Improved ILP Models and Heuristics for Solving Routing and Resource Allocation Problems in Optical Networks

Weichang Zheng, Ming Ke, Jinke Li, Mingcong Yang, Yu Zheng, Yongbing Zhang, Kun Yang *Fellow, IEEE*

***Abstract*—Integer linear programming (ILP) is an exact method for solving optimization problems; however, its application to large-scale scenarios often becomes impractical due to its exponentially increasing time complexity. To overcome this limitation, heuristic algorithms have been developed to efficiently identify near-optimal solutions. Both methods are widely used for routing and resource allocation in optical networks. Elastic Optical Networks (EONs) improve network capacity and spectral efficiency through granular spectrum division using frequency slots (FS), enabling dynamic spectrum allocation. Nevertheless, this flexibility introduces additional complexity to the Routing and Spectrum Assignment (RSA) problem. In this paper, we present improved ILP models for Routing and Wavelength Assignment (RWA) and RSA problems, highlighting how the properties of the model's coefficient matrix, such as total unimodularity and symmetry, influence performance Additionally, we propose a novel Two-Stage Graph Coloring-based Row Generation (TSGCRG) heuristic algorithm tailored for large-scale RWA and RSA issues. Our experiments conducted across networks and service request matrices of varying scales demonstrate that our proposed ILP models and heuristic algorithms yield superior solutions with reduced computation time compared to existing works. Furthermore, we discuss the impact of the number of candidate paths on the ILP path model and find that using 3 or 5 candidate paths is generally the most recommended choice. This integration of exact and heuristic methods offers practical and optimized solutions for managing optical networks across different problem scales.**

***Index Terms*—routing and resource allocation, RWA, RSA, ILP, heuristic algorithm, TSGCRG.**

I. Introduction

With the widespread adoption of bandwidth-intensive services such as cloud computing and video streaming, networks are under pressure to support higher data rates and more flexible traffic management. This has highlighted the need for more adaptable and efficient network architectures. Elastic Optical Networks (EONs), first introduced in [1], provide a promising solution to the challenges of bandwidth and spectral efficiency. By utilizing finer frequency spacing (12.5 GHz), EONs overcome the rigid channel spacing limitations inherent in traditional Wavelength Division Multiplexing (WDM)-based optical networks. The super-channels, which consist of multiple frequency slots (FSs), are used to accommodate varying traffic demands efficiently. This approach not only enhances network capacity but also improves spectral efficiency, effectively meeting the diverse bandwidth requirements of modern applications.

Integer Linear Programming (ILP) has been extensively applied to solve optimization problems in optical networks. As an exact optimization method, ILP finds the optimal solution by expressing both the objective function and constraints in linear forms [2]. A more efficient ILP model is both compact and capable of delivering better solutions within a shorter computation time [3]. The number of variables and constraints directly impacts the model's performance. Additionally, specific properties of the model's coefficient matrix, such as total unimodularity, symmetry, and sparsity, play a crucial role in obtaining solutions.

In routing and wavelength assignment (RWA) and routing and spectrum assignment (RSA) problems, ILP is employed to optimize routing paths and spectrum allocation for service requests. While ILP can provide exact solutions, its efficiency is limited by the non-deterministic polynomial-time hard (NP-hard) nature of these problems, especially for large-scale instances. Therefore, heuristic algorithms are often recommended for complex and large-scale scenarios, as they can quickly identify near-optimal solutions.

Although heuristic algorithms do not guarantee optimal solutions, they can achieve near-optimal results with less computational effort than ILP. Common heuristics include greedy techniques [4], local search [5], and meta-heuristics such as genetic algorithms [6] and simulated annealing [7]. These methods efficiently explore the solution space to find high-quality solutions.

In this paper, we introduce improved ILP models based on flow conservation, focusing on coefficient matrix properties to minimize wavelength or spectrum usage. Additionally, we develop a Two-Stage Graph Coloring-based Row Generation (TSGCRG) heuristic algorithm for solving large-scale RWA and RSA problems, providing a viable alternative to ILP models. In networks and service request matrices of varying scales, we evaluate our proposed ILP models and heuristic algorithms against existing works in terms of solution quality and computational efficiency.

The key contributions of this paper are summarized as follows: 1) We develop improved ILP models that utilize flow conservation and consider the impact of coefficient matrix properties, optimizing solutions effectively for small-scale scenarios. Numerical results demonstrate that our ILP models achieve superior compactness and efficiency compared to existing models. 2) Through a detailed analysis of the model coefficient matrices, we identify and prove that maintaining total unimodularity and symmetry elimination can enhance model quality. 3) We find that total unimodularity has a greater impact when the model is more solvable, while symmetry elimination

improves solving efficiency by reducing the feasible solution space and unnecessary searches. 4) We present a novel heuristic algorithm called TSGCRG, which integrates Tabu Search (TS) and Row Generation (RG) techniques, specifically designed for large-scale problems. Numerical results indicate that the TSGCRG algorithm outperforms other heuristic algorithms in both solution quality and computational efficiency. 5) Notably, TSGCRG can quickly find optimal RWA or RSA solutions for small-scale instances and offers a better trade-off between solution quality and computation time in large-scale scenarios compared to ILP models and existing heuristics. 6) Experiments across varying scales of networks and service request matrices show that our ILP models and TSGCRG algorithm demonstrate strong scalability. Additionally, using 3 or 5 candidate paths are recommended for the path-based ILP models.

The remainder of this paper is organized as follows. Section II reviews relevant literature on solving RWA and RSA problems. Section III presents our developed ILP model tailored for RWA problems, delving into model comparisons and highlighting the beneficial impacts of total unimodularity and the removal of asymmetry in modeling. In Section IV, we introduce the improved ILP model for RSA problems, accompanied by an in-depth comparison with existing ILP models in previous works. Section V outlines the process of our proposed TSGCRG algorithm and gives a simple example to demonstrate it. Section VI conducts simulations to assess the performance of our ILP models and the TSGCRG algorithm, providing a detailed analysis of the results. The paper concludes with Section VII, summarizing our findings and illustrating future research works.

II. Related works

Over the past decades, numerous research efforts have focused on addressing the RWA and RSA problems. In the field of RWA, Banerjee et al. [8] proposed a method to manage large optical networks by breaking the problem into smaller, more manageable sub-problems. Rahbar et al. [9] introduced a quality-of-transmission (QoT)-aware RWA technique for multifiber wavelength-routed all-optical networks. Azodolmolky et al. [10] presented an impairment-aware (IR) RWA algorithm that accounted for physical impairments and QoT estimators. Monoyios et al. [11] addressed physical impairments using two genetic algorithms to optimize RWA solutions. In a more recent study, Nevin et al. [12] applied reinforcement learning (RL) with invalid action masking to tackle the RWA problem in fixed-grid optical networks. Guan et al. [13] proposed a method for successively solving subproblems within the Lagrangian relaxation framework to efficiently approach the RWA problem. Manousakis et al. [14] introduced an energy-aware ILP model and heuristic algorithm that decomposes the problem to minimize energy consumption in RWA scenarios.

With the emergence of EON technology, research shifted toward the RSA problem. Aibin et al. [15] proposed a heuristic algorithm combining simulated annealing (SA) with simple greedy strategies to handle unicast and anycast traffic in EONs. Klinkowski et al. [16] developed an SA-based heuristic to enhance the branch-and-price algorithm for static RSA problems. In a subsequent study, Klinkowski et al. [17] introduced an improved branch-and-price algorithm for large-scale RSA problems defined by a mixed-integer program, incorporating problem relations and cuts to refine lower bounds while leveraging a heuristic to enhance upper bounds. Pederzolli et al. [18] presented a novel path-based metric and a heuristic RSA algorithm aimed at reducing spectral resource fragmentation. Wan et al. [19] proposed a nonlinear programming model to represent the complete RSA problem, along with a decomposition approach that divides the problem into three steps, offering two heuristic algorithms to solve it. Cai et al. [20] developed a node-arc ILP model and an efficient spectrum-window-based greedy heuristic algorithm for RSA problem-solving. Wang et al. [21] introduced two improved channel generation methods to reduce the number of variables in ILP models for RSA problems. Velasco et al. [22] proposed three methods, including mathematical programming, column generation, and metaheuristics for offline RSA problems, as well as two heuristic algorithms for online RSA scenarios.

III. The RWA problem in WDM-based optical networks

WDM enables the transmission of multiple signals on different wavelengths over a single optical fiber. According to the recommendations G.694.2 and G.671 by the Telecommunication Standardization Sector of the International Telecommunication Union (ITU-T) [23], [24], WDM is categorized into two types based on channel spacing: Coarse Wavelength Division Multiplexing (CWDM) and Dense Wavelength Division Multiplexing (DWDM). The main advantage of WDM is its ability to greatly expand the capacity of a single optical fiber.

A. The proposed ILP formulation

In this section, we introduce ILP models for the RWA problem in WDM-based optical networks. The network is modeled as an undirected graph $G = (V, E)$, where V represents the set of network nodes and E denotes the set of network links. Given a predefined service request matrix, the

TABLE I
Parameters used in the ILP model for the RWA problem.

Symbol	Meaning
V	Set of network nodes.
E	Set of network links.
R	Set of requests, where each request r is represent as $r = \{s_r, d_r\}$. s_r and d_r denote the source and destination nodes of r, respectively.
W	Set of wavelengths, where $w \in W$ is the wavelength index.
N_v^+	Set of outgoing links originating from node v.
N_v^-	Set of incoming links terminating at node v.
$N_{w,v}^+$	Set of outgoing links originating from node v in wavelength plane w.
$N_{w,v}^-$	Set of outgoing links terminating at node v in wavelength plane w.
h	A small value.

TABLE II
VARIABLES USED IN THE ILP MODEL FOR THE RWA PROBLEM.

Symbol	Meaning
$\alpha_r^{e,w}$	A binary variable that is equal to 1 if wavelength w in link e is allocated to request r; otherwise, it is 0.
β_w	An binary variable that is equal to 1 if wavelength w is used; otherwise, it is 0.
W_{max}	An integer variable that indicates the maximum number of required wavelengths for the network.

objective is to optimize the wavelength usage while assigning all requests. The parameters and variables used in ILP models are detailed in Tables I and II, respectively.

When solving the RWA problem, two key constraints must be considered: wavelength non-overlapping and wavelength continuity. The non-overlapping constraint ensures that a single wavelength cannot be allocated to more than one request simultaneously. The continuity constraint requires that the same wavelength is maintained across all links in a lightpath for any given request. Taking these constraints into account, we propose an enhanced ILP model for the RWA problem as follows:

ILP-Obj1:

$$\min \quad W_{max}, \tag{1}$$

subject to

$$\sum_{w\in W}\sum_{e\in N^-_{w,s_r}} \alpha_r^{e,w} - \sum_{w\in W}\sum_{e\in N^+_{w,s_r}} \alpha_r^{e,w} = -1,\ \forall r \in R, \tag{2}$$

$$\sum_{w\in W}\sum_{e\in N^-_{w,d_r}} \alpha_r^{e,w} - \sum_{w\in W}\sum_{e\in N^+_{w,d_r}} \alpha_r^{e,w} = 1,\ \forall r \in R, \tag{3}$$

$$\sum_{e\in N_v^-} \alpha_r^{e,w} - \sum_{e\in N_v^+} \alpha_r^{e,w} = 0, \quad \forall r \in R, \forall v \in V\backslash\{s_r, d_r\}, \forall w \in W, \tag{4}$$

$$\sum_{r\in R} \alpha_r^{e,w} \le \beta_w,\ \forall e \in E, \forall w \in W, \tag{5}$$

$$w\beta_w \le W_{max},\ \forall w \in W. \tag{6}$$

The objective in (1) aims to minimize the number of wavelengths required by the network. Constraints (2)∼(4) are designed to manage route selection and ensure wavelength contiguity for each request. The constraint (5) prevents the overlapping of the wavelength and the constraint (6) establishes the lower bound for the objective.

In our proposed ILP model, the objective is defined by the integer variable W_{max}, which is subject to constraint (6). This effectively reduces the model solution space and partially eliminates symmetry within it. In addition, constraints (2)∼(4) are based on flow conservation, where the coefficient matrix is totally unimodular. Next, we will provide a detailed explanation of how properties such as solution space size, total unimodularity, and symmetry elimination of the coefficient matrix impact the model.

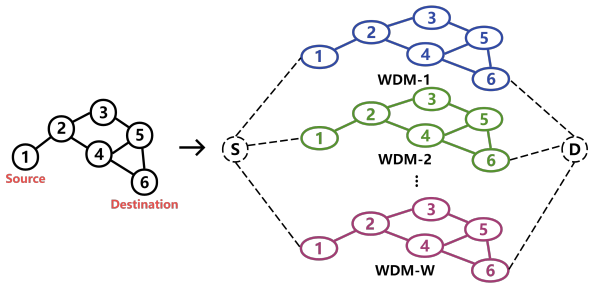


Fig. 1. The integrated network topology with W wavelength planes.

