Survivable Traffic Grooming with Practical Constraints in Large-Scale Optical Network

Jianwei Niu
Theory Lab, 2012 Labs, Huawei Technologies, Co. Ltd
Shenzhen, Guangdong, China
niujianwei7@huawei.com

Junyan Liu
Theory Lab, 2012 Labs, Huawei Technologies, Co. Ltd
Sha Tin, Hong Kong SAR, China
liu.junyan@huawei.com

Fan Zhang
Theory Lab, 2012 Labs, Huawei Technologies, Co. Ltd
Sha Tin, Hong Kong SAR, China
zhang.fan2@huawei.com

Fabo Sun
Optical Product Research Department, Huawei Technologies, Co. Ltd
Dongguan, Guangdong, China
sunfabo@huawei.com

Kerong Yan
Optical Product Research Department, Huawei Technologies, Co. Ltd
Dongguan, Guangdong, China
yankerong@huawei.com

Junqi Ma
Optical Product Research Department, Huawei Technologies, Co. Ltd
Dongguan, Guangdong, China
mark.majunqi@huawei.com

Jie Sun
Theory Lab, 2012 Labs, Huawei Technologies, Co. Ltd
Sha Tin, Hong Kong SAR, China
j.sun@huawei.com

Tieyong Zeng
Department of Mathematics, The Chinese University of Hong Kong
Sha Tin, Hong Kong SAR, China
zeng@math.cuhk.edu.hk

ABSTRACT

This paper investigates the problem of the survivable traffic grooming, routing, and wavelength assignment (GRWA) subject to the no-loop and optical channel (OCh) length constraints. To the best of our knowledge, this complex problem has not been previously studied in the literature. We propose a novel length-constraint and no-loop Dijkstra algorithm, which helps to compute the primary and backup paths for traffic demands based on the augmented-layer graph (ALG). We conduct numerical experiments on both small and large-scale networks and demonstrate the superior efficiency of our approach compared to the results obtained from the integer linear programming formulation, which is solved by the SCIP solver.

KEYWORDS

Survivable Traffic Grooming, No-loop Constraint, OCh Length Constraint, Augmented-Layer Graph, LCNL-Dijkstra Algorithm, Tabu Search

1 INTRODUCTION

The wavelength division multiplexing (WDM) [13],[15] in an optical network is a widely used technology that multiplexes a number of traffic demands onto a single fiber by using different wavelengths. To further improve the bandwidth utilization of the WDM network, the traffic grooming technique has been proposed. The traffic grooming technique packs different traffic demands onto a wavelength using optical-electrical-optical (OEO) converters. Therefore, it can significantly enhance the fiber capacity and pave the way for the development of optical networks. In the WDM network with the traffic grooming, an optical channel (OCh) is a lightpath, which consists of a collection of consecutive fibers that share the same wavelength channel. An OCh originates and terminates at stations with OEO converters.

In an optical network, the failure of a single link can disrupt numerous OChs and connections. Therefore, the survivability in the optical network, which assigns not only the primary path but also the backup path for traffic demands, has been developed as referenced in [13], where the authors have introduced a comprehensive set of protection schemes, including the dedicated link protection, the shared link protection, the dedicated path protection, and the shared path protection. For the high utility of resources, we focus on the shared path-based protection scheme.

In the context of the industry, unnecessary resource wastage, transmission delay, and signal attenuation are intolerable to enterprise customers. Hence, we must consider two important constraints, including the no-loop constraint and the Och length constraint. To further explain, the no-loop constraint also called the simple path constraint in [10] is that the physical path for each demand does not contain any cycles to avoid unnecessary transmission delays. Moreover, we need to consider the length constraint because of the signal attenuation that occurs within an OCh, and the total attenuation or loss is and nonlinearly related to the distance of fibers composing the OCh [7]. To model the attenuation more conveniently, we could set a threshold for the length of an och to represent an accepted attenuation degree.

Due to the high cost of OEO converters and the fact that each OCh requires two converters, our objective is to minimize the number of OChs needed to satisfy a given set of traffic demands while adhering to the shared path protection scheme under the no-loop constraint and the OCh length constraint. We denote the above problem as the survivable multi-constraint traffic grooming (SMTG) problem. Numerous studies have demonstrated a strong interest in the area of the survivable traffic grooming. Authors of [1, 6, 12] have proposed different survivable schemes and objectives with no constraints. In the work of Naser et al. [9], a delay-constrained and survivable scheme is proposed, but it cannot apply to our problem as it focuses on minimizing delay time and resource utilization.

