	0	0.5	0	0.8	0
Large	4.1	0	4.1	0	2.2	0	2.7	0
USA–Can	3.7	0	3.2	0	7.0	0	6.2	0
German	11.9	0	5.42	0	8.9	4.8	7.2	5.1

Table 4
α Performance evaluation in all networks and scenarios.

α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
#Best	2	0	3	1	**5**	4	1	2	3
Score	72	68	58	70	**32**	57	66	60	38
Dev (%)	1.68	1.08	1.03	1.31	**0.43**	1.19	1.41	1.19	0.89

Table 5
Determination of an adequate size for set *ES* using the DPR algorithm.

$\|ES\|$	4	6	8
#Best	9	**17**	4
Score	13	**2**	22
Dev (%)	1.77	**0.42**	2.86

a particular method and *BestVal* for that instance. Finally, we report the statistic called *Score* (Ribeiro et al., 2002).

In our first preliminary experiment, our focus is on tuning α, that is, the value that controls the access into the RCL in the construction algorithm. To this end, we construct, for each α value and network scenario, 200 solutions with GRC. The results are shown in Table 4, where the best global value for each parameter is shown in boldface (and similar for the results provided in Tables 5–7). The results report $\alpha = 0.5$ as the best method, as it provides the best value for each of the statistic parameters evaluated. Therefore, we use *GRC*($\alpha = 0.5$) in the rest of our experiments.

The next preliminary experiment is devoted to the tuning of the two parameters required to set up the VND local search algorithm, that is, *MaxSearch* and N_{MAX}. For this experiment, we make use of the so-called BRKGA tuning as proposed in Festa et al. (2010), where authors use a BRKGA algorithm so as to find adequate parameter values for a GRASP + PR heuristic. In this case, however, and due to the complexity of the instances considered, we only tune the VND parameters. Then, we perform a more exhaustive evaluation of the key PR parameters, namely the size of set *ES* and d_{th}, which manages the access to *ES*.

We test the following values for each parameter: (a) $N_{MAX} = \{5,8,12,15\}$; (b) $MaxSearch = \{15,25,40,50,70\}$. Therefore, a chromosome is defined by 2 parameters. BRKGA in this experiment makes use of: $p = 20$, $p_e = 0.2$, $p_m = 0.2$, $\rho_e = 0.7$. The fitness for each chromosome corresponds to the average obtained over five independent executions of GRC plus VND, each lasting for 56 iterations (seven iterations per thread). The automatic tuning sets parameters $N_{MAX} = 50$ and $MaxSearch = 8$ (i.e., the values with higher frequencies of occurrence).

Next, we conduct two additional preliminary experiments in order to set up the parameters corresponding to the PR procedure, namely the size of set *ES*, and the minimum distance d_{th}. First, we evaluate 4, 6, and 8 as candidate sizes for set *ES*. We note here that the size of set *ES* represents a trade-off between quality and diversity that needs to be evaluated. To this end, we run 20 independent executions of the DPR algorithm for all network scenarios, each lasting for 100 iterations. According to the statistics reported in Table 5, $|ES| = 6$ represents the best trade-off to perform PR, and hence, we consider this value for the rest of our experiments requiring the PR intensification procedure.

To analyze the impact of d_{th}, and given the fact that the maximum distance between two solutions is equal to $|\mathcal{P}^o|$, we compare the performance of three different percentages of this value as possible distance thresholds, that is, 5%, 10%, and 15%. We perform 20 independent executions of the algorithm each lasting for 200 iterations in all network scenarios. According to the statistics reported in Table 6, the best value is $d_{th} = 10\%$, and hence, we select it for the rest of our experiments. Note that both the VND and the PR parameters found will be used by both EPR and BVR algorithms.

Table 6
Study of d_{th} using the DPR algorithm.

d_{th}	$\frac{5}{100}\mathcal{P}^o$	$\frac{10}{100}\mathcal{P}^o$	$\frac{15}{100}\mathcal{P}^o$
#Best	8	**12**	6
Score	14	**6**	16
Dev (%)	1.8	**0.52**	2.1

Finally, to specify the parameters in BVR that deal with the evolution of the population, we consider the values that after some preliminary experimentation we successfully tested in Pedrola et al. (2011). These values are the ones used in Section 5.3.2, though in this case the population size is set to $p = |\mathcal{N}|$ and the chromosome structure as described in Section 4. In the next section, we compare the performance of the metaheuristic hybridizations for the RPD problem proposed in this paper with that of the MILP-based (MP1) and (R-MP1) models.