B. Analysis of total unimodularity in ILP models

To solve the RWA problem, our goal is to identify a routing path and assign an available wavelength for each request, ensuring that no wavelength is used by multiple requests simultaneously. As shown in Figure 1, we integrate all wavelength planes into an updated network topology. This integrated topology spans W wavelength planes, where identical nodes across different planes are merged into a single virtual node.

Constraints (2) and (3) ensure that only one wavelength plane is selected, and that there is exactly one incoming and outgoing link for each request at the source and destination nodes, respectively. These constraints also enforce wavelength continuity, requiring that the same wavelength be used on each link of a request's lightpath. Additionally, constraint (4) guarantees that the total net flow entering any intermediate node is zero on each wavelength plane.

Unlike the path-based and node-arc-based models commonly used in previous works, the routing constraints in our proposed model are based on flow conservation, which is recognized as an efficient modeling approach. This efficiency arises because the coefficient matrix of the routing constraints in our model is totally unimodular. As a result, the solution to the corresponding linear programming (LP) relaxation is guaranteed to be an integer, making the LP solution identical to the ILP solution. The detailed explanation is indicated as follows.

The total count of constraints (2), (3) and (4) is $|R|+|R|+|R|\cdot(|V|-2)\cdot|W|$. Additionally, each constraint utilizes variables $\alpha_r^{e,w}$, with a total of $|R|\cdot|E|\cdot|W|$ such variables. The vector of variables α is shown in (7).

$$\alpha = (\alpha_1^{e_1,w_1},...,\alpha_1^{e_1,w_{|W|}},...,\alpha_1^{e_{|E|},w_1},...,\alpha_1^{e_{|E|},w_{|W|}},..., \alpha_{|R|}^{e_1,w_1},...,\alpha_{|R|}^{e_1,w_{|W|}},...,\alpha_{|R|}^{e_{|E|},w_1},...,\alpha_{|R|}^{e_{|E|},w_{|W|}}). \tag{7}$$

By employing a coefficient matrix A of constraints (2)∼(4), listed in (8), alongside a vector of variables α, and a vector $b = (\overbrace{-1,...,-1}^{|R|}, \overbrace{1,...,1}^{|R|}, \overbrace{0,...,0}^{|R|\cdot(|V|-2)\cdot|W|})$, we can reformulate these three constraints into an ILP problem denoted as $H: A\cdot\alpha^T = b$.

***Theorem** 1:* If the optimal basis matrix B of problem H has a determinant $det(B) = \pm 1$, and B is a $m\times m$ sub-square matrix of (A, I), then the optimal solution α^* of Hwill be an integer solution.

***Proof** 1:* In theory, the optimal solution of problem H is given by $\alpha^* = \binom{B^{-1}b}{0}$. According to Cramer's Rule, $B^{-1} = \frac{B^*}{det(B)}$. In an ILP model, all elements of the adjoint matrix A^* are integers. Since $|det(B)| = 1$, it follows that all elements in B^{-1} are integers. Therefore, α^* is an integer solution because both B^{-1} and b consist solely of integer elements.

***Theorem** 2:* For any matrix $A_{m\times n}$, if it is totally unimodular and the elements of vector b are integers, then all vertices of the feasible region defined by the problem $H : A \cdot \alpha^T \leq b$ are integer points.

***Proof** 2:* When matrix A is totally unimodular, any basis matrix B derived from A also possesses this property, meaning the determinant of B is either 1 or -1. According to Cramer's Rule, which expresses solutions to linear equations in terms of determinants, every component of the vertex (basic feasible solution) is an integer, since both the numerator and denominator in the rule are integers. Therefore, all vertices of the polyhedron are integer points.

A =

$$
\left(\begin{array}{cccc|cc}
\overbrace{1\ \cdots\ 1}^{|W|\cdot|N^-_{s_1}|} & \overbrace{-1\ \cdots\ -1}^{|W|\cdot|N^+_{s_1}|} & \cdots & 0\ \cdots\ 0 & 0\ \cdots\ 0 \\
 & & \vdots & & \\
 & & & \overbrace{1\ \cdots\ 1}^{|W|\cdot|N^-_{s_{|R|}}|} & \overbrace{-1\ \cdots\ -1}^{|W|\cdot|N^+_{s_{|R|}}|} \\
0\ \cdots\ 0 & 0\ \cdots\ 0 & \cdots & & \\
\hline
\overbrace{1\ \cdots\ 1}^{|W|\cdot|N^-_{d_1}|} & \overbrace{-1\ \cdots\ -1}^{|W|\cdot|N^+_{d_1}|} & \cdots & 0\ \cdots\ 0 & 0\ \cdots\ 0 \\
 & & \vdots & & \\
 & & & \overbrace{1\ \cdots\ 1}^{|W|\cdot|N^-_{d_{|R|}}|} & \overbrace{-1\ \cdots\ -1}^{|W|\cdot|N^+_{d_{|R|}}|} \\
0\ \cdots\ 0 & 0\ \cdots\ 0 & \cdots & & \\
\hline
\overbrace{1\ \cdots\ 1}^{|N^-_{v_1}|_1} & \overbrace{-1\ \cdots\ -1}^{|N^+_{v_1}|_1} & \cdots & 0\ \cdots\ 0 & 0\ \cdots\ 0 \\
 & & \vdots & & \\
 & & & \overbrace{1\ \cdots\ 1}^{|N^-_{v_{|R|}}|_1} & \overbrace{-1\ \cdots\ -1}^{|N^+_{v_{|R|}}|_1} \\
0\ \cdots\ 0 & 0\ \cdots\ 0 & \cdots & & \\
\hline
\overbrace{1\ \cdots\ 1}^{|N^-_{v_1}|_{|W|}} & \overbrace{-1\ \cdots\ -1}^{|N^+_{v_1}|_{|W|}} & \cdots & 0\ \cdots\ 0 & 0\ \cdots\ 0 \\
 & & \vdots & & \\
 & & & \overbrace{1\ \cdots\ 1}^{|N^-_{v_{|R|}}|_{|W|}} & \overbrace{-1\ \cdots\ -1}^{|N^+_{v_{|R|}}|_{|W|}} \\
0\ \cdots\ 0 & 0\ \cdots\ 0 & \cdots & &
\end{array}\right) \tag{8}
$$

As shown in (8), the coefficient matrix A for route selection constraints, based on flow conservation, is deemed totally unimodular. This is because it satisfies the following conditions: 1) Every element in matrix A is 0, -1, or 1. 2) Each column of matrix A contains only one non-zero element. The first condition is apparent, while the second is satisfied because the source and destination nodes for each request are unique. For each request r, the non-zero elements in the first and second constraints appear in different columns, and those in the third constraint are separate from the columns for the source and destination nodes. Consequently, each column pertains to only one non-zero element linked to a specific request r.

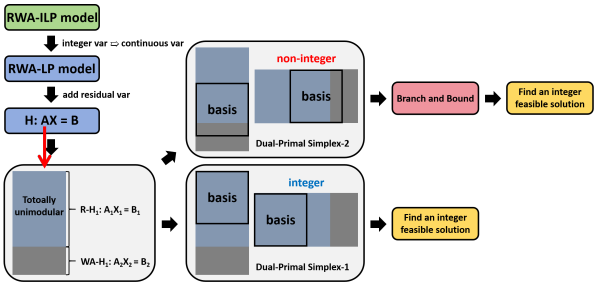


Fig. 2. The detailed solving process of the ILP model using the solver.

***Theorem** 3:* Let H be an integer programming problem, and let H' be the linear programming relaxation of H. If H' has a bounded optimal solution and every vertex of the feasible polyhedron of H' is an integer point, then the optimal solutions of H and H' are identical.

***Proof** 3:* Consider an integer programming problem H with the objective to minimize $c^T x$, subject to $Ax = b$, $x \geq 0$, and $x \in \mathbb{Z}^n$. Let H' be the linear programming relaxation of H, defined by the same objective and constraints, except that $x \in \mathbb{R}^n$. The feasible region of H' is a polyhedron P defined by $Ax = b$ and $x \geq 0$, and we assume it is bounded. By the Fundamental Theorem of Linear Programming, if H' has a bounded feasible region, then an optimal solution must occur at one of the vertices of the polyhedron P. Since it is given that every vertex of P is an integer point, any optimal solution x^* of H' must be an integer point. Since x^* is both feasible for H and optimal for H', it follows that x^* is also an optimal solution for H. Thus, the optimal solutions of H and H' are identical.

As illustrated in our RWA model **ILP-Obj1** in (2) to (6), constraints (2) to (4) is employed for routing (R) based on flow conservation, whose coefficient matrix is totally unimodular. Constraints (5) and (6) are used for wavelength assignment (WA), which breaks the total modularity of the whole matrix. The complexity of combinatorial optimization introduced by the WA problem results in the RWA problem being an NP-hard problem, and it is still challenging to find an efficient way to solve it.

The detailed process of solving the RWA problem as an ILP model using a solver is shown in Figure 2. When solving an ILP problem, the problem is typically relaxed into an LP problem by using continuous variables instead of corresponding integer variables. Then the LP problem is standardized as $H : AX = B$, whose coefficient matrix is composed of routing part $H_1 : A_1X_1 = B_1$ and wavelength assignment part $H_2 : A_2X_2 = B_2$. The dual primal simplex method is usually used for solving the LP problem. When solving this LP problem using the dual primal simplex method, each iteration involves selecting a basis that yields a corresponding feasible solution. This feasible solution acts as a vertex for the branch-and-bound process. If the selected basis is unimodular, which generally occurs in the flow-conservation routing part

of the model, the solution is typically an integer. In this case, the branch-and-bound process can be skipped since an integer solution that meets all the constraints has already been found. On the contrary, if the basis is not unimodular, the resulting solution is more likely to be fractional, which necessitates further branching to resolve these fractional components and eventually reach an integer solution. Consequently, leveraging the unimodular property of the flow-conservation routing model in the RWA problem formulation cannot fully eliminate the need for branch-and-bound but can effectively reduce the number of branching steps, thereby accelerating the overall problem-solving process.

In summary, for NP-hard problems like the RWA problem, utilizing an unimodular coefficient m atrix f or p art of the modeling, such as in the flow-conservation r outing, can significantly s peed u p t he s olution p rocess. B y r educing the number of branching steps required during branch-and-bound, this approach helps improve computational efficiency and shortens the time required to find o ptimal solutions.

C. Analysis of symmetry elimination in ILP models

ILP-Obj2:

$$\min \quad \sum_{w \in W} w\beta_w. \tag{9}$$
$$\text{subject to}$$
$$\text{constraints}(2) \sim (5).$$

ILP-Obj3:

$$\min \quad \sum_{w \in W} \beta_w. \tag{10}$$
$$\text{subject to}$$
$$\text{constraints}(2) \sim (5).$$

Our model tightens the model and eliminates symmetry by introducing constraint (6), which breaks the totally unimodular nature of the flow conservation model. However, this total unimodularity is preserved in models **ILP-Obj2** and **ILP-Obj3**. The total unimodularity ensures that the solution to the counterpart relaxed LP of the ILP is an integer, thereby simplifying computations and improving solution efficiency. Furthermore, objective (9) in model **Obj-2** removes symmetries compared to (10) in model **ILP-Obj3**, which reduces the search space and simplifies the solution process.

Symmetry in ILP models refers to the existence of multiple solutions within the solution space that yield the same objective value. This can negatively affect the model's solving efficiency by introducing redundant work during the optimization process. In our model **ILP-Obj1**, the introduction of constraint (6) breaks the overall total unimodularity but effectively eliminates symmetry. The elimination reduces the number of redundant branches (sub-problems) during the branch-and-bound process, enhancing search efficiency.

Additionally, models **ILP-Obj2** and **ILP-Obj3** maintain total unimodularity by modifying the objective function while keeping the optimization goal unchanged. By removing symmetry in the objective function, model **ILP-Obj2** further narrows the search space compared to model **ILP-Obj3**. This modification accelerates the convergence process, as it guides the solver more directly toward the optimal solution and minimizes unnecessary steps during the search for bounds.

For any given ILP model, consider a feasible domain **D** with two solutions, **x** and **y**. These solutions are defined as symmetric if there exists a symmetry operation σ such that σ**(x) = y**. Furthermore, we have a set of all such symmetry operations denoted as **O**, forming a group under composition.

***Theorem** 4:* A quotient set $\mathbf{D}\backslash\mathbf{O}$ as the set of equivalence classes **[x]**, where each class contains all solutions that can be obtained by applying elements of **O** to **x**. If the search space of the ILP model with symmetry reduction becomes $\mathbf{D'}=\mathbf{D}\backslash\mathbf{O}$, then $|\mathbf{D'}|\leq|\mathbf{D}|$.

Model-1:

$$\min \quad x_1 + x_2 + x_3, \tag{11}$$
$$\text{subject to}$$
$$x_1 + x_2 \leq 1, \tag{12}$$
$$x_2 + x_3 \leq 1, \tag{13}$$
$$x_1 + x_3 \leq 1, \tag{14}$$
$$x_1, x_2, x_3 \in N^+. \tag{15}$$

Model-2:

$$\min \quad x_1 + x_2 + x_3, \tag{16}$$
$$\text{subject to}$$
$$x_1 + x_2 + x_3 \leq 1, \tag{17}$$
$$x_1, x_2, x_3 \in N^+. \tag{18}$$

***Proof** 4:* Here, we illustrate the impact of symmetry reduction on solution space size using two simple ILP models. **Model-1** incorporates three symmetric constraints, making each variable interchangeable. This symmetry results in more combinations of x_1, x_2, and x_3 that satisfy the constraints in the relaxed LP model, thus expanding the solution space. For example, the solution $(0.5, 0.5, 0.5)$ is feasible but not effective, going against the objective of **Model-1**.