In order to solve the SMTG problem, an integer programming problem (ILP) is formulated. However, obtaining the exact solution is challenging since the scale of variables and constraints in the problem can be as large as billions or even more. As a result, we present a novel heuristic algorithm, specifically the augmented-layer graph-based heuristic algorithm, herein referred to as the SMTG-ALGB algorithm. The main framework comprises the construction of the augmented-layer graph (ALG), the primary grooming procedure, the primary tabu search algorithm, the backup grooming procedure, and the backup tabu search algorithm. A crucial component of our grooming algorithm is the length-constraint and no-loop Dijkstra (LCNL-Dijkstra) algorithm based on the ALG, which efficiently tackles the constraints in the SMTG problem. Ultimately, we performed multiple experiments to compare the results of both the ILP formulation and the heuristic algorithm on two artificially-constructed networks and two large-scale, practical networks. The results of our experiments indicate that our novel heuristic algorithm is notably efficient and effective, even when confronted with extensive networks containing hundreds of demands, nodes, and links.

2 PROBLEM STATEMENT

The two-layer network structure[5, 8] models the entire WDM network, consisting of both operational OChs and nodes with OEO converting ability. The physical layer represents the original network, including the physical links and nodes, while the virtual layer models the virtual nodes with OEO converting ability, OEO converters installed on the virtual nodes and established OChs. In Fig.1, an OCh connecting the virtual nodes 1 and 3 is routed by physical nodes 1, 4, and 3 with the red wavelength in the physical layer. From the perspective of the two-layer network, we can formulate the ILP for the SMTG problem below.

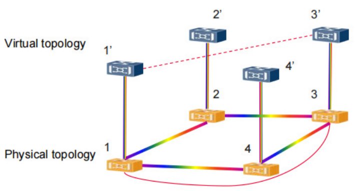

Figure 1: Two-layer network example

2.1 Notation used

In our formulations, we will use the following symbols and variables in TABLE 1 and TABLE 2.

Table 1: Notations

Symbols	Description
V_1	The set of physical nodes in the network.
E	The set of bidirectional physical links.
$E(v_1)$	The set of physical links whose nodes contains v_1.
Λ	The set of available wavelengths, e.g. $\{1, 2, \cdots, 80\}$.
V_2	The set of virtual nodes.
P_{ij}	The set of physical paths connecting virtual nodes i and j, where the overall physical length of each element of P_{ij} adheres to the length constraint.
p, P	The physical path and the whole set, $p \in P = \cup P_{ij}$.
h_e^p	$h_e^p = 1$ if a path p uses the link e; $h_e^p = 0$ if not.
$r^+(p), r^-(p)$	Two directional paths contained in p.
R	The set of directional paths.
$InR(v_2)$	The set of directional paths terminating at $v_2 \in V_2$.
$OutR(v_2)$	The set of directional paths originating at $v_2 \in V_2$.
(p, λ)	Bidirectional OCh with path p and wavelength λ.
(r, λ)	Directional OCh with path r and wavelength λ.
d	A bidirectional demand.
D	The set of bidirectional demands.
s_d	The original node of the demand d.
t_d	The terminal node of the demand d.
B_d	The bandwidth of the demand d.
b_j^d	$b_j^d = -1$, if $j = s_d$; $b_j^d = 1$, if $j = t_d$; $b_j^d = 0$, otherwise.
C	The bandwidth for a wavelength channel.

