International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(2) 2026 pp. 2345-2360
https://doi.org/10.70917/ijcisim-2026-5992
Article

A Hybrid Discrete Grey Wolf Optimization Approach for Spectrum Assignment in Elastic Optical Networks

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

[1,2,3]Networks and Systems Laboratory, Badji Mokhtar Annaba University, Annaba, Algeria
Email: nassima.tsabet@univ-annaba.dz[1], malikababes@yahoo.fr[2], houssem.hadji@gmail.com[3]

Abstract: Future optical networks are expected to face increasing dynamism, heterogeneity, and unpredictability while supporting the growing demand for bandwidth-intensive services. Elastic Optical Networks (EONs) enable flexible and efficient spectrum utilization through fine-grained resource allocation. However, the Routing and Spectrum Assignment (RSA) problem, particularly spectrum allocation in large-scale topologies, remains computationally challenging because of its NP-hard nature and the rapid growth of the search space with network size. This motivates the development of scalable optimization techniques capable of maintaining efficient spectrum utilization in complex environments. To provide a demanding and representative evaluation, this work considers both realistic backbone networks and large-scale synthetic topologies with progressively increasing sizes and different connectivity characteristics for scalability assessment. A hybrid metaheuristic combining the Grey Wolf Optimizer (GWO) with Simulated Annealing (SA) is proposed for spectrum allocation in EONs. The hybridization aims to balance global exploration and local exploitation while mitigating premature convergence. The method is evaluated across benchmark topologies of varying sizes, with the Germany50 and Nobel-EU networks serving as the principal realistic benchmarks. Experimental results demonstrate that GWO–SA reduces the maximum number of frequency slots (FSs) required, improves spectrum utilization, and enhances network capacity, confirming its effectiveness and scalability for large-scale EON spectrum allocation problems.
Keywords: Elastic optical networks, gray wolf optimizer, Network Optimization, meta-heuristics, spectrum assignment (SA), frequency slots (FSs)

1. Introduction

With the prevalence of some high bandwidth demanding applications, Technologies such as cloud computing, the Internet of Things, social media applications, and traditional wavelength division multiplexing (WDM) networks are gradually proving to be inadequate for managing this rapid growth [1]. The 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 adapting data rates and spectrum to the demands of connection requests, flexible optical transmission aims optimize the use of fiber capacity.

However, the technology of EON presents the challenge of RSA, which is considered NP-hard according to [3] and [4]. RSA occurs by choosing the shortest path from the source to the destination, and then allocating the spectrum according to the connection request.This entails assigning a sufficient number of continuous, contiguous, and non-overlapping FSs while establishing an optical link between the source and destination nodes.

Three constraints must be satisfied in the elastic optical network to establish a lightpath for the connection request [5]: (1) To ensure spectrum consistency, the initial frequency slot index on each link of a path must be the same; (2) Spectrum continuity requires that a specific connection request be assigned to a series of contiguous

frequency slots. where a huge connection request cannot be split up into multiple smaller connection requests; (3) There should only be one connection request assigned to a frequency slot on a link at a time.

Concurrently resolving these two issues is a complex task. As a result, they are typically assumed independently by either fixing the routing first and then the spectrum assignment for this chosen route or by giving particular spectral ranges priority and then looking for a route that satisfies those requirements. The vast majority of the papers in the literature separate the assignment of spectrum from routing [6].

This paper is organized as follows: Section 2 presents the related work. Section 3 presents the foundational concepts of the grey wolf optimizer and simulated annealing. In section 4, we describe the SA problem and its mathematical modeling. We also introduce the proposed method in Section 5. The last two sections cover the results of numerical experiments, our conclusions, and a discussion of future work.

2. Related Work

A broad range of heuristic, metaheuristic, and evolutionary methods, together with swarm-intelligence-based approaches and, more recently, machine-learning techniques, has been extensively investigated in the literature.

Early computational-intelligence approaches mainly focused on evolutionary and neural techniques. Silva and Bastos-Filho [9] proposed a Hierarchical Hopfield Neural Network (HHNN) for inter-domain routing in communication networks. The proposed hierarchical architecture substantially reduced the number of neurons and connections required by conventional Hopfield models and improved convergence performance. However, this study focused exclusively on routing in conventional communication networks and did not consider the spectrum-contiguity and spectrum-continuity constraints associated with EONs.

Genetic Algorithms (GAs) have been widely investigated for optical-network resource allocation. Lechowicz and Walkowiak [10] proposed a GA for the RSA problem with multicast traffic in EONs, investigating different selection and crossover mechanisms to identify suitable solutions. Dinarte et al. [11] subsequently introduced the Hybrid Routing and Spectrum Assignment (HRSA) strategy. Rather than jointly optimizing routing and spectrum allocation, HRSA employs a GA to determine, for each source-destination pair, whether routing-first/spectrum-second (R-SA) or spectrum-first/routing-second (SA-R) ordering should be applied. Their results showed that selecting the RSA ordering according to network conditions can significantly reduce request blocking probability. Savva et al. [12] also investigated GA-based optimization in EONs in combination with modified simulated annealing to solve a network-coding-based RSA problem aimed at increasing physical-layer confidentiality while maintaining efficient spectrum utilization.

