RAIRO-Oper. Res. 53 (2019) 1513–1528
https://doi.org/10.1051/ro/2018072

RAIRO Operations Research
www.rairo-ro.org

ON THE PROBLEM OF MINIMIZING THE COST WITH OPTICAL DEVICES IN WAVELENGTH DIVISION MULTIPLEXING OPTICAL NETWORKS: COMPLEXITY ANALYSIS, MATHEMATICAL FORMULATION AND IMPROVED HEURISTICS

Daniel Morais dos Reis[1], Natã Goulart[2], Thiago F. Noronha[3,*] and Sérgio Ricardo de Souza[4]

Abstract. The Fiber Installation Problem (FIP) in Wavelength Division Multiplexing (WDM) optical networks consists in routing a set of lightpaths (all-optical connections) such that the cost of the optical devices necessary to operate the network is minimized. Each of these devices is worth hundreds of thousands of dollars. In consequence, any improvement in the lightpath routing may save millions of dollars for the network operator. All the works in the literature for solving this problem are based on greedy heuristics and genetic algorithms. No information is known on how good are the solutions provided by these heuristics compared to the optimal solution. Besides, no proof that the problem is NP-Hard can be found. In this paper, we prove that FIP is NP-Hard and also present an Integer Linear Programming (ILP) formulation for the problem. In addition, we propose an implementation of the Iterated Local Search (ILS) metaheuristic to solve large instances of the problem. Computational experiments carried out on 21 realistic instances showed that the CPLEX solver running with our ILP formulation was able to solve 11 out of the 21 instances to optimality in less than two minutes. These results also showed that the ILS heuristic has an average optimality gap of 1% on the instances for which the optimal solution is known. For the other instances, the results showed that the proposed heuristic outperforms the best heuristic in the literature by 7%.

Mathematics Subject Classification. 90C59.

Received June 28, 2017. Accepted September 4, 2018.

Keywords. Routing and Wavelength Assignment, optical network optimization, heuristics and metaheuristics, iterated local search.

[1] Departamento de Informática, Gestão e Design, Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG) – Campus V, Divinópolis, Brazil.

[2] Departamento de Tecnologia em Engenharia Civil, Computação e Humanidades, Universidade Federal de São João Del Rei (UFSJ), Ouro Branco, Brazil.

[3] Departamento de Ciência da Computação, Universidade Federal de Minas Gerais (UFMG), Belo Horizonte, Brazil.

[4] Departamento de Computação, Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG) – Campus II, Belo Horizonte, Brazil.

[*]Corresponding author: tfn@dcc.ufmg.br

1. Introduction

The Wavelength Division Multiplexing (WDM) technology allows a more efficient use of the huge capacity of optical fibers. It permits the simultaneous transmission of different optical signals along the same fiber, provided they are multiplexed with different wavelengths. An all-optical point-to-point connection between two nodes is called a *lightpath*. Two lightpaths may be multiplexed with the same wavelength, provided they do not share any common fiber [9, 11].

Given an optical network and a set of lightpaths to be established, the problem of Routing and Wavelength Assignment (RWA) in WDM optical networks consists in routing the set of lightpaths and assigning a wavelength to each lightpath such that lightpaths whose routes share a common fiber are assigned to different wavelengths. Variants of RWA are characterized by different optimization criteria and traffic patterns, see *e.g.*, [8, 22, 30].

We consider in this paper a variant of the RWA problem in which all lightpath requests are known beforehand. The problem of Fiber Installation in optical networks (FIP) consists of routing all lightpaths requests, minimizing the cost of the optical devices necessary to operate the network. These devices are worth hundreds of thousands of dollars and thus, even a small percentage of the network costs is not negligible. In the current paper, we consider the three most expensive devices used in WDM optical networks, *i.e.*, the Reconfigurable Optical Add/Drop Multiplexer (ROADM), the Optical Amplifier (OA) and the Optical Transponder (OT). These devices are described in Section 2.

This problem was firstly introduced in [3] as part of an effort at Bell Labs to develop optimization methods for optical networking [3]. Besides, it is an extension of the patent *System for Routing and Wavelength Assignment in Wavelength Division Multiplexing Optical Networks* [23]. This work advances the problem dealt in this patent by also considering the cost of optical devices in the optimization process.

There are only heuristic algorithms to solve FIP in the literature, and no information is given on how good are the solutions provided by these heuristics compared to optimal solutions. Besides, no result concerning the complexity of the problem was found in the literature. In order to present a complete analysis concerning FIP, this paper introduces a proof that FIP is NP-Hard. Besides, we provide an Integer Linear Programming (ILP) formulation of this problem, in order to find optimal solutions for small and medium size instances, and an Iterated Local Search (ILS) heuristic to find near optimal solutions for large size instances. Computational experiments were carried out on networks obtained from the SNDlib project, which provides realistic network design test instances. The results showed that the ILS heuristic proposed in this paper has an average optimality gap of 1% on the networks for which the optimal solution was found. For the other networks, the results showed that the proposed heuristic outperforms the best heuristic in the literature by 7%, on average.

The remainder of this paper is organized as follows. The next section presents a formal definition of FIP. Then, we propose an ILP formulation for FIP in Section 3 and provide a proof that this problem is NP-Hard in Section 4. Next, Section 5 shows the state-of-the-art heuristics for FIP. Following, we describe our ILS heuristic in Section 6. Finally, computational experiments are presented and discussed in Section 7, and concluding remarks are drawn in the last section.

2. Problem definition

The optical network is modeled by an undirected graph $G = (V, E)$, where V is the set of nodes and E is the set of edges. The edges in E represent the optical fibers available to route the lightpaths, and the nodes in V represent the Optical Add/Drop Multiplexers that connect the optical fibers. Let also $R = \{r_1, r_2, \ldots, r_{|R|}\}$ be the set of lightpath requests, with $r_i \in R$ defined by a tuple $\langle s_i, d_i, w_i \rangle$, where $s_i \in V$ and $d_i \in V$ are the endnodes of the lightpath request r_i, and w_i is the number of wavelengths used by r_i.

In [3], the authors argue that the routing and the wavelength assignment subproblems should be tackled separately. They suggest that solution approaches for FIP should focus only on the routing subproblem, since the wavelength assignment subproblem can be solved by any algorithm for the well studied k-coloring problem [7, 13, 14, 24, 26] in a conflict graph $\hat{G} = (\hat{V}, \hat{E})$. The vertices in this graph correspond to the lightpaths in R, *i.e.*, $\hat{V} = \{\hat{v}_i : r_i \in R\}$, and there is an edge $\hat{e} \in \hat{E}$ between each pair of vertices whose associated routes share a common fiber. Besides, the value of k is equal to the number of wavelength supported by each optical fiber. This subproblem consists in finding a valid k-coloring in $\hat{G} = (\hat{V}, \hat{E})$, where two adjacent nodes do not have the same color. The lightpaths whose corresponding nodes have the same color are assigned to the same wavelength. Therefore, no two lighpaths that share a fiber have the same wavelength. Since k-coloring is a well studied combinatorial problem, and the number of wavelengths supported by the current optical devices are in the order of hundreds of wavelengths, we adopt this approach and focus on the routing problem.

The Fiber Installation Problem (FIP), as defined in [3], consists in deciding a path (route) from s_i to d_i in G for each $r_i \in R$, minimizing the network cost (F_{NC}), described in equation (2.5). In [3], F_{NC} is computed from the three most expensive optical devices necessary to operate the network, which are described below.