Table 2: Variables

Var.	Description
$z^{p,\lambda}$	$z^{p,\lambda} = 1$, if OCh (p, λ) is established; $z^{p,\lambda} = 0$, otherwise.
$x_{p,\lambda}^d$	$x_{p,\lambda}^d = 1$, if the primary path of the bidirectional demand d uses the OCh (p, λ); $x_{p,\lambda}^d = 1$, otherwise.
$x_{r,\lambda}^d$	$x_{r,\lambda}^d = 1$, if the primary path of a directional demand which is contained in the bidirectional demand d and from s_d to t_d uses the directional OCh (r, λ); $x_{r,\lambda}^d = 0$, otherwise.
$y_{p,\lambda}^d$	$y_{p,\lambda}^d = 1$, if the backup path of a bidirectional demand d uses the OCh (p, λ). $y_{p,\lambda}^d = 0$, otherwise.
$y_{r,\lambda}^d$	$y_{r,\lambda}^d = 1$, if the backup path of a directional demand which is contained in the bidirectional demand d and from s_d to t_d uses the directional OCh (r, λ); $y_{r,\lambda}^d = 0$, otherwise.
$w_{p,\lambda}^{d,e}$	$w_{p,\lambda}^{d,e} = 1$, if physical link e has a failure, the demand d is influenced and its backup path uses OCh (p, λ); $w_{p,\lambda}^{d,e} = 0$, otherwise.

2.2 ILP formulation for minimizing the OCh usage

Objective function:

$$\underset{\substack{\{z^{p,\lambda}\},\{x^d_{p,\lambda}\},\{x^d_{r,\lambda}\} \\ \{y^d_{p,\lambda}\},\{y^d_{r,\lambda}\},\{w^{d,e}_{p,\lambda}\}}}{\text{Minimize :}} \sum_{p \in P} \sum_{\lambda \in \Lambda} z^{p,\lambda} \tag{1}$$

Constraints:

$$\sum_{\substack{r\in InR(v_2)\\ \lambda\in\Lambda}} x^d_{r,\lambda} - \sum_{\substack{r\in OutR(v_2)\\ \lambda\in\Lambda}} x^d_{r,\lambda} = b^d_{v_2}, \forall d\in D, v_2\in V_2,$$
$$x^d_{p,\lambda} = x^d_{r^+(p),\lambda} + x^d_{r^-(p),\lambda}, \forall d\in D, p\in P, \lambda\in\Lambda. \tag{2}$$

$$\sum_{\substack{r\in InR(v_2)\\ \lambda\in\Lambda}} y^d_{r,\lambda} - \sum_{\substack{r\in OutR(v_2)\\ \lambda\in\Lambda}} y^d_{r,\lambda} = b^d_{v_2}, \forall d\in D, v_2\in V_2,$$
$$y^d_{p,\lambda} = y^d_{r^+(p),\lambda} + y^d_{r^-(p),\lambda}, \forall d\in D, p\in P, \lambda\in\Lambda. \tag{3}$$

$$\sum_{\lambda\in\Lambda}\sum_{p\in P}\sum_{e\in E(v_1)} h^p_e x^d_{p,\lambda} \le 2, \forall d\in D, v_1\in V_1, \tag{4}$$

$$\sum_{\lambda\in\Lambda}\sum_{p\in P}\sum_{e\in E(v_1)} h^p_e y^d_{p,\lambda} \le 2, \forall d\in D, v_1\in V_1, \tag{5}$$

$$\sum_{\lambda\in\Lambda}\sum_{p\in P} h^p_e x^d_{p,\lambda} + \sum_{\lambda\in\Lambda}\sum_{p\in P} h^p_e y^d_{p,\lambda} \le 1 \tag{6}$$

for $\forall d\in D, e\in E$,

$$\sum_{p\in P} h^p_e z^{p,\lambda} \le 1, \forall e\in E, \lambda\in\Lambda, \tag{7}$$

$$\sum_{d\in D} B_d(1-h^p_e)x^d_{p,\lambda} + \sum_{d\in D} B_d w^{d,e}_{p,\lambda} \le C(1-h^p_e)z^{p,\lambda} \tag{8}$$

for $\forall e\in E, p\in P, \lambda\in\Lambda$,

$$w^{d,e}_{p,\lambda} \le \sum_{p'\in P}\sum_{\lambda'\in\Lambda} h^{p'}_e x^d_{p',\lambda'},\ w^{d,e}_{p,\lambda} \le y^d_{p,\lambda},$$
$$w^{d,e}_{p,\lambda} \ge \left(\sum_{p'\in P}\sum_{\lambda'\in\Lambda} h^{p'}_e x^d_{p',\lambda'}\right) + y^d_{p,\lambda} - 1 \tag{9}$$

for $\forall e\in E, d\in D, p\in P, \lambda\in\Lambda$.