5.3.3. RPD methods performance comparison

In this final experiment, we compare the performances of all the RPD resolution methods proposed throughout this paper. Specifically, the following five algorithms configurations are executed (for each method 20 independent executions are run each with eight threads evolving in parallel):

- *BRKGA*: Run for a minimum of 1000 generations and stopped after a maximum of 200 generations without improvement.
- *BVR*: Run for 500 generations. Three different configurations for this method are tested, namely BVR (1), BVR (3) and BVR (5). The number in brackets corresponds to *GlobItr*, that is, the number of times that the evolution process is stopped so that both PR and VND can be executed. Accordingly, for each method the value of *gens* corresponds to $\lceil \frac{500}{GlobItr} \rceil$.
- *GRASP*: The GRASP multi-start phase (i.e., GRC followed by VND). Each thread performs 36 multi-start iterations.
- *DPR*: The dynamic variant of GRASP + PR, each thread is run with *LocalItr* set to 36 iterations. Note that in this case the evolutionary stage is not executed.
- *EPR*: In this algorithm, each thread is run with the parameters *LocalItr* and *GlobItr* set to 12 and 3 respectively.

The results provided in Table 7, report BVR (3) and BVR (5) as the best methods, thereby showing the benefits achieved by incorporating both VND and PR into the basic BRKGA procedure. Although one can observe that further iterations (BVR (5)) slightly increase *#Best* with respect to BVR (3), it does so at the cost of substantially increasing computation times. Further, in BVR (5), the remaining two parameters do not experience any improvement. Hence, considering that our primary objective is to approach optimality, and that in this respect both BVR (3) and BVR (5) provide the same performance, this experimental analysis reports BVR (3) as the most efficient RPD algorithm among the ones evaluated in this paper. Results obtained by all three GRASP variants also show the impact of introducing PR and in particular of the evolutionary stage. As expected, EPR is able to provide better performance, though requiring more computation time. For the sake of a fair comparison, if a MILP method obtains a *BestVal* for a particular instance, its counter of *#Best* is incremented by 20 units.

Next, in order to be able to numerically compare the heuristic results with those of the MILP methods, in Table 8, we provide the final results for the best methods in terms of the number of regenerators. In this case, the numbers shown in boldface represent the problem instances in which that particular method has not been able to at least equal the result obtained by the best of (MP1) and (R-MP1) for that particular instance. One can note that BVR (3) improves upon the MILP models in all cases except for the

Table 7
Statistic results for all RPD methods.

Method	BRKGA	BVR (1)	BVR (3)	BVR (5)	GRASP	DPR	EPR	(MP1)	(R-MP1)
#Best	40	162	**168**	175	0	0	0	0	140
Score	62	23	**1**	1	142	115	90	102	58
Dev (%)	0.35	0.1	**0.02**	0.02	5.46	2.08	1.5	1.23	2.14
Time (seconds)	100.6	75.9	**88.1**	149.9	1022.3	4008.8	5138.1	24 hours	8236

Table 8
RPD methods results (number of regenerators deployed).

		BRKGA	BVR (1)	BVR (3)	EPR	(MP1)	(R-MP1)
Basic	*A*	384	384	383	**386**	385	384
	B	571	571	571	571	572	574
	C	**582**	579	579	**582**	580	579
	D	**874**	873	873	**875**	874	873
Large	*A*	655	653	652	657	658	661
	B	985	984	983	986	998	1139
	C	937	935	934	**940**	942	937
	D	1419	1418	1417	1423	1432	1430
USA–Can	*A*	**265**	264	264	**265**	269	264
	B	400	400	400	**401**	402	400
	C	465	463	462	471	472	493
	D	701	700	700	703	710	755
German	*A*	**189**	188	188	**191**	193	188
	B	**277**	**276**	**276**	**282**	277	275
	C	**396**	**394**	393	**419**	403	393
	D	589	587	587	**626**	601	589

Table 9
Rank results and pairwise differences of the RPD algorithms ($CD = 3.76$).

