Firefly Algorithm Based Optimization Model for Planning of Optical Transport Networks

Bruno Quirino de OLIVEIRA[1,2], Marcos Antônio de SOUSA[1,2], Flávio Henrique Teles VIEIRA[1]
[1]*School of Electrical, Mechanical and Computer Engineering, Federal University of Goiás, Brazil*
[2]*Engineering School, Pontifical Catholic University of Goiás, Brazil*
brquirino@gmail.com

***Abstract*—The growth in data traffic is raising serious challenges for OTN in terms of improving their capacity efficiency in order to meet the new traffic requirements. Under these circumstances, the task of efficiently utilizing available resources opens opportunities for the development of a variety of techniques for network planning. This paper presents a decision support system for the optical transport network. It is considered the optical transport network planning problem where a traffic interest matrix between the demand nodes is specified. The network is modeled as a graph, through the arc-path approach. An Integer Linear Programming problem solved with a Firefly Algorithm is proposed for network planning, considering cost minimization. The main novelties of the proposed ILP model is that it accomplishes the optical network design with the possibility of multiple destinations of the traffic matrix and with dynamic allocation of the transmission system modularity. To solve the ILP optimization model the firefly algorithm, genetic algorithm and the exact method are used. Simulations are carried out to verify the performance of the bio-inspired algorithms in relation to the exact method. The results obtained with the firefly algorithm surpass those of the genetic algorithm and approximate the optimal result.**

***Index Terms*—artificial intelligence, communication networks, genetic algorithms, optical fiber networks, optimization.**

I. Introduction

The emergence of new technologies and services has imposed substantial changes on telecommunication systems. These transformations are happening on a large scale to meet the development of the Internet of Things (IoT), cloud computing, and 4th and 5th generation mobile communications systems. 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 [1-3].

The Optical Transport Network/Dense Wavelength Division Multiplexing (OTN/DWDM) enables a flexible network infrastructure with high transmission capacity. OTN also provides fault isolation management with advanced techniques for solving any operational issues [4]. This network, coupled with the versatility of dense multiplexing by wavelength division, minimizes the difficulty of data transport as in Time Division Multiplexing (TDM) inside Plesiochronous Digital Hierarchy (PDH) and Synchronous Digital Hierarchy (SDH) networks [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. The research on algorithms to optimize resource capacity is a promising direction to improve decision support systems designed to assist in network planning [6].

The problem of optical transport networks planning (NP-hard) has been widely studied in the literature [7]. The main purpose of planning is often the same: to allocate and size available resources in the most possible efficient way in terms of budget, with optimization models oriented towards minimizing costs. Basically, the methodologies can be differentiated according to: what are the adopted technologies, the most relevant resources for what is desired and a range of the network to be dimensioned [8].

Nature-inspired meta-heuristics are powerful tools for solving NP-hard combinatorial optimization problems. These methods are based on existing mechanisms in nature's biological phenomena [9]. For optical networks, it is possible to highlight the use of the genetic algorithm [10-12] and swarm-based algorithms [13-15], such as the Particle Swarm Optimization algorithm (PSO) [16], ant colony [17] and firefly algorithm [18].

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 [19-20]. 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 [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.

For the flow allocation problem in elastic optical networks (EON) with dedicated path protection, the work [25] proposes two metaheuristics based on optimization, one considering the particle swarm algorithm and another with tabu search. The problem is modeled by integer linear programming with respect to two different optimization

Digital Object Identifier 10.4316/AECE.2020.02007

criteria: with the average and maximum use of the spectrum. Also, regarding the problem of routing in elastic optical networks, in the work [26] it proposed an optimization method based on the Artificial Bee Colony algorithm (ABC) modeled by integer linear programming. The method is developed to be applied in large networks, bringing extensive numerical experiments focused on the performance evaluation of the proposed algorithm in relation to reference methods that solve the same problem.

The work of [27] presents a bi-objective modeling using integer linear programming 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.

This paper aims to develop a decision support system to be used in the strategic planning of optical transport networks. It proposes an integer linear programming model 0-1, which is solved using the exact method and the bio-inspired genetic and firefly algorithms. 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.

It is worth mentioning that the modeling proposed in this work allows us to contemplate the attendance of a peculiar 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 [21-23]. 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 and being able 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 [8], [28]. 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.

