Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

# Hybrid Discrete Grey Wolf Optimization Approach for solving Spectrum Assignment Problem in Elastic Optical Networks

**nassima tsabet**

nassima.tsabet@univ-annaba.dz

Badji Mokhtar Annaba University

**malika babes**

Badji Mokhtar Annaba University

**houssem eddine hadji**

Badji Mokhtar Annaba University

**Research Article**

**Keywords:** elastic optical networks, gray wolf optimizer, Network Optimization, meta-heuristics, spectrum assignment

**Posted Date:** September 24th, 2026

**DOI:** https://doi.org/10.21203/rs.3.rs-10618275/v1

**License:** This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

**Additional Declarations:** No competing interests reported.

# Hybrid Discrete Grey Wolf Optimization Approach for solving Spectrum Assignment Problem in Elastic Optical Networks

Tsabet Nassima[1*], Babes Malika[2*] and Hadji Houssem Eddine[3*]

[1,2,3*]Networks and Systems Laboratory, Badji Mokhtar Annaba University, Annaba, Algeria.

*Corresponding author(s). E-mail(s): nassima.tsabet@univ-annaba.dz; malikababes@yahoo.fr; houssem.hadji@gmail.com;

**Abstract**

The forthcoming optical networks is expected to exhibit characteristics of dynamism, heterogeneity, and unpredictability to effectively address the demand for bandwidth resources in telecommunications. Elastic optical networks (EONs) have emerged as a means to achieve enhanced spectrum allocation capabilities, characterized by increased flexibility and improved efficiency, through the utilization of smaller spectrum granularity. In this context, Routing and Spectrum Assignment (RSA) problem plays a crucial role in EON because it improves overall network performance.However, RSA -particularly spectrum allocation within extensive, large-scale networks- remains computationally challenging because it is an NP-hard optimization problem. Solving a large RSA problem is a complex and computationally intensive task. This work introduces a hybrid optimization approach that combines the Gray Wolf Optimizer(GWO) with Simulated Annealing (SA) to tackle spectrum allocation in large optical networks . The main objective of this hybrid approach is to improve the search process by achieving an effective balance between global exploration and local exploitation. The performance of the proposed technique is assessed through simulations using the 14-node NSFNET and the 28-node US Backbone topologies and several graghs topologies. The obtained results indicate that the method described in this study significantly enhances the network capacity.

**Keywords:** elastic optical networks, gray wolf optimizer, Network Optimization, meta-heuristics, spectrum assignment

# 1 Introduction

With the prevalence of high-bandwidth-demanding applications, such as cloud computing, the Internet of Things, and social media applications, conventional wavelength division multiplexing (WDM) networks are slowly becoming insufficient to support this rapid growth [1]. The Elastic Optical Network (EON) design has been suggested in response to growing network traffic. Its idea is that the spectrum is divided into smaller slices than those now employed in fixed-grid optical orthogonal frequency division multiplexing technology, enabling adjustable bandwidth [2]. By dynamically adapting data rates and spectrum allocation to the demands of connection requests, flexible optical transmission aims to optimize fiber capacity utilization.

However, EON technology introduces the problem of Routing and Spectrum Assignment (RSA), a problem proven to be NP-hard [3] and [4]. The RSA can be resolved by a two-step approach: first, choosing the optimal path connecting the source to the destination nodes, and second, allocating the spectrum according to the connection request. To successfully establish the pathway between the endpoint nodes, the system must provision an appropriate block of frequency slots (FSs) ensuring adherence to spectrum contiguity, path continuity, and non-overlapping constraints.

Three constraints must be satisfied in the EON to create a lightpath for a connection request [5] : (First, the spectrum continuity rule indicates that the assigned frequency slot indices must remain identical across all physical links along the chosen routing path. Second, the spectrum contiguity constraint requires that the allocated bandwidth for a given demand must consist of a continuous block of adjacent frequency slots; a single request cannot be fragmented across isolated spectral bands. Finally, the non-overlapping condition enforces that there should only be one connection request assigned to a frequency slot on a link at a time.

Because optimizing routing and spectrum assignment simultanously results in a complex combinatorial problem which needs significant computational overhead, most existing studies handle the RSA by identifying the routing path first,and then allocating the spectrum. Although alternative formulations may perform spectrum allocation prior to route determination [6].

To overcome the complexity of the routing and spectral assignment problem in optical networks, a wide array of computational intelligence and meta-heuristic techniques have been deployed. These include neural networks (NN) [9],(PSO) [14, 15, 16], bee colony optimization (BCO) [17], genetic algorithms (GA) ) [10, 11, 12, 13], and the grey wolf optimizer (GWO). While swarm intelligence algorithms like GWO have demonstrated exceptional results when implemented in several applications including RSA [19]. Few studies in the literature use the GWO to solve the RSA problem in EON. Notably, a dynamic GWO-based RSA scheme was introduced in [2] to process real-time connection requests as they arrive, yielding a substantial increase in spectral efficiency compared to legacy benchmarks. Similarly, an enhanced GWO variant was proposed in a dynamic Routing and Spectrum Allocation (RSA) technique based on grey wolf optimization is proposed to effectively manage incoming connection requests as they arrive at the network. The results demonstrate a significant improvement in spectrum efficiency compared to reference. In [5], an improved grey wolf optimization method (IGWO) is introduced to efficiently solve the maximum model. This method is