The objective function (1) will count the total used OChs. Flow constraints (2) and (3) ensure that the paths of demands are feasible in the virtual layer. No-loop constraints (4) and (5) are required for physical paths of demands. Even though the OCh paths of demands have no loops, the whole physical path can still have them due to the diverse routing of each OCh. In order to ensure that the primary and backup paths are disjoint in the physical layer, constraint (6) is imposed. Furthermore, different OChs must utilize distinct wavelength channels on the same link, as represented by constraint (7). To ensure that traffic demands do not exceed the bandwidth allotted to each OCh, constraint (8) is included. The linearization of the auxiliary variables is given by constraint (9) according to the description of $w^{d,e}_{p,\lambda}$.

Even though the ILP is a precise formulation and will give an exact solution, practical difficulties may arise when dealing with a large-scale network due to numerous variables and constraints. As an illustration, in a network comprised of 65 nodes, 96 links, 20 wavelength channels, and 122 demands, the combined number of variables and constraints generated by the associated ILP may exceed five billion.

3 HEURISTIC APPROACH

The general survivable traffic GRWA is NP-hard [12], and the additional constraints imposed in our proposed SMTG problem, namely the no-loop and OCh length constraints, increase its computational complexity. In light of this, the ILP formulation also suggests that the problem will be challenging to solve optimally. Consequently, a heuristic approach becomes necessary, and we devise the SMTG-ALGB algorithm.

In the upcoming subsections, any symbols that have not been given explicit definitions can be referenced in TABLE 3.

Table 3: Notations

Symbols	Description
G_p	The physical topology.
G_a	The auxiliary graph.
L	The length constraint for the OCh.
π_w	Weight for WLEs.
π_t	Weight for TrEs.
P_d	The path information for demand d, which only contains OChs. We use P^p_d for the primary path and P^b_d for the backup path.
$H(\cdot)$	Get physical hops for a OCh.
$W(\cdot)$	Get the weight of an edge in ALG.
Num_{opt}	The number of nodes in the physical layer.

3.1 Main framework

To begin with, the main framework will be presented, where the primary process is denoted by 'p' while the backup process is denoted by 'b'. We also simplify inputs and outputs for the functions in SMTG-ALGB.

Algorithm 1: SMTG-ALGB

Input: Physical topology G_p, wavelength set Λ, length constraint L, demands D and so on
Output: G_a, primary paths $\{P^p_d\}$ and backup paths $\{P^b_d\}$ for D

Initialize the ALG G_a according to G_p and Λ;
Sort the demands in a non-increasing order ;
G_a, $\{P^p_d\}$ ← Grooming(G_a, D, 'p') ;
G_a, $\{P^p_d\}$ ← Tabu-search(G_a, D, $\{P^p_d\}$,'p') ;
Sort the demands in a non-decreasing order ;
G_a, $\{P^b_d\}$ ← Grooming(G_a, D, 'b') ;
G_a, $\{P^b_d\}$ ← Tabu-search(G_a, D, $\{P^b_d\}$, 'b') ;

3.2 Construction of the augmented-layer graph

Heuristic algorithms for solving GRWA problems involve the graph construction based on the previous two-layer network structure as the first step. For example, the link bundled auxiliary graph (LBAG) in [11] can be effective with traditional grooming and routing algorithms, albeit with some challenges such as wavelength consistency constraints that need special consideration. While Layered-AG (LAG) [14] provides clear representations of all processes, the abundance of edges, such as mux, demux, and converter edges, may complicate the representation and hinder the efficiency of the grooming algorithm. To address this concern and incorporate the advantages of both approaches, we propose the ALG model. An illustrative example of the ALG corresponding to Fig. 1 is presented in Fig. 2, where we assume that there are two available wavelengths. In order to provide a comprehensive understanding of ALG, a detailed explanation is provided below.