The *Reconfigurable Optical Add/Drop Multiplexer* (ROADM) allows the optical signal to travel through a network node from any incoming fiber to any outgoing fiber with the same wavelength. This device reduces drastically the signal overhead, since it eliminates the need of optical-electrical-optical conversions at intermediate nodes of the lightpath. One ROADM can have many arms, each one supporting a WDM fiber. The cost of a ROADM is linearly proportional to its number of arms. For each WDM fiber used, we need two ROADM arms placed at the endnodes of the fiber. As one ROADM has to be installed at every network node, the optimization model only accounts for the number of ROADM arms. Therefore, for each WDM fiber used, it is charged by:

$$c_1 = 2C_{\text{ROADM}}, \tag{2.1}$$

where C_{ROADM} is the cost of one ROADM arm. The cost of the fiber itself is not considered, because it is assumed that the fibers are already deployed and are inactive or operating with an outdated technology. The number of available fibers in the link $[i,j] \in E$ is denoted by $b_{ij} \in \mathbb{N}$.

The *Optical Amplifier* (OA) compensates the chromatic dispersion, which is a phenomenon that causes the decrease in power of the optical signal, and may prevent it to arrive at its final destination. The cost of an OA can be approximated in the first order by distance-based rules as follows: for each WDM fiber used, it is charged

$$c_2 = \frac{C_{\text{OA}}}{L_{\text{OA}}} \tag{2.2}$$

per kilometer of fiber, where C_{OA} is the cost and L_{OA} is the reach of an OA.

The *Optical Transponder* (OT) converts electrical signals coming from outside the network to optical signals and converts optical signals leaving the network into electrical signals. A lightpath originates in an OT passes through one or more ROADM and ends in another OT. However, there is a limit on the length of a lightpath, given by L_{OT}. If a lightpath is longer than L_{OT} kilometers, it must be split into two lightpaths, and another pair of OT is needed. The cost of an OT is modeled as follows. For each wavelength used in each WDM fiber, it is charged by:

$$c_3 = 2\frac{C_{\text{OT}}}{L_{\text{OT}}} \tag{2.3}$$

per kilometer of fiber, where C_{OT} is the cost of an OT.

Given the route for each lightpath in R, let ω_{ij} be the sum of the wavelengths carried by all the lightpaths that cross the WDM link $[i,j] \in E$, and l_{ij} be the length of this link. Then, the device cost of transmitting ω_{ij} wavelengths through this link is given by:

$$F^{ij}_{\text{WDM}}(\omega_{ij}, l_{ij}) = c_1 \left\lceil \frac{\omega_{ij}}{\mu} \right\rceil + c_2 l_{ij} \left\lceil \frac{\omega_{ij}}{\mu} \right\rceil + c_3 \omega_{ij} l_{ij} \tag{2.4}$$

where μ is the number of wavelengths supported by a WDM fiber, and $\left\lceil \frac{\omega_{ij}}{\mu} \right\rceil$ is the number of WDM fibers needed to carry ω_{ij} wavelengths. This function could also model the cost of deploying the optical fibers, as well as the cost of other optical devices, such was wavelength converters. The cost of deploying an optical fiber can be added to c_1, and the cost of the other optical devices can be added to c_1, c_2, or c_3 depending on the characteristic of optical device.

Finally, the total network cost F_{NC} is given by the sum of the device cost for each individual network link, that is:

$$F_{\mathrm{NC}} = \sum_{[i,j]\in E} F^{ij}_{\mathrm{WDM}}. \tag{2.5}$$

3. Problem formulation

In this section, we formulate FIP as an Integer Linear Programming (ILP) problem. Given an instance $I^{\mathrm{FIP}} = \langle G = (V,E), R, c_1, c_2, c_3, \mu, b_{ij}, l_{ij}\rangle$, we define constants $\pi^r_i \in \{-1,0,1\}$, such that $\pi^r_i = -1$ if vertex $i \in V$ is the source of the lightpath request $r \in R$; $\pi^r_i = 1$, if i is the destination of r, and $\pi^r_i = 0$, otherwise.

The formulation has two sets of decision variables. The variables $y_{ij} \in \mathbb{N}$ represent the number of optical fibers required to accommodate the traffic that passes through the edge $[i,j] \in E$. Besides, for each edge $[i,j] \in E$, we have two variables $x^r_{ij} \in \{0,1\}$ and $x^r_{ji} \in \{0,1\}$, such that $x^r_{ij} = 1$, if the lightpath of request $r \in R$ goes from i to j; $x^r_{ji} = 1$ if the lightpath of request $r \in R$ goes from j to i; and $x^r_{ij} + x^r_{ji} = 0$, otherwise. We note that the number of optical fibers per link is equal to the smallest integer greater than or equal to the ratio between the number of wavelengths crossing the link and the number of wavelengths supported by each fiber, that is

$$y_{ij} = \left\lceil \frac{\sum_{r\in R} w_r (x^r_{ij} + x^r_{ji})}{\mu} \right\rceil, \tag{3.1}$$

where w_r is the number of wavelengths necessary to carry the traffic associated to the lightpath request r and μ is the number of wavelengths supported by each fiber.

The ILP formulation for FIP is given by expressions (3.2)–(3.7).

$$\text{Minimize} \sum_{[i,j]\in E} \left[c_1 y_{ij} + c_2 l_{ij} y_{ij} + c_3 l_{ij} \sum_{r\in R} w_r \left(x^r_{ij} + x^r_{ji}\right)\right] \tag{3.2}$$

subject to:

$$\sum_{[j,i]\in E} x^r_{ji} - \sum_{[i,j]\in E} x^r_{ij} = \pi^r_i \ , \quad \forall r \in R, \forall i \in V \tag{3.3}$$

$$y_{ij} \geq \frac{\sum_{r\in R} \left[w_r (x^r_{ij} + x^r_{ji})\right]}{\mu} \ , \quad \forall [i,j] \in E \tag{3.4}$$

$$y_{ij} \leq b_{ij} \ , \quad \forall [i,j] \in E \tag{3.5}$$

$$y_{ij} \in \mathbb{N} \ , \quad \forall [i,j] \in E \tag{3.6}$$

$$x^r_{ij}, x^r_{ji} \in \{0,1\} \ , \quad \forall [i,j] \in E \tag{3.7}$$

The objective function (3.2) minimizes the sum of $F^{i,j}_{\mathrm{WDM}}$ for all edges $[i,j] \in E$. The flow conservation constraint for each request $r \in R$ is given by equation (3.3). The inequalities (3.4), together with expressions (3.2) and (3.6), enforce that the value of the y_{ij} variables are according to equation (3.1). The constraint (3.5) ensures that no more fibers than the maximum number of optical fibers available is used. Finally, constraints (3.6) and (3.7) define the domain of the variables.

This mathematical formulation allows the application of exact mathematical programming techniques for the solution of FIP instances, which is used in Section 7 to assess the performance of the proposed heuristics.

4. Complexity analysis

In this section, we prove that FIP is NP-Hard by a reduction from the Set Covering Problem [15]. It is worth mentioning that other proofs could also be obtained by reductions from other network design problems such as the Multicommodity Network Loading problem [4,6] and the Capacitated Network Design problem [12,18].

Theorem 4.1. *FIP is NP-Hard.*

Proof. To prove this statement, we show that FIP is as hard as any problem in NP. To this end, we show that the NP-Complete Set Covering Problem [15] can be polynomially reduced to FIP. Consider the Set Cover Problem (SC):

Input: $I^{\rm SC} = \langle U, S, k\rangle$, where U is a set, $S \subseteq 2^U$, and $k \in \mathbb{N}$.
Question: U has a cover $B \subseteq S$ with $|B| \leq k$?

We note that B is a k-cover of U if and only if, $|B| \leq k$ and for all $u_i \in U$, there exists at least one set $B_j \in B$ such that $u_i \in B_j$.

