

A Hybrid Firefly-Genetic Algorithm for Planning of Optical Transport Networks

Bruno Quirino de Oliveira, Marcos Antônio de Sousa and Flávio Henrique Teles Vieira

Abstract—The network design is a well-known combinatorial optimization problem that is classified as NP-Hard. The main purpose of design is often the same, to allocate and size available resources in the most possible efficient way in terms of budget, considering optimization models oriented towards minimizing costs. In this paper, it is proposed to use a hybrid optimization method called Hybrid Firefly-Genetic Algorithm to solve the Integer Linear Programming (ILP) problem, for Optical Transport Network (OTN) planning, considering cost minimization. The method combines the discrete firefly algorithm with the standard genetic algorithm. To apply this method, first it is proposed a mathematical formulation for the optical network planning problem. Two other relevant novelties of the proposed model are: 1 - network modeling that considers a traffic interest matrix in which each predicted demand can be drained from its generating node (origin) to one of the possible network overflow nodes (different destination nodes); this modeling allows the dimensioning of a transport network capable of serving different service profiles at each of its nodes; 2 - network resources that are efficiently allocated with determination of the capacity of each transmission system according to the best solutions for the traffic flow links. This allocation process allows the model to value those solutions with better resource sharing to the detriment of the use of several parallel links operating opportunely with idleness in each one. The results show that the proposed algorithm outperforms other metaheuristics described in the literature.

Index Terms—Algorithm Genetic, Algorithm Firefly, Network Optimization, Optical Network Design.

I. INTRODUCTION

THE emergence of new technologies and services has imposed substantial changes on telecommunication systems. [1–3]. The immediate impact of the deployment and use of these technologies arises with the need to have a telecommunications network with high capacity for data transmission. The Optical Transport Network/Dense Wavelength Division Multiplexing (OTN/DWDM) enables a flexible network infrastructure with high transmission capacity [4]. OTN also provides fault isolation management with

advanced techniques for solving any operational issues [5].

Under these circumstances, the optical network is constantly being subjected to technological innovations in order to take better advantage of the current infrastructure for high transmission rates. On the other hand, as the network becomes more complex, the efficient use of its resources becomes a problem of great concern [6].

The research on algorithms to optimize resource capacity is a promising direction to improve decision support systems designed to assist in network planning [7].

In this scenario, it is natural that different optimization problems have been proposed in the literature for the design of optical transport networks. The applied methodologies in each case are extremely influenced by the network's coverage, the transmission technology to be used and the sources of available information [8] [9]. However, a common feature among these optimization problems is that they absolutely have a traffic matrix that needs to be routed from the source nodes to the destination nodes [10–11].

Nature-inspired metaheuristics are powerful tools for solving NP-hard combinatorial optimization problems. These methods are based on existing mechanisms in nature's biological phenomena [12]. For optical networks, it is possible to highlight the use of the genetic algorithm [13–15] and swarm-based algorithms [16–18], such as the Particle Swarm Optimization algorithm (PSO) [19], ant colony [20] and firefly algorithm [21–22].

With the explosive growth of demands, from various service sources, the performance of the transportation network in the face of possible failures has become a field of study of great relevance. In [23], a heuristic algorithm based on Integer Linear Programming (ILP) [24] is proposed to solve the routing problem in the OTN/DWDM networks context. The application involves multiple network scenarios in relation to topology, traffic distribution and available transmission formats in order to quantify the efficiency benefits of deploying flexible grid formats.

The work of [25] presents a bi-objective modelling using ILP

Bruno Quirino de Oliveira. School of Electrical, Mechanical and Computer Engineering, Federal University of Goiás, Brazil and Engineering School, Pontifical Catholic University of Goiás, Brazil (e-mail: brquirino@gmail.com).

Marcos Antônio de Sousa. School of Electrical, Mechanical and Computer Engineering, Federal University of Goiás, Brazil and Engineering School, Pontifical Catholic University of Goiás, Brazil (e-mail: masousa.damasio@gmail.com).

Flávio H. Teles Vieira. School of Electrical, Mechanical and Computer Engineering, Federal University of Goiás, Brazil (e-mail: flaviohtv@gmail.com).

Digital Object Identifier: 10.14209/jcis.2020.26

for the coding and routing assignment problem in OTN/WDM optical networks with dedicated protection. The objectives are to achieve the lowest cost of routing and, at the same time, employ a minimum number of coding nodes. The proposed formulation uses a weighting method to combine the two objectives into an integrated method and also provides a rigorous analysis in the configuration of the weight coefficients to capture the desired priority of the individual objectives.

For the Elastic Optical Network (EON) flow allocation problem the work [26] proposes two optimization-based metaheuristics, one by the particle swarm algorithm and the other by the taboo search and is represented by ILP. In the paper [27] it is proposed an optimization method based on the Artificial Bee Colony algorithm (ABC) modeled by ILP.

The papers [28–29] present hybrid algorithm proposals, based on the firefly algorithm, using the genetic algorithm crossover and mutation operators, one for the TSP (Traveling Salesman Problem) and another for the CFLP (Capacitated Facility Location Problem), respectively. The crossover operator is used to help the firefly algorithm explore the search for better solutions more efficiently and the mutation operator is used to increase the diversity of fireflies in the population.

This paper aims to develop a decision support system to be used in the strategic planning of optical transport networks. As in [23][24][25][26][27], it is also considered the use of ILP in this work. More specifically, we propose an integer linear programming model 0-1 to model the optical transport network, which is solved using the exact method, the bio-inspired genetic and firefly algorithms and a Hybrid Firefly-Genetic Algorithm. The model selects optimal paths from a set of pre-candidate paths available for the flow of the predicted traffic matrix. The dimensioning of the resources is performed aiming at a minimum cost, that is related to the possibilities of transmission equipment allocation in the nodes and allocation of optical link and regenerators in the network links. Unlike the approaches proposed in [23] [25] [26] [27], one of the remarkable characteristics of our proposal is the dimensioning of the network through a process of efficient resource allocation plus the possibility of draining demands considering multiple overflow nodes.

It is worth mentioning that the modelling proposed in this work allows us to contemplate the attendance of a particular traffic matrix, not yet addressed in the literature. The commonly evaluated matrices are made up of pairs of nodes indicating the source node and destination node [10–11, 23]. In this work, both the developed ILP and the proposed bio-inspired algorithms for the resolution contemplate the possibility of applying a demand traffic matrix with its source node, but with a diversity of destination nodes.

This approach is intended to cover those OTN/DWDM transport network scenarios in which the system operator has the possibility to meet several points of long-distance demand within its area of operation. Therefore, the proposed approach allows the system operator to transfer overflow traffic to another network, that can be carried out by more than one external interconnection point.

II. THE PROBLEM

The problem of planning optical transport networks is complex and difficult to solve [30], [31]. The demands of the traffic interest matrix must be disposed between the nodes through links with an associated capacity. In this context, the emphasis of planning lies in determining the optimal strategy to accommodate the set of demands.

In this work, the flow network model (graphs) [24] is used to represent the problem of planning the optical transport network. Fig. 1 shows an example of a graph, $G = (V, LINK, DEM)$, where V indicates the set of nodes, $LINK$ the set of arcs and DEM the set of demands to be drained by the network. In this work, the arc-path approach is considered [24].

The main elements of this representation are:

Demand nodes: Demand nodes are concentration points and demand generators for the service of the system users. These nodes must be physically associated with a transmission system. Each node is identified by a number $[i]$. The example network of Fig. 1 has five demand nodes and seven links.

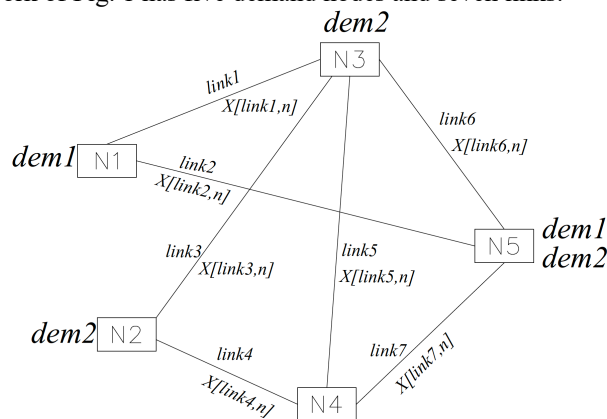


Fig. 1. Transport network representation - arc-path.

Links (arcs) candidates: each link in the network represents the connection between two demand nodes. The demand flow drained by the links is used to guarantee the fulfillment of the demand requirements between the network's nodes. The demand-flow links that connect all the demand nodes to each other are identified by a name and an index, such as *link2*, that identifies the link connecting demand nodes 1 and 5. The total capacity of each link, represented by the allocated transmission equipment capacity, must be able to meet the demand flow chosen to be disposed of by that link. The choice of the capacity of each link is represented by a binary integer type decision variable, for example, $X[link2, n]$, that represents the choice (or not) of a transmission system capacity $[n]$ to be allocated in *link2*.