In contrast, **Model-2** only contains a single constraint, inherently reducing the solution space. It restricts the variables so that only one can be non-zero in any optimal solution, such as $(1, 0, 0)$. Solving the relaxed **Model-2** avoids the redundant searches that violate the optimal solution, unlike in **Model-1**. For example, the feasible solution $(0.5, 0.5, 0.5)$ acceptable in **Model-1** is invalid in **Model-2**. Consequently, the objective value for any feasible solution in relaxed **Model-2** will always be less than that of the optimal solution, which reduces the search overhead in the solution space.

To reduce the solution space in **Model-1**, symmetry can be broken by distinguishing the roles of the variables. For example, the objective (11) can be modified as follows:

$$\min \quad 2^0 x_1 + 2^1 x_2 + 2^2 x_3. \tag{19}$$

This change in the objective (19) introduces asymmetry among the variables x_1, x_2, and x_3, by establishing an order $x_1 \leq x_2 \leq x_3$. Consequently, it reduces the number of feasible solutions by eliminating symmetric alternatives.

TABLE III
PARAMETERS USED IN ILP MODELS FOR THE RSA PROBLEM.

Symbol	Meaning
V	Set of network nodes.
E	Set of network links.
R	Set of requests, where each request r is represent as $r = \{s_r, d_r, t_r\}$. s_r and d_r denote the source and destination nodes of r, respectively, and t_r is the traffic demand of r measured in number of FSs.
R_r	Set of requests excluding request r.
F	Set of FSs.
F_r	Set of the initial FS for request r, i.e., $F_r = \{f \in F \mid f \leq \|F\| - t_r + 1\}$.
$n_r^{p,e}$	A binary constant that is equal to 1 if link e is traversed by path p for request r; otherwise, it is 0.
$m_r^{c,f}$	A binary constant that is equal to 1 if channel c for request r contains FS f; otherwise, it is 0.
N_v^+	Set of outgoing links originating from node v.
N_v^-	Set of incoming links terminating at node v.
C_r	Set of candidate FS channels for request r.
P_r	Set of candidate paths for request r.
h	A small value.
M	A large value.

TABLE IV
VARIABLES USED IN ILP MODELS FOR THE RSA PROBLEM.

Symbol	Meaning
γ_r^e	A binary variable that is equal to 1 if link e is allocated to request r; otherwise, it is 0.
$\delta_{r_1}^{r_2}$	A binary variable that is equal to 1 if the initial FS of request r_1 is greater than that of request r_2; otherwise, it is 0.
ϵ_r	An integer variable that indicates the initial FS of request r.
ζ_r^v	A binary variable that is equal to 1 if node v in is allocated to request r; otherwise, it is 0.
$\eta_r^{e,c}$	A binary variable that is equal to 1 if FS channel c on link e is allocated to request r; otherwise, it is 0.
θ_r^c	A binary variable that is equal to 1 if FS channel c is allocated to request r; otherwise, it is 0.
λ_r^p	A binary variable that is equal to 1 if path p is allocated to request r; otherwise, it is 0.
$\mu_r^{p,c}$	A binary variable that is equal to 1 if FS channel c on path p is allocated to request r; otherwise, it is 0.
F_{max}	An integer variable that indicates the maximum number of required FSs for the network.

IV. EON-BASED OPTICAL NETWORKS AND RSA PROBLEM

As detailed in ITU-T G.694.1 [25], EONs utilize a finer granularity of FS for flexible spacing, unlike the fixed spacing in CWDM or DWDM. However, this flexibility adds complexity to the RSA problem in EONs due to the requirement for spectrum contiguity.

A. The proposed ILP formulation

To address the RSA problem, three key constraints must be considered: spectrum non-overlap, spectrum continuity, and spectrum contiguity. In this section, we present an ILP formulation **RSA-OurILP** for the RSA problem. The parameters and variables utilized in this ILP model are described in Tables III and IV.

RSA-OurILP:

$$\min \quad F_{max}, \tag{20}$$

subject to

$$\sum_{e\in N_{s_r}^-} \gamma_r^e - \sum_{e\in N_{s_r}^+} \gamma_r^e = -1, \ \forall r \in R, \tag{21}$$

$$\sum_{e\in N_{d_r}^-} \gamma_r^e - \sum_{e\in N_{d_r}^+} \gamma_r^e = 1, \ \forall r \in R, \tag{22}$$

$$\sum_{e\in N_v^-} \gamma_r^e - \sum_{e\in N_v^+} \gamma_r^e = 0, \ \forall r \in R, \forall v \in V\backslash\{s_r, d_r\}, \tag{23}$$

$$\delta_{r_1}^{r_2} + \delta_{r_2}^{r_1} = 1, \ \forall r_1, r_2 \in R, r_1 \neq r_2, \tag{24}$$

$$\epsilon_{r_1} + t_{r_1} - 1 - \epsilon_{r_2} \leq |F|(\delta_{r_1}^{r_2} + 2 - \gamma_{r_1}^e - \gamma_{r_2}^e) - 1, \quad \forall r_1, r_2 \in R, r_1 \neq r_2, \forall e \in E, \tag{25}$$

$$\sum_{r\in R} \gamma_r^e t_r \leq F_{max}, \ \forall e \in E, \tag{26}$$

$$\epsilon_r + t_r - 1 \leq F_{max}, \ \forall r \in R. \tag{27}$$

The objective (20) aims to minimize the highest index of required FSs for the network. Constraints (21) to (23) pertain to the route selection based on flow conservation similar to the RWA problem. The key distinction lies in the binary decision variables $\alpha_r^{e,w}$ and γ_r^e, where the initial FS selection is not incorporated into the decision variables due to the extensive size of the FS set (320 vs. 80). Constraints (24) and (25) ensure spectrum non-overlapping, while constraints (26) and (27) limit the maximum index of the required FSs F_{max}.

B. Analysis of ILP models

In this section, we analyze two node-arc-based models: one proposed in [20] (**RSA-NodeFS**) and another in [21] (**RSA-NodeChannel**). In addition, we review two path-based models of **RSA-PathFS** [26] and **RSA-PathChannel** [27].

RSA-NodeFS:

$$\min \quad F_{max}, \tag{28}$$

subject to

$$\sum_{e\in N_{s_r}^+} \gamma_r^e = 1, \ \forall r \in R, \tag{29}$$

$$\sum_{e\in N_{d_r}^-} \gamma_r^e = 1, \ \forall r \in R, \tag{30}$$

$$\sum_{e\in N_v^+} \gamma_r^e - \zeta_r^v = 0, \ \forall r \in R, \forall v \in V\backslash\{s_r, d_r\}, \tag{31}$$

$$\sum_{e\in N_v^-} \gamma_r^e - \zeta_r^v = 0, \ \forall r \in R, \forall v \in V\backslash\{s_r, d_r\}, \tag{32}$$

$$2\gamma_r^e - (\zeta_r^i + \zeta_r^j) \leq 0, \ \forall r \in R, \forall e = (i,j) \in E, \tag{33}$$

$$\delta_{r_1}^{r_2} + \delta_{r_2}^{r_1} = 1, \ \forall r_1, r_2 \in R, r_1 \neq r_2, \tag{34}$$

$$\epsilon_{r_1} + t_{r_1} - 1 - \epsilon_{r_2} \leq M(\delta_{r_1}^{r_2} + 2 - \gamma_{r_1}^e - \gamma_{r_2}^e) - 1, \quad \forall r_1, r_2 \in R, r_1 \neq r_2, \forall e \in E, \tag{35}$$

$$\epsilon_r + t_r - 1 \leq F_{max}, \ \forall r \in R. \tag{36}$$

In model **RSA-NodeFS**, the objective (28) seeks to minimize the highest FS index required. Constraints (29) to (33) handle route selection, including link and node allocations. Constraints (34) and (35) ensure spectrum non-overlapping, contiguity, and continuity. Constraint (36) guarantees that the objective F_{max} covers the final index of every request.

RSA-NodeChannel:

$$\min \quad F_{max}, \tag{37}$$

subject to

$$\sum_{e\in N_{s_r}^-} \eta_r^{e,c} - \sum_{e\in N_{s_r}^+} \eta_r^{e,c} = -\theta_r^c,\ \forall r \in R, \forall c \in C_r \tag{38}$$

$$\sum_{e\in N_{d_r}^-} \eta_r^{e,c} - \sum_{e\in N_{d_r}^+} \eta_r^{e,c} = \theta_r^c,\ \forall r \in R, \forall c \in C_r \tag{39}$$

$$\sum_{e\in N_v^-} \eta_r^{e,c} - \sum_{e\in N_v^+} \eta_r^{e,c} = 0, \quad \forall r \in R, \forall v \in V\backslash\{s_r, d_r\}, \forall c \in C_r, \tag{40}$$

$$\sum_{c\in C_r} \theta_r^c = 1,\ \forall r \in R, \tag{41}$$

$$\sum_{r\in R}\sum_{c\in C_r} \eta_r^{e,c} m_r^{c,f} \le 1,\ \forall e \in E, \forall f \in F, \tag{42}$$

$$\sum_{c\in C_r} \eta_r^{e,c} - 1 \le F_{max},\ \forall r \in R, \forall e \in E. \tag{43}$$

Model **RSA-NodeChannel**, derived from the node-link channel-assignment (NL-CA) formulations proposed in [22] and [27], uses constraints (38) to (40) for route selection. Constraint (41) ensures each request is allocated exactly one FS channel. Constraint (42) prevents spectrum overlap by permitting each FS on any given link to be used by only one request at a time. Finally, constraint (43) calculates the maximum index of required FSs (F_{max}).

RSA-PathFS:

$$\min \quad F_{max}, \tag{44}$$

subject to

$$\sum_{p\in P_r} \lambda_r^p = 1,\ \forall r \in R, \tag{45}$$

$$\delta_{r_1}^{r_2} + \delta_{r_2}^{r_1} = 1,\ \forall r_1, r_2 \in R, r_1 \ne r_2, \tag{46}$$

$$\epsilon_{r_1} + t_{r_1} - 1 - \epsilon_{r_2} \le M(\delta_{r_1}^{r_2} + 2 - \lambda_{r_1}^{p_1} - \lambda_{r_2}^{p_2}) - 1, \quad \forall r_1, r_2 \in R, r_1 \ne r_2, \forall p_1 \in P_{r_1}, \forall p_2 \in P_{r_2}, p_1 \cap p_2 \ne \varnothing, \tag{47}$$

$$\epsilon_r + t_r \lambda_r^p - 1 \le F_{max},\ \forall r \in R, \forall p \in P_r. \tag{48}$$

Models **RSA-PathFS** and **RSA-PathChannel** employ path slot-assignment (P-SA) and path channel-assignment (P-CA) formulations, respectively. In model **RSA-PathFS**, constraint (45) ensures that each request is assigned to a specific path. Constraints (46) and (47) prevent spectrum overlap for any two requests sharing the same link. Additionally, constraint (48) requires that the highest FS index must exceed the final FS index required by any request.

RSA-PathChannel:

$$\min \quad F_{max}, \tag{49}$$

subject to

$$\sum_{p\in P_r}\sum_{c\in C_r} \mu_r^{p,c} = 1,\ \forall r \in R, \tag{50}$$

$$\sum_{r\in R}\sum_{p\in P_r}\sum_{c\in C_r} \mu_r^{p,c} n_r^{p,e} m_r^{c,f} \le 1,\ \forall e \in E, \forall f \in F, \tag{51}$$

$$\sum_{p\in P_r}\sum_{c\in C_r} \mu_r^{p,c}(\theta_r^c + t_r) - 1 \le F_{max},\ \forall r \in R. \tag{52}$$

In model **RSA-PathChannel**, constraint (50) ensures that each request is allocated exactly one path and one FS channel. Constraint (51) guarantees spectrum non-overlapping, while constraint (52) details the computation of the maximum index of required FSs.

We begin by comparing the computational complexities of our proposed model with the other four models mentioned above. The number of variables and constraints of these models is illustrated in Table V.

TABLE V
NUMBER OF VARIABLES AND CONSTRAINTS IN RSA-ILP MODELS.