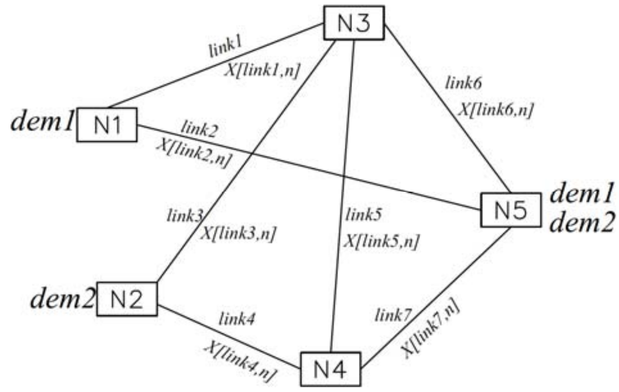

Figure 1. Transport network representation - arc-path

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 modularity transmission system [*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 of Fig. 1 two predicted demand values are indicated: *dem1* (from node 1 to node 5) and *dem2* (from node 2 to 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 are 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 demands are specified, the possible sets of paths to meet the demands *dem1* and *dem2* are indicated in Table I. Notice that the set of paths in Table I is not complete, being just examples of paths for d_1 and d_2 demands. This choice is made through binary type decision variables, *Y[demand, 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	Paths	Links
d_1	p_1	*link2*
	p_2	*link1 → link6*
d_2	p_3	*link4 → link7*
	p_4	*link3 → link6*
	p_5	*link4 → link5 → link6*

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}: a set of OTN/DWDM modules (capacities), [*n*], of the candidate OTN/DWDM transmission systems on the [*i*] ∈ *LINK*;

DEM: set formed by all the predicted demands to be met by the network;

TN: set formed by the possible destination nodes for each network demand;

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

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

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

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

$C_{[i][n]}$: cost of the OTN/DWDM transmission system capacity (in *Gbps*) [*n*] ∈ O_{LINK}, candidate in the link [*i*] ∈ *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*] ∈ *DEM*, in *Gbps*, to be met between the source node and the [*t*] ∈ *TN* possible destination nodes;

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

r: average optical network cost per *km*;

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

ILP 0-1 can then be formally formulated as:

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

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

$$\sum_{t\in TN} \sum_{[k]\in P_j} Y_{[j][k]} = 1, \forall [j] \in DEM \tag{3}$$

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

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 $r.l_{[i]}X_{[i][n]}$ 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 attained uses a single path, which favors the network operation. It is worth mentioning the possibility of evaluating multiple destinations for each demand.

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 proposed model here allocates the modularity of the OTN/DWDM system dynamically. The total demand flow drained by the link guides this decision. This approach is seen as a better sharing of resources when confronted with the proposals presented in [12], [24-26]. In these works, the system transmission modularity and the consequent cost accounting are specified in the pre-processing step 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*| x |O_{LINK}| + (|*DEM*| x |*TN*| x *k*). The amount of constraints is given by (2 x |*LINK*|+|*DEM*|).

IV. GA COMPUTATIONAL IMPLEMENTATION

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 [29]. 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).

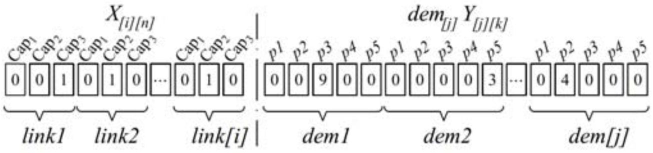

Figure 2. Chromosome encoding and genome structure

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 modularity [*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 ($p_1,...,p_k$) 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 modularity $Cap_{[n]}$ (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. The choice of each flow path must respect this multiplicity. The pseudocode of Genetic Algorithm (GA) is presented as Algorithm 1. Its operators are described further in the text.

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

Initial Population: 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, modularity 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.

Solutions Ranking (Selection): 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: 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: 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.

V. FIREFLY COMPUTATIONAL IMPLEMENTATION

The firefly algorithm is one of the known swarm-based algorithms, having different types of applications [30]. 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 [31].

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. Algorithm 2 presents the basic steps of the firefly algorithm.