inspired by Particle Swarm Optimization and Differential Evolution (DE). The objectives include minimizing network power consumption, the total occupied spectrum, and the highest index of utilized frequency spectrum. GWO is a promising approach due to its efficient search capabilities and adaptability to complex optimization scenarios. For this purpose, in our work, we adapt a grey wolf optimizer to solve the sub-problem of spectrum assignment (SA), where our objective is to minimize the total use of spectrum in large optical networks. This paper is organized as follows: Section 2 presents the foundational concepts of the grey wolf optimizer. In section 3, we describe the SA problem and its mathematical modeling. We also introduce the proposed method in Section 4. The last two sections cover the results of numerical experiments, our conclusions, and a discussion of future work.

# 2 Methods

## 2.1 Grey wolf optimizer

The Grey Wolf Optimizer (GWO) algorithm is modeled by how grey wolves naturally search for the most effective method to hunt prey [22]. The GWO was created based on the leadership hierarchy theory developed by Mirjalili et al., [20]. It was created for numerical optimization issues, It has recently been utilized by several researchers across a variety of domains [21]. It is easily understood, simple to use, versatile, and scalable. It has the unique capacity to find the ideal balance between exploration and exploitation throughout the search, which results in favorable convergence [22]. Numerous empirical studies have shown that the performance of GWO is the best [19].

Grey wolves, also referred to as timber or western wolves, always reside in packs of five to eleven other wolves [20]. Based on the several wolf roles that aid in the advancement of the hunting process. In GWO, the members of the pack are organized into four divisions [22]. The leader wolf, often known as the alpha $\alpha$, is the most important and powerful member of the pack and is in charge of making important decisions [20]. The wolves in the second group are known as betas $\beta$, and they serve as the pack's advisors. Betas are subservient to alphas but can rule the other wolves. The delta makes up the third, it must respect the alpha and beta but it can command the other wolves [19]. The wolves in the final category are known as omega $\omega$ wolves and they have the lowest social standing in their pack. There is a common social hierarchy that establishes rules for domination and power (see Fig. 1).

Three steps are involved in their hunting process, according to Muro et al. [23].The first step involves pursuing, chasing, and approaching the prey. The next step is to encircle and distress the prey until it succumbs. Ultimately, the prey is attacked. To create GWO, Mirjalili et al. modeled social behavior and the hunt [24] as following.

1. `Encircling` As part of their hunting procedure, grey wolves enclose their prey. As shown in Eqs. (1) and (2).

$$\vec{D} = \mid \vec{C} \times \vec{X_p}(t) - \vec{X}(t) \mid \tag{1}$$

$$\vec{X}(t+1) = \vec{X_p}(t) - \vec{A} \times \vec{D} \tag{2}$$

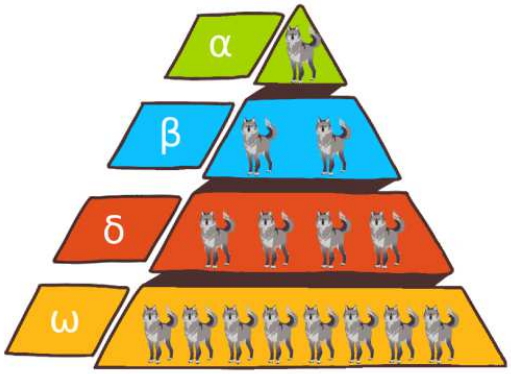



**Fig. 1** Social hierarchy of grey wolves(dominance ranking from top to bottom in decreasing order)

where the position of the wolf is $vecX(t)$, the position of the prey is $\vec{X_p}(t)$, t represents the iteration, coefficients $\vec{A}$ and $\vec{C}$ are two vectors. Following is how they can be calculated by Eqs. (3) and (4):

$$\vec{A} = 2 \times \vec{a} \times \vec{r_1} - \vec{a} \tag{3}$$

$$\vec{C} = 2 \times \vec{r_2} \tag{4}$$

Where the two coefficients $\vec{r_1}$, $\vec{r_2}$ are random vectors with values between 0 and 1, the coefficient $\vec{a}$ differs from them. It reduces linearly from 2 to 0 as shown in Eqs. (5):

$$\vec{a} = 2 - 2 \times \frac{t}{T_{max}} \tag{5}$$

where $t$ represents the current iteration and $T_{max}$ denotes the maximum iteration.