Predicted demand: One of the network's design goals is to meet demand between the network nodes. Each demand is characterized by being our destination and the traffic volume that must be routed through the network. This traffic volume can be expressed in multiples of some base routing unit or even in terms of the required transmission rate. Between the end nodes, one node represents the source node of the demand and the other node the destination node. In the example network depicted by Fig. 1, two profiles of predicted demand are

considered: *dem1* (from node 1 to node 5) and *dem2* (from node 2 to node 3 or node 5).

Candidate paths: a path is defined by a sequence of adjacent nodes without repetition whereby a demand flow can pass, each demand flow being able to use one or more distinct paths. For each predicted demand between two nodes of the network, a set of paths is specified that can be used to flow the demand from its source node to its destination node.

For the example network of Fig. 1, where two profiles of demands are specified, the possible sets of paths to meet the demands *dem1* and *dem2* are indicated in Table I. Note that the *dem2* demand profile modeling allows the traffic flow to two different overflow nodes: *N3* (through paths p_1, p_2, p_3) or *N5* (through paths p_4, p_5, p_6). Note that the set of paths in Table I is not complete, being just examples of paths for *dem1* and *dem2* demands. This choice is made through binary type decision variables, $Y[\text{demand}, \text{path}]$, that specifies the choice (or not) of the path (e.g. path p_1) to flow out the demand (e.g. demand *dem1*).

TABLE I. EXAMPLE OF PATHS FOR DRAINING THE DEMANDS

Demand	Source	Destiny	Paths	Links
<i>dem1</i>	<i>N1</i>	<i>N5</i>	p_1	link2
			p_2	link1 → link6
			p_3	link3
<i>dem2</i>	<i>N2</i>	<i>N3</i>	p_2	link4 → link5
			p_3	link4 → link7 → link6
			p_4	link4 → link7
	<i>N5</i>	p_5	link3 → link6	
		p_6	link4 → link5 → link6	

In studies of medium and large networks, that is the case of the present work, it is essential that the application process of the *k-paths* takes place in an automated way and that it meets some performance criteria. Thus, for each demand source node, a set of shortest *k-paths* is identified for each of the possible destination nodes for this demand. The metric used for the *k-paths* application was the automatic generation using a modified version of the *k-shortest* algorithm [32]. The total distance, in *km*, between the source node and the possible target nodes is used as a decisive factor in the *k-shortest* algorithm.

III. MATHEMATICAL MODEL

In this section, the mathematical formulation for the optical network planning problem is presented. The network is seen as a set of nodes and arcs. The optimization model is based on integer linear programming and uses the arc-path approach [24].

The ILP model uses the following notation:

LINK: set formed by all the arcs of the network used to interconnect the demand nodes;

O_{LINK}: set of OTN/DWDM modules (capacities), $[n]$, of the candidate OTN/DWDM transmission systems on the arc $[i] \in LINK$;

TN: set formed by the possible destination nodes (for example, the overflow nodes of the network) for each network demand. This allows to evaluate scenarios in which each source node may have the demand met by more than one

destination node. For each source node, a set of *k-paths* is generated for each predicted destination node;

DEM: set formed by all the predicted demands to be met by the network. Each element of this set is formed by identifying the traffic interest of one source node for the $[t] \in TN$ possible destination nodes;

P_d : set of candidate paths to meet demand $[d] \in DEM$. This set should contain paths that allow the demand flow $[d]$ for the $[t] \in TN$ possible destination nodes;

Ω_i : set consisting of all the paths that need to use the *link* $[i]$ to drain their demand flow;

$Y_{[i][k][j]}$: a binary variable that counts the use of the path $[k] \in P_j$ to meet demand $[j] \in DEM$;

$X_{[i][n]}$: binary variable that represents the choice of the OTN/DWDM transmission system of $[n] \in O_{LINK}$ capacity, candidate in the link $[i] \in LINK$;

$C_{[i][n]}$: cost of the OTN/DWDM transmission system capacity (in *Gbps*) $[n] \in O_{LINK}$, candidate in the link $[i] \in LINK$. $C_{[i][n]}$ encompass the regenerator cost, if necessary. This fact is evaluated during the application process of the *k-paths*;

$dem_{[j]}$: predicted demand $[j] \in DEM$, in *Gbps*, to be met between the source node and the $[t] \in TN$ possible destination nodes;

$Cap_{[n]}$: the capacity of the OTN/DWDM transmission system, of modularity $[n] \in O_{LINK}$ in *Gbps*;

r : average optical network cost per *km*;

$l_{[i]}$: length in *km* of the *link* $[i] \in LINK$;

ILP 0-1 can then be formally formulated as:

$$\text{Min} \sum_{[i] \in LINK} \sum_{[n] \in O_{LINK}} (C_{[i][n]} + r \cdot l_{[i]}) X_{[i][n]} \quad (1)$$

$$\sum_{[n] \in O_{LINK}} Cap_{[n]} X_{[i][n]} - \sum_{[k] \in \Omega_i} dem_{[j]} Y_{[j][k]} \geq 0, \forall [i] \in LINK \quad (2)$$

$$\sum_{[k] \in P_j} Y_{[j][k]} = 1 \forall [j] \in DEM \quad (3)$$

$$\sum_{[n] \in O_{LINK}} X_{[i][n]} \leq 1, \forall [i] \in LINK \quad (4)$$

The objective function (1) accounts for cost of the network. The product $C_{[i][n]} X_{[i][n]}$ refers to the transmission equipment allocation costs, while $r \cdot l_{[i]} X_{[i][n]}$ notes the costs associated with the optical network segment.

The constraints of technical capacity (2) occur in each link provided by the planner for the demands flow. They ensure that the capacity of the transmission system allocated to the link is sufficient to drain the entire demand flow that uses that link. This set of constraints allows adapting the model to address some specificities of the used technology in the optical transmission system.

The set of flow constraints (3) ensures that each demand predicted to be obeyed uses a single path, which favors the network operation. It is worth mentioning the possibility of evaluating multiple destinations for each demand. The process of generating the *DEM* set, with each element consisting of a source node and *TN* destination nodes, makes it possible to

evaluate this type of scenario.

The set of exclusivity constraints (4) ensures that only one optical transmission system must be allocated on each link. The goal is to privilege the gain of scale commonly practiced in the market, avoiding unnecessary searches for solutions that use parallel links. Naturally, the use of this set of restrictions can be done in an optional manner, according to the interests of the planner.

It is worth mentioning that the model proposed here efficiently allocates the capacity of the OTN / DWDM system. The total flow of demand drained on each link guides this decision. The transmission systems are sized simultaneously with the process of choosing the best traffic flow solutions on each link, the set of restrictions (2) ensures this. This process allows the dimensioning of resources in an aggregate way, valuing the policies of scale gain opportunely practiced in the market. This approach is seen as capable of providing a better resource sharing when confronted with the proposals presented in [10] [11] [13] [23]. In these works, the transmission capacity of the system and the consequent cost accounting are specified in the pre-processing stage of the *k*-paths for each demand.

The proposed model scalability is compromised by its complexity, according to the number of variables and required constraints. Assuming a worst-case scenario, where all the demands can be met, each via *k* feasible paths, through links that present all candidate OTN/DWDM modules, the number of required variables is ($|LINK| \times |O_{LINK}| + (|DEM| \times |TN| \times k)$). The amount of constraints is given by $(2 \times |LINK| + |DEM|)$.

IV. GENETIC ALGORITHM

In the considered problem of this work, the solution must include the chosen routes to flow each demand, as well as the transmission systems allocated in each link. In order, the candidate solution is encoded in a genome, known as chromosomes [33]. Each chromosome represents the demand to be met, as well as the codification of the possible paths to be used to flow the demand. The network cost is not directly encoded since its evaluation is performed by the fitness function, modeled in (1).

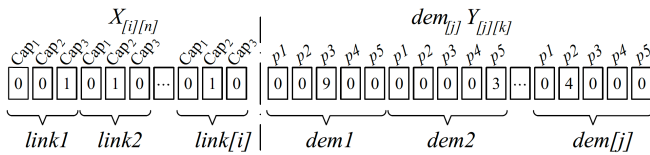


Fig. 2. Chromosome encoding and genome structure.

Fig. 2 shows the chromosomal representation used in this work. Each demand can choose a flow path from a set of precalculated paths. The binary variables $X_{[i][n]}$ model the possibility of allocating (or not) an OTN/DWDM transmission system of capacity $[n]$ ($n=3$ in Fig. 2). The binary variables $Y_{[j][k]}$ represent the choice (or not) of the path p_k ($k=5$, in Fig. 2).

Linked to the choice path (p_1, \dots, p_k) to flow the predicted demand $[k] \in DEM$ (variable $Y_{[j][k]}$) is in $dem_{[j]}$. With the total demand flow at $link_{[i]}$ its capacity $Cap_{[n]}$ (variables $X_{[i][n]}$) is chosen. The structure created for the variables $Y_{[j][k]}$ in Fig. 2 has TN multiplicity of destination nodes for each demand.