Model	Variables	Constraints
OurILP	$O(\|R\|\cdot\|E\|+\|R\|^2+2\cdot\|R\|)$	$O(\|R\|\cdot\|V\|+\|R\|^2+\|E\|\cdot\|R\|^2)$
NodeFS	$O(\|R\|\cdot\|E\|+\|R\|\cdot\|V\|+\|R\|^2+2\cdot\|R\|)$	$O(2\cdot\|R\|\cdot\|V\|+\|R\|\cdot\|E\|+\|R\|^2+\|R\|^2\cdot\|E\|+3\cdot\|R\|)$
NodeChannel	$O(\|R\|\cdot\|E\|\cdot\|C\|+\|R\|\cdot\|C\|)$	$O(\|R\|\cdot\|C\|\cdot\|V\|+\|R\|+\|R\|\cdot\|E\|+\|E\|\cdot\|F\|)$
PathFS	$O(\|R\|\cdot\|P\|+\|R\|^2+2\cdot\|R\|)$	$O(\|R\|+\|R\|^2+\|R\|^2\cdot\|P\|^2+\|R\|\cdot\|P\|)$
PathChannel	$O(\|R\|\cdot\|P\|\cdot\|C\|+\|R\|\cdot\|C\|)$	$2\|R\|+\|E\|\cdot\|F\|$

Due to the limited number of candidate paths per request, the value of $|P|$ is relatively small. Therefore, the path-based models (**RSA-PathFS** and **RSA-PathChannel**) involve fewer variables and constraints compared to the node-arc-based models (**RSA-NodeFS** and **RSA-NodeChannel**), as well as our proposed model (**RSA-OurILP**). Furthermore, the channel-based models (**RSA-NodeChannel**, and **RSA-PathChannel**) contain more variables but fewer constraints than their FS-based counterparts (**RSA-NodeFS** and **RSA-PathFS**), owing to the use of three-dimensional variables. In contrast, models **RSA-OurILP**, **RSA-NodeFS**, and **RSA-PathFS** separate routing and spectrum assignment, leading to a reduced number of variables and constraints.

V. THE PROPOSED TSGCRG ALGORITHM

A. Algorithm overview

Our proposed heuristic algorithm, called the Two-Stage Graph Coloring-based Row Generation (TSGCRG), initially solves the RWA and RSA problems focusing on load balancing without considering wavelength/spectrum non-overlapping constraints, as illustrated in **RWA-LB** and **RSA-LB**, respectively. Then, the wavelength/spectrum assignment problem is formulated as a graph coloring problem for further solving.

Once we obtain the routing results for each request from the **RWA-LB** or **RSA-LB** solutions, we represent each request as a node in a graph. Connections are established between nodes representing requests that share common links in their paths. The wavelength/spectrum assignment challenge of the RWA/RSA is represented as a graph coloring problem, where assigning different colors to adjacent nodes ensures that requests sharing links do not overlap in wavelengths/FSs. Therefore, minimizing the number of colors used is analogous to minimizing the wavelengths/FSs required for the requests.

RWA-LB:

$$\min \quad W_{max}, \tag{53}$$

subject to

flow conservation constraints$(21) \sim (23)$,

$$\sum_{r \in R} \gamma_r^e \leq W_{max}, \ \forall e \in E. \tag{54}$$

RSA-LB:

$$\min \quad F_{max}, \tag{55}$$

subject to

flow conservation constraints$(21) \sim (23)$,

$$\sum_{r \in R} \gamma_r^e t_r \leq F_{max}, \ \forall e \in E. \tag{56}$$

The graph coloring problem is initially solved using the Welch Powell (WP) algorithm, and the results are refined through iterative updates via a two-stage Tabu Search (TS) algorithm. For illustration, the workflow of our proposed algorithm, specifically for solving the RWA problem, is detailed in Algorithm 1.

In Algorithm 1, we initially determine the routing paths for requests and compute the initial maximum wavelength index, denoted as $Routes$ and W_{max}^{init}, by solving the **RWA-LB** model. This model aims to minimize the load balance on each link. Following this, as detailed from lines 5 to 11, we construct a graph **G-R** and apply the Welch-Powell (WP) algorithm to assign colors to this graph.

The WP algorithm is a heuristic method used for graph coloring, as detailed in Algorithm 2. It assigns colors to the vertices of a graph ensuring that adjacent vertices receive different colors, while minimizing the total number of colors used. This approach is particularly relevant to the RWA problem, where the wavelength non-overlapping constraint resembles a graph coloring challenge. Each RWA request represents a vertex in the graph, and wavelengths correspond to colors. Requests sharing common network links are adjacent in the graph, necessitating distinct colors (wavelengths) to avoid overlaps. Thus, effectively coloring the graph—minimizing the number of colors used—directly corresponds to reducing the number of wavelengths needed in the RWA problem.

Coloring the graph **G-R** generates a color map $ColorMap$, a wavelength assignment scheme $WAlloc$, and the maximal wavelength index used W_{max}^{wp}. $ColorMap$ indicates how colors (each representing a unique wavelength) are assigned to the requests within the graph **G-R**. $WAlloc$ specifies the particular wavelength allocated to each request according to $ColorMap$, and W_{max}^{wp} is the highest wavelength index used

Algorithm 1 The TSGCRG algorithm.

1: **procedure** $TSGCRG$(**R**,**W**) ▷ Input
2: Run **RWA-LB**
3: $Routes \leftarrow$ routing paths from **RWA-LB**
4: $W_{max}^{init} \leftarrow$ the initial solution for $W_{\max}$ from **RWA-LB**
5: Initialize a graph **G-R** with each request as a node
6: **for** each request r in **R do**
7: $p_1 \leftarrow$ path of request r_1 from $Routes$
8: **for** each request r_2 in **R do**
9: $p_2 \leftarrow$ path of request r_2 from $Routes$
10: **if** p_1 and p_2 share common links **then**
11: Add an edge between r_1 and r_2 in **G-R**
12: **end if**
13: **end for**
14: **end for**
15: Run **Algo.2** for graph coloring
16: $ColorMap \leftarrow$ coloring results from **Algo.2**
17: Assign wavelengths to colored requests
18: $WAlloc \leftarrow$ wavelength assignment results
19: $W_{max}^{wp} \leftarrow$ maximal wavelength index used
20: **if** $W_{max}^{wp} > W_{max}^{init}$ **then**
21: **if** $W_{max}^{init} < 3$ **then**
22: Find odd cycles in **G-R**
23: $OC \leftarrow$ set of the odd cycles that $|oc| > 3$
24: **while** $OC \neq \varnothing$ **do**
25: Cut off an edge of the node with the least degree in the largest cycle
26: Add a constraint to **RWA-LB**
27: Run the updated **RWA-LB**
28: **end while**
29: $Routes \leftarrow$ the updated routing paths
30: Run **Algo.2** for graph coloring
31: $ColorMap \leftarrow$ coloring results from **Algo.2**
32: Assign wavelengths to colored requests
33: $WAlloc \leftarrow$ wavelength assignment results
34: **else**
35: Run **Algo. 3** for iterative improvement
36: $ColorConf \leftarrow$ updated color conflicts
37: $ColorMap \leftarrow$ updated coloring results
38: $WAlloc \leftarrow$ updated wavelength assignment results
39: $W_{max}^{wp} \leftarrow$ updated maximal wavelength index used
40: **if** $ColorConf > 0$ and $W_{max}^{wp} > W_{max}^{init}$ **then**
41: Run **Algo. 4** for row generation
42: **else**
43: **return** $Routes$ and $WAlloc$
44: **end if**
45: **end if**
46: **else**
47: **return** $Routes$ and $WAlloc$
48: **end if**
49: **return** $Routes$ and $WAlloc$ ▷ Output
50: **end procedure**

Algorithm 2 The Welch-Powell (WP) algorithm.

procedure WP(G-R) ▷ Input
  Sort all nodes of **G-R** in descending order of their degrees
  Assign the first color to the first node
  for each uncolored node in the sorted list **do**
    if the node is not adjacent to any node with the current color **then**
      Assign the current color to this node
    end if
  end for
  Repeat with the next color until all nodes are colored
  ColorMap ← Final coloring results and sequences of requests per color
  return ColorMap ▷ Output
end procedure

in the coloring process, ensuring no wavelength overlaps.

Should W_{max}^{wp} exceed W_{max}^{init}, further optimization is required. Two main factors influencing graph coloring results are the largest odd cycle and the maximum clique. If W_{max}^{init} is less than 3, the largest odd cycle is dominant; in such cases, we remove all odd cycles greater than 3 from the graph. Otherwise, we proceed with an iterative improvement process using the Tabu Search (TS) algorithm, as described in Algorithm 3. This phase focuses on adjusting the wavelength assignment ($WAlloc$) and reducing the color conflicts ($ColorConf$), with the goal of minimizing the maximal wavelength index used.

The Tabu Search (TS) algorithm, outlined in Algorithm 3, is designed for optimizing wavelength assignments. It begins with the input graph **G-R**, the color map $ColorMap$, and the initial solution W_{max}^{init} derived from the first stage of Algorithm 1. Given a predefined Tabu list size T and an iteration count threshold Ω, the algorithm initializes an empty Tabu list H to store recently visited solutions and avoid repeated iterations. The iteration count $IterCount$ starts at 0.

In lines 5 to 8, colors (representing wavelengths) are reassigned for requests where the wavelengths exceed $W_{\mathbf{max}}^{\mathbf{init}}$. This recoloring process allows for color conflicts and aims to align the total color sum with the initial solution. This phase focuses on reducing color conflicts, ultimately yielding the minimum conflicts $ColorConf$ and the updated color map $ColorMap$ for the requests.

The algorithm continues with a while loop until the iteration count $IterCount$ reaches the threshold Ω. Each iteration generates neighborhood solutions by varying the colors in the current $ColorMap$. These solutions, along with their corresponding color conflicts CF, are added to a solution set $Neigh$ and sorted in ascending order according to the number of color conflicts.

For each solution in the sorted $Neigh$, the algorithm checks if it is not in the Tabu list H or if it reduces conflicts below $ColorConf$. If either condition is satisfied, tentative updates for $ColorMap^*$ and $W_{max}^{wp}{}^*$ are calculated. As outlined in lines 23 to 27, if a solution surpasses the current best, it is recognized as the new optimal and then updates the relevant parameters.

Algorithm 3 The Tabu Search (TS) algorithm.

1: **procedure TS(G-R, ColorMap, $W_{\mathbf{max}}^{\mathbf{init}}$, T, Ω)** ▷ Input
2: Initialize Tabu list H and iteration count $IterCount$
3: $H \leftarrow$ empty list
4: $IterCount \leftarrow 0$
5: **for** each request r in Set $\{r \in R | WAlloc_r > |W_{max}^{init}|\}$ **do**
6: Reassign colors to r within $|W_{max}^{init}|$ to minimize color conflicts
7: $ColorConf \leftarrow$ minimum color conflicts
8: $ColorMap \leftarrow$ updated coloring results for **ColorMap**
9: **end for**
10: **while** $IterCount \leq \Omega$ **do**
11: Generate neighborhood solutions by changing the color of each request in $ColorMap$
12: $Neigh \leftarrow$ set of solutions $Neigh = (NS, CF)$
13: $NS \leftarrow$ set of neighborhood searches
14: $CF \leftarrow$ set of color conflicts
15: Sort $Neigh$ by number of color conflicts in ascending order
16: **for** each solution (NS, CF) in sorted $Neigh$ **do**
17: **if** NS not in H or $CF < ColorConf$ **then**
18: Update $ColorMap$ with NS
19: $ColorMap^* \leftarrow$ tentative updated coloring results
20: $W_{max}^{wp}{}^* \leftarrow$ tentative updated maximal wavelength index used
21: **if** $|H| \geq$ **T then**
22: Remove the oldest entry from H
23: **end if**
24: Add NS to H
25: **if** $CF \leq ColorConf$ **then**
26: $ColorConf \leftarrow CF$
27: $ColorMap \leftarrow ColorMap^*$
28: $W_{max}^{wp} \leftarrow W_{max}^{wp}{}^*$
29: **break** ▷ Exit the current loop
30: **end if**
31: **end if**
32: **end for**
33: **if** $ColorConf = 0$ and $W_{max}^{wp} = W_{\mathbf{max}}^{\mathbf{init}}$ **then**
34: **break** ▷ Terminate if optimal solution found
35: **end if**
36: $IterCount \leftarrow IterCount + 1$
37: **end while**
38: **return** $ColorConf$ and $ColorMap$ ▷ Output
39: **end procedure**

If the updated solution results in zero color conflicts and the used maximal wavelength index matches the initial maximum, the algorithm terminates, signifying that an optimal solution has been found. If not, the iteration count is incremented. Should color conflicts remain in $ColorMap$ after the TS iterative improvements, the Row Generation (RG) algorithm, detailed in Algorithm 4, is employed for further optimization. If no conflicts are present, the process concludes, yielding

Algorithm 4 The Row Generation (RG) algorithm.