2. `Hunting` Grey wolves are usually adept at locating their prey to execute encirclement. The hunting phase is mostly mentored by the alphas. It can also occasionally be attributed to beta and delta. Nevertheles, finding the prey location in an unknown search area is challenging. it is hought that the alpha , beta, and delta would provide the most accurate information regarding the potential location of the prey. Consequently, the top three results are retained, aiding other search agents (omegas) in adjusting their rankings by following the leading search agents. The process of hunting is represented mathematically as follows:

$$\vec{D_\alpha} = \mid \vec{C_1} \times \vec{X_\alpha} - \vec{X} \mid \tag{6}$$

$$\vec{D_\beta} = \mid \vec{C_2} \times \vec{X_\beta} - \vec{X} \mid \tag{7}$$

$$\vec{D_\Delta} = \mid \vec{C_3} \times \vec{X_\Delta} - \vec{X} \mid \tag{8}$$

$$\vec{X_1} = \vec{X_\alpha} - \vec{A_1} \times \vec{D_\alpha} \tag{9}$$

$$\vec{X_2} = \vec{X_\beta} - \vec{A_2} \times \vec{D_\beta} \tag{10}$$

$$\vec{X_3} = \vec{X_\Delta} - \vec{A_3} \times \vec{D_\Delta} \tag{11}$$

$$\vec{X_{(t+1)}} = \frac{\vec{X_1} + \vec{X_2} + \vec{X_3}}{3} \tag{12}$$

where $\vec{X_\alpha}$ , $\vec{X_\beta}$, $\vec{X_\Delta}$ are the positions of alpha, beta, and delta, $\vec{A_1}$, $\vec{A_2}$, $\vec{A_3}$, $\vec{C_1}$, $\vec{C_2}$, $\vec{C_3}$ are six vectors. They're calculable by Eqs. (3–4) respectively.

The location of the prey should be assessed by the alpha, beta, and delta wolves to adjust the positions of the other search agents in their hunt.

3. `Attacking` When the target prey stops moving, the wolves initiate their final attack. To model this convergence, the searching boundaries are tightened by decreasing the value of $\alpha$ and $\vec{A}$ linearly from 2 down to 0 across successive iterations. Because the vector $\vec{A}$ is randomly generated within the shrinking interval $[-\alpha, \alpha]$ this decay gradually decreases the step size of the wolves.

Algorithm 1 provides the pseudo-code for the Grey Wolf Optimization.

**Algorithm 1** Grey Wolf Optimizer algorithm (GWO)

initialization phase;
initialize the parameters $a$, $A$ and $C$;
create an initial population of random solutions;
evaluate the fitness value for every grey wolf;
choose the best three grey wolves as $X_\alpha$, $X_\beta$ and $X_\Delta$ respectively;
$t \leftarrow 0$
**while** $t \leq tmax$ **do**
adjust the position of each grey wolf utilizing Eq. 12;
evaluate the fitness score of each grey wolf;
update $a, A$ and $C$;
evaluate the fitness score of each grey wolf;
update $X_\alpha$, $X_\beta$ and $X_\Delta$;
$t \leftarrow t + 1$
**end while**
return $X_\alpha$

GWO is typically designed for continuous optimization problems; however, it can be adapted for discrete problems, as demonstrated in our approach.

# 3 Simulated annealing

The simulated annealing heuristic (SA) was proposed by Kirkpatrick in 1983 [7], inspired by the metallurgy annealing process in physics. It has been applied extensively to solve both continuous and discrete optimization issues. This type of local search strategy is known as "hill climbing," where a probability measure is utilized to determine whether to accept a bad result or not. The fundamental idea of SA, which simulates the best possible solution procedure, is founded on physics's solid annealing process. To form a solid with a regular microstructure, the material is initially heated to a designated temperature to facilitate melting. Subsequently, the material is gradually cooled. As the temperature of the solid increases, the particles within the solid accelerate and move in a non-uniform manner. Conversely, as the solid cools, the particles decelerate. When the temperature attains ambient levels, the particles exist in their lowest energy state and achieve thermal equilibrium. Similar to other heuristic methods, in simulated Annealing the population is initiated and assigned an initial temperature, Subsequently, the fitness of all individuals in the population is evaluated. Once the new population is generated using the temperature and a random number, its fitness value is evaluated. The best individual is incorporated into the current population based on the ratio of the fitness differences between the old and new populations relative to the current temperature, which is assessed using a random number. The temperature is then updated by applying a cooling. Until convergence or maximum generations are reached, this procedure is repeated.
The acceptance criterion is the Metropolis function. The possibility of an increase in temperature $T$ of size $\Delta E$ is given by:

$$P = e^{-\Delta S/T} \tag{13}$$

Once a random number $R$ is generated between 0 and 1, the random number is compared to the $P$. A move upwards is only chosen if $R < P$, and smaller uphill movements have a greater probability of being chosen than bigger ones. As a result, there is occasionally an opportunity to accept a highly unlikely move while missing a very likely one [8].On the other hand, when the temperature $T$ lowers, the probability of selecting an upward movement $\Delta S$ decreases.

The simulated annealing pseudo-code is detailed in Algorithm 2.

# 4 Problem Description

This section describes the SA problem and its mathematical model in EONs.