3.2.1 Edges in the ALG. In ALG, we classify the edges into three categories as follows.

- WLE: Assuming that each physical link has $|\Lambda|$ wavelengths, $|\Lambda|$ wavelength layers will be generated accordingly. The notation $E_w(i, j, m)$ denotes the wavelength edge that forms the connection between nodes i and j in the m^{th} wavelength layer, while its weight is represented by $W_w(i, j, m)$.
- TrE: Edges representing transmitters and receivers are a crucial component of ALG. The notation $E_t(i, \lambda_m)$ is used to denote the transceiver edges connecting the virtual layer to the m^{th} wavelength layer, with $W_t(i, \lambda_m)$ representing their respective weights.
- OCh: Edges within the virtual layer are crucial in the establishment of an OCh, utilizing designated physical links and a specific wavelength. If an OCh denoted as $E_o(i, j, \lambda_m)$ is established to interconnect nodes i and j using wavelength channel λ_m, the corresponding wavelength edges in the m^{th} wavelength-layer will be removed. The weight for $E_o(i, j, \lambda_m)$ is characterized by $W_o(i, j, \lambda_m)$. The number of physical hops when traversing an OCh through the physical layer can be represented by $H(i, j, \lambda_m)$.

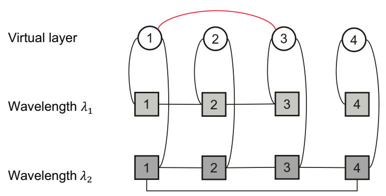

Figure 2: ALG

3.3 Grooming algorithm based on the ALG

After the construction of ALG, we can groom the traffic demands one by one based on it. For example, if we want to establish a connection between nodes 3 and 4, we can find a possible path $E_t(3, \lambda_2)$-$E_w(3, 4, 2)$-$E_t(4, \lambda_2)$, after which we should add an OCh $E_o(3, 4, \lambda_2)$ and delete $E_w(3, 4, 2)$ in the ALG. Basically, there exist the primary and the backup routing algorithms, and these two versions exhibit minor differences in their approach towards updating the graph for the LCNL-Dijkstra.

3.3.1 Grooming algorithm. The main grooming procedure is shown in Algorithm 2. It should be noted that after the LCNL-Dijkstra algorithm section, we perform iterations over both the edge set and the node set of the path. However, this path may not necessarily be the final OCh path for the given demand. We need

Algorithm 2: Grooming

```
Input: G_a, D, the type of grooming T, 'p' for primary and
       'b' backup.
Output: G_a, {P_d}
for all d ∈ D do
    if T == 'p' then
        Delete the OChs where B_a < B_d ;
    else if T == 'b' then
        Delete all WLEs and OChs wich have joint parts
          with the primary path of d ;
        Delete the OCh edges where B_s + B_a < B_d ;
        Update G_a ;
    end
    if LCNL-Dijkstra(G_a, s_d, t_d, Num_opt, L, Path) then
        Get the returned edge set Edge ;
        for all e ∈ Edge do
            Determine the type of e ;
            Update G_a and P_d ;
        end
    else
        Failed grooming process for d ;
    end
    Restore the deleted edges before LCNL-Dijkstra ;
end
```

to transfer the wavelength sub-paths with TrEs into a newly established OCh and save it in P_d. Besides, WLEs used by the new OCh will be deleted from G_a.

3.3.2 LCNL-Dijkstra algorithm. In the LCNL-Dijkstra shown in Algorithm 3, we endeavor to explicate the variables and functions that will be mentioned. $dist[\cdot]$ denotes the distance from a node to the source node, which is determined by the weights in G_a. Additionally, $len[\cdot]$ keeps track of the physical length of the most recent contiguous subpath that exists within a single wavelength layer. Furthermore, $prev[\cdot]$ identifies the parent node and edge of a given node. It should be noted that the nodes with smaller distances are prioritized in Q, which serves as a priority queue. The vector $omsNodes[v]$ stores all non-repeating physical nodes from the original node s_d to v. The function `CheckNoOverlap`$(omsNodes[u], l)$ returns *true* if edge l has no repeating physical nodes with $omsNodes[u]$ except the joint node; it returns $false$ otherwise. The function `CheckLength`(L, u, l) returns *true* if $len[u] + \texttt{OmsLen}(l) \geq L * (1 + \texttt{floor}(len[u]/L))$ and false otherwise. Here, `OmsLen`$(\cdot)$ is used to denote the physical length of a wavelength edge. If node u is relaxed with edge l, and the resulting subpath exceeds the designated limit, this sub-path will be flagged for further splitting. Its weight may be subject to increase during the subsequent `Relax()` part if required.