Other evolutionary approaches have addressed specific aspects of the RSA problem. Goścień [14] formulated resource allocation in survivable EONs carrying unicast, anycast, and multicast traffic and investigated Particle Swarm Optimization (PSO) and Tabu Search approaches. The problem incorporates dedicated path protection and spectrum-utilization objectives, making it substantially more complex than conventional unprotected RSA. The methods were evaluated on NSF15, Euro16, and Euro28 network instances. In the context of bio-inspired algorithms, Pérez-López et al. [17] investigated bee-inspired techniques for semi-dynamic RSA with variable-bandwidth requests. These studies demonstrate the potential of population-based optimization but also indicate that performance strongly depends on the problem representation, traffic assumptions, and optimization objectives.

Among swarm-intelligence algorithms, Grey Wolf Optimization (GWO) has attracted attention because of its relatively simple population structure and its balance between exploration and exploitation. Xuan et al. [5] proposed an Improved Grey Wolf Optimizer (IGWO) for the manycast RSA problem in EONs. Their model considers three objectives: network power consumption, total occupied spectrum, and the maximum index of the utilized frequency spectrum. These objectives are transformed into a single objective through a weighted-sum formulation. The IGWO incorporates mechanisms inspired by Particle Swarm Optimization and Differential Evolution to improve the distribution and search ability of the wolf population. Although the reported results demonstrate improvements over

the compared methods, the weighted aggregation of multiple objectives requires appropriate parameter selection and does not directly preserve the trade-offs among the individual objectives.

Mahala and Thangaraj [2] also applied GWO to dynamic RSA in EONs in the context of high-volume ecosystem-monitoring data transmission. Their approach aimed primarily at improving spectrum efficiency when dynamically serving incoming connection requests. The results demonstrated the suitability of GWO for optical resource allocation; however, the work addressed the complete RSA process within a specific application scenario rather than investigating the Spectrum Assignment (SA) subproblem independently.

More recent studies have increasingly focused on fragmentation, dynamic traffic, Quality of Transmission (QoT), and adaptive decision-making. Kumar et al. [15] proposed a traffic- and fragmentation-aware IWO-RSA method combining Invasive Weed Optimization for routing and spectrum allocation with a Spotted Hyena Optimizer for spectrum-fragmentation management. This hybrid strategy addresses blocking probability and spectrum utilization under dynamic and non-uniform traffic conditions.

Deep Reinforcement Learning (DRL) has also emerged as an alternative to conventional metaheuristics. Asiri and Wang [26] proposed DeepRMSA-QoT, a DRL framework for QoT-aware Routing, Modulation, and Spectrum Assignment. QoT information is incorporated into both the state representation and reward functions, enabling the agent to identify feasible lightpaths under physical-layer constraints. Although DRL provides adaptive decision-making under dynamic traffic, its effectiveness depends on the design of the state/action representation, reward function, training procedure, and network scenarios used during learning.

Recent fragmentation-aware methods provide another important research direction. Ujjwal et al. [27] proposed PFA-SM-RSA, which uses a two-stage spectrum allocation mechanism and a spectrum-block quality metric to proactively avoid fragmentation. The approach was evaluated on NSFNET, USNET, and COST-239. Jiang et al. [28] proposed TFF-RSA, which jointly considers frequency-domain and time-domain fragmentation as well as link-load information. Simulations on NSFNET and USNET demonstrated improvements in both request blocking and spectrum utilization.

A recent joint optimization strategy was proposed by Kiran et al. [29], who introduced an adaptive Ant Colony Optimization approach, A^3G, for online Routing, Modulation-Level, and Spectrum Assignment (RMLSA). In contrast to conventional sequential approaches, A^3G employs an augmented graph to represent EON constraints and performs routing and spectrum decisions jointly. More recently, Zheng et al. [30] proposed an integrated RSA framework combining Floyd-Warshall-assisted A* k-shortest-path routing with binary-encoded spectrum searching. This approach particularly targets computational scalability and fast identification of contiguous spectrum blocks in large EON instances.

These studies indicate that existing RSA research predominantly focuses on jointly optimizing routing and spectrum allocation, selecting an appropriate RSA ordering, reducing blocking probability, mitigating fragmentation, satisfying QoT constraints, or simultaneously optimizing several network-level objectives. In contrast, comparatively less attention has been devoted to investigating the optimization capability of GWO specifically for the Spectrum Assignment subproblem when routing decisions are already available.