## 4.1 Problem Description

Formally, consider a physical optical network topology as a graph G (V, E), where V is the node-set, and E is the vertices set. Moreover, in graph G, the vertices denote the requests (demands), with each vertex's weight corresponding to the required bandwidth for that request. The edges connecting the vertices symbolize the guard band necessary to prevent cross-talk. The weight of each edge reflects the size of the guard

**Algorithm 2** Simulated annealing (SA)

1: Generate an initial random solution set $S$
2: Evaluate the fitness function of $S$ i.e $Cost(S)$
3: Initialize the temperature
4: while temperature is remains elevated do
5: for $L$ iterations do
6: Choose a neighboring solution $S^{'}$ of $S$(generated by making a minor adjustment to the current solution).
7: calcule $S = Cost(S^{'}) - Cost(S)$
8: if $S^{'} \leq S$ then
9: Assign $S = S^{'}$ (indicating a downhill move)
10: else
11: (uphill move)
12: accept this based on probability $P = e^{-\Delta S/T}$
13: end if
14: end for
15: decrease the temperature
16: end while

band. The goal is to identify a short path in G that encompasses each request exactly once while minimizing the bandwidth usage of the optical network.

## 4.2 Modelization for SA problem in EON

In letterature Numerous integer programming models for RSA are presented. In this work, we focus on the so-called the distance spectrum assignment (DSA) model [25], which use different-size guardbands to adapt to the crosstalk level or the security requirement for various pairs of lightpaths in an EON. Also, in this study they show that it can be solved optimally in polynomial time by simplifying it to an ordered SA issue, which imposes a new vertex order constraint. This model was among the ones with the best performance in the experiments described.

The objective function of the Spectrum Assignment (SA) is defined as minimizing the maximum utilization of frequency slots (FSs), as follows:

$$Minimiser\ max\ (FS) \tag{14}$$

Frequency slots (FS) denote the spectral intervals allocated to the vertices in the Spectrum Assignment graph.

The Spectrum Assignment (SA) problem is subject to certain constraints that must be adhered to, including the following:

- **The constraint on required bandwidth**
  The bandwidth size $w_{v_i}$ allocated to each vertex $v_i$ in the set of vertices $V$ must be equal to its weight $v_i^w$, i.e.:

$$\mid w_{v_i} \mid = v_i^w, \forall v_i \in V \tag{15}$$

- **The constraint on spectrum continuity**
  This constraint ensures that the allocation of spectrum intervals to a request remains consistent across all links in the optical path. It is fundamentally guaranteed, as each optical path is represented by a vertex in the DSA graph.
- **The constraint on spectrum adjacency**
  Consists of ensuring that the FSs assigned to a vertex must be contiguous in the spectral domain. Knowing that $v_i^b$ and $v_i^a$ are the first and the last index of $FS$ assigned to $v_i$, we can express this constraint as follows :

$$\{ v_1^b, v_{i+1}^b, ....., v_{i-1}^a, v_i^a \}, v_i^b, v_i^a \in \mathbb{N}^+$$

- **The constraint on guardband**
  To meet the guardband requirements between two vertices,$v_i$ and $v_j$, the number of frequency slots (FS) assigned to the edge connecting $v_i$ and $v_j$ must be at least equal to the weight of that edge, denoted as $d_{v_i v_j}$. Thus, the following condition must hold:

$$FS(v_i v_j) \leq d_{v_i v_j} \tag{16}$$

- **The constraint on order**
  The indices of the spectral intervals assigned to the frequency slots (FSs) for the vertices must be arranged in ascending order.

We evaluate each solution within the population using the evaluation procedure outlined in [25] (refer to Algorithm 1)

**Algorithm 3** Evaluation procedure

**Inputs** the graph $G(V, E)$ ; the weight of vertex: $v_i^w$, the weights of edge: $d_{v_i v_j}$, the sequence of the vertices: $O_i = (v_{i1}, v_{i2}, ....., v_{in})$.
**Outputs**: assignment strategy $Seq$ ; and the maximum number of frequency slots utilized.
$v_{i1}^b \leftarrow 1$ ;
$v_{i1}^a \leftarrow v_{i1}^w$;
$Seq \leftarrow v_{i1}^b$;
$j \leftarrow 2$;
**while** $j \leq |V|$ **do** ;
    $s1 \leftarrow max(v_{ik}^a + d_{v_{i_k} v_{j_k}} + 1)$;
    $s2 \leftarrow v_{ij-1}^b$;
    $v_{ij}^b \leftarrow max(s1, s2)$ ;
    $v_{ij}^a \leftarrow v_{ij}^b + v_{ij}^w - 1$;
    $Seq \leftarrow Seq \cup v_{ij}^b$;
    $j \leftarrow j + 1$;
**end while**
**return** $Seq$ and $max v_{i1}^a; 1 \leq j \leq |V|$

**Table 1** NOTATIONS

| | |
|---|---|
| $FS$ | Spectrum interval indices used, $FS \in \mathbb{N}^+$ |
| $v_i$ | Number of requests $i$, $v_i \in V$ |
| $v_i^w$ | The request weight $i$ in bandwidth |
| $w_{v_i}$ | The set of $FS$ assigned to $v_i$ |
| $v_i^b$ | The initial $FS$ index assigned to $v_i$ |
| $v_i^a$ | The final $FS$ index assigned to $v_i$ |
| $d_{v_i v_j}$ | Weight of the edge between $v_i$ and $v_j$ |

# 5 The proposed discrete grey wolf optimizer

This section provides a description of the proposed GWO–SA algorithm for the spectrum assignment problem. As noted earlier, GWO excels at global search due to its strong exploration capability; however, in large search spaces it can exhibit premature convergence and become trapped in local minima. To address this limitation, we propose a hybrid method that integrates GWO with SA. The integration strengthens exploration, facilitates escape from local optima, and improves overall optimization efficiency. Specifically, the SA mechanism is embedded to refine the wolves' position updates within the Grey Wolf Optimizer. The principal steps of the algorithm comprise initialization, distance calculation, position updating, and exploitation. Each step is described in more detail in the following subsections.