An interpretation of the chromosome for Fig. 1/Table I is:

dem1 has $TN=1$, that is, the possibility of draining the demand from the originating node $N1$ is by 2 paths (p_1 and p_2) to a single destination node $N5$ and *dem2* presents $TN=2$, that is, it has the possibility to drain the demand from node $N2$ (source) to 2 destinations, through 3 possible paths (p_1 , p_2 and p_3) to destination 1 ($N3$) or through 3 paths (p_1 , p_2 and p_3) to destination 2 ($N5$). The choice of each flow path must respect this multiplicity. The pseudocode of Genetic Algorithm (GA) is presented in Fig. 3. First, the initial parameters of the algorithm are declared (Step01) and after the objective function is defined (Step02). The evolutionary operators are detailed below.

Standard Genetic Algorithm	
Step01.	Initialize algorithm parameters Population number (<i>nPop</i>) Generation number (<i>nGer</i>) Crossover probability (<i>Pc</i>) Probability of mutation (<i>Pm</i>)
Step02.	Objective function $f(x), x = (x_1, \dots, x_d)^T$
Step03.	Generates initial population <i>P</i> of chromosomes $x_i (i = 1, 2, \dots, n)$
Step04.	The fitness function is determined by $f(x_i)$
Step05.	While ($t < nGer$) Fitness calculation - determined by $f(x_i)$ Performs tournament selection Apply crossover mechanism with probability <i>Pc</i> Apply mutation mechanism with probability <i>Pm</i>
Step06.	Rank individuals and find the best global solution
Step07.	End while

Fig. 3. Pseudocode of genetic algorithm.

Initial Population (Step03): the number of individuals that will be part of the initial population is defined by the planner. The size of this population is maintained throughout the iterative process, and each individual of the population is created in two steps:

1. *Step of the $Y_{[j][k]}$ variables*: It consists of allocating randomly, for each demand, one of the available paths for flow. Thus, it is possible to check the total flow being drained through each link.
2. *Step of the $X_{[i][n]}$ variables*: With the flows of each link calculated in the previous step, capacity is assigned to each arc (variable $X_{[i][n]}$ assume value 1) equal to or greater than the total flow. For those cases where the link flow is identified as null, the variable $X_{[i][n]}$ is set as 0.

Fitness function (Step04): determined by the objective function set in Step03. Solutions Ranking/Selection (Step05): the initial population represents a set of feasible solutions to the presented problem. Then it is necessary that each element of this set be evaluated quantitatively so that the quality of each solution can be defined. For the developed genetic algorithm, the objective function of ILP 0-1 is used as a fitness function. In this way, the solutions have their quality evaluated according to the presented cost. So, the lower the cost the better an individual.

Now, for the crossover process, it is necessary to select the individuals to be crossed. The tournament selection is performed by drawing two sets of solutions taken from the current population. In each set, a tournament is held where only the best is selected. With the best individual from each set

selected, there is a couple of individuals for the crossover. This selection process is performed for each crossover that will occur in each algorithm generation.

The number of individuals drawn from the initial population is also defined by the planner. However, the larger the set the greater the selection pressure since the chances increase that the same individuals will be selected, and this fact decreases the population diversity.

Crossover (Step05): The implemented crossing, as well as the initial population, happens in two stages, and the step of the $X_{[i][n]}$ variables does not change. Thus, a cut-off point in the chain of $Y_{[j][k]}$ variables, with the restriction that paths of the same demand cannot be separated by this cut-off point. Indeed, this process could represent the duplicating of demand within the network against the proposed model, and that would make the individual inevitably infeasible. Thus, the possibilities of cutoff points are limited to the number of demands to be drained in the network, respecting the number of candidate paths for each demand.

The cut-off point is responsible for dividing the portion $Y_{[j][k]}$ of the individual into two parts. The left part of the first individual will be concatenated with the right part of the second, and vice versa. In this way two children are generated, that is, two possible new network solutions, and these ones tend to be better than the previous two.

After the crossover procedure executed in the solution portion $Y_{[j][k]}$ each generated child has its $X_{[i][n]}$ portion created, allowing its evaluation. Only one child can remain in the next generation. The choice of the individual follows the following priority:

1. two feasible solutions: the lowest cost child individual will compose the next generation.
2. only one feasible solution: the feasible child individual is selected.
3. no feasible solution: the crossover operator is performed again with other cutoffs until a feasible solution, in other words, a feasible child individual is found.

It is worth remembering that the crossover operator described above is controlled by an occurrence probability defined by the planner. The crossover operator only happens if the probability is met. Therefore, in the event of no crossover occurring, the parent individual is simply kept in the next generation. The crossover process can alter both the demand destination and the used path for its flow.

Mutation (Step05): the new population module (composed of new individuals generated in the crossover process) may undergo slight random changes in the chosen paths to flow each demand. This procedure is performed to ensure population diversity over the generations, reducing the likelihood of convergence to a local minimum.

Mutation does not occur in all individuals. A probability of occurrence is also defined to perform this operator control. Since a very high probability rate may represent a deconstruction of the solution over the generations.

The mutation also occurs in the portion of the $Y_{[j][k]}$ variables. Consequently, the portion $X_{[i][n]}$ needs to be defined again, as

performed in one of the initial population creation steps. The mutation only changes the path but does not modify the demand destination. After the mutation mechanism, a ranking of the solutions is performed (Step06), finding the best solution, and then, the stopping criterion ($nGer$) is checked.

V. FIREFLY ALGORITHM

The Firefly Algorithm (FA) is one of the known swarm-based algorithms, having different types of applications [34]. This algorithm is a metaheuristic inspired by nature to solve optimization continuous problems, especially NP-hard problems and was motivated by the simulation of the social behavior of fireflies. It is possible to use them to formulate optimization algorithms because the flashing of the light can be used in such a way that it is associated with the objective function of the considered problem in order to obtain optimal solutions [35].

For maximization problems, the brightness can be proportional to the objective function value. For minimization problems, the brightness may be the inverse of the objective function value.

In the firefly algorithm, there are two important issues: the light intensity variation and the attractiveness formulation. For simplicity, it can be assumed that the firefly attractiveness is determined by its brightness, which in turn is associated with the encoded objective function.

The attractiveness function $\beta(r)$ can be any monotonically decreasing function, as the following generalized form given by:

$$\beta(r) = \beta_0 e^{-\gamma r^m}, m \geq 1 \quad (5)$$

where β_0 is the attractiveness at $r = 0$ and r is the distance between two fireflies.

Since it is generally faster to calculate $1/(1+r^2)$ than the exponential function [10] equation (5) can be approximated as equation (6).

$$\beta(r) = \frac{\beta_0}{1 + \gamma r^2} \quad (6)$$

The distance between any two fireflies i and j , in the position x_i and x_j , respectively, can be defined as a cartesian distance, according to (7), where x_{ik} is the component k of the spatial coordinate x_i of the firefly i and d denotes the number of dimensions.

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^d (x_{ik} - x_{jk})^2} \quad (7)$$

The random move of a firefly i to another brightest firefly j is determined by (8), where the second term considers the firefly attractiveness, the third term is random with α being a random parameter, and ϵ_i is a vector of random numbers drawn from a Gaussian distribution or a uniform distribution.

$$x_i = x_i + \beta_0 e^{-\gamma r_{ij}^2} (x_i - x_j) + \alpha \epsilon_i \quad (8)$$

In a simpler way, ϵ_i can be replaced by $rand - 1/2$, where $rand$ is a random number generator evenly distributed in $[0,1]$. For most applications $\beta = 1$ and $\alpha \in [0,1]$. In practice, the light absorption coefficient γ ranges from 0,1 to 10. This parameter describes the variation in attractiveness and its value is responsible for the convergence speed of the algorithm [11].

The firefly was originally developed to solve continuous optimization problems and cannot be applied directly to solve discrete problems. The main challenges of using firefly to solve discrete problems are in calculating the distance between two discrete fireflies and how they will move [36].

The distance between two fireflies is defined by the distance between the permutation of their sequences. There are two possible ways to measure the distance between any two fireflies i and j , in the positions x_i and x_j : (a) the Hamming distance [37] and (b) the Swap distance [38]. The Hamming distance between two permutations is the number of non-matching elements in the sequence. The Swap distance is the number of minimum exchanges required for one permutation to obtain the other.

The attraction and movement must be implemented and interpreted for the discrete firefly in the same way that is done in the continuous firefly algorithm. Thus, the move given in (8) is divided into two sub-steps: β -step and α -step as shown in (9) and (10), respectively.

$$x_i = \beta(r)(x_i - x_j) \tag{9}$$

$$x_i = x_i + \alpha(rand - 1/2) \tag{10}$$

The parameters β and α used in (8) are the same used in (9) and (10). The attraction steps β and α are not interchangeable, so the β -step must be calculated before the α -step while the new position is found. The β -step always brings the firefly j closer to the firefly i . In other words, after applying the β -step in a firefly towards the other firefly, its distance is always diminished, and the decrease is proportional to its previous distance. For this, Hamming distance is used as the distance function. This means that for the permutation to get closer to the other permutation, the amount of its common elements has to increase. Firstly, in the β -step process, what is common in both fireflies is extracted.