Next, we briefly describe the relaxing part in Algorithm 4, where we ignore some necessary input variables.

Within the Algorithm 4, we make use of the `ceil()` function when $newOch \neq 0$. It aims to make $len[\cdot]$ accurately reflect the true number of OCh if we consider the physical length constraint. Once a path has been determined, the function `Split()` will be employed to examine the TrEs and the physical length of WLEs between

Algorithm 3: LCNL-Dijkstra

Input: Auxiliary graph G_a, original node s_d, terminal node t_d, Num_{opt}, L, $Path$ (including $Edge$ and $Node$)
Output: $flag$
$flag \leftarrow false$;
$newOch \leftarrow 0$;
for *each node* $v \in G_a$ **do**
  $dist[v] \leftarrow INF, len[v] \leftarrow 0$;
  $prev[v].node \leftarrow -1, prev[v].edge \leftarrow -1$;
end
$dist[s_d] \leftarrow 0$;
Q.insert(s_d) ;
$omsNodes[s_d]$.push(mod(s_d, Num_{opt})) ;
while $Q \neq \emptyset$ **do**
  $u \leftarrow Q$.pop() ;
  if $u == t_d$ **then**
    $flag \leftarrow true$;
    break ;
  end
  for *each* l *from adjacent edges of* u **do**
    if CheckNoOverlap($omsNodes[u], l$) **then**
      continue ;
    end
    if CheckLength(len, L, u, l) **then**
      $newOch \leftarrow 1$;
    end
    Get the other endpoint of l and denoted as v ;
    Relax($dist, u, v, l, newOch, \pi_t$) ;
  end
end
if $flag$ **then**
  Generate a complete path $tmpPath$ based on $prev$;
  $Path \leftarrow$ Split($tmpPath$) ;
end

Algorithm 4: Relax

Input: $dist, u, v, l, newOch, \pi_t$
if $dist[v] \neq INF$ *and* $dist[u] + \mathrm{W}(l) + 2 * newOch * \pi_t < dist[v]$ **then**
  $dist[v] \leftarrow dist[u] + W(l) + 2 * newOch * \pi_t$;
  $prev[v].node \leftarrow u, prev[v].edge \leftarrow l$;
  Q.insert(v) ;
end
if $newOch \neq 0$ **then**
  $len[v] \leftarrow$ ceil($len[u]/L$) + OmsLen(l) ;
else if $l \in E_w$ **then**
  $len[v] \leftarrow len[u]$ + OmsLen(l) ;
else if $l \in E_t$ **then**
  $len[v] \leftarrow 0$;
end
Update $len[v]$ and $omsNodes[v]$;

them. In the event that a decision is made to split the corresponding wavelength subpath, the splitting node can be chosen flexibly, taking into account factors such as the reusability and availability of wavelength channels. It is important to note, however, that the number of splits should not exceed ceil($len[u]/L$).

3.3.3 *Grooming policy and weights setting.*

Before we talk about the setting of weight, we go through the bandwidth of a channel first. The bandwidth for a wavelength channel can be categorized into free bandwidth B_a, dedicated bandwidth, and spare bandwidth. The spare bandwidth can further be divided into sharable spare bandwidth B_s and non-sharable bandwidth.

In our survivable protection scheme, we will identify a risk-disjoint backup path for the primary path of each demand, taking link failure scenarios into consideration. Given that the backup routing process entails the utilization of sharable bandwidth, it is judicious to prescribe distinct weight settings for the primary and backup grooming, respectively. Besides, In order to minimize the usage of OChs, which constitute the cost component of our task, it is imperative to enhance the sharable ability of the routing algorithm. Considering the aforementioned background and the insights provided in [12], we propose the following weight setting and diverse grooming processes can be delineated by varying the weights in ALG.