1: **procedure** RG(**G-R**, **Routes**, **ColorConf**, $W_{\mathbf{max}}^{\mathbf{wp}}$, $W_{\mathbf{max}}^{\mathbf{init}}$, **T**, Ω) ▷ Input
2: Initialize Tabu list H and iteration count $IterCount$
3: $H \leftarrow$ empty list
4: $IterCount \leftarrow 0$
5: **while IterCount** $\leq \Omega$ **do**
6: Find the maximum clique MC in **G-R**
7: Generate neighborhood solutions by cutting off |**ColorConf**| edges in MC
8: $Neigh \leftarrow$ set of solutions $Neigh = (NS, NN)$
9: $NS \leftarrow$ set of neighborhood searches
10: $NN \leftarrow$ set of node numbers for severed edges
11: Sort $Neigh$ by node numbers of severed edges in ascending order
12: **for** each solution (NS, NN) in sorted $Neigh$ **do**
13: **if** NS not in H **then**
14: Add constraints of avoiding using common links for the request(s) in NN to **RWA-LB**
15: Solve the updated **RWA-LB**
16: $Routes^* \leftarrow$ tentative updated routing paths
17: $W_{\mathbf{max}}^{\mathbf{init}\,*} \leftarrow$ tentative updated initial solution
18: **end if**
19: **if** $|H| \geq$ **T then**
20: Remove the oldest entry from H
21: **end if**
22: Add NS to H
23: Run **Algo.2** for graph coloring by $Routes^*$
24: $ColorMap^* \leftarrow$ Coloring results from **Algo.2**
25: Assign wavelengths to colored requests
26: $WAlloc^* \leftarrow$ wavelength assignment results
27: $W_{max}^{wp\,*} \leftarrow$ maximal wavelength index used
28: **if** $W_{max}^{wp\,*} = W_{\mathbf{max}}^{\mathbf{init}\,*}$ **then**
29: $Routes \leftarrow Routes^*$
30: $WAlloc \leftarrow WAlloc^*$
31: **return** $Routes$ and $WAlloc$
32: **else**
33: **if** $W_{max}^{wp\,*} \leq W_{\mathbf{max}}^{\mathbf{wp}}$ **then**
34: $Routes \leftarrow Routes^*$
35: **G-R** $\leftarrow$ update the graph
36: $W_{max}^{wp} \leftarrow W_{max}^{wp\,*}$
37: **break** ▷ Exit the current loop
38: **end if**
39: **end if**
40: **end for**
41: **if** $W_{max}^{wp} = W_{\mathbf{max}}^{\mathbf{init}}$ **then**
42: **break** ▷ Terminate if optimal solution found
43: **end if**
44: $IterCount \leftarrow IterCount + 1$
45: **end while**
46: **return** $Routes$ and $WAlloc$ ▷ Output
47: **end procedure**

optimal $Routes$ and $WAlloc$.

The RG algorithm simultaneously optimizes routing and wavelength assignments. It begins with an empty Tabu list and sets the iteration count to 0. The algorithm proceeds through the following steps within a while loop until the iteration threshold Ω is reached:

- Identifying the maximum clique MC in **G-R**, which represents the largest set of interconnected requests and directly affects the maximal wavelength index used.
- Generating neighborhood solutions by removing |**ColorConf**| edges from MC to reduce conflicts. These variations are then prioritized based on the node numbers of the severed edges.
- For each solution, constraints are added to **RWA-LB** to prevent the use of common links among selected requests. The routing path and initial solution are updated accordingly. The graph coloring algorithm then reassigns wavelengths based on these updates, resulting in a new color map $ColorMap^*$, wavelength allocation $WAlloc^*$, and maximal wavelength index $W_{max}^{wp\,*}$.
- If $W_{max}^{wp\,*}$ equals $W_{max}^{init\,*}$, the process terminates and outputs the optimal $Routes$ and $WAlloc$. Conversely, if $W_{max}^{wp\,*}$ is equal to or less than W_{max}^{wp}, the current routing paths $Routes$ and graph **G-R** are updated, and the loop cycles back to line 5 for the next iteration. The loop breaks when an optimal solution is achieved, with W_{max}^{wp} matching the initial maximal wavelength index W_{max}^{init}.

B. Algorithm time complexity analysis

In this section, we give a detailed time complexity analysis for our proposed TSGCRG algorithm, which is indicated as follows:

1 Initialization (Lines 1-4): The **RWA-LB** procedure initializes the routing paths and wavelength assignment. The complexity of this step is $O(R)$.

2 Constructing the Request Graph (Lines 5-13): This step involves comparing each pair of requests to check if their paths share common links. The double loop iterates over R^2 pairs, and for each pair, the path comparison is $O(1)$. Thus, the total complexity of this step is $O(R^2)$.

3 Welch-Powell Graph Coloring (Lines 14-18, Lines 30-33): The complexity of the Welch-Powell algorithm depends on the graph's structure. Sorting nodes based on their degree takes $O(R \log R)$, and coloring each node involves checking adjacency, which is $O(R^2)$. Therefore, the overall complexity of this step algorithm is $O(R^2)$.

4 Tabu Search for Conflict Resolution (Lines 35-39): The Tabu Search algorithm iteratively searches for better solutions in the neighborhood. For each request, reassigning colors involves checking conflicts, which is $O(R)$. Each iteration of the Tabu Search explores neighborhood solutions and sorts them, which adds $O(R^2 \log R)$ complexity. The number of iterations is bounded by Ω, resulting in an overall complexity of $O(\Omega \cdot R^2 \log R)$.

5 Row Generation (Lines 41): In the Row Generation algorithm, the maximum clique is identified, and new constraints are added to the RWA problem based on severed edges. Identifying the maximum clique and solving the updated RWA problem and coloring for each iteration

TABLE VI
THE NODE PAIRS AND ROUTING PATHS OF INPUT REQUEST MATRIX **R-1** AND **R-2**.

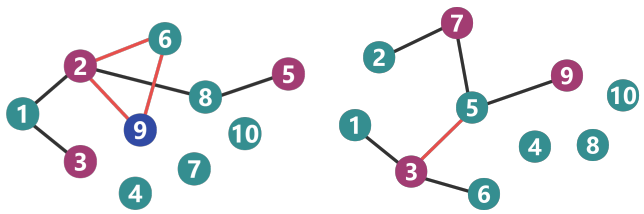
ID	R-1			R-2		
	SD Pair	Path	$W_{\max}^{\text{init}}$	SD Pair	Path	$W_{\max}^{\text{init}}$
1	(1, 7)	1→12→26→25→24→11→10→9→8→7	2	(1, 17)	1→12→26→25→24→11→10→9→8→23→22→21→20→19→18→17	2
2	(12, 5)	12→15→10→9→8→7→6→5		(1, 17)	1→27→28→2→3→4→17	
3	(27, 24)	27→1→12→26→25→24		(27, 8)	27→1→12→15→10→9→8	
4	(2, 27)	2→28→27		(2, 13)	2→13	
5	(3, 14)	3→14		(3, 11)	3→14→15→10→11	
6	(13, 16)	13→12→15→14→16		(13, 15)	13→12→15	
7	(28, 17)	28→2→3→4→17		(28, 14)	28→2→3→14	
8	(4, 11)	4→3→14→15→10→11		(4, 15)	4→5→21→22→23→8→9→10→15	
9	(14, 5)	14→16→6→5		(14, 15)	14→15	
10	(5, 7)	5→6→7		(5, 6)	5→6	

takes $O(R^2 \cdot E)$. Given the iteration count Ω, the overall complexity for RG is $O(\Omega \cdot (R^2 \cdot E))$.

Considering all components, the overall worst-case time complexity of the TSGCRG algorithm is dominated by the Row Generation algorithm, which has the highest complexity. Thus, the total complexity is approximately $O(\Omega \cdot (R^2 \cdot E))$.

C. Algorithm demonstration

In this part, we demonstrate how our proposed TSGCRG algorithm works through two simple examples of solving the RWA problem. We consider two sets of request matrices, **R-1** and **R-2**, each containing 10 requests with different source-destination (SD) pairs. Our experiments are conducted on a simulated European Backbone Network (EBN) featuring 28 nodes and 68 bidirectional links. The table VI presents the SD pairs along with their corresponding routing paths, as derived from the initial **RWA-LB** solution for the two request matrices.



(a) Coloring result of R-1. (b) Coloring result of R-2.

Fig. 3. The initial coloring results of input matrices **R-1** and **R-2**.

Based on the initial solutions obtained, we establish the color mappings for the request matrices **R-1** and **R-2**. These color mappings are depicted in Figure 3, showing the allocation results for both matrices. Each node represents a service request, connected nodes indicate that the two requests share a link, and the color of each node represents the corresponding wavelength. It is observed that the requests in **R-1** require three different wavelengths, while those in **R-2** need only two, in accordance with the wavelength non-overlapping constraint. This difference is attributed to the maximum cliques identified in the graphs for **R-1** and **R-2**, highlighted by red lines in Figure 3a and 3b, which are 2-6-9 and 3-5, respectively. This leads to different results in required wavelengths with the input of **R-1** and **R-2**. Consequently, the size of the maximum clique determines the least number of wavelengths needed: three for **R-1** due to a clique size of 3, and two for **R-2**, in line with its corresponding clique size.

The procedures described above align with lines 2 to 18 of Algorithm 1. Following this, we assess the initial solution, W_{max}^{init}, against the solution obtained by the WP algorithm, W_{max}^{wp}, for both **R-1** and **R-2**. For **R-1**, W_{max}^{init} is found to be less than $W_{max}^{wp}(2 < 3)$, necessitating further optimization to attain a routing and wavelength allocation that meets the initial solution criteria. Conversely, for **R-2**, the routing paths and wavelength assignments can be finalized directly, as W_{max}^{init} and W_{max}^{wp} are equal $(2 = 2)$.

Since the size of the largest odd cycle in **G-R** of **R-2** does not exceed 3, we proceed directly with applying the TS algorithm, as shown in Algorithm 3, to improve W_{max}^{wp} for **R-1**. During this process, we set the iteration count threshold, denoted by Ω, to 10, and limit the size of the Tabu list to 5. In each iteration, we maintain the color conflicts $ColorConf$ and the coloring results $ColorMap$ of **R-1** as a tentative label, updating them whenever a better solution is found. The final results for **R-1** are summarized in the first row of Table VII.

TABLE VII
THE OPTIMAL COLOR CONFLICT AND COLORING RESULT OF **R-1**.

Iter.	**ColorConf**	**ColorMap**
0	1	{1: 1, 2: 0, 3: 0, 4: 1, 5: 0, 6: 1, 7: 1, 8: 1, 9: 1, 10: 1}
1	0	{1: 1, 2: 0, 3: 0, 4: 1, 5: 0, 6: 1, 7: 1, 8: 1, 9: 0, 10: 1}

Algorithm 3 is employed to improve the coloring solutions obtained by the WP algorithm through iterations. This method is especially beneficial for large-scale problems where the WP algorithm might not achieve the optimal solution. However, it should be noted that for **R-1**, this method may not lead to improvements due to the small scale of the problem, with only 10 requests and a maximum clique size of 3 in its graph **G-R**, which inherently leads to color conflicts.

The RG algorithm, as demonstrated in Algorithm 4, is employed for final improvement based on the TS algorithm. In the graph **G-R** of **R-1**, the maximum clique is 2-6-9, resulting in a single color conflict given that W_{max}^{wp} is equal to W_{max}^{init}. Our objective is to reduce the size of the maximum clique by eliminating the link between request 2 and request

9. According to Table VI, the shared link between request 2 and request 9 is 6-5. We introduce a constraint to **RWA-LB** ensuring that prevents request 2 from traversing link 6-5, which is shown as follows:

$$\epsilon_2^{(6,5)} = 0. \tag{57}$$

We then run the updated **RWA-LB** and employ the WP algorithm to determine the coloring results for **R-1**. Compared the routing results of the updated **R-1** with the original, we note a single change: the routing path for request 2 has been adjusted to $12 \rightarrow 15 \rightarrow 10 \rightarrow 9 \rightarrow 8 \rightarrow 23 \rightarrow 22 \rightarrow 21 \rightarrow 5$. Therefore, the coloring results for **R-1** after the first iteration are presented in the second row of Table VII. The number of color conflicts has been reduced to 0, and an optimal solution is achieved after just one search iteration. At this point, We conclude the iterations of Algorithm 4, and the final results are compiled. Figure 4 illustrates the changes in color mappings for **R-1** across iterations, allowing for color conflicts.

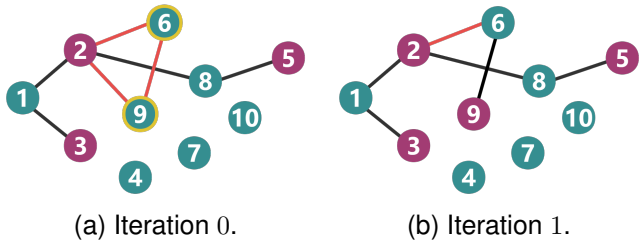


Fig. 4. The coloring results of **R-1** with iterations.

D. Statement of the TSGCRG algorithm for the RSA problem

Due to additional requirements for spectrum contiguity, the graph coloring challenge in the RSA problem is more complex than in the RWA problem. Our proposed TSGCRG algorithm for RSA largely aligns with solving the RWA problem, with the following distinctions:

(1) Each node in the graph represents a request and is weighted according to the number of FSs required by that request.
(2) Nodes are assigned a contiguous sequence of colors from a predefined RGB color table, which includes 320 colors, reflecting the node's weight in terms of the number of colors needed.
(3) The largest clique in the graph is defined as the one with the greatest total weight of its node, with the clique size determined by the sum of these weights.