Therefore, the present work focuses specifically on the SA stage and adapts the Grey Wolf Optimizer to minimize the overall spectrum usage in large optical-network instances. By isolating spectrum assignment from routing, the proposed approach enables a more direct investigation of the ability of GWO to explore the combinatorial spectrum-allocation space while simultaneously satisfying the spectrum continuity, contiguity, and non-overlapping constraints inherent to elastic optical networks (EONs). In addition, this formulation facilitates a systematic assessment of the trade-off between spectrum compactness, solution quality, and computational scalability when addressing large-scale network instances.

3. Methods

3.1 Grey wolf optimizer

The Grey Wolf Optimizer (GWO) is a population-based metaheuristic inspired by the social hierarchy and cooperative hunting behavior of grey wolves [22]. In the algorithm, candidate solutions are represented as wolves and organized according to four hierarchical levels: alpha, beta, delta, and omega [20]. The alpha wolf represents the best solution found so far, while the beta and delta wolves correspond to the second- and third-best solutions. The remaining candidate solutions are considered omega wolves and are guided by these three dominant individuals. The optimization process reproduces the main stages of grey-wolf hunting, namely searching for the prey, approaching and encircling it, and finally attacking it It was created for numerical optimization issues, It has recently been utilized by several researchers across a variety of domains [21]. During the search, the alpha, beta, and delta wolves are assumed to have the most reliable information about the location of the prey, which corresponds to the optimal solution in the optimization problem. Therefore, the other wolves progressively update their positions according to the positions of these three leaders. At the beginning of the optimization, the wolves are allowed to explore a relatively large region of the search space in order to identify promising solutions. As the iterations progress, their movements become increasingly restricted around the leading wolves, which gradually shifts the algorithm from exploration toward exploitation of the most promising region. At each iteration, the fitness of all wolves is evaluated, the three best solutions are updated, and the population continues to move according to these leaders until the stopping condition, generally the maximum number of iterations, is reached. The final alpha wolf is then considered the best solution produced by the algorithm. This mechanism allows GWO to maintain a balance between global exploration of the search space and local exploitation around high-quality solutions, while keeping the optimization procedure relatively simple and scalable [24]. It was created for numerical optimization issues, It has recently been utilized by several researchers across a variety of domains [21].

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

Algorithm 1. Grey Wolf Optimizer algorithm (GWO)

1. Initialization phase.
2. Initialize the parameters a, A and C.
3. Create an initial population of random solutions.
4. Evaluate the fitness value for every grey wolf.
5. Choose the best three grey wolves as X_α, X_β and X_Δ, respectively.
6. $t \leftarrow 0$.
7. **While** $t \leq t_{max}$ **do**:
8. Adjust the position of each grey wolf utilizing Eq. 12.
9. Evaluate the fitness score of each grey wolf.
10. Update a, A, and C.
11. Evaluate the fitness score of each grey wolf.
12. Update X_α, X_β, and X_Δ.
13. $t \leftarrow t + 1$.
14. **End while**.
15. Return X_α.

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

3.2 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 ΔE is given by:

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

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 ΔS decreases.

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

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** the temperature remains elevated **do**:
5. For L iterations, choose a neighboring solution S' of S by making a minor adjustment to the current solution.
6. Calculate $\Delta S = Cost(S') - Cost(S)$.
7. If $S' \leq S$, assign $S = S'$ (downhill move).
8. Otherwise, accept the uphill move with probability $P = e^{-\Delta S/T}$.
9. End if; end for.
10. Decrease the temperature.

11. **End while**.

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 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 is defined as minimizing the maximum utilization of FSs, as follows:

$$Minimiser\ max\ (FS)$$

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

The 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.:

$$| w_{v_i} | = v_i^w, \forall v_i \in V$$

- **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 :

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

- **The constraint on guardband**

To meet the guardband requirements between two vertices, v_i and v_j, the number of 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:

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

- **The constraint on order**

  The indices of the spectral intervals assigned to the frequency slots 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 3)

Table I. NOTATIONS

FS	Spectrum interval indices used, $FS \in N^+$
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

Algorithm 3. Evaluation procedure

1. **Inputs:** the graph $G(V, E)$; vertex weights v_i^w; edge weights $d_{v_i v_j}$; vertex sequence $O_i(v_{i_1}, v_{i_2}, \ldots, v_{i_n})$.
2. **Outputs:** assignment strategy Seq and the maximum number of frequency slots utilized.
3. $v_{i_1}^b \leftarrow 1$.
4. $v_{i_1}^a \leftarrow v_{i_1}^w$.
5. $Seq \leftarrow v_{i_1}^b$.
6. $j \leftarrow 2$.
7. **While** $j \leq |V|$:
8. $s1 \leftarrow max\{v_{i_k}^a + d_{v_{i_k} v_{j_k}} + 1\}$.
9. $s2 \leftarrow v_{i_{j-1}}^b$.