ALGORITHM 2. STANDARD FIREFLY 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 $f(x)$, $x=(x_1,\ldots,x_d)^T$
Step03.	Generates initial population P of fireflies x_i(i= 1,2,…,n)
Step04.	Light intensity I_j in x_j 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.	Rank the fireflies and find the best global solution
Step07.	End while

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

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

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] (5) can be approximated as equation (6).

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

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.

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

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.

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

In a simpler way, ε_i can be replaced by *rand-1/2*, where *rand* 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 [9].

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 [32].

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 [33] and (b) the Swap distance [34]. 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 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 (11):

$$\beta = \frac{1}{1+\gamma d_{i,j}^{\ 2}} \tag{11}$$

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()≤β* 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 *α.random()* to perform many exchanges of two randomly chosen elements or choose many elements through

α*.random()* and shuffle their positions. The first option is easier to implement, but the results are not as good as the second [35]. 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 $\{1,\ldots,n\}$. Then $\alpha=1$ means that no step is done and $\alpha=n$ means to shuffle all elements of the permutation.

The coding routines, initial population generation, and ranking follow the process described in Section GA Computational Implementation. The following is a description of the firefly movement routine.

Firefly movement: 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).

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 three-modularity transmission system ($n=3$) and five candidate paths to flow each demand ($k=5$) each *TN* destination is shown in Fig. 3.

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 or 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.

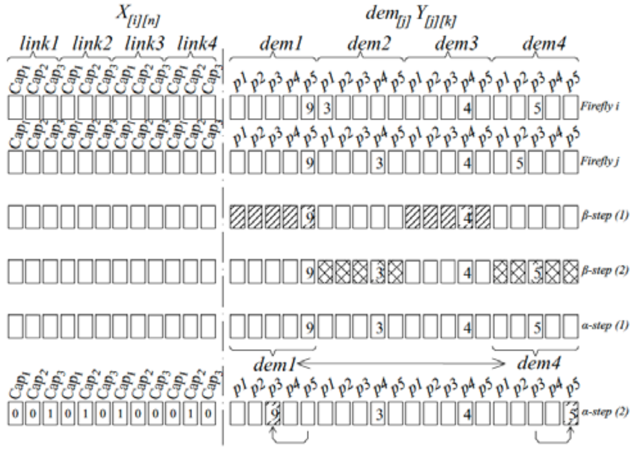

Figure 3. Firefly movement

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. 3 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 modularity ($X_{[i][n]}$ 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 is performed.

VI. 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 this partnership, relevant information was collected, such as demand points location, distances between locations, network topology, equipment costs, and optical network.

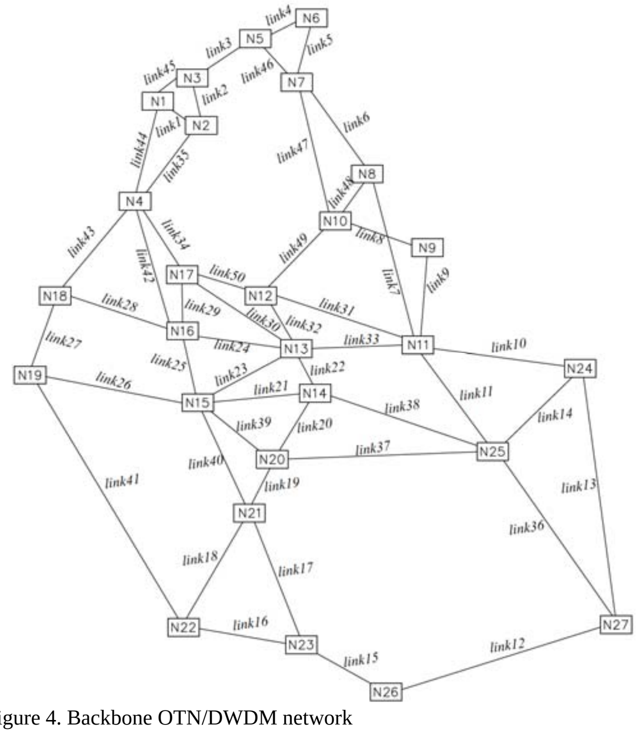

Figure 4. Backbone OTN/DWDM network

The studied OTN/DWDM network can assume three hierarchical models, with a transmission capacity of 40, 100 and 400 *Gbps*. The real raised network is composed of 27 individual nodes and 50 links. Fig. 4 shows the network topology, including possible links to be allocated and scaled.