To answer the SC question, we define a function $g(\cdot) : I^{\rm SC} \mapsto I^{\rm FIP}$ that receives an instance $I^{\rm SC} =< U, S, k >$ of SC and returns an instance $I^{\rm FIP} = \langle G = (V, E), R, c_1, c_2, c_3, \mu, b_{ij}, l_{ij}\rangle$ of FIP such that if the total network cost of the optimal solution for $I^{\rm SC}$ is smaller than or equal to k, $I^{\rm SC}$ answers *YES*; otherwise, $I^{\rm SC}$ answers *NO*. First, the set of nodes is defined as $V = \{d\} \cup V^U \cup V^S$, where:

$$V^U = \{u_i \ : \ u_i \in U\},$$
$$V^S = \{S_j \ : \ S_j \in S\},$$

and d is the destination node of all lightpaths in this instance. Next, the set of edges is defined as $E = E^{uS} \cup E^{Sd}$, where:

$$E^{uS} = \{[u_i, S_j] \ : \ (S_j \in S) \ \wedge \ (u_i \in S_j)\}, \text{ and}$$
$$E^{Sd} = \{[S_j, d] \ : \ S_j \in S\}.$$

Then, the set of lightpath requests is fixed to:

$$R = \{r_i = \langle u_i, d, 1\rangle \ : \ u_i \in U\}.$$

Constants c_1, c_2, c_3 and μ are defined respectively as $c_1 = 0$, $c_2 = 1$, $c_3 = 0$, $\mu = |U|$, and the maximum number of fibers per link is set to $b_{ij} = 1$, $\forall [i, j] \in E$.

Finally, the link length l_{ij} is set to:

$$l_{ij} = \begin{cases} 0, \forall\, [i, j] \in E^{uS} \\ 1, \forall\, [i, j] \in E^{Sd}. \end{cases}$$

An example of the network obtained from the SC instance $U = \{1, 2, 3, 4, 5\}$ and $S = \big\{S_1 = \{1, 2\}, S_2 = \{1, 2, 3\}, S_3 = \{2, 3, 4\}, S_4 = \{3, 4, 5\}, S_5 = \{4, 5\}\big\}$ is given in Figure 1. This transformation is such that if the edge $[S_j, d] \in E^{Sd}$ is used by the lightpath request $r_i \in R$, then the item $u_i \in U$ is covered by the set $S_j \in S$ in the corresponding SC instance.

The values chosen for c_1, c_2, c_3 and μ result in

$$F^{ij}_{\rm WDM}(\omega_{ij}, l_{ij}) = l_{ij} \cdot \left\lceil \frac{\omega_{ij}}{|U|} \right\rceil,$$

for all $[i, j] \in E$. Accordingly, for all $[u_i, S_j] \in E^{uS}$, $F^{u_iS_j}_{\rm WDM} = 0$, because $l_{u_iS_j} = 0$. Besides,

$$F^{S_jd}_{\rm WDM} = \left\lceil \frac{\omega_{S_jd}}{|U|} \right\rceil \leq 1,$$

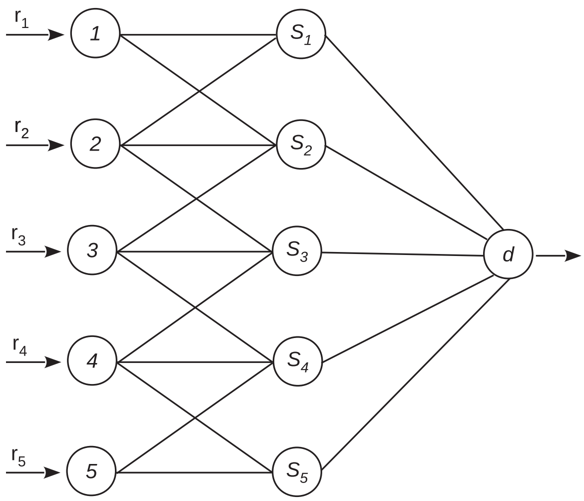

FIGURE 1. FIP network obtained with the transformation from the SC instance $U = \{1,2,3,4,5\}$ and $S = \{S_1 = \{1,2\}, S_2 = \{1,2,3\}, S_3 = \{2,3,4\}, S_4 = \{3,4,5\}, S_5 = \{4,5\}\}$.

because $l_{S_j d} = 1$ for all $[S_j, d] \in E^{Sd}$, and $\omega_{S_j d} \leq \sum_{r_i \in R} w_i = |R| = |U|$. Therefore, $F_{\text{WDM}}^{S_j d} = 0$, if $\omega_{S_j d} = 0$, *i.e.* if the edge $[S_j, d] \in E^{Sd}$ is not used by a lightpath, and $F_{\text{WDM}}^{S_j d} = 1$, if $\omega_{S_j d} > 0$, *i.e.* if this edge is used by at least one lightpath. Consequently, the cost of a solution $P = \{P_i : r_i \in R\}$ is equal to the number of edges in E^{Sd} used by the lightpaths in P.

Given the optimal solution $P^* = \{P_i^* : r_i \in R\}$ of I^{FIP}, if the cost of P^* is less than or equal to k, one can make a k-cover $B \subseteq S$ for I^{SC} with $B = \{S_j \in S : [S_j, d] \in E^{Sd} \cap E^{P^*}\}$, where $E^{P^*} = \bigcup_{P_i^* \in P^*} P_i^*$ is the set of all the edges used by the lightpaths in P^*. Since the cost of P^* is less than or equal to k, it uses at most k edges in E^{Sd}, so $|B| \leq k$, and the answer to I^{SC} in this case is *YES*.

It remains to prove that if the cost of P^* is larger than k, the corresponding SC instance answers *NO*. If the cost of P^* is larger than k and the answer to I^{SC} was *YES*, it would be possible to build a solution $\overline{P} = \{\overline{P}_i = \{[u_i, S_j], [S_j, d]\} : S_j \in B \text{ and } u_i \text{ is covered by } S_j \text{ in } B\}$. Since it was assumed that $|B| \leq k$, $\overline{P}$ would use at most k edges in E^{Sd}, and its cost would be less than or equal to k. As this contradicts the hypothesis that P^* is optimal, if the cost of P^* is larger than k, the corresponding SC instance answers *NO*, which completes the proof. □

5. Related works

This section shows the previous works concerning FIP. First, in Section 5.1, the constructive heuristic proposed in [3] are revised. Then, in Section 5.2, the Biased Random-Key Genetic Algorithm (BRKGA) proposed in [17] is summarized.

5.1. Constructive heuristics

Two heuristics were proposed in [3] for solving FIP. The so-called `Greedy` algorithm is a greedy heuristic followed by a local search procedure, while the `PSC` algorithm is similar to `Greedy`, but uses the piecewise strongly concave cost function instead of the cost presented in expression (2.4), as the cost of the link $[i, j] \in E$ for routing the lightpaths.

Algorithm 1 Pseudo-code of the Greedy heuristic for FIP.

Require: : $(G, R, \mu\ , \text{ROADM}_{\text{cost}}, \text{OT}_{\text{cot}}, \text{OA}_{\text{cot}})$
Ensure: : s
1: **for all** $r \in R$ **do**
2:     **for all** $[i, j] \in E$ **do**
3:         $C_{ij}^{\text{inc}} \leftarrow F_{\text{WDM}}^{ij}(\omega_{ij} + \alpha, l_{ij}) - F_{\text{WDM}}^{ij}(\omega_{ij}, l_{ij})$
4:     **end for**
5:     Recompute the lightpath of r according to the new incremental costs
6: **end for**
7: $s \leftarrow$ LocalSearch(1-opt, s)
8: **return** s;

5.1.1. *Greedy heuristic*

The Greedy heuristic, introduced by [3], is showed in Algorithm 1. The input parameters are: (i) the graph G with the physical topology of the network, (ii) the requests $r \in R$, in the order in which they will be evaluated, (iii) the wavelength limit per fiber μ and (iv) the costs of optical devices. At each iteration of the loop given by lines 1–6, a request $r \in R$ is allocated. In the loop given by lines 2–4, the incremental cost C_{ij}^{inc} of the function (2.4) in relation to the number of wavelengths is calculated, in the form:

$$C_{ij}^{\text{inc}} = F_{\text{WDM}}(\omega_{ij} + \alpha, l_{ij}) - F_{\text{WDM}}(\omega_{ij}, l_{ij}) \tag{5.1}$$

where α is the number of wavelengths to be added in the optical network links. This value represents the incremental cost of allocating each request. After this, the heuristic executes the Dijkstra Algorithm [10] at line 5 to find the lightpath with the least incremental cost for the request. In the following, after all the requests are allocated in their respective lightpaths, the solution s is returned and the heuristic is closed.