Method *Avg. Rank*	BVR (3) (2.125)	BVR (5) (2.125)	BVR (1) (2.93)	BRKGA (4.53)	(R-MP1) (4.781)	EPR (5.937)	(MP1) (6.5)	DPR (7.25)	GRASP (8.875)
BVR (3)	–	0	0.82	2.41	2.66	3.81	4.38	5.13	6.75
BVR (5)	–	–	0.82	2.41	2.66	3.81	4.38	5.13	6.75
BVR (1)	–	–	–	1.59	1.84	3	3.56	4.31	5.94
BRKGA	–	–	–	–	0.25	1.4	1.97	2.72	4.34
(R-MP1)	–	–	–	–	–	1.15	1.72	2.47	4.09
EPR	–	–	–	–	–	–	–	1.31	2.93
(MP1)	–	–	–	–	–	–	–	0.75	2.38
DPR	–	–	–	–	–	–	–	–	1.63
GRASP	–	–	–	–	–	–	–	–	–

German (*B*) scenario, and that even EPR reports best values in some instances compared to the MILP methods. In fact, considering all the problem instances, BVR (3) using much less computation time, provides a reduction of 272 regenerators when compared with (R-MP1), 106 with (MP1), and of 60 if compared with the best of both MILP methods. These results also allow us to analyze the effect of the problem complexity on the performance of the best BRKGA-based algorithm (BVR (3)), and the best GRASP-based method (EPR). Using the values reported in Table 8, we compute the difference between the results of both heuristics (i.e., regs (EPR)-regs (BVR (3))). Whilst the total difference in the number of regenerators for scenarios (*A,B*) is 22, it increases up to 94 for scenarios (*C,D*). Further, we confirm that the difference between both algorithms does not change significantly under a load variation, as the aggregated difference is 56 and 60 respectively, for scenarios (*A,C*) and (*B,D*).

Hence, considering only solution quality, these results report that BRKGA-based heuristics are more appropriated for the RPD problem because of both their simple decoding algorithm and ability to obtain high-quality solutions in short computational times. Further, we can notice that despite the benefits generated by both VND and PR in GRASP are noticeable, they are not able to match the fast genetic evolution. The size of both $\mathcal{P}^o$ and $\mathcal{C}^*$ make it necessary to introduce memory into the process, and in this aspect, BVR clearly outperforms EPR thanks to the joint operation of the genetic evolution and PR.

5.4. Statistical analysis of the results

In this section, we aim at confirming the results obtained in the last section. To this end, we conduct tests to analyze whether the performance differences found among the RPD algorithms are statistically significant.

Demsar (2006) tackled the issue of statistical tests for comparison of algorithms on multiple problem instances. In this paper, we use the non-parametric Friedman Test (Friedman, 1940), and the Nemenyi Post hoc test (Nemenyi, 1963) to evaluate our $k = 9$ algorithms under the $N = 16$ different problem instances as reported in Table 1.

The Friedman test ranks the algorithms for each problem instance separately. By comparing the average ranks of the algorithms, the statistical significance of differences between the methods is examined. In this work, we use the enhanced version of the Friedman test developed by Iman and Davenport (1980), which uses the test statistic $\mathcal{F}_F$ based on the *F*-distribution with degrees of freedom $((k-1),(k-1)(N-1))$. If the equivalence of the algorithms is rejected, the Nemenyi post hoc test is applied in order to perform pairwise comparisons. The average ranks are reported in Table 9 for each of the nine different RPD methods. Given the ranking obtained, BVR (3) and BVR (5) represent the best performing algorithms closely followed by BVR (1). Then, we find BRKGA and (R-MP1) providing quite similar performance. The next group is formed by EPR and (MP1), and finally the DPR and GRASP algorithms reporting the worst results. The next step is devoted to

analyzing the statistical significance of differences between these ranks. In our scenario, the $\mathcal{F}_F$ test statistic is distributed according to the F-distribution with (8,120) degrees of freedom. In this case, $\mathcal{F}_F = 44.543$, a value which is fairly greater than the critical value 2.66 obtained with the F-distribution and a significance level of the test $\alpha = 0.01$. Taking into account this test, a significant difference between the performance of the different RPD methods exists, and thus, the equivalence can be rejected. Hence, we can proceed with the Nemenyi Post hoc test to determine differences between the average ranks for every pair of algorithms. To this end, we compute the critical difference (CD) (see Demsar, 2006) between algorithm ranks. For a significance value $\alpha = 0.01$, we have that $CD = 3.76$. Given the results provided in Table 9, we can clearly identify two different groups. A first group consisting of BRKGA-based methods plus (R-MP1), and a second one with the remaining algorithms. Note that algorithms within a group differ from best to worst in less than CD. Hence, the Friedman and Nemenyi tests confirm the results obtained in the experimental analysis and reinforce the conclusion that BRKGA-based algorithms are effective metaheuristics for the RPD problem.