Secondly, there is a need to fill in the gaps in relation to the previous distance from the permutations. This can be achieved with the probability $\beta = \frac{1}{1+\gamma.d_{i,j}^2}$ where $d_{i,j}$ is the Hamming distance between fireflies i and j . Through probability β a firefly element i or j will be inserted into the new firefly. After calculating β a random number $rand()$ is generated in the $[0,1]$ range. If $rand() \leq \beta$ then the insertion is performed with the element of i , otherwise than j . This process is carried out until all the gaps in the new firefly are filled. It is worth mentioning that after each filling, the new distance must be checked for the next gap insertion.

After filling all the gaps, the α -step is performed, which is simpler than β -step. This step will change the elements of this new formation to the neighboring elements. The smallest change corresponds to two elements.

There are two ways to apply the α -step: make a $\alpha.Random()$ to perform many exchanges of two randomly chosen elements, or choose many elements through

$\alpha.Random()$ and shuffle their positions. The first option is easier to implement, but the results are not as good as the second [39].

The α parameter represents a maximum allowed step for the permutation, which consists of n elements. To accomplish this there is a need for α to be from the set $\{1, \dots, n\}$. Then $\alpha = 1$ means that no step is done and $\alpha = n$ means to shuffle all elements of the permutation. Fig. 4 presents the basic steps of the firefly algorithm.

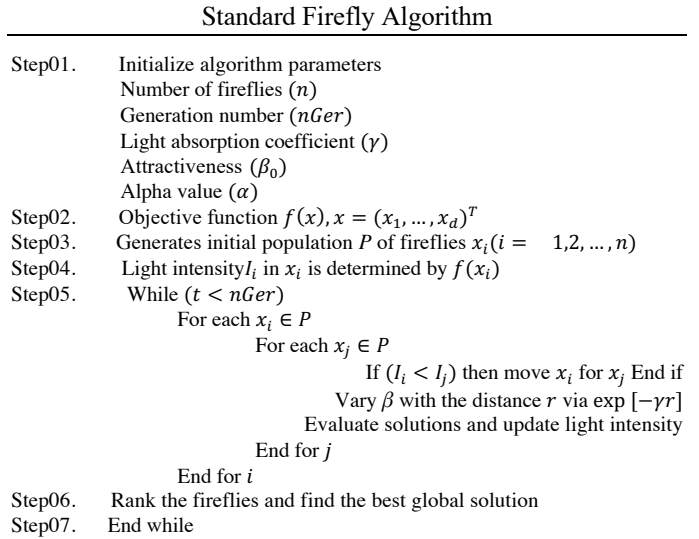


Fig. 4. Pseudocode of firefly algorithm.

Initially, the parameters of the algorithm are declared (Step01) and next, the objective function is defined (Step02). The coding routines, initial population generation (Step03), and ranking (Step04) follow the process described in Section IV. Next, we describe the firefly movement routine.

Firefly movement (Step05): after the creation of the firefly population, the light intensity (cost) value, that is obtained by evaluating the objective function of the ILP 0-1. With this, it is possible to evaluate the quality of the solution (ranking) according to the presented cost. Hence, the lower the cost the better the individual (greater light intensity).

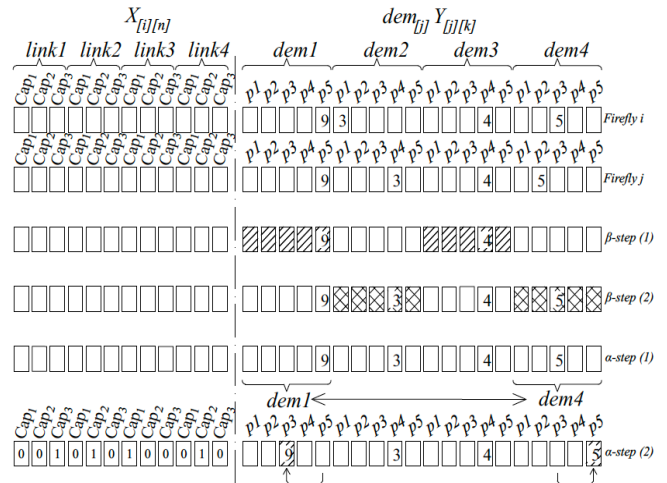


Fig. 5. Firefly movement.

From this set of solutions containing the lowest cost fireflies to the highest cost a copy is created and in which the fireflies are

positioned from the highest to the lowest cost (performed to obtain greater diversity), to enter the movement stage. The first firefly i_1 is compared with all fireflies j (copy). This procedure is performed with the firefly i_2 up to firefly i_n . If the selected firefly j has a lower intensity, i.e. higher implementation cost, it will be moved towards the firefly i , which has the highest intensity and this firefly will be placed in the firefly i position.

To perform the movement, first there is a need to calculate the distance between the fireflies. For this calculation, the Hamming distance will be used. The movement is applied to the firefly's binary variables ($Y_{[j][k]}$). An example with a transmission system with capacity equals to 3 ($n=3$) and five candidate paths to flow each demand ($k=5$) each TN destination is shown in Fig. 5.

In the example, the Hamming distance between firefly i and firefly j is 2 (demands flowed by different paths in $dem2$ and $dem4$). To perform the movement, the β -step is executed, that is, a new firefly β -step (1) is created, maintaining the demands with identical paths ($dem1$ and $dem3$). After the gaps are filled in, the probability β is calculated on each gap and compares this value with the random number $rand()$ generated in the $[0,1]$ interval. Note that β is controlled by the light absorption coefficient (γ), defined by the user, and Hamming distance. If $rand() \leq \beta$ then the insertion is performed with the firefly i element, otherwise the firefly j . In the β -step (2) it can be verified that $dem2$ was selected from firefly j , and for $dem4$ firefly, i was chosen.

Then, the α -step will be performed. The maximum amount of (α -step) permutations is controlled by α parameter, previously defined. Thus, the permutations quantity is arbitrarily determined by choosing an integer random value generated in the $[1, \alpha]$ interval and after, the demands for performing the permutation are chosen, also randomly. In the example, the α value was 2. In α -step (1) of Fig. 5 the permutations quantity was equal to 1 and the chosen demands for the exchange were $dem1$ and $dem4$. What will be changed is the demand path to flow and not the demands values, so that $dem1$ that was flowing its demand of 9 Gbps by the path p_5 , is now drained by p_3 , which was the path that drained $dem1$ demand of 5 Gbps. And in $dem4$ the exchange was performed from p_3 to p_5 .

Once the β -step and α -step procedures are performed, the total flow of each link can be calculated and thus allocate its capacity ($X_{[i][n]}$ variables step). If the firefly is infeasible, the movement process is performed again, until the creation of a feasible firefly. After the completion of the comparison/movement of all fireflies, a new ranking (Step06) is performed.

VI. HYBRID FIREFLY-GENETIC ALGORITHM

The optimization algorithm performance can be improved by transforming the current solution into one or more enhanced solutions. To perform this improvement, the method proposed in [28] is used. The category of this hybrid metaheuristic is the integrative [40], in which an algorithm is considered a subordinate component and incorporated into that of a master metaheuristic, which governs the search process. In this work, the master meta-heuristic is the firefly algorithm and the genetic algorithm is subordinate to it. Fig. 6 presents the steps of the hybrid discrete Firefly-Genetic Algorithm (HFA).

The firefly algorithm (shown in Fig. 4) was extended by inserting crossing and mutation mechanisms. This extension corresponds to the Improvement method (Step06) in Fig. 6.

Two solutions are used at each iteration, the first and second ranking solutions (highest light intensity) provided by the firefly algorithm. The selected fireflies are always submitted to the genetic algorithm crossover and mutation operators, i.e., the probabilities of crossover and mutation are both 100%. The two best possible fireflies between parents and children are kept in the first and second position of the population and the rest are eliminated.

Hybrid Firefly-Genetic Algorithm	
Step01.	Initialize algorithm parameters Number of fireflies (n) Generation number ($nGer$) Light absorption coefficient (γ) Attractiveness (β_0) Alpha value (α)
Step02.	Objective function $f(x), x = (x_1, \dots, x_d)^T$
Step03.	Generates initial population P of fireflies $x_i (i = 1, 2, \dots, n)$
Step04.	Light intensity I_i in x_i is determined by $f(x_i)$
Step05.	While ($t < nGer$) For each $x_i \in P$ For each $x_j \in P$ If ($I_i < I_j$) then move x_i for x_j End if Vary β with the distance r via $\exp[-\gamma r]$ Evaluate solutions and update light intensity End for j End for i
Step06.	Improvement Method Rank the fireflies and find the current bests Apply crossover mechanism in the two best current solutions Apply mutation mechanism
Step07.	End while

Fig. 6. Pseudocode of hybrid-firefly algorithm.

VII. EVALUATED SCENARIO

The methodology can be applied to different scenarios of the optical transport network, different types of transmission technologies and different traffic matrices. Naturally, for the same input variables, if the data parameters are different, for example, the precomputed paths for each demand, the reached solutions will also change.