Once all the requests in R are routed, it is possible that cheaper routes for each lightpath may exist as a result of the additional devices installed to satisfy the subsequent requests. Therefore, these heuristics perform a local search in an 1-opt neighborhood. Let s be a feasible solution for FIP. A solution s' is a neighbor of s if and only if it differs from s by the route of at most one lightpath. The local search procedure subsequently passes over the lightpath permutation, removing the present route for the current lightpath and computing the least expensive route for this lightpath under the present circumstances (possibly the same route). After each pass over the lightpath permutation, if at least one improving neighbor is found, the procedure repeats from the beginning of the permutation. It stops when a local optimum is reached, *i.e.*, when no better solution can be obtained by re-routing a single lightpath.

5.1.2. *PSC heuristic*

While the `Greedy` heuristic uses expression (2.4) as its greedy cost function, the `PSC` heuristic uses a piecewise strongly concave cost function. Let μ be the number of wavelengths supported by an optical fiber, and p be the current pass of the heuristic over the permutation of lightpath requests in the local search procedure described above. The greedy function used by `PSC` is given by

$$\overline{F}_{\text{WDM}}^{ij}(\omega_{ij}, l_{ij}, p, P) = (c_1 + c_2 l_{ij}) \left[\frac{p}{P} \left\lceil \frac{\omega_{ij}}{\mu} \right\rceil + \frac{P-p}{P} f\left(\frac{\omega_{ij}}{\mu}\right) \right] + c_3 \omega_{ij} l_{ij} \tag{5.2}$$

where P is an integer parameter to be tuned and $f(x) = \lfloor x \rfloor + \sqrt{x - \lfloor x \rfloor}$ is the piecewise strongly concave function.

The pseudo-code of the `PSC` heuristic is shown in Algorithm 2. Note that the pseudo-code of `PSC` is similar to that of `Greedy`. The main difference is the function $\overline{F}_{\text{WDM}}^{ij}$, used to calculate the incremental cost at line 5 of this algorithm. In the first pass of the `PSC` heuristic over the lightpath request permutation, p is equal to zero. Therefore, the expression (5.2) assumes the value:

$$\overline{F}_{\text{WDM}}^{ij}(\omega_{ij}, l_{ij}, 0, P) = (c_1 + c_2 l_{ij})\, f\left(\frac{\omega_{ij}}{\mu}\right) + c_3 \omega_{ij} l_{ij}$$

At every pass of `PSC` over the lightpath permutation, the value of p is incremented and the function $\overline{F}_{\mathrm{WDM}}$ gradually converges to the function F_{WDM}. The speed of the convergence is defined by the parameter P, which was set to $P = 5$ in [3]. The experiments reported in [3] show that `PSC` outperforms `Greedy` when the traffic demands incur heavy loads on the network.

Algorithm 2 Pseudo-code of the PSC heuristic for FIP.

Require: : $(G, R, \mu\ , \mathrm{ROADM}_{\mathrm{cost}}, \mathrm{OT}_{\mathrm{cot}}, \mathrm{OA}_{\mathrm{cot}}, P)$
Ensure: : s
$p \leftarrow 0$;
while $p < P$ **do**
    for all $r \in R$ **do**
        for all $[i, j] \in E$ **do**
            $C_{ij}^{\mathrm{inc}} \leftarrow \overline{F}_{\mathrm{WDM}}^{ij}(\omega_{ij} + \alpha, l_{ij}, p, P) - \overline{F}_{\mathrm{WDM}}^{ij}(\omega_{ij}, l_{ij}, p, P)$
        end for
        Recompute the lightpath of r according to the new incremental costs
    end for
    $s \leftarrow$ LocalSearch(1-opt, s)
    $p \leftarrow p + 1$
end while
return s

5.2. Biased Random-Key Genetic Algorithm

A Biased Random-Key Genetic Algorithm (BRKGA) was proposed for solving FIP in [17]. The BRKGA metaheuristic was proposed in [16] as an extension of the genetic algorithms with random keys of [5]. Solutions are represented as vectors of randomly generated real numbers called keys. A deterministic algorithm, called a decoder, takes as input a solution vector and associates with it a feasible solution of the combinatorial optimization problem, for which an objective value or fitness can be computed. To implement the crossover operation, one parent is selected at random from a set of elite solutions (the top best solutions in the population), while the other is selected at random from the remaining non-elite solutions. Parents are allowed to be selected for mating more than once in a given generation. It is said that the selection is biased since one parent is always an elite solution and because this elite solution has a higher probability of passing its genes to the offsprings. This genetic algorithm does not make use of the standard mutation operator, where parts of the chromosomes are changed with small probability. Instead, at each generation, a fixed number of new random generated solutions are introduced in the next population.

In the BRKGA of [17], called `GA-FIP`, there is one key in the range $[0, 1]$ for each lightpath request in R. The solution, represented by a chromosome, is decoded by a two step procedure. First, the lightpath requests are sorted in non-increasing order of their key values. The resulting order is used as the permutation in which the lightpath requests are processed by the heuristic `PSC`. The solution returned by `PSC` is the solution represented by the chromosome, and the corresponding solution cost is used as the fitness of the chromosome. `GA-FIP` uses the *Parametric Uniform Crossover* scheme proposed in [29] to combine two parent solutions and produce an offspring solution. In this scheme, the offspring inherits each of its keys from the best fit of the two parents with probability 0.7 and from the least fit parent with probability 0.3.

Algorithm 3 shows the pseudo-code of GA-FIP. The input parameters are the graph G and the set of lightpath requests R. The chromosomes in the initial population are randomly generated in line 1. Each iteration of the loop in lines 2–12 computes a new generation, until a stopping condition is met. At each generation, the population is partitioned in line 3 into two sets: TOP and REST. Consequently, the size of the population is $|\mathrm{TOP}| + |\mathrm{REST}|$. Subset TOP contains the best solutions in the population. Subset REST is formed by two disjoint subsets: MID and BOT, with subset BOT being formed by the worst elements on the current population. The chromosomes in TOP are simply copied to the population of the next generation in line 4, while the elements in BOT are replaced by newly created mutants in line 5. The remaining $|\mathrm{MID}| = |\mathrm{REST}| - |\mathrm{BOT}|$ elements of the new

Algorithm 3 Pseudo-code of the GA-FIP procedure.

```
Require: G and R
 1: Add |TOP| + |REST| randomly generated chromosomes to the initial population
 2: while Stopping condition is not met do
 3:     Split current population in sets TOP and REST according to their fitness
 4:     Copy to the next generation the TOP portion of the population
 5:     Add |BOT| randomly generated chromosomes to the next generation
 6:     for n from 1 to |MID| do
 7:         Let p1 be a randomly selected chromosome from TOP
 8:         Let p2 be a randomly selected chromosome from REST
 9:         p3 ← Parametric-Uniform-Crossover(p1, p2)
10:         Add p3 to the next generation.
11:     end for
12: end while
13: return the best solution in the last generation
```

population are obtained by crossover with one parent randomly chosen from TOP in line 7 and the other from REST in line 8. The resulting chromosome p_3 is added to the new population in lines 9 and 10. As the TOP chromosomes are always copied to the next generation, when the algorithm stops the best solution can be found in the population of the last generation and is returned in line 13.