VI. Numerical Results and Performance Analysis

In this section, we conduct simulations on a 28-node, 68-link European Backbone Network (EBN) topology, depicted in Figure 5, to assess the performance of ILP models and heuristic algorithms. We simulate service requests ranging from 10 to 100, increasing in increments of 10. The distribution of demand for these requests is standardized across different matrices with consistent mean values and variances.

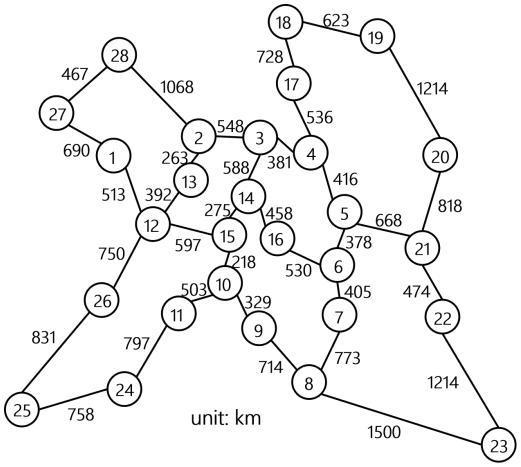


Fig. 5. The 28-node 68-link EBN topology.

The benchmark algorithms are: 1) the Shortest Path and First Fit (SPFF) algorithm; 2) the K Shortest Path and First Fit (KSPFF) algorithm; 3) Fragmentation-Aware Load-Balanced K Shortest Path Routing and First Fit (FL-KSPRFF) algorithm [28]; 4) Balanced Load Spectrum allocation (BLSA) algorithm [29]; 5) Decomposed algorithm [30]. Results are averaged over 50 simulation iterations. The simulations are executed on Microsoft Windows 10, powered by a 13th Gen Intel(R) Core(TM) i7-13700K CPU with 64 GB of memory. ILP models are solved using Gurobi Optimizer 10.0.3 on Python 3.7.

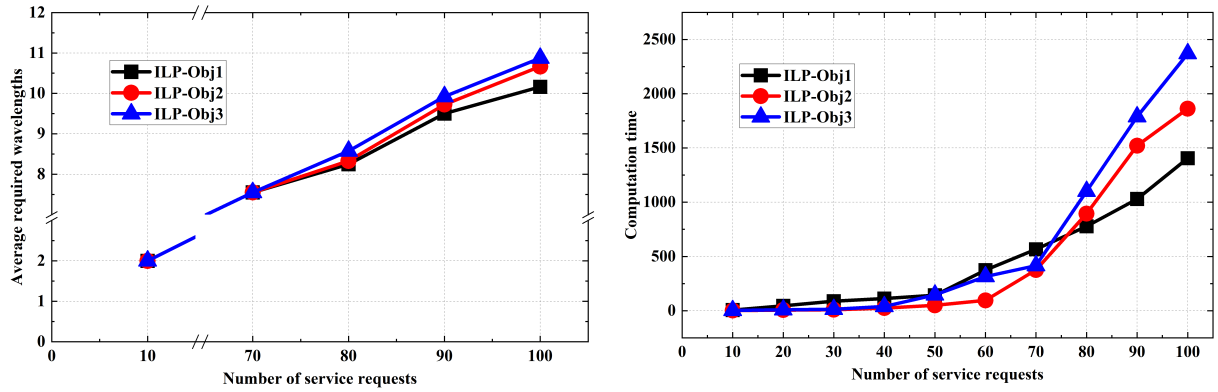
A. Results for the RWA problem

In this part, we first compare the ILP models for the RWA problem by examining their optimal solution values and computation times to highlight how matrix properties affect model performance. Figure 6 shows the average required wavelengths and computation time for different ILP models.

As shown in Figure 6(a), the average required wavelengths to satisfy service request matrices increase with their size. For a range of service requests from 10 to 70, the optimal solutions for all three ILP models are identical, as the small size of the request matrix allows solutions to be found within a reasonable time limit (3600s). As the matrix size grows, models **ILP-Obj2** and **ILP-Obj3** can not find the optimal solution within an hour in some instances, whereas our model consistently achieves it. Therefore, our proposed ILP model **ILP-Obj1** outperforms the other two models.

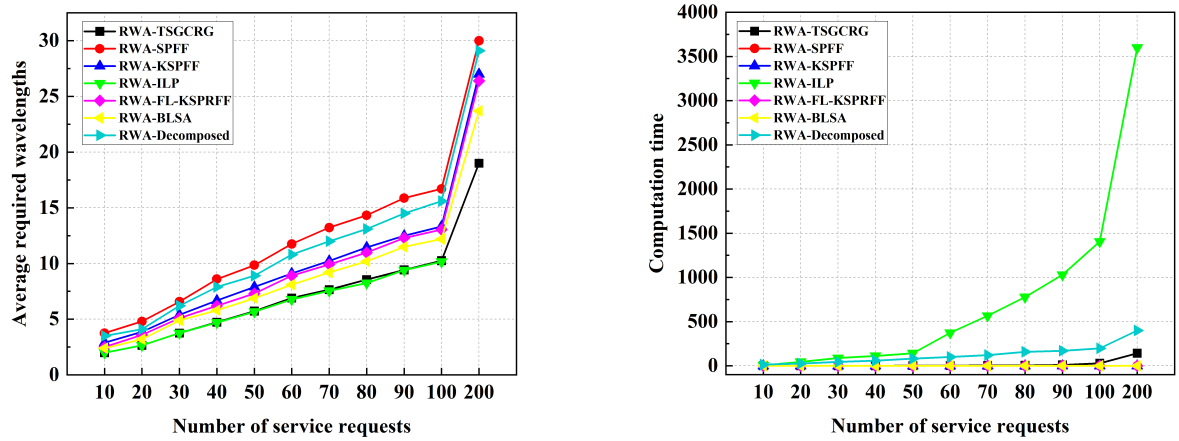
Model **ILP-Obj2** reduces symmetry in the objective function, leading to better optimal solutions compared to model **ILP-Obj3**, which demonstrates the positive impact of symmetry elimination on ILP models. This effectively reduces the search space during the solving process. Additionally, while model **ILP-Obj2** maintains the total unimodularity, it still has a symmetric constraint matrix, unlike model **ILP-Obj1**. The additional constraint in model **ILP-Obj1** further reduces redundant branches and tightens the solution space, yielding superior optimal solutions.

Figure 6(b) indicates how the computation time for the ILP models changes with the number of service requests. Due to



(a) Average required wavelengths (b) Computation time

Fig. 6. Comparison of numerical results for different ILP models with varying numbers of service requests: (a) Average required wavelengths; (b) Computation time.



(a) Average required wavelengths (b) Computation time

Fig. 7. Comparison of numerical results for different heuristic algorithms with varying numbers of service requests: (a) Average required wavelengths; (b) Computation time.

the efficiency gains from symmetry elimination, model **ILP-Obj2** consistently outperforms model **ILP-Obj3**, regardless of the request count. For small-scale request matrices (10–70), the ILP models are relatively easy to solve, making total unimodularity the more influential factor. Consequently, model **ILP-Obj2** requires less computation time than our model **ILP-Obj1**. As the number of service requests increases, solving ILP models becomes more challenging, reducing the impact of total unimodularity. In this case, the extra constraint in our model, which reduces redundant branches and tightens the feasible solution space, improves solving efficiency.

Figure 7 presents the comparison results of the average required wavelengths and computation time for different heuristic algorithms. To better illustrate the performance of the heuristics, we include the ILP model results as a reference. Regardless of the number of service requests, our TSGCRG algorithm consistently outperforms the other five algorithms, which suffer from their limited solution spaces. Specifically, when the number of service requests ranges from 10 to 100, the maximum gap between the solutions from our TSGCRG algorithm and the ILP model's optimal results is only 3%. Furthermore, when the request count reaches 200, the ILP model becomes difficult to solve within 3600 seconds due to its growing complexity. In this case, our proposed algorithm provides the best solution among all the methods.

While the ILP model can achieve optimal solutions, it requires significantly more computation time than the heuristic algorithms. The SPFF, KSPFF, FL-KSPFF, and BLSA algorithms have shorter computation times due to their limited candidate paths. The computation times for the Decomposed algorithm and our TSGCRG algorithm remain within an acceptable range, while the Decomposed algorithm takes slightly longer due to its iterative optimization steps. Overall, our TSGCRG algorithm demonstrates strong performance, with minimum deviation from the optimal solution, superior scalability as the problem size grows, and low computation time.

B. Results for the RSA problem

In this section, we evaluate the performance of our proposed ILP model and heuristic algorithm for the RSA problem by

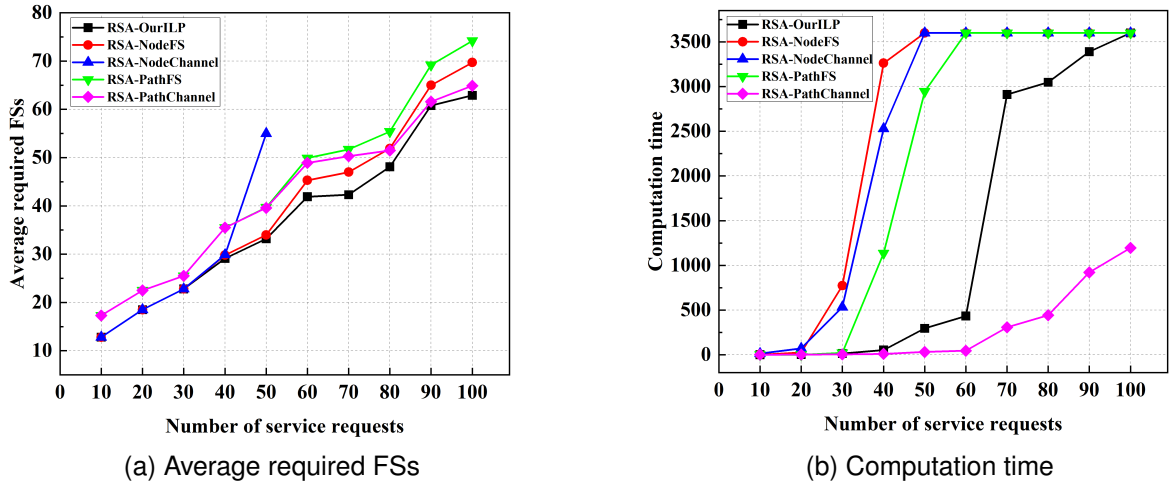


Fig. 8. Comparison of numerical results for different ILP models with varying numbers of service requests: (a) Average required FSs; (b) Computation time.

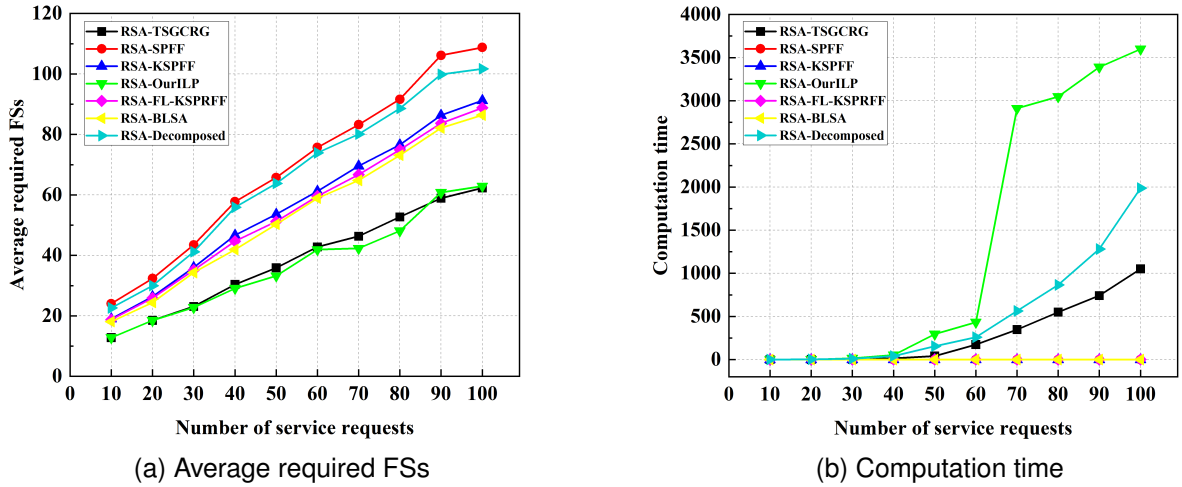


Fig. 9. Comparison of numerical results for different heuristic algorithms with varying numbers of service requests: (a) Average required FSs; (b) Computation time.

comparing them with the existing methods. Figure 8 shows the results for the average required FSs and computation time across different ILP models. As depicted in Figure 8(a), all ILP models show an increase in the average required FSs as the number of service requests grows. Due to its tighter solution space, fewer variables, and the benefits of flow conservation, our proposed ILP model consistently outperforms the other four models. Specifically, when the request matrix is relatively small, the path-based model performs worse than the node-arc-based and flow conservation-based models because of its limited feasible solution space. When the number of requests exceeds 50, model **RSA-NodeChannel** fails to reach a solution within 3600 seconds.