6. Conclusions

The purpose of this paper has been the development of efficient metaheuristic methods to solve the RPD problem found in translucent sub-wavelength switching optical networks. This problem deals with the minimization of the number of regenerators required to mitigate the impact of the PLIs in the network. Due to both their high cost and power-consumption, this problem is of great interest for network operators which strive for cost-effective, energy-efficient architectures. We have developed two different hybridized metaheuristics based on both GRASP and BRKGA algorithms. Moreover, we have introduced VND and PR into their basic procedures, and finally implemented the EPR and BVR algorithms to solve the RPD problem. We have compared their results with those of both an optimal and a heuristic MILP formulation using CPLEX. Among them, BVR reported the best overall results in all the scenarios evaluated except for one, thereby standing as an efficient and competitive algorithm to be taken into consideration for the planning and design of future sub-wavelength OTNs. Further, we also observed that genetic methods such as BVR are particularly recommended for the RPD problem because of their efficient decoding algorithm and fast genetic evolution.

Acknowledgement

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under Grant agreement n 247674 (STRONGEST Project), the Spanish Ministry of Science through the FPU Program and the DOMINO Project (TEC2010-18522). It was also supported in part by the NSF Engineering Research Center for Integrated Access Networks (CIAN) (sub-award Y503160).

References

Agrawal, G.P., 2002. Fiber-Optic Communications Systems. Wiley-Interscience, New York, NY, USA., pp. 490–497.
Ben Yoo, S.J., 2006. Optical packet and burst switching technologies for the future photonic internet. IEEE/OSA Journal of Lightwave Technology 24 (12), 4468–4492.
Demsar, J., 2006. Statistical comparison of classifiers over multiple data sets. Journal of Machine Learning Research 7, 130.
Feo, T.A., Resende, M.G.C., 1995. Greedy randomized adaptive search procedures. Journal of Global Optimization 6, 109–133.
Festa, P., Gonçalves, J.F., Resende, M.G.C., Silva, R.M.A., 2010. Automatic tuning of GRASP with path-relinking heuristics with a biased random-key genetic algorithm. In: Festa, P. (Ed.), Experimental Algorithms, Lecture Notes in Computer Science, vol. 6049, pp. 338–349.
Friedman, F., 1940. A comparison of alternative tests of significance for the problem of m rankings. The Annals of Mathematical Statistics 11 (1), 86–92.
Glover, F., 1996. Tabu search and adaptive memory programing: advances, applications and challenges. In: Barr, R.S., Helgason, R.V., Kennington, J.L. (Eds.), Interfaces in Computer Science and Operations Research. Kluwer Academic Publishers, pp. 1–75.
Gonalves, J., Resende, M.G.C., 2010. Biased random-key genetic algorithms for combinatorial optimization. Journal of Heuristics 17 (5), 487–525.
González de Dios, O., Bernini, G., Zervas, G., Basham, M., 2011. Framework for GMPLS and path computation support of sub-wavelength switching optical networks. IETF draft, draft-gonzalezdedios-subwavelength-framework-00.
Hansen, P., Mladenovic, N., Moreno-Pérez, J.A., 2010. Variable neighbourhood search: methods and applications. Annals of Operations Research (1), , 367–40.
Höller, H., Melián, B., Voß, S., 2008. Applying the pilot method to improve VNS and GRASP metaheuristics for the design of SDH/WDM networks. European Journal of Operational Research 191, 691–704.
Iman, R.L., Davenport, J.M., 1980. Approximations of the critical region of the Friedman statistic. Communications in Statistics-Theory and Methods 9, 571–595.
Laguna, M., Martí, R., 1999. GRASP and path relinking for 2-layer straight line crossing minimization. INFORMS Journal on Computing 11, 44–52.
Lee, Y., Bernstein, G., Imajuku, W., 2011. Framework for GMPLS and PCE control of wavelength switched optical networks (WSONs). IETF draft, draft-ietf-ccamp-rwa-wson-framework-12.