With the goal of evaluating an application scenario similar to reality, the methodology used to elaborate the network data was the integration of the team with a telecommunications company in Brazil. In this partnership, relevant information was collected, such as demand points location, distances between locations, network topology, equipment costs, and optical network.

The studied OTN/DWDM network can assume three hierarchical models, with a transmission capacity of 40, 100 and 400 Gbps. The actual network is composed of 27 individual nodes and 50 links. Fig. 7 shows the network topology (without considering the actual scale), including possible links to be allocated and dimensioned.

Traffic interest is defined from cities where the carrier attends, merging concession and expansion areas. Each city (demand node) has a service forecast for the clients of that region. A particularity of this network is that the traffic generated at each demand node (source node) needs to be

drained to the communication (or overflow) points with another WAN (Wide Area Network). In Fig. 7 network, nodes N26 and N27 are the destination nodes.

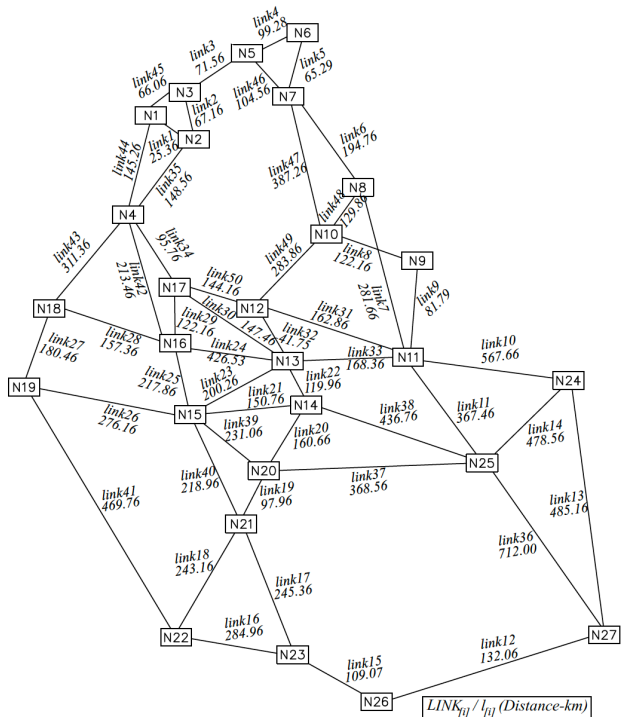


Fig. 7. Backbone OTN/DWDM – Network 1.

The network presents a total of 27 predicted demands that must fully met and directed to the destination nodes. Table II lists an example of a randomly generated traffic matrix in the [0.5, 50] Gbps range. The 27 demands in Table II have N26 or N27 as destination nodes (TN).

TABLE II. OTN/DWDM NETWORK DEMANDS – NETWORK 1

Demand/ Source Node	(Gbps)	Demand/ Source Node	(Gbps)
dem1/N1	13	dem15/N15	4
dem2/N2	0.5	dem16/N16	10
dem3/N3	1	dem17/N17	22.5
dem4/N4	5	dem18/N18	11.5
dem5/N5	0.5	dem19/N19	12.5
dem6/N6	37.5	dem20/N20	7.5
dem7/N7	1.5	dem21/N21	19
dem8/N8	2.5	dem22/N22	3.5
dem9/N9	12	dem23/N23	25
dem10/N10	2.5	dem24/N24	30
dem11/N11	7.5	dem25/N25	15
dem12/N12	4	dem26/N26	13
dem13/N13	20	dem27/N27	30
dem14/N14	50		

Some modeling characteristics that are obeyed in Fig. 7 network evaluation: the demands and the links obey the same form of routing, being bidirectional; the capacity of the links is explicit; all existing paths can be used and the predicted demand cannot be drained by more than one path.

The network cost was determined from a detailed survey of each service and assets used to implement a link. After collecting the separated costs of each item, the total deploying cost of each link with the three OTN / DWDM modules being

evaluated (40, 100, 400 Gbps) was calculated. The costs of regenerators have also been specified according to the capacity of transmission and its range (used for distances greater than 80 km). Table III presents these values. The used values are relative. The cost of the 40 Gbps capacity OTN/DWDM system (\$ 89000.00) is used as a basis. The considered optical network cost represents a consolidated calculation of the average network cost per kilometer, which is 0.012.

TABLE III. COSTS OF OTN/DWDM SYSTEMS

Capacity (Gbps)	Distance	Cost
40	Up to 80km	1.00
	Greater than 80km	1.32
100	Up to 80km	1.90
	Greater than 80km	2.22
400	Up to 80km	3.92
	Greater than 80km	4.24

The genetic algorithm and the firefly algorithm are implemented in MATLAB® software, running in a microcomputer with macOS® environment, Intel Core i5 2.3GHz processor and 8GB RAM. The performance of the presented approach was verified computationally, and the results were compared with the exact method, where the *intlinprog* solver was used, which makes use of the branch-and-bound solution method (B&B) [41].

The initial experiment was performed on the exact ILP model to verify the influence of the number of candidate paths for each demand. The quality of the obtained solution, as well as the required computational effort, are evaluated.

Table IV presents the results (cost) according to the number of candidate paths for each demand between $k = 10$ (5 paths for each destination) and $k = 20$ (10 paths for each destination). It is also shown in Table IV: the total network capacity (sum of capacities (40, 100 or 400 Gbps) of all links allocated in the network), the processing time and the capacity used which represents the percentage of the total flow drained in the network in relation to the total capacity, that is, it represents the average network occupation.

TABLE IV. EXACT METHOD COMPARISON – NETWORK 1

Number of Candidate Paths (k)	Cost	Total Capacity (Gbps)	Capacity Used (%)	Execution Time (s)
10	559.63	3340	47.40	2460
12	537.75	3040	52.20	56077
14	531.67	3340	49.70	35566
16	530.96	3340	50.30	64376
18	530.96	3340	50.30	290739
20	530.96	3340	50.30	318140

The ILP model obtained the optimal solution in all cases. It can be observed in Table IV that the optimal solution and the execution time are strongly influenced by the precomputed paths. This demonstrates that the large network solution obtained with exact methods requires a great computational effort and consequently a high value in time processing. For 16, 18 and 20 candidate paths the solution is the same, however, the processing time increases considerably.

The network topology with the optimal solution (exact method) for $k = 16, 18$ and 20 can be checked in Fig. 8. Each

chosen link indicates the flows drained and the capacity allocated (flow/capacity) in *Gbps*.

The solution network prioritized the flow of demands through node N26. In the obtained result of the 50 links that were candidates, 25 were installed, being 16 links of 40 *Gbps*, 3 links of 100 *Gbps* and 6 links of 400 *Gbps*. The utilized capacity of the network solution is 50.30%.

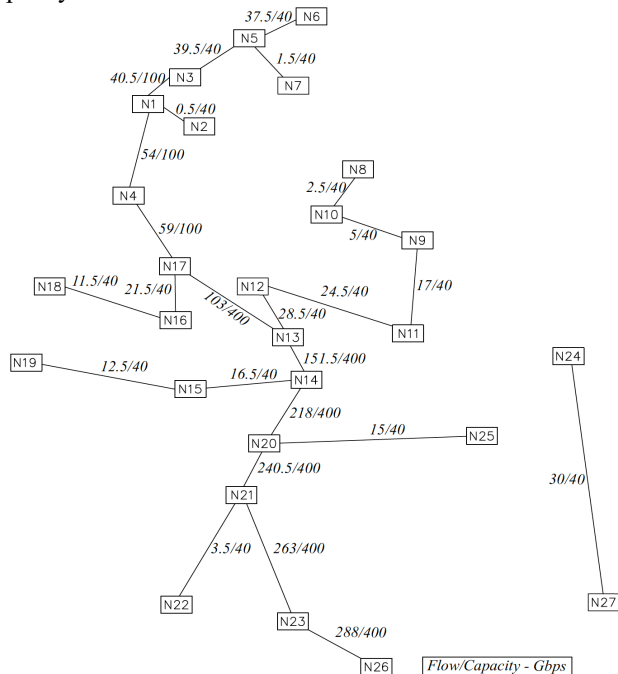


Fig. 8. Solution network for $k=16, 18$ and 20 .

Table V and VI describe the five configurations of the FA, HFA, and GA algorithms used to evaluate the network. The parameters β and γ of the firefly algorithm are fixed at $\beta = 1$ and $\gamma = 0.1$ [12]. These configurations were chosen in order to evaluate the performance of the methodology by privileging some of its parameters, besides the main objective that is to dimension the network at minimum cost.

TABLE V
CONFIGURATIONS FA AND HFA

Configurations	Parameters		
	Fireflies	Generation	α
1	100	50	4
2	100	50	8
3	100	100	8
4	200	100	4
5	500	500	8

For each value of k 10 replicates were performed (each algorithm was executed 10 times). The execution time presented is the sum of the 10 replicas made. The stopping criterion is the number of generations.