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. 4 network, nodes N26 and N27 are the destination nodes.

TABLE II. OTN/DWDM NETWORK DEMANDS

Demand	**(*Gbps*)**	**Demand**	**(*Gbps*)**
dem1	13	dem15	4
dem2	0.5	dem16	10
dem3	1	dem17	22.5
dem4	5	dem18	11.5
dem5	0.5	dem19	12.5
dem6	37.5	dem20	7.5
dem7	1.5	dem21	19
dem8	2.5	dem22	3.5
dem9	12	dem23	25
dem10	2.5	dem24	30
dem11	7.5	dem25	15
dem12	4	dem26	13
dem13	20	dem27	30
dem14	50		

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.

Some modeling characteristics that are obeyed in Fig. 4 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 modularity of transmission and its range. Table III presents these values. The used values are relative. The cost of the 40 *Gbps* modularity OTN/DWDM system ($ 79000.00) is used as a basis. The considered optical network cost represents a consolidated calculation of the average network cost per kilometer, which is 0.00012.

TABLE III. COSTS OF OTN/DWDM SYSTEMS

Modularity (*Gbps*)	**Distance**	**Cost**
40	Up to 80*Km*	1.00
	Greater than 80*Km*	1.32
100	Up to 80*Km*	1.90
	Greater than 80*Km*	2.22
400	Up to 80*Km*	3.92
	Greater than 80*Km*	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) [36].

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 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).

The metric used for the *k-paths* application was the automatic generation using a modified version of the k-shortest algorithm [37-38]. The total distance between the source node and the target node, in *km*, is used as a deciding factor.

TABLE IV. EXACT METHOD COMPARISON

Number of Candidate Paths	**Cost**	**Total Capacity (*Gbps*)**	**Capacity Used (%)**	**Execution Time (*s*)**
10	70.84	3340	47.40	2460
12	68.07	3040	52.20	56077
14	67.30	3340	49.70	35566
16	67.21	3340	50.30	64376
18	67.21	3340	50.30	290739
20	67.21	3340	50.30	318140

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

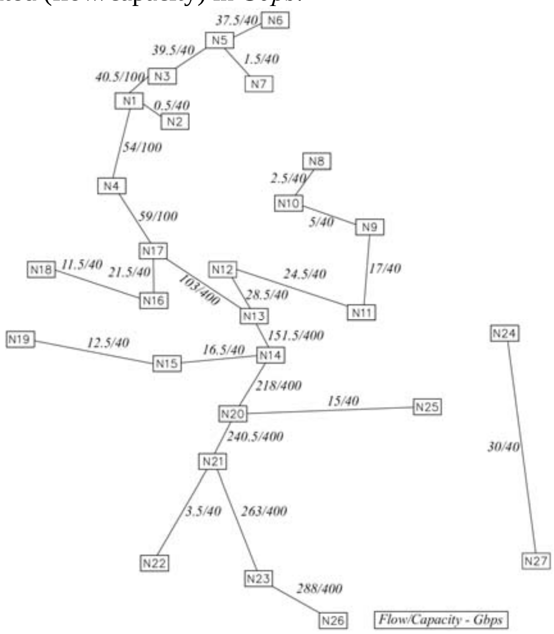

Figure 5. Solution network for k=16, 18 and 20

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%.

Table V presents the performance comparison between the genetic algorithm and the firefly algorithm with the change in the number of candidate paths.

TABLE V. COMPARISON OF BIO-INSPIRED METHODS

Number of Candidate Paths	**Method**	**Cost**	**Exact Method (Cost)**	**Gap**
10	Genetic Algorithm	79.14	70.84	11.72%
	Firefly	70.84		00.00%
12	Genetic Algorithm	78.16	68.07	14.82%
	Firefly	70.32		03.30%
14	Genetic Algorithm	73.79	67.30	09.64%
	Firefly	67.30		00.00%
16	Genetic Algorithm	78.84	67.21	17.30%
	Firefly	69.46		03.35%
18	Genetic Algorithm	80.70	67.21	20.07%
	Firefly	69.55		03.48%
20	Genetic Algorithm	78.82	67.21	17.27%
	Firefly	69.85		03.92%

The Gap represents the solution percentage that is above the optimal value. The stopping criterion was the execution time of 3600 seconds. For the genetic algorithm, 100 population individuals were used, 0.6 crossover probability and 0.10 mutation probability. In the firefly, 100 population fireflies were used, the value of β was 1, α with value 8 and 0.1 for the light absorption coefficient (γ).

