



Two-stage EDA-based approach for all optical WDM mesh network survivability under SRLG constraints

Jianyong Sun*

CPIB, School of Bioscience, The University of Nottingham, Nottingham, UK

ARTICLE INFO

Article history:

Received 22 August 2008

Received in revised form 6 January 2010

Accepted 17 January 2010

Available online 25 January 2010

Keywords:

Shared-risk-link-group

Routing and wavelength assignment problem

Greedy heuristic

Estimation of distribution algorithm

ABSTRACT

In this paper, a two-stage evolutionary algorithm is proposed to solve an \mathcal{NP} -complete telecommunication problem—all optical wavelength-division multiplexing (WDM) mesh network survivability under shared-risk-link-group (SRLG) constraints. First of all, a novel greedy heuristic with two control parameters is developed to construct feasible solutions of the telecommunication problem. An estimation of distribution algorithm (EDA) with guided mutation is applied to search for optimum settings of the two control parameters in respective two stages. Given the found best control parameters, an optimal solution of the considered problem can be constructed by the greedy heuristic. Experimental results show that the proposed approach compares favorably against the best-known evolutionary-based algorithm in 26 out of 30 test instances in terms of solution quality within given time limit.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

Wavelength-division multiplexing (WDM) technology greatly increases the network capacities. It is generally believed that the new generation Internet will be mainly based on WDM technology [30]. The network capacity is greatly increased, survivability of an optical WDM mesh network becomes more and more crucial both for customers and providers since a link or node failure could cause huge damages. Therefore, network providers are eager to design a survivable network to minimize the loss of data.

Many protection schemes have been developed for network survivability. Among these schemes, the shared path-protection scheme has been found that can use less network sources [35]. In the path-based protection scheme, a lightpath¹ that transports connection requests by wavelengths from the source node to the destination node is called as a *primary* or *working* path.² To protect the working path against single link failure,³ a *protection* or

backup path disjointed with the working path is routed between the source and destination nodes. Wavelength channels should be assigned on links involved in the working and backup paths. Hence, the network survivability problem is also known as the routing and wavelength assignment problem (RWAP).

A number of heuristics have been proposed for the routing and wavelength assignment problem, for examples in [1,29,28,45] to just name a few, based on different protection schemes. But most of them do not consider the shared-risk-link-group (SRLG) constraints. In a WDM mesh network, fibers containing in the same duct, or cable, belong to the same SRLG. Fibers in the same SRLG could fail simultaneously because of destructive events, such as earthquake. These fibers share the same “risk”. Due to the existence of SRLG, the RWAP becomes much more complicated. Actually, it has been proved to be \mathcal{NP} -complete [19]. Fig. 1 shows a simple example of WDM network under SRLG constraints.

In the figure, the example WDM network has five nodes and seven ducts in the duct-layer. In Fig. 1(a), the grey rectangles represent the ducts used to wrap the fibers (the solid lines in the plots). Fig. 1(b) shows the network in the link-layer. In the plot, two fibers (1,2) and (2,3) are bundled together by the same duct. In Fig. 1(c), the protection scheme is shown for two connection requests, which are represented by solid and dashed arrow lines respectively. To transmit the request from node 2 to node 3, in case that we do not consider the duct topology, the two paths $2 \rightarrow 3$ and $2 \rightarrow 0 \rightarrow 1 \rightarrow 3$ can be used as the working path and the protection path, respectively considering single-link-failure protection. Alternative protection path is $2 \rightarrow 1 \rightarrow 2$. In case that the duct between node 2 and 3 fails, the alternative path cannot be used as the pro-

* Corresponding author at: CPIB, School of Bioscience, The University of Nottingham, Sutton Bonington, UK.

E-mail address: j.sun@cpib.ac.uk.

¹ In a WDM network without wavelength converter, a lightpath is a connection between a source node and a destination node occupying the same wavelength. Many fiber links may be involved in a lightpath to provide an all-optical connection between the source and destination nodes.

² We do not differentiate the two terms in the paper.

³ Since the optical cross-connect (OXC), a device to switch an optical signal from an incoming fiber to an outgoing fiber on the same wavelength, in an all-optical WDM mesh network is seldom broke, we only consider link failure in this paper.

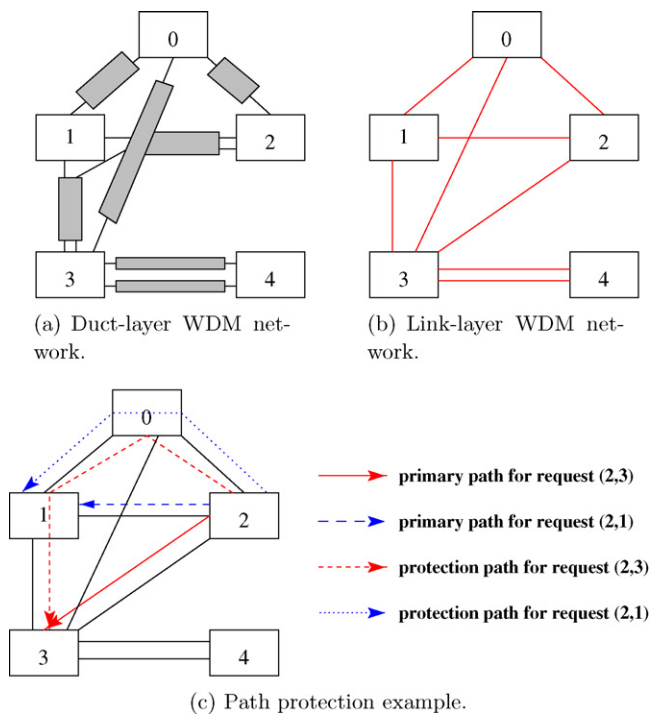


Fig. 1. Example of a duct-layer WDM network.