6. Iterated local search for FIP

In this section, we propose two heuristics, named ILS-GREEDY and ILS-PSC, both based on Iterated Local Search (ILS) metaheuristic [19, 20]. The ILS metaheuristic performs a search in the space of local optima. It starts from a local optimum s and, at each iteration, performs a *perturbation* in this solution followed by a local search in the perturbed solution, in order to produce a new local optimum s'. The perturbation consists of a random move in a given neighborhood. It must be strong enough to generate a solution outside the region of attraction of the current local optimum, otherwise the following local search would lead back to the current local optimum. However, it should not be too strong so that the heuristic resembles a random search. If s' meets some acceptance criterion, the procedure is repeated from s'; otherwise, the heuristic resumes from s. More details about ILS can be found in [20].

Our approach for solving FIP consists of a greedy construction heuristic followed by an implementation of the ILS metaheuristic. We propose two versions of the algorithm: one based on the `Greedy` heuristic, called `ILS-GREEDY`, and another based on the `PSC` heuristic, called `ILS-PSC`. These heuristics make use of a perturbation procedure based on the k-opt neighborhood. A random solution in this neighborhood is obtained by removing k lightpaths from the current solution s, and then reinserting them in the solution with the `Greedy` heuristic (in the case of `ILS-GREEDY`) or with the `PSC` heuristic (in the case of `ILS-PSC`). The lightpaths are selected for removal with probability proportional to the number of wavelengths that they require. The size k of the neighborhood is a parameter to be tuned. The local search applied to the perturbed solution is the same 1-opt local search proposed in [3], and described in Section 5.1.1. At each ILS iteration, the local search passes over the lightpath requests, removing the present route for the current lightpath and computing the least expensive route under the present circumstances. It stops when no better solution can be obtained by re-routing a single lightpath.

The pseudo-code of the ILS procedure for FIP is given in Algorithm 4. The algorithm starts from the initial local optimum s^i, that consists of the solution provided by the the `Greedy` heuristic (in the case of `ILS-GREEDY`) or by the `PSC` heuristic (in the case of `ILS-PSC`). First, the current solution s and the best known solution s^* are initialized with s^i. The loop in lines 2–11 is performed until a stopping condition is met. A new local optimum s' is obtained by performing the k-opt perturbation procedure to s in Line 3, followed by the 1-opt local search procedure to the perturbed solution in Line 4. The best known solution is updated in lines 5–7, and

Algorithm 4 Pseudo-code of the ILS procedure for FIP.

Require: : (s^i) a solution generated from `Greedy` (Algorithm 1) or `PSC` (Algorithm 2) heuristics
Ensure: : s^*

```
1: s, s* ← s^i;
2: while Stopping condition is not met do
3:     s' ← Perturbation(K-opt, s);
4:     s' ← LocalSearch(1-opt, s');
5:     if s' better than s* then
6:         s* ← s';
7:     end if
8:     if AcceptanceCriterion(s, s') then
9:         s ← s';
10:    end if
11: end while
12: return s*;
```

the acceptance criterium states that the current local optimum s is replaced by s' in lines 8–10, if the cost of the latter is smaller than that of the former. The ILS procedure stops when a maximum elapsed time is reached or when a solution as good as a given target is found. Then, the best known solution is return in Line 12.

7. Computational experiments

The heuristics `Greedy`, `PSC`, `GA-FIP`, `ILS-GREEDY` and `ILS-PSC` were coded in `C++` and compiled with the *GNU GCC* compiler version 4.4.3. As in [17], the size of sets TOP, REST and BOT in `GA-FIP` was set to $0.25 \cdot |R|$, $0.70 \cdot |R|$ and $0.05 \cdot |R|$, respectively. Preliminary experiments set the value of k to $0.3 \cdot |R|$ in the k-opt perturbation procedure of `ILS-GREEDY` and `ILS-PSC`. As a benchmark, the integer linear programming formulation of FIP, given by expressions (3.2)–(3.7), was coded in OPL language and run over CPLEX version 12.5. The random number generator used was the `Mersenne Twister` [21]. All experiments showed in this paper were performed on an Intel (R) Core (TM) 2 Quad Q9650 with 3.00 GHz of clock speed and 4 GB of RAM.

The 21 instances used in the experiments are part of the SNDlib library [25]. This library provides realistic instances with geographical information of the location of the nodes, edges and the total requests between pairs of nodes. In our experiments, traffic demands for each request were randomly generated between 0 and 99 wavelengths with the purpose of generating challenging instances. Besides, as suggested in [3], we set $C_{\text{ROADM}} = 1000$, $C_{\text{OA}} = 0.5 \cdot C_{\text{ROADM}}$, and $C_{\text{OT}} = 0.1 \cdot C_{\text{ROADM}}$. In addition, the number of wavelengths supported by each fiber and the maximum number of fibers per link were set respectively to $\mu = 100$ and $b_{ij} = 8$, for all $[i, j] \in E$, which are typical values for these parameters. These instances are used in the three sets of experiments described below.

7.1. First set of experiments

The first set of experiments aims at evaluating the OPL/CPLEX branch-and-bound algorithm, based on formulation (3.2)–(3.7), within a time limit of 7200 s. The results are presented in Table 1. The first four columns display the name, the number of nodes, the number of edges, and the number of lightpath requests for each instance, respectively. The fifth and sixth columns show, respectively, the lower bound lb and the upper bound ub obtained by the algorithm. For the sake of a better visualization, these values are displayed divided by 10^3. When no feasible solution is found within the time limit, the corresponding entries are filled with a dash. The seventh column shows the relative optimality gap $\Big((ub - lb)/lb\Big)$. The number of nodes explored and the running time of the algorithm are reported in the last two columns, respectively.

It can be seen from Table 1 that, using the ILP formulation (3.2)–(3.7), it was possible to find optimal solutions for 11 (out of the 21) instances in less than two minutes. Feasible solutions were obtained for only three out of the 10 instances with 25 or more nodes, which shows that this approach can only efficiently solve to optimality small instances of FIP, and that heuristics are needed to tackle larger instances.

TABLE 1. Performance of the OPL/CPLEX branch-and-bound algorithm, based on formulation (3.2)–(3.7), within a time limit of 7200 s.

Name	|V|	|E|	|R|	lb	ub	Gap (%)	Nodes	Time(s)
dfn-bwin	10	45	45	340.32	340.32	0.00	97	3.13
dfn-gwin	11	47	55	449.64	449.64	0.00	98	4.87
di-yuan	11	42	22	5 485.83	5 485.83	0.00	98	0.45
pdh	11	34	24	117.71	117.71	0.00	97	0.64
abilene	12	15	66	2 120.42	2 120.42	0.00	99	1.95
polska	12	18	66	731.19	731.19	0.00	99	3.15
nobel-ger	14	21	121	706.51	706.51	0.00	140	20.69
nobel-us	14	21	91	4 393.38	4 393.38	0.00	99	5.58
atlanta	15	22	105	29 087.24	29 087.24	0.00	145	3.73
newyork	16	49	120	38 864.12	38 864.12	0.00	100	5.54
geant	22	36	231	8 125.05	8 125.05	0.00	165	76.00
france	25	45	300	1 354.30	85 102.72	6 183.89	141	7 200.00
norway	27	51	351	–	–	–	–	7 200.00
sun	27	102	65	34 591.02	43 481.01	25.70	88	7 200.00
nobel-eu	28	41	378	2 241.30	5 173.29	130.82	107	7 200.00
cost266	37	57	666	–	–	–	–	7 200.00
janos-us-ca	39	122	741	–	–	–	–	7 200.00
pioro40	40	89	780	–	–	–	–	7 200.00
germany50	50	88	662	–	–	–	–	7 200.00
zib54	54	81	626	–	–	–	–	7 200.00
ta2	65	108	807	–	–	–	–	7 200.00

7.2. Second set of experiments

The second set of experiments aims at assessing the average solution cost of GREEDY, PSC, GA-FIP, ILS-PSC and ILS-GREEDY. As in [17], the stopping condition of the heuristics was set at 600 s, and each heuristic was run 10 times with different seeds for the pseudo-random number generator. For the sake of a better visualization, the network costs are displayed divided by 10^3.