10. $v_{i_j}^b \leftarrow max\{s1, s2\}$.
11. $v_{i_j}^a \leftarrow v_{i_j}^b + v_{i_j}^w - 1$.
12. $Seq \leftarrow Seq \cup \{v_{i_j}^b\}$.
13. $j \leftarrow j + 1$.
14. **End while**.
15. **Return** Seq and $maxv_{i_j}^a$, $1 \leq j \leq |V|$

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. 5.

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

Hamming distance [31] 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 (ΔE) 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_α is returned as the outcome of the entire optimization procedure.

Algorithm 4 presents the pseudo-code of the discrete GWO–SA algorithm, whereas Fig. 1 illustrates the overall workflow of the proposed approch.

Algorithm 4. The discrete GWO-SA algorithm

3. Randomly initialize the population of wolves.
4. Initialize the temperature for SA, T.
5. **While** iteration $<$ max-iteration:
6. Calculate the fitness value of each wolf according to Algorithm 3.
7. Select the three wolves with the best fitness $(X_\alpha, X_\beta, X_\delta)$.
8. For each wolf in the population:
9. Calculate Hamming distance according to Eq. 5.
10. Generate a neighbor of the current wolf by swap mutation.
11. Evaluate the fitness of the new wolf.
12. Update position by applying simulated annealing.
13. If the new position is better, accept it; otherwise accept it with a probability based on temperature.
14. Decrease temperature over iterations.
15. End for.
16. Update the alpha, beta, and delta wolves after position updates.
17. Stop if the stopping criterion is met.
18. **End while**.
19. Return the best solution X_α.

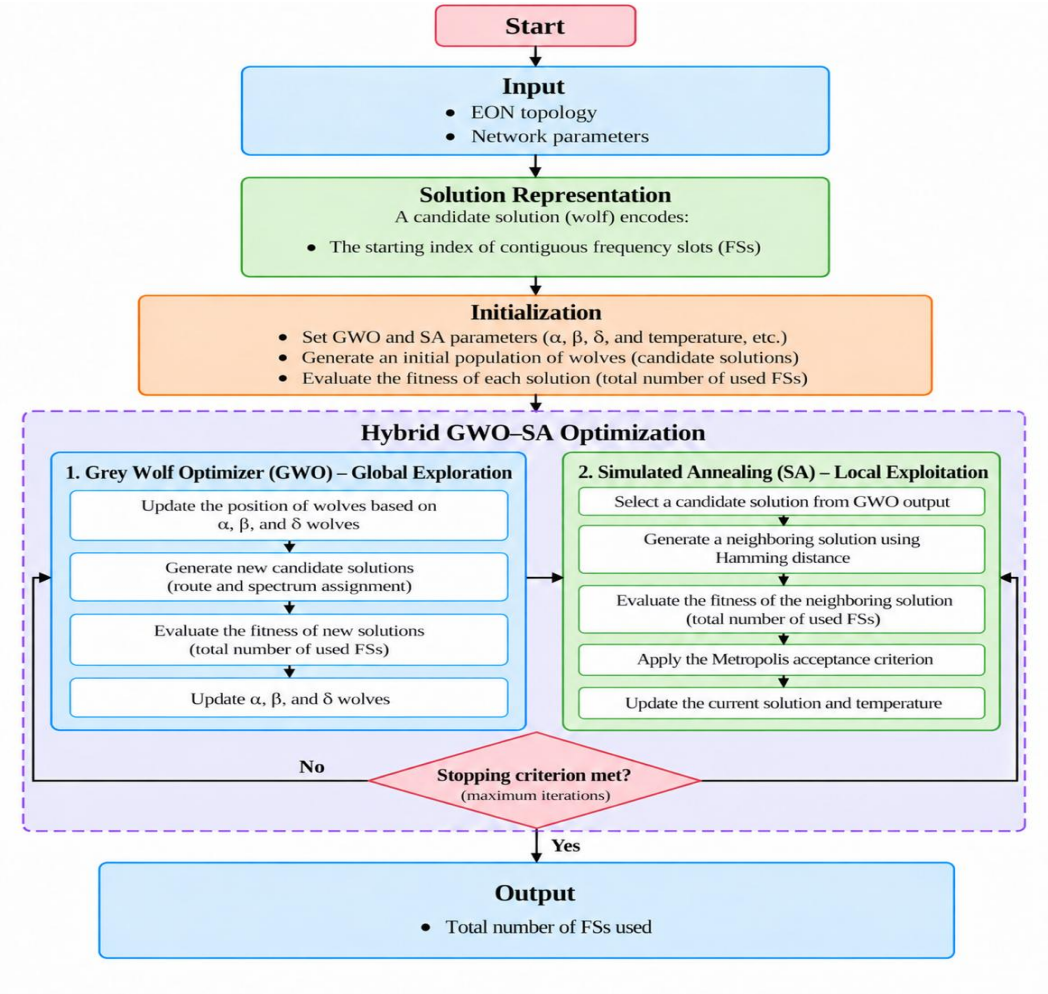


Fig. 1. Hybrid GWO–SA Optimization Flowchart

6. Experiments and Analysis

To substantiate the improvements introduced by the proposed algorithm, its performance was compared with that of the Two-Phase algorithm [25], the standard GWO, and the optimal solutions obtained using the CPLEX 12.9.0 solver through the Python API for larger problem instances. All methods were evaluated using the same dataset to ensure a fair comparison. Section 6.1 describes the simulation topologies, while Section 6.2 presents and analyzes the experimental results.