Fig. 6, 7, 8 and 9 present the comparison for each Table V case (number of candidate paths variation) of the cost as a function of the execution time. The firefly algorithm proved to be more effective in all cases, with emphasis on 10 and 14 candidate paths, where firefly obtained the same value as the branch-and-bound exact method. The firefly algorithm in most cases reaches convergence before 500 seconds. The best genetic algorithm solution was for 14 candidate paths, with 09.64% greater than the optimal value. The genetic algorithm results were on average 15.14% higher than the exact method, and the firefly with 2.34%.

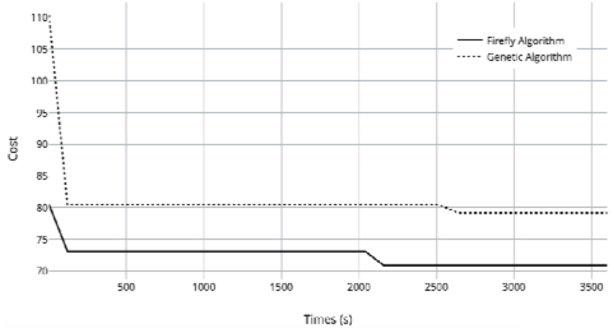

Figure 6. The convergence of algorithms for k=10

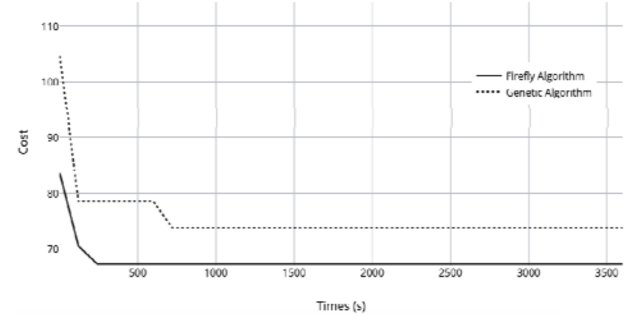

Figure 7. The convergence of algorithms for k=14

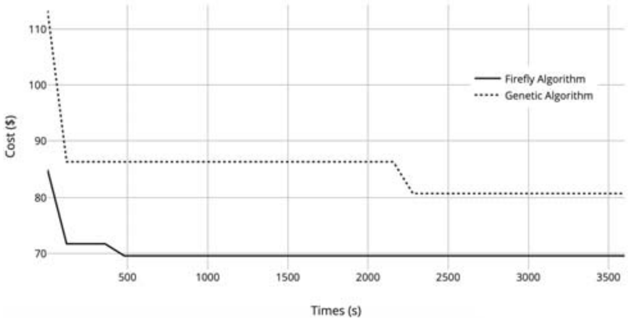

Figure 8. The convergence of algorithms for k=18

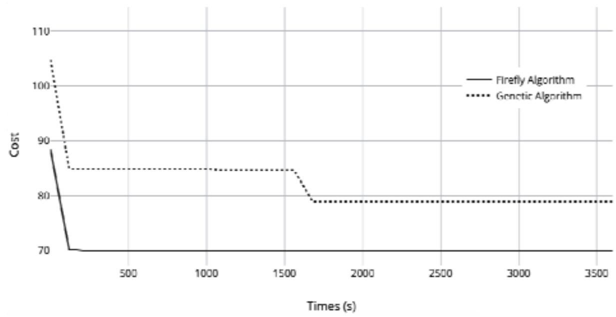

Figure 9. The convergence of algorithms for k=20

Table VI presents another comparison between the genetic algorithm and the firefly methods, where for each one, 10 replicates were performed (each algorithm was executed 10 times), and the stopping criterion was the number of generations (50). For the expected demands, k=10 paths were generated (5 for each destination). This k value allows good paths combinations variety to be analyzed by keeping time at a relatively low processing value (as seen in Table IV).