TABLE VI
CONFIGURATIONS OF AG

Configuration	Parameters			
	Population	Generation	Prob. Crossover	Prob. Mutation
1	100	50	0.6	0.05
2	100	50	0.9	0.10
3	100	100	0.9	0.10
4	200	100	0.6	0.05
5	500	500	0.9	0.10

Table VII presents the performance comparison, bringing the best solutions of the exact, GA, FA and HFA methods. The *Gap* represents how many percent the solution is above the optimal value found by the exact method. The *Config.* shows in which configurations the lowest deployment cost values were found.

The HFA algorithm proved to be effective in all cases, as its solutions reached the same value as the exact method (branch-and-bound), while the FA reached the value in only 2 cases ($k=14$ and $k=20$). The HFA, in configuration 5 (Table IV), reached the optimal value in all scenarios evaluated.

TABLE VII
ALGORITHM COMPARATIVES – NETWORK I

Number of Candidate Paths (k)	Exact Method (Cost)	Method	Cost	Gap	Config.
10	559.63	GA	578.12	03.30%	3 to 5
		FA	577.41	03.17%	2 to 5
		HFA	559.63	00.00%	4, 5
12	537.75	GA	556.63	03.51%	4, 5
		FA	538.78	00.19%	5
		HFA	537.75	00.00%	3, 5
14	531.67	GA	549.44	03.34%	5
		FA	531.67	00.00%	5
		HFA	531.67	00.00%	5
16	530.96	GA	548.73	03.34%	4, 5
		FA	532.85	00.35%	4, 5
		HFA	530.96	00.00%	4, 5
18	530.96	GA	555.53	04.62%	3 to 5
		FA	538.07	01.34%	5
		HFA	530.96	00.00%	5
20	530.96	GA	548.50	03.30%	5
		FA	530.96	00.00%	3, 5
		HFA	530.96	00.00%	4, 5

The performance in terms of computational time (Table VIII) required to find the solution with the exact method was high and it is strongly influenced by the value of k . Based on all k values and configuration 5 of the bio-inspired algorithms, the genetic algorithm, despite having excellent results in terms of time, presented the highest cost values. HFA had slightly higher processing time values than FA but achieved better cost results (Table VII).

TABLE VIII
COMPARISONS OF PROCESSING TIME BETWEEN METHODS – NETWORK I

Number of Candidate Paths (k)	Execution Time (s)			
	Exact Method	GA	FA	HFA
10	2460	7060	29651	29874
12	56077	7342	28746	29965
14	35566	8608	30987	31675
16	64376	8923	32287	33934
18	290739	10006	33178	34365
20	318140	9466	35971	37537

Fig. 9, 10 and 11 show the comparative for the cases of $k=10$, $k=14$ and $k=20$ of Table VII, for the three methods, GA, FA and HFA, in configuration 5, of cost as a function of number of generations.

It is worth noting that in the three cases, that the increase in computational effort (around 3.7% on average) caused by the solution improvement method proposed in pseudocode of hybrid-firefly algorithm (Step06), is compensated by faster convergence to the optimal solution.

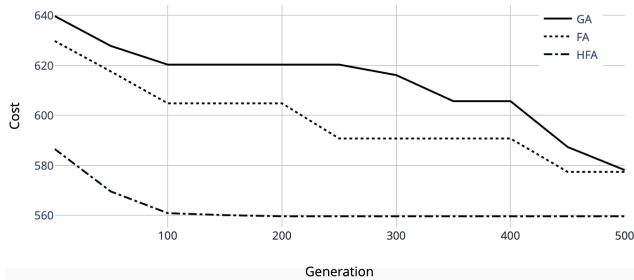


Fig. 9. Convergence of methods for $k=10$.

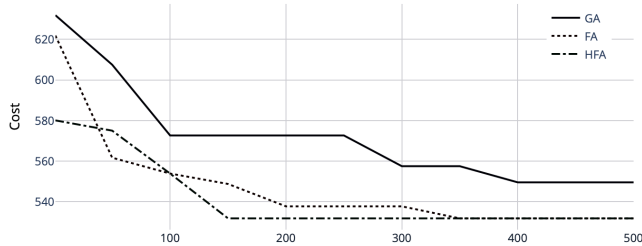


Fig. 10. Convergence of methods for $k=14$.

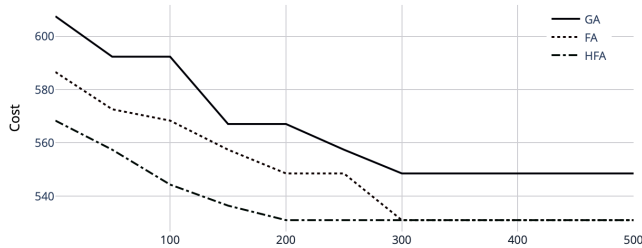


Fig. 11. Convergence of methods for $k=20$.

Table IX presents the comparison between method performances considering change in the number of candidate paths and assuming as stopping criterion, the execution time of 3600 seconds. The configuration used for the bio-inspired algorithms is 5 (Table V and VI).

TABLE IX
COMPARISONS OF COST BETWEEN METHODS – NETWORK 1

Number of Candidate Paths (k)	Cost			
	Exact Method	GA	FA	HFA
10	559.63	625.20	559.63	559.63
12	631.37	617.46	555.53	537.75
14	620.54	582.94	531.67	531.67
16	622.75	622.83	548.73	538.79
18	566.43	637.53	549.44	549.44
20	553.63	622.91	551.81	548.50

Considering all k values, HFA provided the best cost values, with an average of 544.39, followed by the FA method, with an average of 549.44. The exact method reached an average of 592.34 and GA achieved the highest implementation costs, with an average of 618.10. It is worth noting that the exact method searches for the best solution in a space less adequate than that considered by HFA, being less efficient for the available time of 3600 seconds. Considering the stopping criteria of the algorithms, the generation number and the processing time (3600 seconds), the HFA performance can be highlighted once it provided the lowest cost values in all cases.

The variability between the solutions obtained by each method, taking into account all configurations, can be seen in Table X. Listed are: Lowest Cost (Best Solution), Worst Cost, Average Cost, and Average Gap. In all scenarios the HFA hybrid method proposed in this work achieved the best results also in terms of little variability between the obtained solutions.

TABLE X
COMPARISONS BETWEEN METHODS – NETWORK 1

Method	$k=10$			
	Best Cost	Worst Cost	Average Cost	Average Gap
Exact	559.63	-	-	-
GA	578.12	629.79	598.35	06.91%
FA	577.41	578.12	577.49	03.19%
HFA	559.63	577.41	568.96	01.67%
$k=14$				
Exact	531.67	-	-	-
GA	549.44	616.12	566.59	06.56%
FA	531.67	556.63	545.65	02.63%
HFA	531.67	549.44	543.52	02.22%
$k=20$				
Exact	530.96	-	-	-
GA	548.50	576.15	563.07	06.05%
FA	530.96	568.33	550.47	03.67%
HFA	530.96	568.33	543.44	02.35%

In order to better verify and compare the performance of the methods, it was also considered a smaller network, that is, the OTN/DWDM network shown in Fig. 12. This network is contained in the SNDlib repository [42]. The network is made up of 11 nodes, 34 links and 11 demands that can be drained by destination nodes (TN) N2 or N8, with the possibility of allocating 40, 100, or 400 Gbps transmission systems capacity.

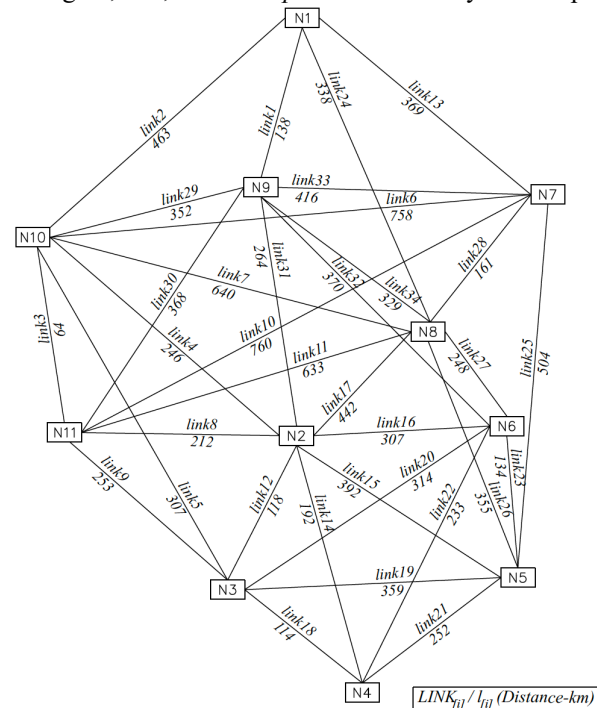


Fig. 12. Backbone OTN/DWDM – Network 2.

Fig. 12 shows the network topology (without considering the real scale), including possible links to be allocated and dimensioned and also the distance of the links in km. Table XI

shows the demands generated randomly in the [0.5, 50] Gbps range.