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.

6.1 The simulation topologies

Considering that the routes associated with the traffic demands were precomputed and fixed, the optimization process focused exclusively on spectrum allocation.

To provide a convincing large-scale evaluation of the proposed algorithm, the simulation study considers a diverse set of graph topologies, including realistic benchmark networks as well as several reproducible synthetic topologies of increasing size.

For the experimental evaluation, two realistic network topologies, namely Germany50 and Nobel-EU, were selected from the SNDlib repository. These topologies are commonly used in the literature to assess optimization and resource-management approaches in transport networks. The Germany50 topology consists of 50 nodes interconnected by 88 bidirectional links. It has an average node degree of 3.52 and a network diameter of nine hops. The Nobel-EU topology, on the other hand, consists of 28 nodes and 41 links.

We utilized TopoHub Python package for loading the Germany50 and Nobel-EU network topologies directly from SNDlib Library.

We adopt a progressive evaluation framework based on network topologies of increasing size, comprising 25, 50, 75, and 100 nodes. 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

Random graphs

Table II presents the average number of FSs required by the Two-Phase algorithm, standard GWO, the proposed SA-GWO algorithm, and the CPLEX solver for random graphs containing between 14 and 19 nodes. The results demonstrate that the proposed SA-GWO consistently outperforms both standard GWO and the Two-Phase algorithm across all graph configurations. Compared with standard GWO, SA-GWO reduces the average FS requirement range from 4.75 % to 5.72 %, with an overall reduction of approximately 5.21%. The largest absolute improvement is observed for the 19-node graph, where SA-GWO saves 6.54 frequency slots compared with standard GWO. Nevertheless, SA-GWO remains close to the CPLEX reference results.

From image 2, we observed that all four curves follow a broadly similar upward trend, showing that spectrum requirements increase with graph size. Most importantly, the SA-GWO curve remains below the curves of standard GWO and the Two-Phase algorithm over the entire range of graph sizes. This consistent ordering shows that the improvement achieved by SA-GWO is not limited to an isolated instance. Instead, the proposed method maintains its advantage as the problem size increases. The relatively stable separation between SA-GWO and standard GWO also indicates that the proposed modifications systematically enhance the search capability of the original GWO. Overall, the random-graph experiments provide evidence that the proposed SA-GWO offers an effective compromise between solution quality and applicability to increasing graph sizes.

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

Nodes	Two-phases	GWO	SA-GWO	Cplex	
14	72.7	75.2	71.63	71.6	
15	72.5	74.8	71.13	70.1	
16	75.5	78.3	74.1	73.6	
17	91.0	94	89.1	87.5	
18	99.4	101.4	96.23	94.5	

19	110.0	114.4	107.86	105	

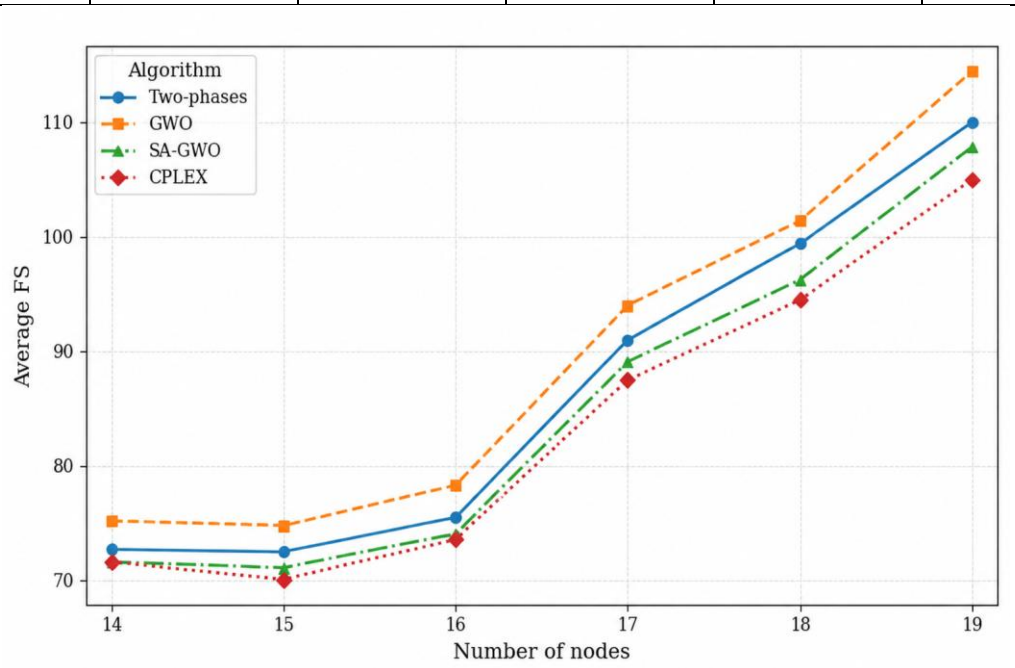


Fig. 2. The average FSs computed for various algorithms: Two-phases algorithm,the GWO standard, SA-GWO algorithm and CPLEX solver