First, we report the results for the 11 instances whose optimal solutions are known in Table 2. The first column displays the name of the instance. The second and third columns show the average solution cost and the average relative optimality gap for the GREEDY heuristic. The same data is respectively reported for PSC in the fourth and fifth columns, for GA-FIP in the sixth and seventh columns, for ILS-PSC in the eighth and ninth columns, and for ILS-GREEDY in the last two columns. It can be seen from Table 2 that the average optimality gaps of Greedy and PSC are approximately 30%, while that of GA-FIP is close to 10%. The smallest average optimality gaps were obtained by ILS-PSC (1.90%) and ILS-GREEDY (1.51%). The largest optimality gap of ILS-PSC and ILS-GREEDY was respectively 4.45% and 4.48%, on the instance nobel-ger, while that of GA-FIP was 27.18%, on the same instance. These results point out to the fact that the heuristics proposed in this paper provide near optimal solutions that are significantly better than those obtained by the heuristics found in the literature.

Next, we compare the performance of ILS-PSC, ILS-GREEDY, and the best heuristic in the literature of FIP (GA-FIP) for the 10 instances whose optimal solutions are not known. The results are presented in Table 3. The first column displays the name of each instance, while the second column shows the average cost of the solutions found by GA-FIP (c^{ga}). The third and the forth columns give, respectively, the average solution cost of ILS-PSC (c^{ils}) and the average relative improvement of ILS-PSC over GA-FIP ($(c^{ga} - c^{ils})/c^{ga}$). The same data is reported for ILS-GREEDY in the fifth and sixth columns. We note that the larger is the relative improvement of ILS-PSC or ILS-GREEDY, the better are their solutions compared to those of GA-FIP. Besides, negative values for this metric

Table 2. Average solution cost and relative optimality gap of GREEDY, PSC, GA-FIP, ILS-PSC and ILS-GREEDY on the 11 instances whose optimal solutions are known.

Instance	GREEDY		PSC		GA-FIP		ILS-PSC		ILS-GREEDY	
name	cost	gap (%)	cost	gap (%)	cost	gap (%)	cost	gap (%)	cost	gap (%)
dfn-bwin	473.30	39.07	481.64	41.52	340.33	0.00	340.33	0.00	340.33	0.00
dfn-gwin	575.24	27.93	578.98	28.76	449.65	0.00	449.65	0.00	449.65	0.00
di-yuan	6 471.73	17.97	6 525.64	18.95	5 485.84	0.00	5 485.84	0.00	5 485.84	0.00
pdh	180.59	53.42	182.04	54.65	117.71	0.00	117.71	0.00	117.71	0.00
abilene	2 979.38	40.51	3 035.07	43.13	2 590.25	22.16	2 207.32	4.10	2 189.48	3.26
polska	852.76	16.63	855.73	17.03	822.79	12.53	747.86	2.28	747.57	2.24
nobel-ger	971.59	37.52	980.45	38.77	898.52	27.18	737.93	4.45	738.15	4.48
nobel-us	5 279.62	20.17	5 376.69	22.38	5 044.05	14.81	4 546.84	3.49	4 395.06	0.04
atlanta	39 050.65	34.25	39 188.86	34.73	34 912.89	20.03	29 751.74	2.28	30 295.45	4.15
newyork	45 709.15	17.61	45 658.95	17.48	44 665.30	14.93	39 403.30	1.39	39 143.69	0.72
geant	9 526.18	17.24	10 187.53	25.38	8 790.05	8.18	8 363.40	2.93	8 263.77	1.71
Average		29.30		31.16		10.89		1.90		1.51

Table 3. Comparison of ILS-PSC and ILS-GREEDY against GA-FIP on the 10 instances with more than 25 nodes for which the optimal solution is not known.

Instance	GA-FIP	ILS-PSC		ILS-GREEDY		$\frac{i^g - i^p}{i^p}$
name	cost	cost	imp (%)	cost	imp (%)	
france	72 192.85	69 974.08	3.07	66 335.40	8.11	10.99
norway	138 812.99	122 448.89	11.79	128 545.39	7.40	18.78
sun	42 572.00	36 715.21	13.76	36 700.49	13.79	14.93
nobel-eu	4 471.13	4 010.32	10.31	3 980.93	10.96	8.00
cost266	6 578.96	6 767.09	−2.86	6 165.12	6.29	13.65
janos-us-ca	8 606.66	8 182.40	4.93	8 192.41	4.81	8.77
pioro40	161 409.03	157 274.41	2.56	156 974.41	2.75	12.05
germany50	3 105.48	2 907.48	6.38	2 910.74	6.27	5.78
zib54	123 350.82	124 785.03	−1.16	124 585.03	−1.00	12.93
ta2	144 892.50	130 476.71	9.95	127 750.03	11.83	16.73
Average:			5.87		7.12	12.26

indicate that GA-FIP outperformed the other heuristics. The last column display the ratio $((i^g - i^p)/i^p)$ between the number of iteration performed by ILS-GREEDY with 600 s of running time (i^g) and that of ILS-PSC (i^p). The larger is this ratio, the greater is the number iterations of ILS-GREEDY with respect to that of ILS-PSC. It can be seen that the average relative improvement of ILS-PSC over GA-FIP was 5.87%, while that of ILS-GREEDY was 7.12%. GA-FIP outperformed ILS-PSC on instances cost266 and zib54, while ILS-GREEDY was outperformed by GA-FIP only on instance zib54. The better performance of ILS-GREEDY over ILS-PSC can be explained by the fact that the former runs, on average, 12.26% more iterations than the latter, which significantly increases the chances of ILS-GREEDY finding better solutions than those of ILS-PSC.

Then, we plot the evolution of the best solution cost of GA-FIP, ILS-PSC and ILS-GREEDY throughout the 600 s of processing time. Results are reported for instances zib54 and pioro-40, because (i) zib54 is the only instance for which GA-FIP outperformed both ILS-PSC and ILS-GREEDY and (ii) pioro40 is an instance that reflects the characteristic behavior of the remaining 20 instances. It can been seen from Figure 2 that the results of ILS-PSC and ILS-GREEDY for instance zib54 are competitive with those of GA-FIP, despite the fact that the latter outperformed the formers in this instance. On the other hand, one can see from Figure 3 that the average cost of the best solution found by ILS-GREEDY for instance pioro40 is always smaller than that of

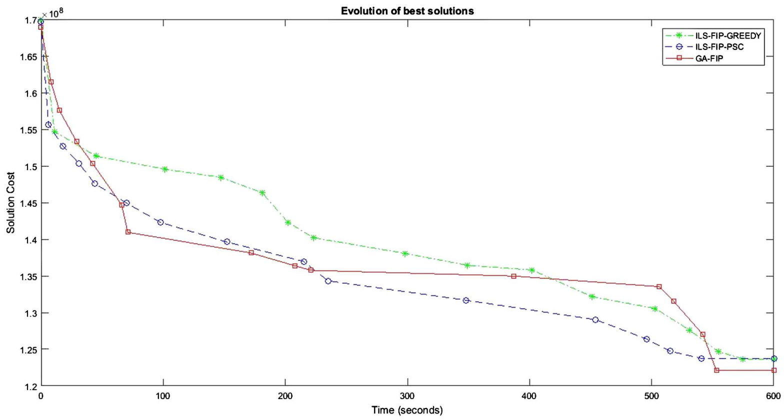

FIGURE 2. Evolution of the best solutions found by `GA-FIP` `ILS-PSC` and `ILS-GREEDY` for instance zib54.