TABLE XI. OTN/DWDM NETWORK DEMANDS – NETWORK 2

Demand/Source Node	(Gbps)
dem1/N1	19
dem2/N2	42
dem3/N3	25
dem4/N4	46
dem5/N5	21
dem6/N6	33
dem7/N7	36
dem8/N8	08
dem9/N9	44
dem10/N10	06
dem11/N11	47

Table XII presents the performance comparison, bringing the best solutions of the exact method, GA, FA and HFA, to the network of Fig. 12. The configurations used in the bio-inspired algorithms are those of Table V and VI.

TABLE XII
ALGORITHM COMPARATIVES – NETWORK 2

Number of Candidate Paths (<i>k</i>)	Exact Method (Cost)	Method	Cost	Gap	Config.
10	98.44	GA	102.93	04.56%	4, 5
		FA	102.93	04.56%	1 to 5
		HFA	102.93	04.56%	1 to 5
12	98.44	GA	102.93	04.56%	3 to 5
		FA	98.44	00.00%	3, 5
		HFA	102.93	04.56%	1 to 5
14	98.44	GA	102.93	04.56%	1 to 5
		FA	102.93	04.56%	1 to 5
		HFA	98.44	00.00%	3, 5
16	97.24	GA	102.93	05.85%	4, 5
		FA	98.44	01.23%	3, 5
		HFA	97.24	00.00%	3, 5
18	97.24	GA	102.93	05.85%	4, 5
		FA	97.24	00.00%	3, 5
		HFA	97.24	00.00%	3, 5
20	97.24	GA	102.93	05.85%	1 to 5
		FA	98.44	01.23%	3, 5
		HFA	97.24	00.00%	3, 5

In terms of response performance, the HFA algorithm also proved to be effective, as its solutions matched the exact method in 4 cases ($k=14$ to $k=20$ - *Config.* 3 and 5), while the FA reached the value in only 2 cases, for ($k=12$ and $k=18$ - *Config.* 3 and 5). It is worth mentioning that for configurations 3 and 5, HFA reached the optimum solution. In the large network, the optimal solution was achieved only for configuration 5. These results show that for smaller networks, HFA does not need to have its parameters intensely or excessively adjusted to obtain a good solution.

Table XIII shows the comparison of the processing time between the exact methods, GA, FA and HFA for the network of Fig. 12. The execution times are presented in terms of the number of candidate paths (*k*) in Table XIII. The configurations used to present the processing times of the bio-inspired algorithms are 3 and 5 (Table V and VI).

TABLE XIII
COMPARISONS OF PROCESSING TIME – NETWORK 2

<i>k</i>	Exact	Execution Time (s)					
		GA		FA		HFA	
		Config.		Config.		Config.	
		3	5	3	5	3	5
10	04	267	12904	1996	23299	1959	23830
12	08	554	13600	2144	24178	2019	25257
14	12	573	13910	2172	24814	2130	25781
16	14	609	14825	2128	25219	2168	26579
18	20	631	10294	2138	29809	2194	25699
20	28	664	11828	2208	27779	2270	25791

The exact method proved to be more effective than the other methods (GA, FA and HFA) in terms of processing time for the smaller network, as the search space of the algorithm is small. This method for the largest network in Fig. 7, which has a larger volume of data, showed a high processing time (large search space). It is worth noting that the processing time of the exact method is less influenced by the number of candidate paths when considering the network in Fig. 12 than for the larger network.

The parameters of the bio-inspired algorithms influence their performance, mainly the number of individuals and generations. This can be seen in Table XIII, because even for a small network the algorithms obtained a high value in the processing time. This is due to the fact that even though the network has a lower volume of information, the algorithms have to process all individuals up to the established generation number. It is worth noting that, while observing a great increase in computational time for the exact method (from 28 to 318140 seconds), the proposed HFA processing time increases by only 45.5% in the worst case, when tuning the parameter values of the algorithm for the configuration 5 and $k=20$, in both networks.

TABLE XIV
COMPARISONS BETWEEN METHODS – NETWORK 2

Method	Best Cost	Worst Cost	Average Cost	Average Gap
Exact	98.44	-	-	-
GA	102.93	113.20	109.09	10.82%
FA	102.93	102.93	102.93	04.56%
HFA	102.93	102.93	102.93	04.56%
<i>k=14</i>				
Exact	98.44	-	-	-
GA	102.93	102.93	102.93	04.56%
FA	102.93	102.93	102.93	04.56%
HFA	98.44	102.93	101.12	02.72%
<i>k=20</i>				
Exact	97.24	-	-	-
GA	102.93	102.93	102.93	05.85%
FA	98.44	102.93	102.38	05.28%
HFA	97.24	98.44	97.96	00.74%

The variability between the solutions obtained by each method, taking into account all configurations, can be seen in Table XIV. In all scenarios, the hybrid method proposed in this work obtained good results with little variability between the solutions obtained for the two evaluated networks. As with the large network, the convergence of HFA to the optimal solution is faster than the GA and FA algorithms for the smaller

network.

VIII. CONCLUSION

This work presented a strategic planning proposal oriented towards minimizing the costs for the OTN/DWDM optical transport network infrastructure. The network is seen as a graph and, following the arc-path approach, the modeling has been translated as a whole linear programming problem with binary variables, which obeys the demand constraints, technical equipment capacity, and attendance exclusivity requirements. The resolution techniques used to solve the ILP 0-1 optimization model were the genetic algorithm, firefly algorithm and the hybrid firefly algorithm.

The preliminary obtained results regarding the developed computational tool performance are encouraging, highlighting the model ease of adaptation to meet new requirements and/or network and technology specificities to be evaluated. The modeling proposed in this work presents the following most relevant differences from others in the literature: (1) optical network design with demand matrix with multiple destinations; and (2) determination of capacities of transmission systems with more efficient traffic aggregation.

The solution quality and computational time are strongly influenced by the number of candidate paths and the parameters of the bio-inspired algorithms. The genetic algorithm, even with the shortest processing times, presented the costliest solutions. Regarding the hybrid firefly-genetic algorithm, it stands out for the outstanding results for a smaller network, and mainly for large network once in this case it reached the optimal value with computational time lower than that of the exact method for all considered configurations.

The activities that can be pointed out as extensions of this work are: analysis using fuzzy modeling to represent inaccurate data, including more optimistic demand scenarios; and studies on the possibility of multi-objective modeling, comparing minimum cost and maximum network capacity.