TABLE VI. COMPARISON OF FIREFLY AND GENETIC ALGORITHM

Method	**Genetic Algorithm**		**Firefly**	
Replica	**Cost**	**Time (s)**	**Cost**	**Time (s)**
1	80.98	34	70.84	143
2	78.52	33	73.09	145
3	79.72	38	73.32	139
4	77.99	31	73.09	141
5	79.47	31	73.30	141
6	80.66	32	73.23	134
7	81.85	30	73.23	138
8	81.58	36	73.18	141
9	80.34	35	73.32	144
10	82.84	29	73.09	148

The two methods have low values in terms of the processing time of each replica, highlighting that the genetic algorithm provided the smallest values. However, regarding the network deployment cost, the firefly algorithm proved to be more effective in all replicas, with replica 1 obtaining the same value of the exact method.

Fig. 10 shows the comparative costs of these methods in relation to the number of generations (with *k*= 10 generated paths). It can be observed that both methods in the first generations show a high drop, but the firefly has better values.

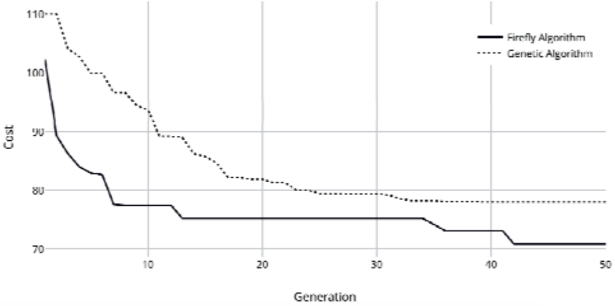

Figure 10. Convergence comparative for *k*=10

Table VII presents the comparison between the three methods for 10 candidate paths, taking the number of generations as the stopping criterion.

TABLE VII. COMPARISON BETWEEN METHODS FOR 10 PATHS

Method	Exact	Genetic Algorithm	Firefly
Best Cost	70.84	77.99	70.84
Worst Cost	-	82.84	73.32
Average Cost	-	80.39	72.97
Average Gap	-	13.52%	3.00%
Execution Time (s)	2460	329	1414

The execution time presented for the genetic and firefly algorithms is the sum of the 10 performed replicas. The performance in terms of computational time required to find the solution with the exact method presented a high value when compared to the other two methods.

The genetic algorithm in spite of having excellent results in terms of time, the solution had a higher network implantation value (approximately 10% greater than the optimal solution). The firefly algorithm stands out for its efficiency because it reached the same optimum value of the exact method with a shorter execution time (about 42.50%).

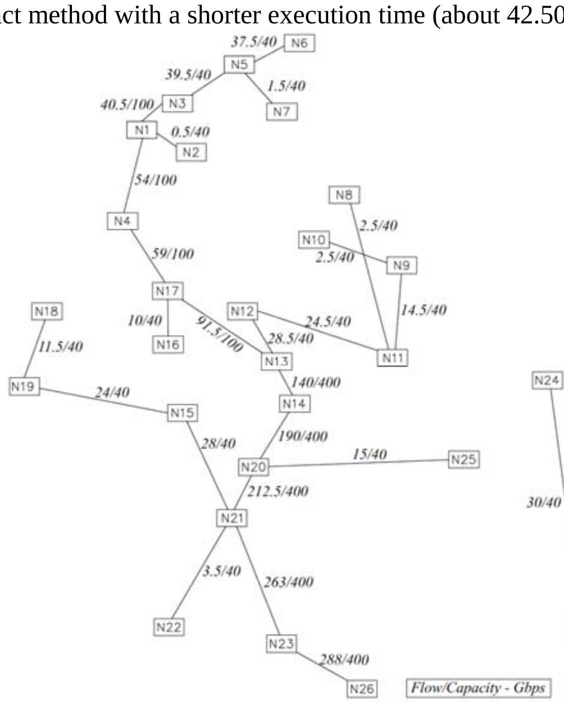

Figure 11. Solution network for *k*= 10

The network topology with the optimal solution (exact method and firefly), for *k*=10, obtained with the simulations, can be verified in Fig. 11.