## 5.1 Initialization

In the initialization phase, a population of wolves is generated at random, with each wolf encoded as a vector in the solution space. This vector is an integer sequence that specifies the ordering of requests, which in turn determines the assignment of spectrum intervals. Subsequently, the initial temperature T, cooling rate, and other key parameters of GWO—including a, A, and the positions of the alpha, beta, and delta wolves—are initialized.

## 5.2 Distance calculation

Within GWO, the alpha, beta, and delta wolves guide the population by inducing position updates based on distance-difference vectors relative to their locations. In our discrete formulation of GWO, distance is computed using Hamming distance rather than the difference vector eq. 17.

$$\vec{D_j} = random[1, hamming - disatnce(X_{i \in N}, X_{j \in \alpha,\beta,\delta}) \tag{17}$$

Hamming distance [15] serves as a quantitative metric for assessing the dissimilarity between two vectors of equal length. It is defined as the count of positions at which the corresponding symbols in the vectors differ.

## 5.3 Position update using simulated annealing

Rather than directly updating the positions of wolves based solely on the alpha, beta, and delta positions, we incorporate simulated annealing as a mechanism to refine and

adjust the position of each wolf, thus introducing a probabilistic approach to enhance exploration and avoid premature convergence.

For each wolf in the population :

1. **Generation of a neighboring solution via Swap mutation controlled by the Hamming distance:**
   Starting from the current solution of wolf $X_i$, generate a neighboring solution $X_i^{'}$ by applying a swap mutation to a number of positions proportional to the Hamming distance computed between $X_i$ and the position of the leader wolf. This ensures that the perturbation is consistent with the proximity to the best solutions.
2. **Simulated Annealing Update Step:**
   - Calculate the fitness value for the new candidate solution.
   - Calculate the difference in fitness ($\Delta E$) between the new solution and the current solution.
   - If the new solution is better ($\Delta E < 0$), accept it as the new position.
   - If it is worse ($\Delta E > 0$), accept it with a probability $e^{-\Delta E/T}$, where $T$ is the current temperature. This allows for escaping local optima by occasionally accepting worse solutions.

### 5.4 Temperature update

The decrease in temperature follows the fractional reduction function at each iteration to progressively limit random moves.

### 5.5 Termination Criteria and Final Outcome

Repeat the above steps (exploration with GWO and position updates with SA) until a stopping criterion is met. In the proposed method, a defined maximum number of iterations is taken as the termination criterion. Finally, the best agent $X_\alpha$ is returned as the outcome of the entire optimization procedure.

The discrete GWO-SA algorithm pseudo-code is shown in Algorithm 4.

## 6 Experiments and Analysis

To substantiate the improvements introduced by the proposed algorithm, To demonstrate the efficiency of the proposed algorithm, we compared its results with those obtained from the Two-Phase algorithm [25] using the same dataset.We also compared our results with the optimal results generated by the CPLEX 12.9.0 solver utilizing the Python API for the larger instances. This section presents the results. In Section 5.1, the simulation topologies are detailed, while Section 5.2 provides the experimental results and analysis.

All digital experiments were conducted on a computer featuring an Intel® Core i5 processor with a 2.40 GHz CPU and 8 GB of RAM.

**Algorithm 4** The discrete GWO-SA algorithm

Randomly initialize the population of the wolves;
Initialize the temperature for SA $T$;
while (iteration ¡ max-iteration)
Calculate the fitness value of each wolf according to algorithm 3;
Select the three wolves with the best fitness $(X_\alpha, X_\beta, X_\delta)$;
FOR(each wolf in the population)
Calculate distance Hamming according to eq. 17;
Generate neighbor of current wolf by Swap mutation;
Evaluate the fitness of the new wolf;
Update position by applying simulated annealing;
IF(new position is better)
Accept the new position;
ELSE
Accept the new position with a probability based on temperature;
Decrease temperature over iterations;
ENDFOR
Update the alpha, beta, delta wolves after position updates;
IF(stopping criterion met)
ENDWHILE
Return the best solution $X_\alpha$;

## 6.1 Simulation graphs

To demonstrate the improvements introduced by the proposed algorithm, the simulation study targets diverse graph types, including random and complete graphs. For each graph, we conduct 30 independent simulation runs and report the averaged outcomes to enhance statistical reliability.

We utilized the NetworkX and Matplotlib libraries in Python to create various types of graphs. Random graphs were employed, where each pair of vertices is directly connected with a probability of 0.5. The weights for the vertices and edges were randomly selected from the range [1, $|V|$], where $|V|$ denotes the number of vertices. Additionally, we utilized a complete graph, in which the weights of the vertices and edges were also randomly selected from the range [1, $|V|$].