As the number of service requests increases, the efficiency of model **RSA-NodeChannel** declines significantly due to the sharp rise in the number of variables $\eta_r^{e,c}$. In contrast, this issue does not occur in model **RSA-PathChannel**, which also uses a three-dimensional variable $\mu_r^{p,c}$. Model **RSA-PathChannel** performs better in terms of average required FSs for a larger number of requests. This is because the number of candidate paths in path-based models ($p = 3$) is significantly lower than the number of candidate channels in channel-based models ($c = 310$), resulting in a much smaller scale for the variable $\mu_r^{p,c}$ compared to $\eta_r^{e,c}$. This difference enhances the solving efficiency of model **RSA-PathChannel**. Moreover, the computation time for our proposed ILP model is the shortest except for model **RSA-PathChannel**. Overall, our proposed ILP model demonstrates a better trade-off in both solving efficiency and solution quality compared to other existing ILP models.

Figure 9 compares the solution quality and solving speed of different heuristic algorithms. We use the ILP model results as a benchmark to more clearly illustrate the efficiency and effectiveness of these heuristics. As depicted in Figure 9(a), our proposed TSGCRG algorithm outperforms the other five heuristic algorithms across varying scales of service request matrices in the RSA problem. Notably, the gap between our TSGCRG result and the optimal solutions remains under 10%

TABLE VIII
PERFORMANCE EVALUATIONS FOR ILP MODELS **RSA-PATHFS** AND **RSA-PATHCHANNEL** WITH DIFFERENT NUMBERS OF CANDIDATE PATHS AND SERVICE REQUESTS IN EBNS.

Number of requests	Metrics	RSA-OurILP	RSA-PathFS					RSA-PathChannel				
			Path-1	Paths-3	Paths-5	Paths-7	Paths-9	Path-1	Paths-3	Paths-5	Paths-7	Paths-9
10	Req. FSs	14	30*	20*	20*	14*	14*	30*	20*	20*	14*	14*
	Time(unit: s)	0.12	0.01	0.09	0.45	0.66	3.35	0.17	0.49	1.03	3.18	2.77
20	Req. FSs	20	34*	24*	20*	20*	20*	34*	24*	20*	20*	20*
	Time	0.53	0.02	1.48	2.31	8.62	25.18	0.26	1.21	2.92	3.41	6.98
40	Req. FSs	31	54*	40	35	32	32	54*	40*	35*	32*	32*
	Time	58.49	0.33	1074.10	3600.00	3600.00	3600.00	1.00	7.29	12.67	17.53	20.65
80	Req. FSs	38	66	51	49	46	46	66*	51*	40*	40*	40*
	Time	1374.84	3600.00	3600.00	3600.00	3600.00	3600.00	17.35	35.25	674.74	1279.08	1649.98

and our method even surpasses the ILP solutions in scenarios with a high number of requests.

In Figure 9(b), although the SPFF, KSPFF, FL-KSPRFF, and BLSA algorithms have shorter computation times, they result in a higher average required FSs due to their limited solution spaces. As the number of service requests increases, the complexity of the ILP model grows exponentially, resulting in a significant rise in computation time. Furthermore, the Decomposed algorithm, despite achieving fewer average required FSs through its iterative optimization process, requires more computation time than the other four algorithms based on predefined candidate paths. Overall, our TSGCRG yields better performance than the existing heuristic algorithms.

Table VIII illustrates the impact of the number of candidate paths on the performance of the RSA-Path models. We evaluate the performance of models **RSA-PathFS** and **RSA-PathChannel** with 1, 3, 5, 7, or 9 candidate paths. Increasing the number of candidate paths expands the feasible solution space of the RSA-Path models, thereby enhancing the potential for optimizing the objectives. However, this increase also significantly improves the computational complexity of the models, which may cause a failure in obtaining the optimal solution within an acceptable time. Therefore, finding a suitable number of candidate paths is crucial for model performance. The results indicate that using 3 or 5 shortest paths strikes a beneficial balance between solution quality and computation time, making them the most recommended choices for achieving effective optimization.

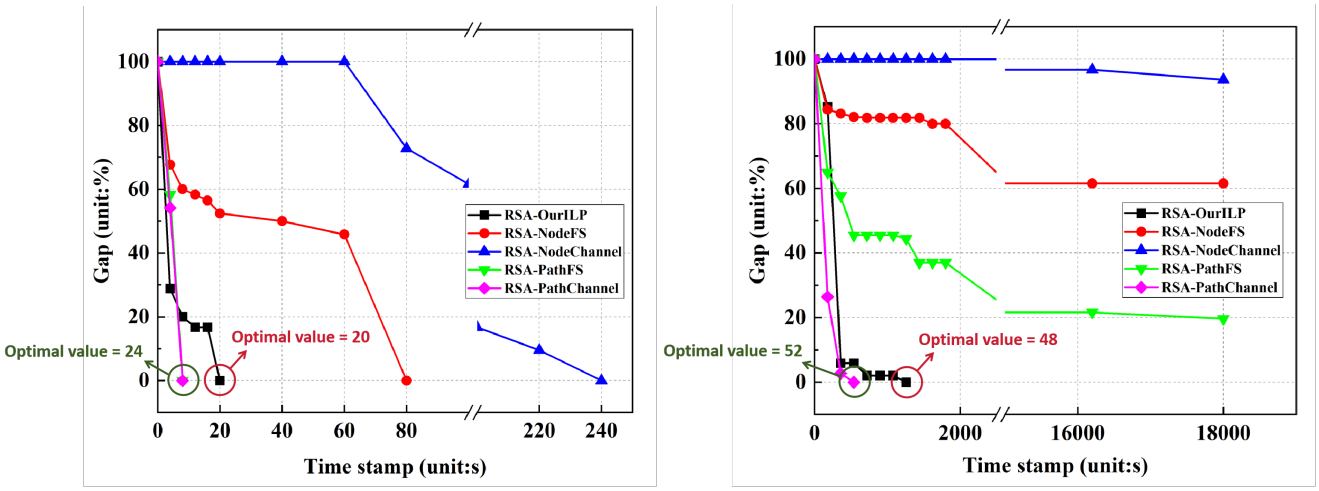
Table IX presents a comparative analysis of the performance of various ILP models for the RSA problem across different network scales. The scale of the network in this paper is defined by the number of nodes and their average degree. The results indicate that, regardless of the network size, path models tend to yield slightly inferior optimal solutions compared to other ILP models due to the limited feasible solution space. Notably, model **RSA-PathChannel** excels in solution time, but it requires a considerable amount of time for solvers to set up the model before computation begins. Interestingly, our findings reveal that ILP models perform more efficiently in networks with a higher average node degree when the number of nodes is constant. This is attributed to the solver's ability to more easily identify optimal path solutions with higher average node degrees, thereby enhancing convergence speed. Overall, our proposed ILP model showcases a commendable trade-off between solution quality and computation time across all network scales, underscoring its scalability.

TABLE IX
PERFORMANCE EVALUATIONS FOR ILP MODELS FOR THE RSA PROBLEM IN DIFFERENT SCALES OF NETWORKS.

Number	Degree	Metrics	RSA-ILP				
			OurILP	NodeFS	NodeChannel	PathFS	PathChannel
20	3	Req. FSs	24*	24*	24*	26*	26*
		Time(s)	49.40	579.14	467.62	279.22	5.08
	5	Req. FSs	10*	10*	10*	18*	18*
		Time	22.38	101.81	66.80	48.01	1.69
40	3	Req. FSs	18*	18*	18*	21*	20*
		Time	203.53	850.5	574.63	355.47	5.91
	5	Req. FSs	10*	10*	10*	18*	18*
		Time	42.29	200.76	87.20	66.52	1.98
80	3	Req. FSs	12*	12*	12*	20*	20*
		Time	302.4	1815.85	953.41	596.24	6.07
	5	Req. FSs	10*	10*	10*	18*	18*
		Time	64.34	154.86	111.52	88.65	2.31

TABLE X
PERFORMANCE EVALUATIONS FOR HEURISTIC ALGORITHMS FOR THE RSA PROBLEM IN DIFFERENT SCALES OF NETWORKS.

Number	Degree	Metrics	RSA-Heuristic					
			TSGCRG	SPFF	KSPFF	FL-KSPRFF	BLSA	Decomposed
20	3	Req. FSs	24	44	36	37	36	41
		Time(s)	9.38	0.001	0.01	0.03	0.01	14.21
	5	Req. FSs	10	41	23	26	23	38
		Time	11.84	0.002	0.02	0.04	0.02	23.64
40	3	Req. FSs	20	38	30	33	31	39
		Time	19.25	0.004	0.03	0.03	0.02	31.67
	5	Req. FSs	13	30	21	24	26	33
		Time	30.12	0.006	0.03	0.03	0.02	44.21
80	3	Req. FSs	15	35	26	29	30	35
		Time	35.67	0.01	0.10	0.09	0.06	55.79
	5	Req. FSs	14	33	23	24	27	33
		Time	41.29	0.02	0.10	0.10	0.11	73.21



(a) Small-scale problem (b) Large-scale problem

Fig. 10. Comparison of solution process track of different ILP models by the solver with different problem scales: (a) small-scale problem; (b) large-scale problem.

The performance evaluation of heuristic algorithms in different network scales is illustrated in Table X. Unlike the ILP models, the computational efficiency of heuristic algorithms tends to decrease as the average node degree increases. The SPFF, KSPFF, FL-KSPRFF, and BLSA algorithms exhibit remarkably short computation times due to their reliance on (K)SP routing. However, this efficiency comes at the expense of solution quality. In contrast, our proposed TSGCRG algorithm significantly outperforms these alternatives in terms of solution quality, achieving results closely aligned with the optimal solutions provided by ILP models. While the computation time for the TSGCRG is longer than that of the (K)SP-based heuristics due to its RG step, it remains shorter than that of the Decomposed algorithm and is well within acceptable limits. Thus, the TSGCRG demonstrates adaptability across various network scales, consistently delivering superior performance in the balance of solution quality and computation time.

Figure 10 presents the solution tracking for different ILP models under varying RSA problem sizes. As the problem size increases (with the number of requests rising from 30 to 80), the computation time for the RSA problem grows exponentially due to its NP-hard nature. As shown in Figure 10(a), for the small-scale problem, all five ILP models eventually converge to the optimal solution within different time stamps. Among them, the path-based models require the least computation time, closely followed by our proposed model with a slight difference. The node-arc-based models, however, take the longest to solve. Although the path-based models have the shortest solving time, their optimal solutions are not as competitive as the other three models due to their limited solution space.

Figure 10(b) illustrates the solution process for the large-scale problem. It is evident that only model **RSA-PathChannel** and our proposed model can achieve optimal solutions within the 5-hour time limit, while the other three models show different optimality gaps, ordered from smallest to largest as models **RSA-PathFS**, **RSA-NodeFS**, and **RSA-NodeChannel**. Additionally, our model outperforms model **RSA-PathChannel** in terms of the quality of the optimal solution. In summary, our proposed ILP model achieves superior performance in both solution time and quality, primarily attributed to the unimodularity of the flow-conservation-based constraints and the elimination of symmetry in the solution space.

VII. Conclusion

In this paper, we focus on developing ILP models and heuristic algorithms for RWA and RSA problems in optical networks. By examining properties of the model's coefficient matrix, such as total unimodularity and symmetry, we propose improved ILP models that adhere to flow conservation principles. Our findings demonstrate the advantages of maintaining total unimodularity and eliminating symmetry within ILP models, both of which enhance model performance. Notably, total unimodularity has a greater impact when the model is more solvable, while symmetry elimination reduces the feasible solution space and unnecessary searches. Additionally, we introduce the TSGCRG heuristic algorithm, which efficiently addresses RWA and RSA problems, particularly in large-scale scenarios. Compared to existing methods across different scales of networks and service request matrices, our TSGCRG exhibits superior solving efficiency and effectiveness, with feasible solutions that closely approximate the optimal solutions derived from ILP models. In our future work, we plan to explore a multi-objective optimization problem that aims to minimize both wavelength resources and network deployment costs in scenarios where interference is considered. Furthermore, we will investigate heuristic designs that integrate routing and spectrum assignment decisions. Our future research will also expand to encompass optical network planning, taking into account C+L bands and asymmetric traffic patterns.

ACKNOWLEDGMENTS

This work has been partially supported by the Municipal Government of Quzhou under Grant No.2023D028 and JSPS Grant-in-Aid for Scientific Research (C) 21K04544.

REFERENCES

[1] O. Gerstel, M. Jinno, A. Lord, and S. B. Yoo, "Elastic optical networking: A new dawn for the optical layer?" *IEEE communications Magazine*, vol. 50, no. 2, pp. s12–s20, 2012.

[2] J. E. Graver, "On the foundations of linear and integer linear programming i," *Mathematical Programming*, vol. 9, no. 1, pp. 207–226, 1975.

[3] D.-S. Chen, R. G. Batson, and Y. Dang, *Applied integer programming: modeling and solution*. John Wiley & Sons, 2011.

[4] J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Transactions on Information theory*, vol. 50, no. 10, pp. 2231–2242, 2004.

[5] D. S. Johnson, C. H. Papadimitriou, and M. Yannakakis, "How easy is local search?" *Journal of computer and system sciences*, vol. 37, no. 1, pp. 79–100, 1988.