The solution network also prioritized the demands to be drained by 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 solution network for *k*=10 (Fig. 11) is 5.40% more expensive than the solution network for *k* = 16, 18 and 20 (Fig. 5). As the allocated OTN/DWDM systems are the same in the two networks, the budget difference is due to the network cost and/or regenerators due to the changes in demand nodes 8, 15 and 18 service. Although the two networks have the same total available capacity (Table IV), Fig. 11 network shows a larger gap than that presented in Fig. 5.

VII. CONCLUSION

The constant telecommunication demand evolution has forced the sector operators to seek new solutions to plan their networks. The objectives to be achieved can be addressed by optimization techniques that can combine both cost minimization and revenue maximization.

This work presented a strategic planning proposal oriented towards the 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 and the 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 proposed modeling in this work presents the following most relevant differences: optical network design with demand matrix with multiple destinations; and dynamic allocation of transmission systems modularity with more efficient traffic aggregation.

The solution quality and computational time are strongly influenced by the number of candidate paths. The genetic algorithm, even with the shortest processing times, presented the costliest solutions. Regarding the firefly algorithm, it stands out for the good results both in terms of solution and computational time.

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, M. Gupta, "A review on activities of fifth generation mobile communication system," Alexandria Engineering Journal, 2017. doi:10.1016/j.aej.2017.01.043.

[2] S. Mumtaz, A. Morgado, K. M. S. Huq, J. Rodriguez, "A survey of 5G technologies: regulatory, standardization and industrial

perspectives," Digital Communications and Networks, 2017. doi:10.1016/j.dcan.2017.09.010.

[3] S. Li, L. Da Xu, S. Zhao, "5G internet of things: a survey," Journal of Industrial Information Integration, 2018. doi:10.1016/j.jii.2018.01.005.

[4] Q. Wang, G. Ying, "OTN for the future transmission network," Symposium on Photonics and Optoelectronics, 2012. doi:10.1109/SOPO.2012.6270500.

[5] T. G. Robertazzi. Optical Networks for Telecommunications. Introduction to Computer Networking. Springer, Cham, pp. 67-79, 2017.

[6] Y. S. Kavian. Intelligent Systems for Optical Networks Design: Advancing Techniques: Advancing Techniques. IGI Global, pp. 153-174, 2013.

[7] F. Musumeci, et al., "An overview on application of machine learning techniques in optical networks," IEEE Communications Surveys & Tutorials, 2018. doi:10.1109/COMST.2018.2880039.

[8] J. Simmons. Optical Network Design and Planning. Springer International Publishing Switzerland, pp. 10-15, 2014.

[9] X. S. Yang. Nature-Inspired Metaheuristic Algorithms. Luniver Press, Second Edition, pp. 1-5, 2010.

[10] A. Eira, J. Santos, J. Pedro, 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. doi:10.1364/JOCN.6.000326.

[11] 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. doi:10.1688/JISE.2017.33.2.3.

[12] D. Din, "Genetic algorithm for virtual topology design on MLR WDM networks," Optical Switching and Networking, vol. 18, pp. 20-34, 2015. doi:10.1016/j.osn.2015.03.003.

[13] S. A. Fernandez, A. A. Juan, J. A. Adrián, J., D. G. Silva, D. R. Terrén, "Metaheuristics in telecommunication systems: network design, routing, and allocation problems," IEEE Systems Journal, vol. 12, pp. 3948-3957, 2018. doi:10.1109/JSYST.2017.2788053.

[14] J. Mata, et al., "Artificial intelligence (AI) methods in optical networks: A comprehensive survey," Optical Switching and Networking, vol. 28, pp. 43-57, 2018. doi:10.1016/j.osn.2017.12.006.

[15] X. S. Yang, S. F. Chien, T. O. Ting. Bio-inspired computation in telecommunications. Morgan Kaufmann, pp. 23-38, 2015.

[16] C. Papagianni,et al., "Communication network design using particle swarm optimization," IEEE International Multiconference on Computer Science and Information Technology, pp. 915-920, 2008. doi:10.1109/IMCSIT.2008.4747351.

[17] 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. doi:10.1109/JSAC.2010.100504.

[18] 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. doi:10.1016/j.asoc.2014.03.046.