## 6.2 Experimental results and analysis

1. Random graphs

   Table 2 presents the results of the comparison of two algorithms (Two-Phase Algorithm, SA-GWO Algorithm) with the results of the CPLEX Solver based on their average number of frequency slots used across different random graph configurations, measured by varying the number of nodes (from 14 to 19). From Fig. 2, we observe that for smaller node counts (14-15), all algorithms show comparable performance, reflecting their effectiveness in simpler scenarios. However, as

the graph size increases (especially from 17 nodes onward), the SA-GWO Algorithm demonstrates significant superiority, validating its robustness in managing increased complexity.

| Number of Nodes | Two-phases | SA-GWO | Cplex |
|---|---|---|---|
| 14 | 72.7 | 71.63 | 71.6 |
| 15 | 72.5 | 71.13 | 70.1 |
| 16 | 75.5 | 74.1 | 73.6 |
| 17 | 91.0 | 89.1 | 87.5 |
| 18 | 99.4 | 96.23 | 94.5 |
| 19 | 110 | 107.86 | 105 |

**Table 2** The average FSs computed for various algorithms: two-phase algorithm, SA-GWO algorithm and the CPLEX solver.

The results in Table 3 indicate that the proposed SA-GWO algorithm minimizes the maximum number of frequency slots (FSs) utilized in the network compared to the Two-Phase reference algorithm, and it approaches the optimal results produced by the CPLEX solver in most cases (see Fig. 3).

| Number of Nodes and edges | Two-phases | SA-GWO | Cplex |
|---|---|---|---|
| 14 nodes,15 edges | 35 | 33 | 31 |
| 14 nodes,30 edges | 48 | 50 | 39 |
| 14 nodes,45 edges | 59 | 52 | 45 |
| 14 nodes,60 edges | 85 | 80 | 75 |
| 14 nodes,75 edges | 106 | 99 | 94 |
| 14 nodes,90 edges | 140 | 132 | 123 |

**Table 3** The average FSs computed for various algorithms based on the number of nodes and edges.

2. Random complete graphs Table 4 presents the average of FSs derived from the six random complete graphs. In complete graphs, we find that the number of FSs sets utilized is greater than that in random graphs. This can be interpreted as the requirement for the FS sets assigned to each vertex to be mutually disjoint. However, the SA-GWO algorithm allows us to obtain solutions almost identical to those of CPLEX.
3. Large Graphs

   To illustrate the effectiveness of our algorithm for large graphs that have not been addressed in the existing literature, we performed an analysis of six types of random complete graphs, with the number of vertices ranging from 20 to 25, as detailed in Table 5. In this case, we compared the results of SA-GWO with those obtained using the standard GWO method, as well as with those from the CPLEX solver.

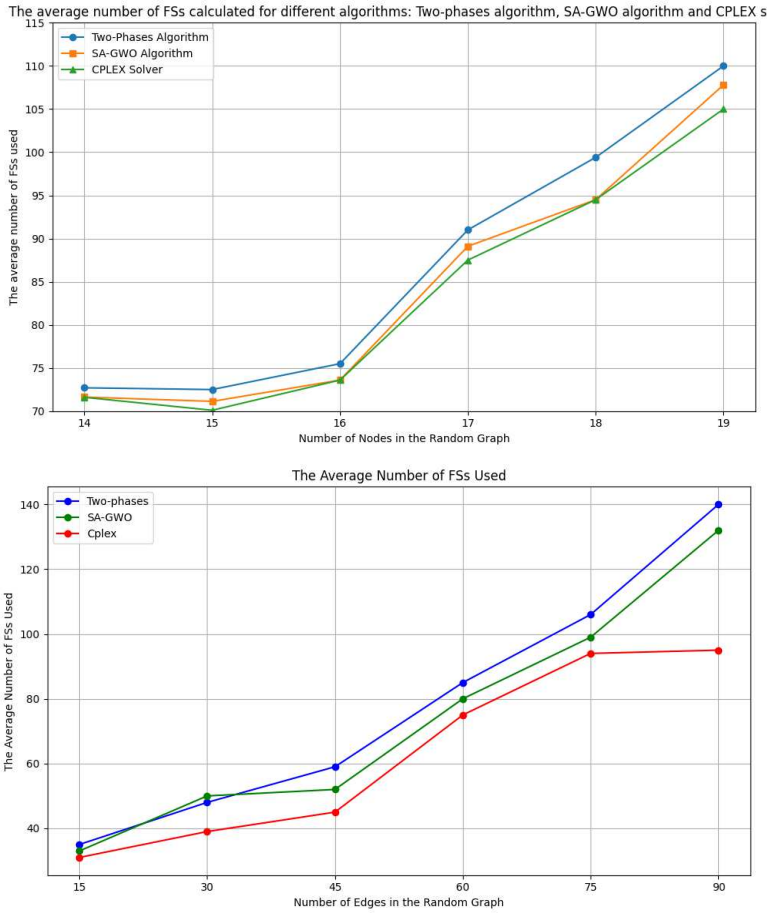



**Fig. 2** The average FSs computed for various algorithms: Two-phases algorithm, SA-GWO algorithm and CPLEX solver

| Number of Nodes | Two-phases | SA-GWO | Cplex |
|---|---|---|---|
| 14 | 143.6 | 142 | 142.4 |
| 15 | 163.6 | 164 | 160.5 |
| 16 | 196.6 | 193 | 191.1 |
| 17 | 213.3 | 206 | 207.6 |
| 18 | 231.3 | 225 | 223.7 |
| 19 | 241.5 | 233 | 231.8 |

**Table 4** The average number of FSs calculated for different algorithms in complete graphs

In Table V and Fig 4, it is evident that the SA-GWO algorithm does not reach the optimal results achieved by CPLEX. However, it significantly reduces the maximum number of FSs used when compared to the standard GWO algorithm.