Random complete graphs

Table III presents the average number of frequency slots obtained from six random complete-graph configurations with sizes ranging from 14 to 19 nodes. Complete graphs represent particularly restrictive instances because every pair of nodes is connected. The results show that the average number of frequency slots increases substantially for all the evaluated methods, confirming that the observed increase is primarily caused by the density and size of the complete graphs rather than by a particular optimization method.

Figure 3 visually confirms the numerical observations reported in Table III. All four curves increase as the number of nodes grows, reflecting the increasing difficulty of frequency assignment in complete graphs. Thus, SA-GWO provides the best overall performance among the heuristic algorithms and produces average solutions close to the CPLEX reference.

Table III. The average number of FSs calculated for different algorithms in complete graphs

Number of Nodes	Two-phases	GWO	SA-GWO	Cplex
14	143.6	152	142	142.4
15	163.6	169	164	160.5
16	196.6	201	193	191.1
17	213.3	215	212	207.6
18	231.3	230	229	223.7
19	241.5	244	236	231.8

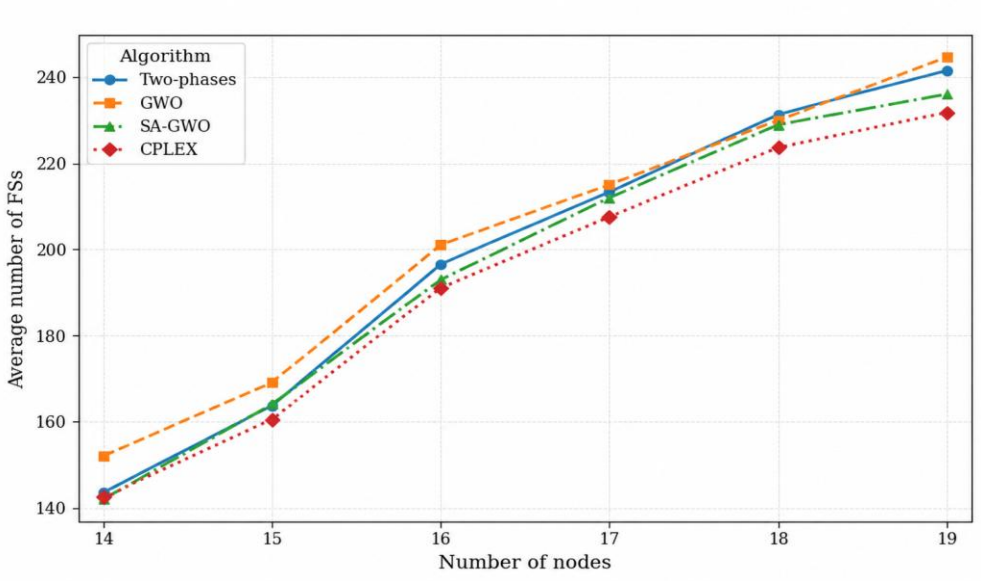


Fig. 3. The average FSs computed for various algorithms: Two-phases algorithm,the GWO standard, SA-GWO algorithm and CPLEX solver in complete graphs

Large-scale Graphs

We start with synthetic large-scale graphs, where we used complete graphs of increasing size, comprising 25, 50, 75, and 100 nodes.

Table IV. The average FSs computed for large-scale graphs

Number of Nodes	**25**	**50**	**75**	**100**
GWO	574.68	2602.3	5990.0	10957.23
SA-GWO	523.04	2314.5	5416.63	10110.03
CPLEX solver	450.7	2011.3	5029.88	9628.0

The results in Table IV and the grouped bar chart demonstrate a substantial increase in frequency-slot (FS) requirements as the network size grows, reflecting the higher allocation complexity and greater number of resource conflicts in large-scale topologies. For instance, the average FS requirement of SA-GWO increases from 523.04 for the 25-node network to 10,110.03 for the 100-node network. Despite this growth, SA-GWO consistently requires fewer FSs than the standard GWO across all evaluated network sizes, confirming the effectiveness of the proposed improvement. Moreover, the absolute reduction achieved by SA-GWO becomes more pronounced for larger networks, highlighting its scalability and increasing practical benefit as the problem size increases. At the same time, SA-GWO remains comparatively close to the CPLEX reference solutions, indicating that it can maintain competitive solution quality even for large and computationally demanding instances.