[19] J. Pedro, "Designing transparent flexible-grid optical networks for maximum spectral efficiency," IEEE/OSA Journal of Optical Communications and Networking, 2017. doi:10.1364/JOCN.9.000C35.

[20] H. Liu, C. Xiong, Y. Chen, C. Li, 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. doi:10.1016/j.yofte.2018.01.004.

[21] A. Eira, J. Pedro, 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. doi:10.1109/HPSR.2012.6260848.

[22] X. Chen, 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. doi:10.1364/JOCN.4.000248.

[23] J. R. Santos, A. Eira, J. Pires, "A Heuristic Algorithm for Designing OTN Over Flexible-Grid DWDM Networks," Journal of Communications, 2017. doi:10.12720/jcm.12.9.500-509.

[24] M. S. Bazaraa, J. J. Jarvis, H. D. Sherali. Linear Programming and Network Flows. 4a ed., Willey, pp. 1-35, 2010.

[25] R. Goscien, "Two metaheuristics for routing and spectrum allocation in cloud-ready survivable elastic optical networks," Swarm and Evolutionary Computation, vol. 44, pp. 388-403, 2019. doi:10.1016/j.swevo.2018.04.013.

[26] R. Goscien, M. Lozano, "Artificial bee colony for optimization of cloud-ready and survivable elastic optical networks," Computer Communications, vol. 128, pp. 35-45, 2018. doi:10.1016/j.comcom.2018.07.011.

[27] 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. doi:10.1016/j.comcom.2018.08.006.

[28] K. D. R. Assis, I. Queiroz, R. C. Almeida, H. Waldman, "MILP formulation for resource optimization in Spectrum-Sliced Elastic Optical Path Networks," Microwave & Optoelectronics Conference (IMOC), SBMO/IEEE MTT-S International, 2013. doi:10.1109/IMOC.2013.6646582.

[29] A. Ghosh, S. Tsutsui. Advances in Evolutionary Computing: Theory and Applications. Springer Science & Business Media, pp. 441-461, 2012.

[30] X. S. Yang. Cuckoo Search and Firefly Algorithm: Theory and Applications. Vol. 516, Springer, pp. 315-331, 2013.

[31] X. S. Yang, X. He, "Firefly algorithm: recent advances and applications." International Journal of Swarm Intelligence, vol. 1, pp. 36-50, 2013. doi:10.1504/IJSI.2013.055801.

[32] W. T. Lunardi, V. Holger, "Comparative study of genetic and discrete firefly algorithm for combinatorial optimization." Proceedings of the 33rd Annual ACM Symposium on Applied Computing. ACM, 2018. doi:10.1145/3167132.3167160.

[33] A. M. Mohsen, W. Al-Sorori, "A new hybrid discrete firefly algorithm for solving the traveling salesman problem." Applied Computing and Information Technology." Springer, Cham, 2017. p. 169-180. doi:10.1007/978-3-319-51472-7_12.

[34] G. K. Jati, R. Manurung, "Discrete firefly algorithm for traveling salesman problem: A new movement scheme." Swarm Intelligence and Bio-Inspired Computation. Elsevier, pp. 295-312, 2013. doi:10.1016/B978-0-12-405163-8.00013-2.

[35] X. S. Yang, X. He, "Applications of nature-inspired algorithms." In Mathematical Foundations of Nature-Inspired Algorithms, Springer, pp. 87-97, 2019. doi:10.1007/978-3-030-16936-7_6.

[36] D. R. Morrison, S. H. Jacobson, J. J. Sauppe, E. C. Sewell, "Branch-and-bound algorithms: a survey of recent advances in searching, branching, and pruning." Discrete Optimization, vol. 19, pp. 79-102, 2016. doi:10.1016/j.disopt.2016.01.005.

[37] A. Schickedanz, D. Ajwani, U. Meyer, P. Gawrychowski, "Average case behavior of k-shortest path algorithms." In International Conference on Complex Networks and their Applications, Springer, pp. 28-40, 2018. doi:10.1007/978-3-030-05411-3_3.

[38] F. Gang, "Finding k-shortest simple paths in directed graphs: A node classification algorithm." Networks, vol. 64, pp. 6-17, 2014. doi:10.1002/net.21552