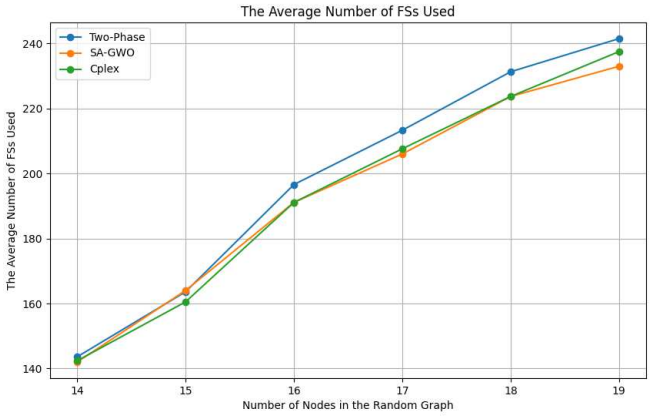



**Fig. 3** The average number of FSs calculated for different algorithms: Two-phases algorithm, SA-GWO algorithm and CPLEX solver

| Number of Nodes | GWO | SA-GWO | Cplex |
|---|---|---|---|
| 20 | 336.08 | 308.53 | 260.7 |
| 21 | 399.04 | 349.11 | 299.11 |
| 22 | 480.96 | 440.06 | 340.36 |
| 23 | 532.0 | 494.30 | 398.70 |
| 24 | 560.62 | 504.39 | 470.0 |
| 25 | 580.19 | 529.0 | 498.0 |

**Table 5** The average FSs computed for large graphs

Based on our observations, we can assert that the proposed SA-GWO algorithm effectively approximates optimal solutions for both small and large graphs.

# 7 Conclusion

In this study, we introduced a hybrid optimization methodology that integrates Grey Wolf Optimization (GWO) with Simulated Annealing (SA) to address the Spectrum Assignment problem in Elastic Optical Networks (EONs). The proposed approach aims to enhance network capacity by optimizing the allocation of frequency slots, while effectively balancing exploration and exploitation within the search space.

The simulation results across various network topologies demonstrated that the proposed SA-GWO hybrid algorithm achieves superior performance compared to traditional approaches. By incorporating Simulated Annealing into the optimization process, the hybrid approach enhanced search efficiency and solution quality, effectively minimizing spectrum fragmentation and maximizing resource utilization.

For future research, we plan to extend the proposed algorithm to dynamic spectrum assignment scenarios that account for real-time traffic demands. Additionally, we aim to explore multi-objective optimization techniques to simultaneously address power

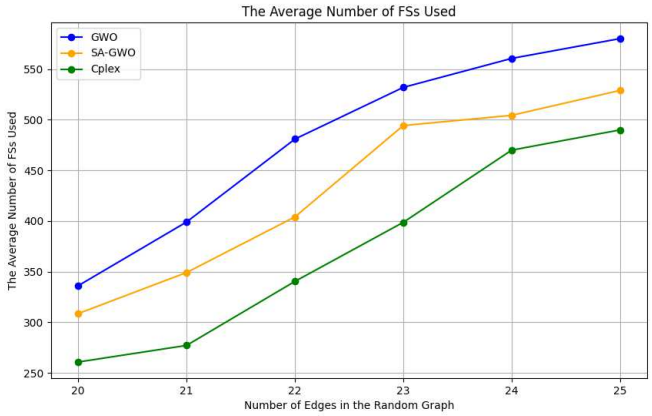



**Fig. 4** The average FSs computed for different algorithms for large graphs

consumption and latency. Furthermore, integrating machine learning-based heuristics will be investigated to enhance predictive spectrum allocation and improve decision-making in Elastic Optical Networks.

# 8 Declarations

## 8.1 Ethics approval and consent to participate

Our manuscript does not have both human and animal studies.

## 8.2 Consent for publication

Our manuscript does not contain any individual person's data in any form (including any individual details, images or videos)


## 8.3 Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.


## 8.4 Availability of data and materials

No datasets used in our work


## 8.5 Acknowledgements

not applicable

# References


[1] Mukherjee, J. et al. Routing and Spectrum Allocation Using Elastic Optical Network. Am. J. Electron. Commun. 3, 12–15 (2022).

[2] Mahala, N. and Thangaraj, J. Routing and Spectrum Allocation in Elastic Optical Networks for Ecosystem Monitoring. IEEE Sens. J. 21, 17620–17628 (2021).