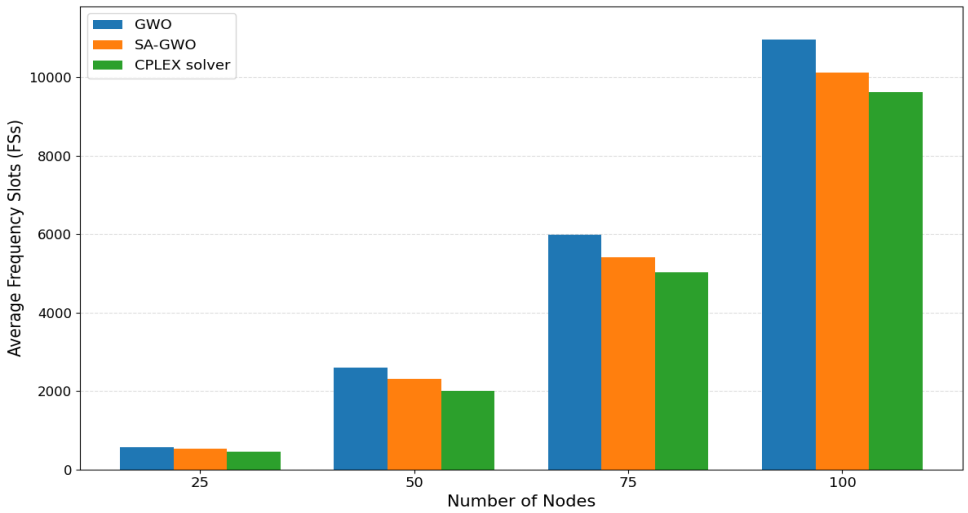


Fig. 4. Germany50 and Nobel-EU network topologies from the SNDlib dataset

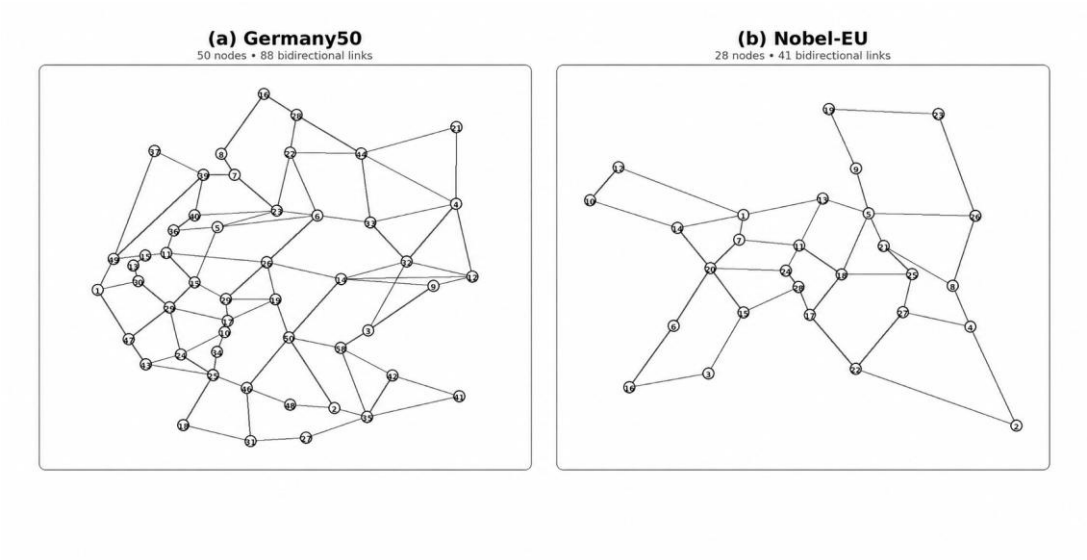


Fig. 5. Germany50 and Nobel-EU network topologies from the SNDlib dataset

Table V. The average FSs computed for Germany50 and Nobel-EU network topologies

topologies	**Nobel-EU**	**Germany50**
GWO	721.9	1829.18
SA-GWO	620.1	1513.23
CPLEX solver	606.3	1479.27