[6] J. H. Holland, "Genetic algorithms," *Scientific american*, vol. 267, no. 1, pp. 66–73, 1992.

[7] D. Bertsimas and J. Tsitsiklis, "Simulated annealing," *Statistical science*, vol. 8, no. 1, pp. 10–15, 1993.

[8] D. Banerjee and B. Mukherjee, "A practical approach for routing and wavelength assignment in large wavelength-routed optical networks," *IEEE Journal on selected areas in communications*, vol. 14, no. 5, pp. 903–908, 1996.

[9] A. G. Rahbar, "Dynamic impairment-aware rwa in multifiber wavelength-routed all-optical networks supporting class-based traffic," *Journal of Optical Communications and Networking*, vol. 2, no. 11, pp. 915–927, 2010.

[10] S. Azodolmolky, Y. Pointurier, M. Angelou, D. Careglio, J. Solé-Pareta, and I. Tomkos, "A novel impairment aware rwa algorithm with consideration of qot estimation inaccuracy," *Journal of Optical Communications and Networking*, vol. 3, no. 4, pp. 290–299, 2011.

[11] D. Monoyios and K. Vlachos, "Multiobjective genetic algorithms for solving the impairment-aware routing and wavelength assignment problem," *Journal of optical communications and networking*, vol. 3, no. 1, pp. 40–47, 2011.

[12] J. W. Nevin, S. Nallaperuma, N. A. Shevchenko, Z. Shabka, G. Zervas, and S. J. Savory, "Techniques for applying reinforcement learning to routing and wavelength assignment problems in optical fiber communication networks," *Journal of Optical Communications and Networking*, vol. 14, no. 9, pp. 733–748, 2022.

[13] X. Guan, S. Guo, Q. Zhai, W. Gong, and C. Qiao, "A new method for solving routing and wavelength assignment problems in optical networks," *Journal of lightwave technology*, vol. 25, no. 8, pp. 1895–1909, 2007.

[14] K. Manousakis, A. Angeletou, and E. M. Varvarigos, "Energy efficient rwa strategies for wdm optical networks," *Journal of Optical Communications and Networking*, vol. 5, no. 4, pp. 338–348, 2013.

[15] M. Aibin and K. Walkowiak, "Simulated annealing algorithm for optimization of elastic optical networks with unicast and anycast traffic," in *2014 16th International Conference on Transparent Optical Networks (ICTON)*. IEEE, 2014, pp. 1–4.

[16] M. Klinkowski and K. Walkowiak, "A simulated annealing heuristic for a branch and price-based routing and spectrum allocation algorithm in elastic optical networks," in *Intelligent Data Engineering and Automated Learning–IDEAL 2015: 16th International Conference, Wroclaw, Poland, October 14-16, 2015, Proceedings 16*. Springer, 2015, pp. 290–299.

[17] M. Klinkowski, M. Żotkiewicz, K. Walkowiak, M. Pióro, M. Ruiz, and L. Velasco, "Solving large instances of the rsa problem in flexgrid elastic optical networks," *Journal of Optical Communications and Networking*, vol. 8, no. 5, pp. 320–330, 2016.

[18] F. Pederzolli, D. Siracusa, A. Zanardi, G. Galimberti, D. La Fauci, and G. Martinelli, "Path-based fragmentation metric and rsa algorithms for elastic optical networks," *Journal of Optical Communications and Networking*, vol. 11, no. 3, pp. 15–25, 2019.

[19] X. Wan, N. Hua, and X. Zheng, "Dynamic routing and spectrum assignment in spectrum-flexible transparent optical networks," *Journal of Optical Communications and Networking*, vol. 4, no. 8, pp. 603–613, 2012.

[20] A. Cai, G. Shen, L. Peng, and M. Zukerman, "Novel node-arc model and multiiteration heuristics for static routing and spectrum assignment in elastic optical networks," *Journal of Lightwave Technology*, vol. 31, no. 21, pp. 3402–3413, 2013.

[21] J. Wang, M. Shigeno, and Q. Wu, "Ilp models and improved methods for the problem of routing and spectrum allocation," *Optical Switching and Networking*, vol. 45, p. 100675, 2022.

[22] L. Velasco, A. Castro, M. Ruiz, and G. Junyent, "Solving routing and spectrum allocation related optimization problems: From off-line to in-operation flexgrid network planning," *Journal of Lightwave Technology*, vol. 32, no. 16, pp. 2780–2795, 2014.

[23] I.-T. S. Sector, "Spectral grids for wdm applications: Cwdm wavelength grid," *ITU-T Recommendation G. 694.2*, 2002.

[24] ——, "Transmission characteristics of optical components and subsystems," *ITU-T Recommendation G. 671*, 2012.

[25] I. Recommendation *et al.*, "Spectral grids for wdm applications: Dwdm frequency grid," *ITU-T G*, vol. 694, no. 694.1, 2006.

[26] K. Christodoulopoulos, I. Tomkos, and E. A. Varvarigos, "Elastic bandwidth allocation in flexible ofdm-based optical networks," *Journal of Lightwave Technology*, vol. 29, no. 9, pp. 1354–1366, 2011.

[27] K. Walkowiak, *Modeling and optimization of cloud-ready and content-oriented networks*. Springer, 2016, vol. 56.

[28] X. Chen, J. Li, P. Zhu, R. Tang, Z. Chen, and Y. He, "Fragmentation-aware routing and spectrum allocation scheme based on distribution of traffic bandwidth in elastic optical networks," *Journal of Optical Communications and Networking*, vol. 7, no. 11, pp. 1064–1074, 2015.

[29] Y. Wang, X. Cao, Q. Hu, and Y. Pan, "Towards elastic and fine-granular bandwidth allocation in spectrum-sliced optical networks," *Journal of Optical Communications and Networking*, vol. 4, no. 11, pp. 906–917, 2012.

[30] K. Christodoulopoulos, K. Manousakis, and E. Varvarigos, "Comparison of routing and wavelength assignment algorithms in wdm networks," in *IEEE GLOBECOM 2008-2008 IEEE Global Telecommunications Conference*. IEEE, 2008, pp. 1–6.

APPENDIX

In this section, we use two simple examples to illustrate how unimodularity can simplify the solution process of the ILP model using the branch-and-bound.

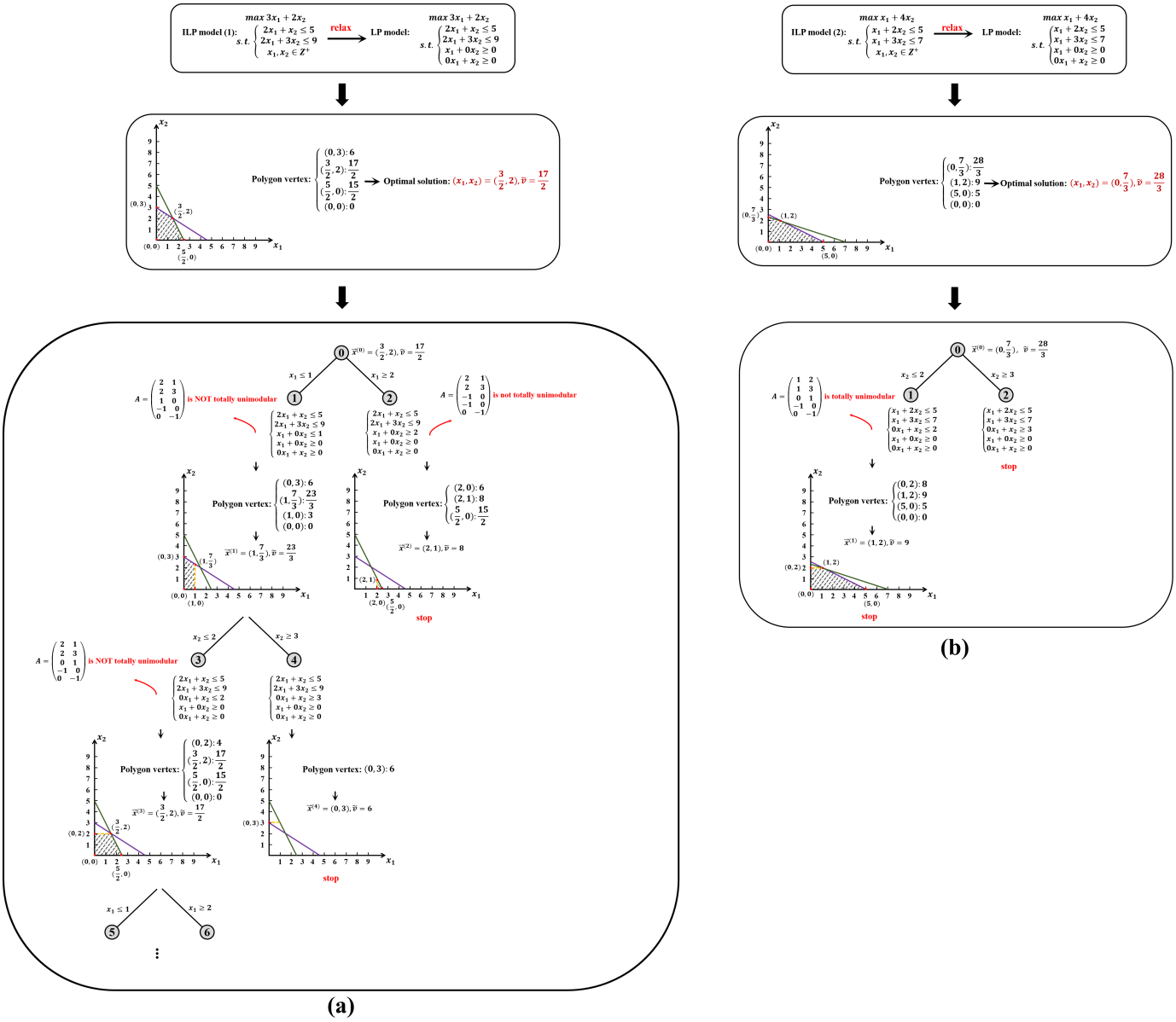


Fig. 11. The detailed process of branch and bound for solving two ILP models: (a) an ILP model whose coefficient matrix is not unimodular (b) an ILP model whose coefficient matrix is unimodular.

We argue that when an ILP model's constraint coefficient matrix contains totally unimodular sections, the need for additional branching during the solution process is reduced. This leads to fewer subproblems and increases the efficiency of the algorithm. Essentially, the more a model exhibits totally unimodular properties, the fewer branching steps are needed, which speeds up the solution process and improves its accuracy.

As shown in Figure 11, we demonstrate how the presence of totally unimodular portions in the constraint coefficient matrix affects the performance of the branch-and-bound algorithm. In Figure 11 (a), the ILP model's coefficient matrix lacks totally unimodular properties. In contrast, in Figure 11 (b), the coefficient matrix contains totally unimodular parts. These two different structures significantly impact the behavior of the branch-and-bound algorithm, which is commonly used to solve ILP models.

The branch-and-bound algorithm works by iteratively solving relaxed versions of the ILP, generating bounds, and dividing the problem into smaller subproblems (branching). As the algorithm progresses, branches that lead to suboptimal solutions are pruned. In Figure 11 (a), the relaxed ILP solution provides an optimal value of $(x_1, x_2) = (\frac{3}{2}, 2)$ with an objective value of $\frac{17}{2}$. To proceed, we need to branch on x_1 by setting $x_1 \leq 1$ and $x_1 \geq 2$.

Since the constraint coefficient matrix in ILP model (1) is not totally unimodular, the solution space contains non-integer vertices. If one of these vertices is an extreme point, further branching is required. As shown in Figure 11 (a), the branches for $x_1 \leq 1$ and $x_1 \geq 2$ lead to feasible solutions of $(x_1, x_2) = (1, \frac{7}{3})$ with an objective value of $\frac{23}{3}$, and $(x_1, x_2) = (2, 1)$ with an objective value of 8. The branch $x_1 \leq 1$ further branches on x_2, splitting into $x_2 \leq 2$ and $x_2 \geq 3$. However, since the solutions found are still non-integer, the process continues, with bounds remaining at $(8, \frac{17}{2})$. This loop continues as integer solutions cannot be obtained.

On the other hand, ILP model (2) shown in Figure 11 (b) has a totally unimodular constraint matrix, enabling the branch-and-bound algorithm to prune branches more effectively. After solving the relaxed ILP, we get an optimal solution of $(x_1, x_2) = (0, \frac{7}{3})$ with an objective value of $\frac{28}{3}$. We then branch on x_2, but since the matrix is totally unimodular, all vertices in the solution space are integer points. As a result, we quickly identify the optimal solution $(x_1, x_2) = (1, 2)$ with an objective value of 9, without further branching.

In summary, ILP models with unimodular submatrices require less computational time to reach the optimal solution. This advantage becomes even more pronounced in the branch-and-bound search process, especially for more complex models. When the vertices in the solution space are entirely integer (totally unimodular) or mostly integer (unimodular), the branch-and-bound algorithm can prune non-promising branches, explore fewer nodes, and in some cases, may eliminate the need for further pruning. This results in a faster and more efficient optimization process, which is particularly beneficial for large-scale ILP problems where computational resources and time are limited.