[3] Shirazipourazad, S., Zhou, C., Derakhshandeh, Z. and Sen, A. On routing and spectrum allocation in spectrum-sliced optical networks. Proc. - IEEE INFOCOM 385–389 (2013) doi:10.1109/INFCOM.2013.6566800.

[4] Talebi, S., Bampis, E., Lucarelli, G., Katib, I., and Rouskas, G. N. Spectrum assignment in optical networks: A multiprocessor scheduling perspective. J. Opt. Commun. Netw. 6, 754–763 (2014).

[5] Xuan, H., Lin, L., Qiao, L., and Zhou, Y. Grey wolf algorithm and multi-objective model for the manycast RSA problem in EONs. Inf. 10, 1–16 (2019).

[6] Dinarte, H. A., Correia, B. V. A., Chaves, D. A. R., Almeida, R. C. Routing and spectrum assignment: A metaheuristic for hybrid ordering selection in elastic optical networks. Comput. Networks 197, (2021).

[7] Kirkpatrick S, D Jr Gelatt C, P. Jr. Vecchi M. Optimization by simulated annealing. Science (1983) ;220,671–80.https://doi.org/10.1126/science.220.4598.671

[8] C. Low, "Simulated annealing heuristic for flow shop scheduling problems with unrelated parallel machines," Computers and Operations.Research, vol. 32, pp. 2013-2025, (2005).

[9] H. O. Silva, C. J. Bastos-Filho, Inter-domain routing for communication networks using hierarchical hopfield neural networks, Engineering Applications of Artificial Intelligence 70(2018) 184 – 198.

[10] P. Lechowicz, K. Walkowiak, Genetic algorithm for routing and spectrum allocation in elastic optical networks, in: 2016 Third European Network Intelligence Conference (ENIC), IEEE, 2016, pp. 273-280.

[11] Dinarte, H. A., Correia, B. V. A., Chaves, D. A. R. and Almeida, R. C. Routing and spectrum assignment: A metaheuristic for hybrid ordering selection in elastic optical networks. Comput. Networks 197, (2021).

[12] G. Savva, K. Manousakis, G. Ellinas, Providing confidentiality in optical networks: Metaheuristic techniques for the joint network coding-routing and spectrum allocation problem, in: 2020 22nd International Conference on Transparent Optical Networks (ICTON), IEEE, 2020, pp. 1-4.

[13] Agarwal, R. and Bhatia, R. Multipath routing using genetic algorithm in elastic optical network. J. Opt. 53, 2316–2321 (2024).

[14] 1. Goścień, R. Two metaheuristics for routing and spectrum allocation in cloud-ready survivable elastic optical networks. Swarm Evol. Comput. 44, 388–403 (2019).

[15] Shi, T., Yang, Y., Huang, S., Chen, L., Kuang, Z., Heng, Y., Mei, H. (2019). Molecular image-based convolutional neural network for the prediction of ADMET properties. Chemometrics and Intelligent Laboratory Systems, 194. https://doi.org/10.1016/j.chemolab.2019.103853

[16] M. Cavalcante, H. Pereira, D. Chaves, R. Almeida, Optimizing the cost function of power series routing algorithm for transparent elastic optical networks, Optical Switching and Networking 29 (2018) 57 - 64.

[17] D. Christian, P. López, M. Soto-Bovó, J. Colbes, P. Pinto-Roa, Semi-dynamic Routing and Spectrum Assignment with variable bandwidth in Elastic Optical Networks. Bee-inspired Algorithms approach.(2021)

[18] B. C. Chatterjee, N. Sarma, E. Oki, Routing and spectrum allocation in elastic optical networks: A tutorial, IEEE Communications Surveys Tutorials 17 (3) (2015) 1776-1800.

[19] Yu, X. and Wu, X. Ensemble grey wolf Optimizer and its application for image segmentation. Expert Syst. Appl. 209, 118267 (2022).

[20] Gupta, S. Deep, K. A novel Random Walk Grey Wolf Optimizer. Swarm Evol. Comput. 44, 101–112 (2019).

[21] Sopto, D. S., Ayon, S. I., Akhand, M. A. H. and Siddique, N. Modified Grey Wolf Optimization to Solve Traveling Salesman Problem. 2018 Int. Conf. Innov. Eng. Technol. ICIET 2018 1–4 (2019) doi:10.1109/CIET.2018.8660872.

[22] Faris, H., Aljarah, I., Al-Betar, M. A. Mirjalili, S. Grey wolf optimizer: a Review of recent variants and applications. Neural Comput. Appl. 30, 413–435 (2018).

[23] C. Muro, R. Escobedo, L. Spector, R.P. Coppinger, Wolf-pack (Canis lupus) hunting strategies emerge from simple rules in computational simulations, Behav. Process. 88 (2011) 192–197.

[24] Mirjalili, S., Mirjalili, S. M. and Lewis, A. Grey Wolf Optimizer. Adv. Eng. Softw. 69, 46–61 (2014).

[25] H. Wu, F. Zhou, Z. Zhu, and Y. Chen, "On the distance spectrum assignment in elastic optical networks," IEEE/ACM Trans. Netw., vol. 25, no. 4, pp. 2391–2404, Aug. 2017.