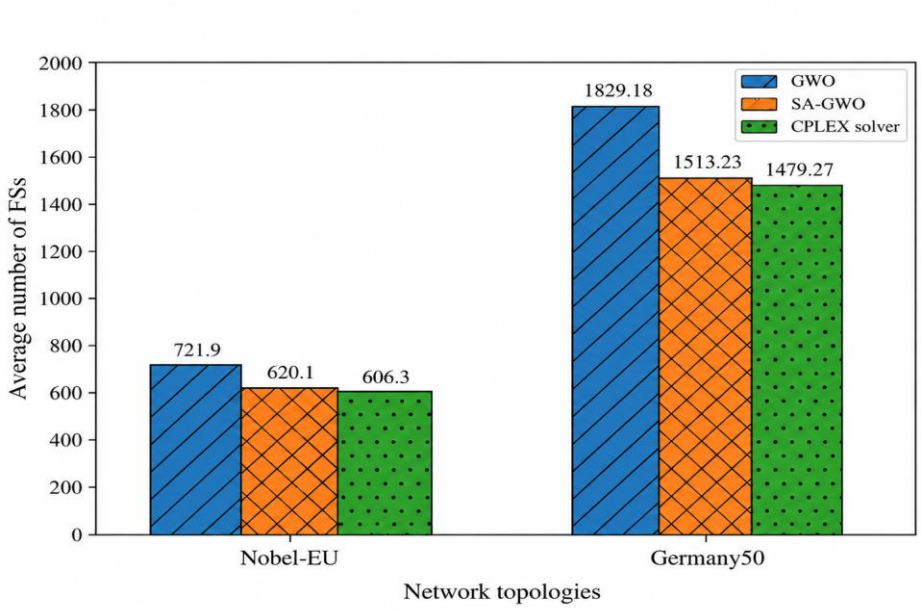


Fig. 6. Germany50 and Nobel-EU network topologies from the SNDlib dataset

Table V and the corresponding grouped bar chart compare the average number of FSs obtained by standard GWO, the proposed SA-GWO, and CPLEX for the realistic Nobel-EU and Germany50 network topologies. Considering both topologies together, SA-GWO reduces the aggregate number of required FSs by approximately 16.38% relative to standard GWO, while remaining only 2.29% above the aggregate CPLEX result. Therefore, the results consistent evidence that the proposed SA-GWO significantly improves the original GWO and produces near-CPLEX solutions across different realistic network topologies.

7. Conclusion

In this study, we proposed a hybrid GWO–SA optimization approach for the Spectrum Assignment problem in EONs. The method combines the global exploration capability of Grey Wolf Optimization with the local refinement mechanism of Simulated Annealing to mitigate premature convergence and improve frequency-slot allocation in large search spaces. Experimental evaluation on realistic backbone networks, including Germany50 and Nobel-EU, together with progressively larger synthetic topologies, showed that GWO–SA improves spectrum utilization and network capacity while maintaining effective performance as network size increases. These results demonstrate the suitability of the proposed method for computationally demanding spectrum-assignment problems and confirm the benefit of balancing exploration and exploitation in large-scale EONs. Future work will extend the approach to dynamic and traffic-adaptive RSA, considering fragmentation, blocking probability, and real-time demand variations. Further research will investigate multi-objective optimization involving energy consumption, latency, and quality-of-transmission constraints, as well as emerging multi-band and space-division-multiplexed EONs. Learning-assisted mechanisms will also be explored to support predictive, adaptive, and autonomous spectrum allocation.

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. & 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. & 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. & 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. & 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. & 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. & 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. S. Selva Kumar, S. Kalaivani, S. P. Siddique Ibrahim, and G. Swathi, "Traffic and fragmentation aware algorithm for routing and spectrum assignment in Elastic Optical Network (EON)," Optical Fiber Technology, vol. 81, Art. no. 103480, 2023, doi: 10.1016/j.yofte.2023.103480.
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. & 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. & 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. & 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.
26. A. Asiri and B. Wang, "Deep Reinforcement Learning for QoT-Aware Routing, Modulation, and Spectrum Assignment in Elastic Optical Networks," Journal of Lightwave Technology, vol. 43, no. 1, pp. 42–60, 2025, doi: 10.1109/JLT.2024.3446762.
27. U. Ujjwal, K. Surtani, and B. Padma, "A proactive, fragmentation-aware spectrum management algorithm for routing and spectrum assignment in elastic optical networks," Optical Fiber Technology, vol. 95, Art. no. 104445, 2025, doi: 10.1016/j.yofte.2025.104445.
28. . Jiang, M. Yu, and T. Shang, "Joint time-frequency fragmentation-aware routing and spectrum assignment algorithm for elastic optical network," Optical Fiber Technology, vol. 94, Art. no. 104358, 2025, doi: 10.1016/j.yofte.2025.104358.
29. M. Jyothi Kiran, V. Chebolu, G. Das, and R. Datta, "Online RMLSA in EONs With A³G: Adaptive ACO With Augmentation of Graph," Journal of Lightwave Technology, vol. 44, no. 1, pp. 180–193, 2026.
30. W. Zheng et al., "Accelerating RSA in elastic optical networks with Floyd-driven A* KSP routing and binary-encoded spectrum search," Optical Switching and Networking, vol. 61, Art. no. 100874, 2026, doi: 10.1016/j.osn.2026.100874.
31. R. W. Hamming, "Error detecting and error correcting codes," Bell Syst. Tech. J., vol. 29, no. 2, pp. 147–160, Apr. 1950.