REFERENCES

- [1] A. Kumar and M. Gupta, "A review on activities of fifth generation mobile communication system," *Alexandria Engineering Journal*, 2017, 10.1016/j.aej.2017.01.043.
- [2] S. Mumtaz, A. Morgado, K. M. S. Huq and J. Rodriguez, "A survey of 5G technologies: Regulatory, standardization and industrial perspectives," *Digital Communications and Networks*, 2017, 10.1016/j.dcan.2017.09.010.
- [3] S. Li, L. Da Xu and S. Zhao, "5G internet of things: A survey," *Journal of Industrial Information Integration*, 2018, 10.1016/j.jii.2018.01.005.
- [4] Q. Wang and G. Ying, "OTN for the Future Transmission Network," *Symposium on Photonics and Optoelectronics (SOPO)*, 2012, 10.1109/SOPO.2012.6270500.
- [5] ITU-T Recommendation G.709. Interfaces for the Optical Transport Network (OTN). 2012.
- [6] Y. S. Kaviani, "Intelligent Systems for Optical Networks Design: Advancing Techniques: Advancing Techniques," *IGI Global*, 2013.
- [7] F. Musumeci, et al., "An Overview on Application of Machine Learning Techniques in Optical Networks," *IEEE Communications Surveys & Tutorials*, 2018, 10.1109/COMST.2018.2880039.
- [8] J. Pedro, "Designing Transparent Flexible-grid Optical Networks for Maximum Spectral Efficiency," *IEEE/OSA Journal of Optical Communications and Networking*, 2017, 10.1364/JOCN.9.000C35.
- [9] H. Liu, C. Xiong, Y. Chen, C. Li and D. Chen, "An Optimization Method of VON Mapping for Energy Efficiency and Routing in Elastic Optical Networks," *Optical Fiber Technology*, vol. 41, pp. 173-181, 2018, 10.1016/j.yofte.2018.01.004.
- [10] A. Eira, J. Pedro and J. Pires, "Cost-optimized Dimensioning of Translucent WDM Networks with Mixed-Line-Rate Spectrum-Flexible Channels," *IEEE 13th International Conference on High Performance Switching and Routing*, 2012, 10.1109/HPSR.2012.6260848.
- [11] X. Chen and J. Admela, "Optimized Parallel Transmission in OTN/WDM Networks to Support High-Speed Ethernet with Multiple Lane Distribution," *Journal of Optical Communications and Networking*, pp. 248-258, 2012, 10.134/JOCN.4.000248.
- [12] X. S. Yang, "Nature-Inspired Metaheuristic Algorithms," *Luniver Press*, Second Edition, 2010.
- [13] A. Eira, J. Santos, J. Pedro and J. Pires, "Multi-Objective Design of Survivable Flexible-Grid DWDM Networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 3, pp. 326-339, 2014, 10.1364/JOCN.6.000326.
- [14] D. Din, "Heuristic and Genetic Algorithms for Solving the Virtual Topology Design Problem on Elastic Optical Networks," *Journal of Information Science & Engineering*, vol. 33, 2017, 10.1688/JISE.2017.33.2.3.
- [15] D. Din, "Genetic algorithm for virtual topology design on MLR WDM networks," *Optical Switching and Networking*, vol. 18, pp. 20-34, 2015, 10.1016/j.osn.2015.03.003.
- [16] S. A. Fernandez, A. A. Juan, J. A. Adrián, J., D. G. Silva and D. R. Terrén, "Metaheuristics in Telecommunication Systems: Network Design, Routing, and Allocation Problems," *IEEE Systems Journal*, vol. 12, pp. 3948-3957, 2018, 10.1109/JSYST.2017.2788053.
- [17] J. Mata, et al., "Artificial Intelligence (AI) Methods in Optical Networks: A Comprehensive Survey," *Optical Switching and Networking*, vol. 28, pp. 43-57, 2018, 10.1016/j.osn.2017.12.006.
- [18] X. S. Yang, S. F. Chien, T. O. Ting, "Bio-inspired Computation in Telecommunications," *Morgan Kaufmann*, 2015.
- [19] C. Papagianni, et al., "Communication Network Design Using Particle Swarm Optimization," *IEEE International Multiconference on Computer Science and Information Technology*, pp. 915-920, 2008, 10.1109/IMCSIT.2008.4747351.
- [20] J. Triay, C. Cervello, "An Ant-Based Algorithm for Distributed Routing and Wavelength Assignment in Dynamic Optical Networks," *IEEE Journal on Selected Areas in Communications*, vol. 28, pp. 542-552, 2010, 10.1109/JSAC.2010.100504.
- [21] A. Rubio, M. A. Veja, and D. L. González, "An Improved Multiobjective Approach Inspired by the Flashing Behaviour of Fireflies for Traffic Grooming in Optical WDM Networks," *Applied Soft Computing*, pp. 617-636, 2014, 10.1016/j.asoc.2014.03.046.
- [22] M. Haruna, M. Jinno, and Y. Nishio, "Application of Firefly Algorithm to Optimization of Translucent Elastic Optical Network," *Proc. of Nonlinear Theory and its Applications Conference*, 2015.
- [23] J. R. Santos, A. Eira and J. Pires, "A Heuristic Algorithm for Designing OTN Over Flexible-Grid DWDM Networks," *Journal of Communications*, 2017, 10.12720/jcm.12.9.500-509.
- [24] M. S. Bazaraa, J. J. Jarvis and H. D. Sherali, "Linear Programming and Network Flows," *Wiley*, 2010.
- [25] D. T. Hai, "A bi-objective Integer Linear Programming Model for the Routing and Network Coding Assignment Problem in WDM Optical Networks with Dedicated Protection," *Computer Communications*, vol. 133, pp. 51-58, 2019, 10.1016/j.comcom.2018.08.006.
- [26] R. Goscién, "Two Metaheuristics for Routing and Spectrum Allocation in Cloud-Ready Survivable Elastic Optical Networks," *Swarm and Evolutionary Computation*, vol. 44, pp. 388-403, 2019, 10.1016/j.swevo.2018.04.013.
- [27] R. Goscién and M. Lozano, "Artificial Bee Colony for Optimization of Cloud-Ready and Survivable Elastic Optical Networks," *Computer Communications*, vol. 128, pp. 35-45, 2018, 10.1016/j.comcom.2018.07.011.
- [28] A. Rahmani and S. A. MirHassani, "A hybrid Firefly-Genetic Algorithm for the Capacitated Facility Location Problem," *Information Sciences*, vol. 283, pp. 70-78, 2014, 10.1016/j.ins.2014.06.002.
- [29] A. M. Mohsen and W. Al-Sorori, "A New Hybrid Discrete Firefly Algorithm for Solving the Traveling Salesman Problem," *Applied Computing and Information Technology*, Springer, 2017, 10.1007/978-3-319-51472-7_12.
- [30] J. Simmons, "Optical Network Design and Planning," *Springer International Publishing Switzerland*, 2014.

- [31] K. D. R. Assis, I. Queiroz, R. C. Almeida and H. Waldman, "MILP Formulation for Resource Optimization in Spectrum-Sliced Elastic Optical Path Networks," *Microwave & Optoelectronics Conference (IMOC)*, SBMO/IEEE MTT-S International, 2013, 10.1109/IMOC.2013.6646582.
- [32] A. Schickedanz, D. Ajwani, U. Meyer and P. Gawrychowski, "Average Case Behavior of K-Shortest Path Algorithms," *International Conference on Complex Networks and their Applications*, Springer, pp. 28-40, 2018, 10.1007/978-3-030-05411-3_3.
- [33] A. Ghosh and S. Tsutsui, "Advances in Evolutionary Computing: Theory and Applications," *Springer Science & Business Media*, 2012.
- [34] X. S. Yang, "Cuckoo Search and Firefly Algorithm: Theory and Applications," *Springer*, Vol. 516, 2013.
- [35] X. S. Yang and X. He, "Firefly algorithm: Recent Advances and Applications," *International Journal of Swarm Intelligence*, vol. 1, pp. 36-50, 2013, 10.1504/IJSI.2013.055801.
- [36] W. T. Lunardi and V. Holger, "Comparative study of genetic and discrete firefly algorithm for combinatorial optimization," *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, 2018, 10.1145/3167132.3167160.
- [37] A. M. Mohsen and W. Al-Sorori, "A New Hybrid Discrete Firefly Algorithm for Solving the Traveling Salesman Problem," *Applied Computing and Information Technology*, Springer, Cham, 2017, 10.1007/978-3-319-51472-7_12.
- [38] G. K. Jati and R. Manurung, "Discrete Firefly Algorithm for Traveling Salesman Problem: A new movement scheme," *Swarm Intelligence and Bio-Inspired Computation*, Elsevier, pp. 295-312, 2013, 10.1016/B978-0-12-405163-8.00013-2.
- [39] X. S. Yang and X. He, "Applications of Nature-Inspired Algorithms, Mathematical Foundations of Nature-Inspired Algorithms," *Springer*, pp. 87-97, 2019, 10.1007/978-3-030-16936-7_6.
- [40] F. J. Rodriguez, C. Garcia and M. Lozano, "Hybrid Metaheuristics Based on Evolutionary Algorithms and Simulated Annealing: Taxonomy, Comparison, and Synergy Test," *IEEE Transactions on Evolutionary Computation*, 2012, 10.1109/TEVC.2012.2182773.
- [41] D. R. Morrison, S. H. Jacobson, J. J. Sauppe, and E. C. Sewell, "Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning," *Discrete Optimization*, 19, pp. 79-102, 2016, 10.1016/j.disopt.2016.01.005.
- [42] S. Orlowski, R. Wessaly, M. Pióro and A. Tomaszewski, "SNDlib 1.0: Survivable Network Design Library," *Networks*, pp. 276-286, 2010, 10.1002/net.20371.



Bruno Q. de Oliveira holds a degree in Electrical Engineering from the Pontifical Catholic University of Goiás (2011), a master's degree in Production and Systems Engineering from the Pontifical Catholic University of Goiás (2015) and a doctorate degree in Electrical and Computer Engineering from the Federal University of Goiás (2020). He is currently a professor at the Pontifical Catholic University of Goiás and at the Senai Faculty of Technology. He has experience in the areas of electrical/mechatronics engineering and production engineering, with an emphasis on electrical installations, automation, telecommunications systems and operational research, working mainly on the following topics: electrical projects, industrial automation, building automation, programmable logic controllers, industrial networks, strategic planning, mathematical modeling, systems optimization, artificial intelligence and decision support systems.



Marcos A. de Sousa has a degree in Electrical Engineering from the Federal University of Goiás (1996), a master's degree in Electrical Engineering from the State University of Campinas (1999) and a doctorate in Electrical Engineering from the State University of Campinas (2004). He is currently an adjunct professor at the School of Engineering at PUC Goiás and an associate professor at the School of Electrical, Mechanical and Computer Engineering at UFG. He has experience in the field of Electrical Engineering and Production Engineering, with an emphasis on systems engineering, acting mainly on the following themes: modeling and optimization of systems, decision support systems, applied computational intelligence, artificial intelligence, economic engineering and telecommunications systems.



Flávio H. T. Vieira holds a degree in Electrical Engineering from the Federal University of Goiás (2000), a master's degree in Electrical and Computer Engineering (EEEC) from the Federal University of Goiás (2002), a doctorate degree in Electrical Engineering from the State University of Campinas (2006) and a post-doctorate in Electrical Engineering from the State University of Campinas (2008). He is currently an Associate Professor at the Federal University of Goiás (UFG), Coordinator of the Graduate Program in Electrical and Computer Engineering at UFG. He has experience in the area of Electrical and Computer Engineering, with emphasis on Telecommunications, acting mainly on the following themes: Wireless Networks, Communication Networks, Network Traffic, Intelligent Systems applied to telecommunications and energy systems, Smart Grid, Pattern Recognition and Artificial Intelligence.