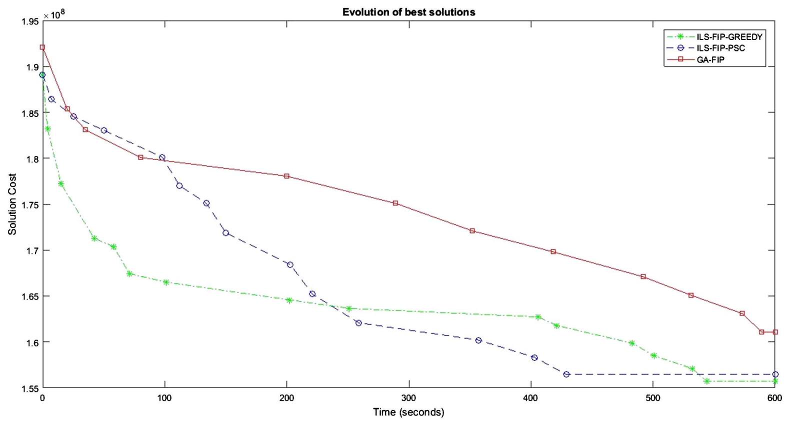

FIGURE 3. Evolution of the best solutions found by `GA-FIP`, `ILS-PSC` and `ILS-GREEDY` for instance pioro40.

GA-FIP, after the first seconds of running time. These results indicate that ILS-GREEDY is the best among the heuristics studied in this work. This hypothesis is further investigated in the next set of experiments.

7.3. Third set of experiments

In the third and last set of experiments, we evaluate the empirical probability of having `GA-FIP`, `ILS-PSC`, and `ILS-GREEDY` finding a solution with a cost at least as good as some target value within a given running time. The heuristics were run 200 times on each instance, with different seeds for the pseudo-random number generator.

First, we display time-to-target plots [2], which give, on the vertical axis, the probability that an algorithm finds a solution at least as good as a given target value within a given running time, shown on the horizontal axis. The plots show the empirical probability distributions of the CPU time required by each heuristic to find the target solution value. To plot this distribution, we followed the methodology described in [1, 2]. We associate a probability $p_i = (i - \frac{1}{2})/200$ with the ith smallest running time t_i and plot the points $z_i = (t_i, p_i)$, for $i = 1, \ldots, 200$. The more to the left a plot is, the better the algorithm corresponding to it. For the same reasons

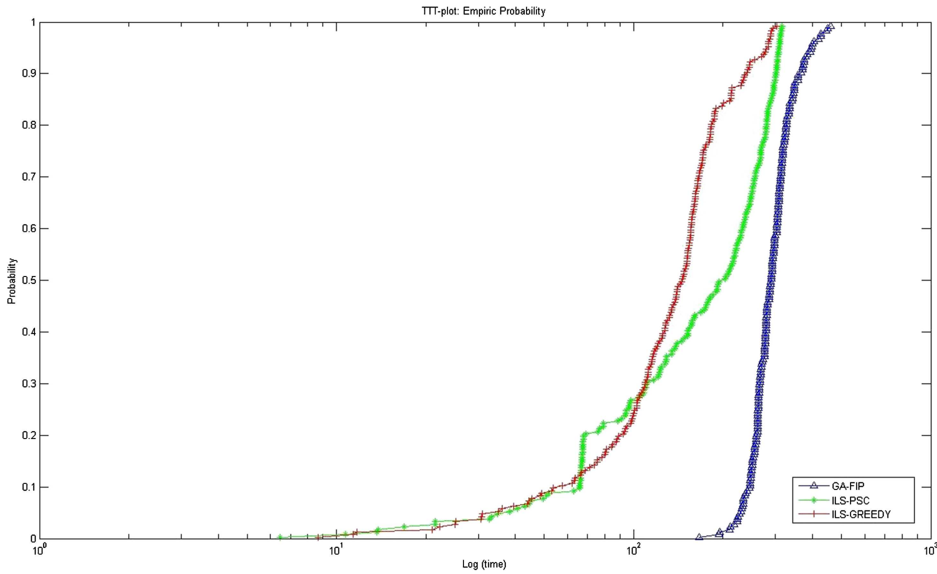

Figure 4. *TTT-plot* `GA-FIP`, `ILS-PSC` and `ILS-GREEDY` for pioro40 instance.

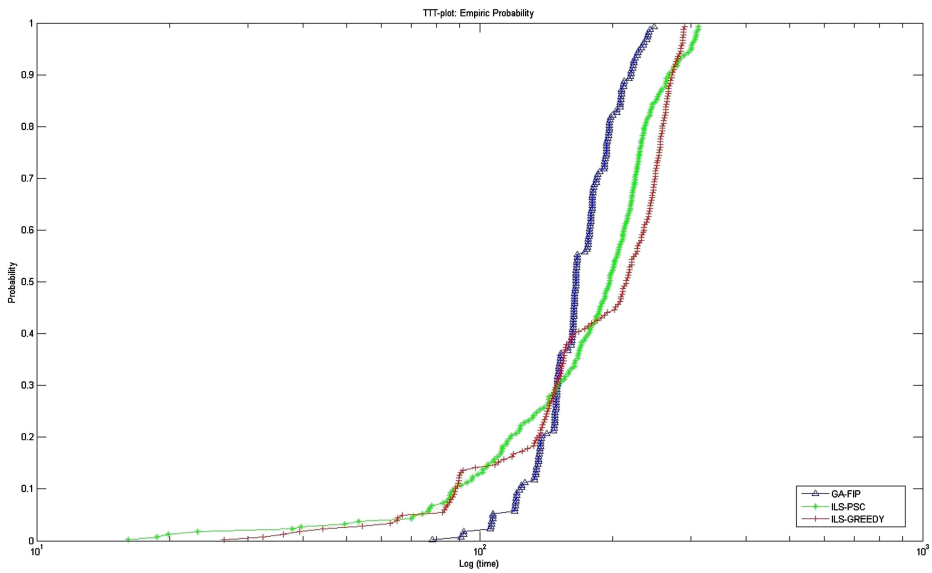

Figure 5. *TTT-plot* `GA-FIP`, `ILS-PSC` and `ILS-GREEDY` for zib54 instance.

previously explained, the results are reported only for instances pioro40 and zib54 and are plotted in Figures 4 and 5, respectively. It can be seen in Figure 4 that, after 100 s, `ILS-GREEDY` have a higher probability of reaching the target than the other two heuristics on instance pioro40. The probability of `ILS-GREEDY` reaching the target around 300 s is approximately 90%, while that is 50% for PSC and 1% for GA-FIP. On the other hand, one can see in Figure 5 that, for instance zib54, `ILS-PSC` and `ILS-GREEDY` have a higher probability of reaching the target than `GA-FIP` before 200 s. However, the probability of the former reaching the target around 300 s is approximately 50%, while that of the latter is 90%.