Liang, Z., Chaovalitwongse, W.A., Cha, M., Moon, S.B., 2010. Redundant multicast routing in multilayer networks with shared risk resource groups: complexity, models and algorithms. Computers & Operations Research 37, 1731–1739.
Manousakis, K., Kokkinos, P., Christodoulopoulos, K., Varvarigos, E., 2010. Joint online routing, wavelength assignment and regenerator allocation in translucent optical networks. IEEE/OSA Journal of Lightwave Technology 28 (8), 1152–1163.
Martins, A.X., Duhamel, C., Mahey, P., Saldanha, R.R., de Souza, M.C., 2012. Variable neighborhood descent with iterated local search for routing and wavelength assignment. Computers & Operations Research 39 (9), 2133–2141.
Nemenyi, P.B., 1963. Distribution-Free Multiple Comparisons. Ph.D. thesis, Princeton University, New Jersey.
Noronha, T., Resende, M.G.C., Ribeiro, C.C., 2010. A biased random-key genetic algorithm for routing and wavelength assignment. Journal of Global Optimization 50 (3), 503–518.
Palmieri, F., Fiore, U., Ricciardi, S., 2010. A GRASP-based network re-optimization strategy for improving RWA in multi-constrained optical transport infrastructures. Computer Communications 33 (15), 1809–1822.
Pedrola, O., Careglio, D., Klinkowski, M., Solé-Pareta, J., 2011. Regenerator placement strategies for translucent OBS networks. IEEE/OSA Journal of Lightwave Technology 29 (22), 3408–3420.
Pedrola, O., Careglio, D., Klinkowski, M., Solé-Pareta, J., 2011. Offline routing and regenerator placement and dimensioning for translucent OBS networks. IEEE/OSA Journal of Optical Communications and Networking 3 (9), 651–666.
Pedrola, O., Ruiz, M., Velasco, L., Careglio, D., Gonzlez de Dios, O., Comellas, J., 2012. A GRASP with path-relinking heuristic for the survivable IP/MPLS-over-WSON multi-layer network optimization problem. Computers & Operations Research. http://dx.doi.org/10.1016/j.cor.2011.10.026.
Ramamurthy, B., Fenq, H., Datta, D., Heritage, J.P., Mukherjee, B., 1999. Transparent vs. opaque vs. translucent wavelength-routed optical networks. In: Proceedings of IEEE/OSA OFC 1999, San Diego, CA, USA.
Reeves, C.R., Yamada, T., 1998. Genetic algorithms, path relinking and the flowshop sequencing problem. Evolutionary Computation 6, 230–234.
Reis, R., Ritt, M., Buriol, L., Resende, M., 2010. A biased random-key genetic algorithm for OSPF and DEFT routing to minimize network congestion. International Transactions in Operational Research 18 (3), 295–423.
Resende, M.G.C., Ribeiro, C.C., 2005. GRASP with path-relinking: Recent advances and applications. In: Ibaraki, T., Nonobe, K., Yagiura, M. (Eds.), Metaheuristics: Progress as Real Problem Solvers. Springer, pp. 29–63.
Resende, M.G.C., Werneck, R.F., 2004. A hybrid heuristic for the p-median problem. Journal of Heuristics 10, 59–88.
Resende, M.G.C., Martí, R., Gallego, M., Duarte, A., 2010. GRASP and path relinking for the max-min diversity problem. Computers & Operations Research 37, 498–508.
Ribeiro, C.C., Uchoa, E., Werneck, R.F., 2002. A hybrid GRASP with perturbations for the Steiner problem in graphs. INFORMS Journal on Computing 14, 228–246.
Rochette, M., Blows, J.L., Eggleton, B.J., 2006. 3R optical regeneration: an all-optical solution with BER improvement. OSA Optics Express 14 (14), 6414–6427.
Shen, G., Tucker, R.S., 2007. Translucent optical networks: the way forward. IEEE Communications Magazine 45 (2), 48–54.
Skorin-Kapov, N., 2007. Routing and wavelength assignment in optical networks using bin packing based algorithms. European Journal of Operational Research 177, 1167–1179.
Villegasa, J.G., Prinsa, C., Prodhona, C., Medaglia, A.L., Velasco, N., 2012. A GRASP with evolutionary path relinking for the truck and trailer routing problem. Computers & Operations Research 38 (9), 1319–1334.
Zhang, G.Q., Lai, K.K., 2006. Combining path relinking and genetic algorithms for the multiple-level warehouse layout problem. European Journal of Operational Research 169, 413–425.