Table 4: Weight setting for primary and backup routing

Primary, edge e	WLE	$W_w(e) = \begin{cases} \pi_w, & \text{if } E_w(e) \text{ is unused} \\ \infty, & \text{otherwise} \end{cases}$
	TrE	$W_t(e) = \pi_t$
	OCh	$W_o(e) = \begin{cases} \alpha \times H(e), & \text{if } B_a \geq B_d \\ \infty, & \text{otherwise} \end{cases}$, where $0 < \alpha \leq 1$ and α is tunable.
Backup, edge e	WLE	$W_w(e) = \begin{cases} \pi_w, & \text{if } E_w(e) \text{ is unused} \\ \infty, & \text{otherwise} \end{cases}$
	TrE	$W_t(e) = \pi_t$
	OCh	$W_o(e) = \begin{cases} \alpha \times H(e), & \text{if } B_s \geq B_d \\ (\beta + \alpha(1-\beta)) \times H(e), & \text{if } B_s < B_d \leq B_s + B_a \\ \infty, & \text{otherwise} \end{cases}$, where $0 < \alpha \leq 1$ and $\beta = (B_d - B_s)/B_d$.

3.4 Tabu Search Algorithm

Tabu search is a widely employed meta-heuristic method. For a comprehensive understanding of it, interested readers are advised to refer to [4]. Our proposed tabu search algorithm entails the removal of a pre-existing OCh and subsequently rerouting its associated demands to other extant OChs, the complete removal of an OCh may not be a feasible option in every instance. Consequently, we adopt a strategy which accepts a move that either preserves the extant number of OChs or augments it. We delineate the following significant measures.

Firstly, thoroughly examining every OCh in each iteration would be exceedingly time-consuming. To overcome this issue, we propose generating a relatively small set of OCh edges, then selecting a single edge for manipulation in each subsequent iteration. Secondly, to enhance the algorithm's flexibility, we adopt a hybrid approach involving both fixed and variable tabu steps. Moreover, we integrate the OCh into the tabu list and remove the corresponding OCh from the list after the later tabu steps. Thirdly, when it is necessary to remove an OCh and re-groom the associated demands, it is probable that they will be assigned to the same physical links that were previously allocated for the deleted OCh. To circumvent this issue, the weights of all relevant WLEs can be raised after removing the OCh. Lastly, an important measure is inspired by previous work in [2] and [3], where a cost function is defined for each move of re-grooming demand. In the end, we opt to choose the slightest cost move. We omit the intricate details, which resemble those of the preceding algorithm.

3.5 Experiments

In this section, we provide the details of our experiments involving four distinct networks. The first two are synthetic small-scale networks, while the latter two correspond to large practical networks of cities in China. Table 5 presents the statistical characteristics of networks and demands. Specifically, in the network under consideration, each physical link can accommodate up to 80 wavelengths, and each wavelength channel has a bandwidth of 100G. The traffic demands being considered have diverse bandwidth requirements, e.g. $\{2.5, 10, 100\}$G; The physical length of links in Network 1 and 2 varies between 1, 2, and 3 km, whereas in Network 3, it is mainly 1 km. The physical length of links in Network 4 is much longer, from 1 to 800 km. For the weights of WLEs and TrEs in our heuristic algorithm, we set $\pi_w = 1$ and $\pi_t = 10$, respectively. Besides, we limit each vertex pair to a maximum of 20 paths for the computation of the ILP. Our experiments are running in an Ubuntu server with an Intel(R) Xeon(R) Gold 6266C CPU of 3.00GHz and a RAM of 64GB.

Table 5: Network topology

Network	$\|V_1\|$	$\|E\|$	$\|V_2\|$	$\|\Lambda\|$	$\|D\|$	L	$\|P\|$
1	4	4	4	3	4	10	8
2	6	8	6	4	6	10	73
3	65	96	65	20	122	6	6429
4	227	409	227	40	578	1000	33537

Table 6 presents the ILP and heuristic results (OCh number and time consumption). For the heuristic algorithm with tabu search, we achieve the same optimal values as we get from the ILP for small-scale networks. Moreover, the time consumption of the heuristic algorithms for small-scale networks is almost negligible. Besides, it was not possible to construct the ILP using the SCIP solver for networks 3 and 4, since the variables and constraints involved in network 3 are estimated to be approximately 1.5 billion and 4.5 billion, and they will be large for the network 4. However, even when conducting hundreds of rounds of tabu search on large-scale networks, the heuristic remains effective and highly competitive.