Then, we perform a direct numerical comparison of the time-to-target-value of `GA-FIP`, `ILS-PSC` and `ILS-GREEDY`, as suggested by [27]. Let t^{ga}, t^{g}, and t^{psc} be respectively the continuous random variables representing the time needed by `GA-FIP`, `ILS-GREEDY` and `ILS-PSC` to find a solution as good as a given target value. Let also $P(t^{psc} \leq t^{ga})$ be the empirical probability that `ILS-PSC` converges faster than `GA-FIP`, and $P(t^{g} \leq t^{ga})$ be the empirical probability that `ILS-GREEDY` converges faster than `GA-FIP`. The values of $P(t^{psc} \leq t^{ga})$ and $P(t^{g} \leq t^{ga})$ were computed using the software made available in [28]. The results show that, for instance pioro40, $P(t^{psc} \leq t^{ga}) = 0.837$ and $P(t^{g} \leq t^{ga}) = 0.960$, *i.e.*, the probability that ILS-PSC converges faster than GA-FIP is greater than 83%, whereas the probability that `ILS-GREEDY` converges faster than GA-FIP is greater than 96%. For instance zib54, the results show that $P(t^{psc} \leq t^{ga}) = 0.393$ and that $P(T^{g} \leq t^{ga}) = 0.360$. Therefore, despite the fact that `GA-FIP` outperforms both `ILS-PSC` and `ILS-GREEDY` in this single instance, the latter have still a probability close to 40% of converging faster to the target solution than `GA-FIP`.

8. Concluding remarks

In this paper, we presented a study on the Fiber Installation Problem in WDM optical Networks. We proved that the problem is NP-Hard and proposed an ILP formulation that allows solving to optimality 11 out of the 21 realistic instances tested. We presented two heuristics, based on Iterated Local Search, for tackling large instances of the problem. The computational experiments showed that best results were obtained by our `ILS-GREEDY` heuristic, which has an average optimality gap of 1% on the instances whose optimal solutions are known. Besides, this heuristic found solutions 7.12% better, on average, than those obtained by the best existing heuristic in the literature.

Future works may focus on developing better heuristics and exact algorithms for solving larger instances of FIP. Since each of the optical devices considered is worth hundreds of thousands of dollars, any reduction in the optimality gap of the network cost can result in savings of millions of dollars. Besides, alternative models and algorithms that include the cost of other optical devices can also be studied. In addition, models and algorithms that solve the routing and the wavelength assignment problems together can also be investigated.

Acknowledgements. The authors thank the Federal Center of Technological Education of Minas Gerais, the Federal University of São João Del Rei and the Federal University of Minas Gerais. This work was partially supported by the Brazilian National Council for Scientific and Technological Development (CNPq), the Foundation for Support of Research of the State of Minas Gerais (FAPEMIG), and Coordination for the Improvement of Higher Education Personnel (CAPES) - Finance Code 001.

References

[1] R. Aiex, M.G.C. Resende and C.C. Ribeiro, Probability distribution of solution time in GRASP: an experimental investigation. *J. Heuristics* **8** (2002) 343–373.

[2] R. Aiex, M.G.C. Resende and C.C. Ribeiro, TTT-plots: a perl program to create time-to-target plots. *Optim. Lett.* **1** (2007) 355–366.

[3] S. Antonakopoulos and L. Zhang, Heuristics for fiber installation in optical network optimization. In: *Proceedings of 2007 IEEE Global Telecommunications Conference. Washington, USA* (2007) 2342–2347.

[4] P. Avella, S. Mattia and A. Sassano, Metric inequalities and the network loading problem. *Dis. Optim.* **4** (2007) 103–114.

[5] J.C. Bean, Genetic algorithms and random keys for sequencing and optimization. *ORSA J. Comput.* **2** (1994) 154–160.

[6] D. Bienstock, S. Chopra, O. Günlük and C. Tsai, Minimum cost capacity installation for multicommodity network flows. *Math. Prog.* **81** (1998) 177–199.

[7] J.A. Bondy and U.S.R. Murty, Graph Theory with Applications. Macmillan, London (1976).

[8] J.S. Choi, N. Golmie, F. Lapeyrere, F. Mouveaux and D. Su, A functional classification of routing and wavelength assignment schemes in DWDM networks: static case. In: *Proceedings of the 7th International Conference on Optical Communication and Networks, Paris* (2000) 1109–1115.

[9] Cisco. Introduction to DWDM technology. *Technical Report.* Cisco Systems Inc., San Jose, CA (2001).

[10] E.W. Dijkstra, A note on two problems in connection with graphs. *Numer. Math.* **1** (1959) 269–271.

[11] A. Dutta, N. Dutta and M. Fujiwara, WDM Technologies: Optical Networks. Elsevier, Amsterdam (2004).

[12] A. Frangioni and B. Gendron, $0-1$ reformulations of the multicommodity capacitated network design problem. *Dis. Appl. Math.* **157** (2009) 1229–1241.

[13] Y. Frota, N. Maculan, T.F. Noronha and C.C. Ribeiro, A branch-and-cut algorithm for partition coloring. *Networks* **55** (2010) 194–204.

[14] P. Galinier and A. Hertz, A survey of local search methods for graph coloring. *Comput. Oper. Res.* **33** (2006) 2547–2562.

[15] M.R. Garey and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York, NY, USA (1990).

[16] J.F. Gonçalves and M.G.C. Resende, Biased random-key genetic algorithms for combinatorial optimization. *J. Heuristics* **17** (2011) 487–525.

[17] N. Goulart, L.G.S. Dias, S.R. de Souza and T.F. Noronha, Biased random-key genetic algorithm for fiber installation in optical network optimization. In: *2011 IEEE Congress on Evolutionary Computation. New Orleans* (2011) 2267–2271.

[18] K. Holmberg and D. Yuan, A lagrangian heuristic based branch-and-bound approach for the capacitated network design problem. *Oper. Res.* **48** (2000) 461–481.

[19] H.R. Lourenço, O.C. Martin and T. Stützle, Iterated local search. In: Handbook of Metaheuristics, edited by F. Glover, G. Kochenberger and F.S. Hillier, 1st edition. Springer (2003) 320–353.

[20] H.R. Lourenço, O.C. Martin and T. Stützle, Iterated Local Search: Framework and Applications. 2nd edition. Springer US, Boston, MA (2010) 363–397.

[21] M. Matsumoto and T. Nishimura, Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul.* **8** (1998) 3–30.

[22] T.F. Noronha, M.G.C. Resende and C.C. Ribeiro, A biased random-key genetic algorithm for routing and wavelength assignment. *J. Global Optim.* **50** (2011) 503–518.

[23] T.F. Noronha, M.G.C. Resende and C.C. Ribeiro, System for Routing and Wavelength Assignment in Wavelength Division Multiplexing Optical Networks (2014). US8693871B2.

[24] T.F. Noronha and C.C. Ribeiro, Routing and wavelength assignment by partition coloring. *Eur. J. Oper. Res.* **171** (2006) 797–810.

[25] S. Orlowski, R. Wessäly, M. Pióro and A. Tomaszewski, Sndlib 1.0 – survivable network design library. *Networks* **55** (2010) 276–286.

[26] R. Read, The number of fc-colored graphs on labelled nodes. *Can. J. Math.* **12** (1960) 410–414.

[27] C.C. Ribeiro, I. Rosseti and R. Vallejos, Exploiting run time distributions to compare sequential and parallel stochastic local search algorithms. *J. Global Optim.* **54** (2012) 405–429.

[28] C.C. Ribeiro, I. Rosseti and R. Vallejos, TTTplots-compare: a perl program to compare time-to-target plots or general runtime distributions of randomized algorithms. *Optim. Lett.* **9** (2014) 601–614.

[29] W. Spears and K. De Jong, On the virtues of parameterized uniform crossover. In: Proceedings of the Fourth International Conference on Genetic Algorithms. San Mateo, CA (1991) 230–236.

[30] H. Zang, J.P. Jue and B. Mukherjee, A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks. *Opt. Netw. Mag.* **1** (2000) 47–60.