4 CONCLUSION

In this paper, we have presented the SMTG-ALGB algorithm, which tackles the survivable GRWA problem with the no-loop and OCh length constraints and aims to minimize the usage of OCHs. We first construct the ALG and then devise grooming algorithms. Since our primary and backup grooming algorithms' effectiveness is highly dependent on the weights configured within the ALG, a detailed exposition on weight setting is also provided. Lastly, a tabu search algorithm is proposed to further improve the grooming result.

The experiments have demonstrated that our algorithm can achieve the optimal solution for a small network and provide effective grooming results for large practical networks, while the ILP-based method can only be applied to small networks.

Table 6: Overall Results

Network	ILP		Our algorithm (without TS)		Our algorithm (with TS)	
	OCh	Time	OCh	Time	OCh	Time
1	4	1s	5	0.001s	4	0.001s
2	8	4221s	9	0.001s	8	0.003s
3	None	None	113	0.2s	102	20s
4	None	None	1104	8s	1060	766s

REFERENCES

[1] Asima Bhattacharya, Malabika Sarder, Monish Chatterjee, and Diganta Saha. 2020. Efficient integration of logical topology design and survivable traffic grooming for throughput enhancement in WDM optical networks. *IEEE Access* 8 (2020), 58155–58170.
[2] Ilfat Ghamlouche, Teodor Gabriel Crainic, and Michel Gendreau. 2003. Cycle-based neighbourhoods for fixed-charge capacitated multicommodity network design. *Oper. Res.* 51, 4 (Aug. 2003), 655–667.
[3] Ilfat Ghamlouche, Teodor Gabriel Crainic, and Michel Gendreau. 2004. Path relinking, cycle-based neighbourhoods and capacitated multicommodity network design. *Ann. Oper. Res.* 131, 1-4 (Oct. 2004), 109–133.
[4] Fred Glover and Manuel Laguna. 1998. *Tabu search.* Springer.
[5] Arunita Jaekel, Ataul Bari, and Subir Bandyopadhyay. 2008. Resilient traffic grooming for WDM networks. *J. Opt. Netw.* 7, 5 (May 2008), 378.
[6] Arunita Jaekel, Ataul Bari, Quazi Rahman, Ying Chen, Subir Bandyopadhyay, and Yash Aneja. 2012. Resource efficient network design and traffic grooming strategy with guaranteed survivability. *Optical Switching and Networking* 9, 4 (2012), 271–285.
[7] Gerd Keiser. 2021. *Optical Signal Attenuation and Dispersion.* Springer Singapore, Singapore, 93–145. https://doi.org/10.1007/978-981-33-4665-9_3
[8] Junyan Liu, Fan Zhang, Kerong Yan, Junqi Ma, and Bo Bai. 2022. An ILP-based approach for dual-layer optimization in large-scale optical networks. In *2022 31st Wireless and Optical Communications Conference (WOCC).* IEEE.
[9] Hassan Naser and Ming Gong. 2007. Link-disjoint shortest-delay path-pair computation algorithms for shared mesh restoration networks. In *2007 12th IEEE Symposium on Computers and Communications.* IEEE, 269–274.
[10] Xinyun Wu, Tao Ye, Qi Guo, and Zhipeng Lü. 2015. GRASP for traffic grooming and routing with simple path constraints in WDM mesh networks. *Comput. Netw.* 86 (July 2015), 27–39.
[11] Wang Yao and B Ramamurthy. 2005. A link bundled auxiliary graph model for constrained dynamic traffic grooming in WDM mesh networks. *IEEE J. Sel. Areas Commun.* 23, 8 (Aug. 2005), 1542–1555.
[12] Wang Yao and B Ramamurthy. 2005. Survivable traffic grooming with path protection at the connection level in WDM mesh networks. *J. Lightwave Technol.* 23, 10 (Oct. 2005), 2846–2853.
[13] Dongyun Zhou and Suresh Subramaniam. 2000. Survivability in optical networks. *IEEE network* 14, 6 (2000), 16–23.
[14] Hongyue Zhu, Hui Zang, Keyao Zhu, and B Mukherjee. 2003. A novel generic graph model for traffic grooming in heterogeneous WDM mesh networks. *IEEE ACM Trans. Netw.* 11, 2 (April 2003), 285–299.
[15] K Zhu and B Mukherjee. 2003. A review of traffic grooming in WDM optical networks: Architectures and challenges. *Optical Networks Magazine* 4, 2 (2003), 55–64.