tection path. Moreover, in assigning wavelength for the paths, the duct-layer topology should also be considered. For example, to protect the requests from node 2 to node 3, and from node 2 to node 4, we use the paths $2 \rightarrow 0 \rightarrow 1 \rightarrow 3$ and $2 \rightarrow 0 \rightarrow 1$ as shown in Fig. 1(c), respectively. We may want to use the same wavelength on links $(2, 0)$ and $(2, 1)$ for the protection of the primary paths $2 \rightarrow 3$ and $2 \rightarrow 1$ because they are fiber-disjoint. However, notice that the two primary paths go through the same duct $(2, 3)$, the two paths may fail at the same time. Therefore, we should use different wavelengths for the two protection paths.

From the simple example, one can see that the routing and wavelength assignment under the SRLG constraints is much more difficult than the problem without considering the SRLG constraints. More problems should be considered when carrying out routing and wavelength assignment. Zang and Mukherjee [46] used integer linear programming (ILP) to model the RWAP in WDM mesh network under SRLG constraints with various protection schemes. According to their analysis, the complexity of integer linear programming (ILP) formulation for the shared-path protection of the RWAP under SRLG constraints is $\mathcal{O}(N^4WD)$ in terms of the number of decision variables, where N is the number of nodes, W is the number of wavelengths available on each link and D is the number of ducts. For the simple example, if we set $W = 10$, the number of variables is in the magnitude of 777,600. Therefore, it is not practical to use exact algorithms (such as branch and bound algorithm) for even a small-size network RWAP due to the inhibitive computational cost to be used.

Fortunately, the development of nature-inspired optimization algorithms, such as genetic algorithm [11], estimation of distribution algorithm [24], and many others, provides us a possibility to find a near global optimum in a reasonable short time for the \mathcal{NP} -complete problems. We could follow the ILP formulation of the considered problem to represent a solution as a vector of integer variables. However, the size of the solution vector will be as many as millions even for a medium-sized network instance. This will cause low-efficiency of the nature-inspired algorithm if the integer solution representation is adopted. In light of this, to make the

nature-inspired algorithm efficient on the considered problem, we have to design suitable solution representation.

In this paper, we propose an evolutionary-based approach with deliberate solution representations for the RWAP under SRLG constraints. First of all, a parameterized greedy heuristic is developed to construct solutions of the problem. The greedy heuristic includes mainly three parts, i.e. the working paths routing procedure, the protection path routing procedure and the wavelength assignment procedure since a solution of the RWAP includes the routing paths (the working paths and protection paths) for the connection requests and the wavelengths assigned on these paths. In the greedy heuristic, to route the working paths, a set of candidate paths (called *PathList*) for each connection request is firstly constructed. The working path routing procedure selects paths from the prepared *PathList* as the working paths. The protection path routing procedure constructs protection paths iteratively in an order of the connection requests. The wavelength assignment procedure assigns wavelengths for the working and protection paths, and returns a network cost as a measure of the solution.

There are two control parameters, i.e. the combination of the available working paths for the connection requests and the order of connection requests for protection paths, in the greedy heuristic. Given a pair of control parameters, a feasible solution of the considered problem can be uniquely constructed. Therefore, we can optimize the considered problem through searching optimal control parameters. In this paper, the estimation of distribution algorithm [24] with guided mutation [49] is applied to find optimal parameters of the heuristic. The proposed algorithm includes two stages. The first stage searches for the optimal combination of the candidate working paths for the connection requests. The second stage searches for the optimal order of the connection requests for the protection paths routing.

The proposed algorithm was compared with the best-known EA-based algorithm (called BH/EA/G) [49] and a pure random algorithm based on the proposed greedy heuristic, on a suite of test networks with different network topologies. The comparison results showed that the proposed algorithm outperforms BH/EA/G in 26 out of 30 test instances, and the pure random algorithm on all the test instances, in terms of solution quality with the same computational cost.

The rest of the paper is organized as follows. Section 2 gives a review on the related heuristics/meta-heuristics work to the telecommunication optimization problems. The main contribution of the paper is also presented. Section 3 defines the considered shared-path protection RWAP under SRLG constraints. Section 4 presents the proposed greedy heuristic. The EDA-based algorithm for optimizing the problem w.r.t. the control parameters is introduced in Section 5. Computational results are given in Section 6. Section 7 concludes the paper.

2. Related work

Optimization problems in telecommunication have been extensively studied in the literature from different perspectives. Readers are referred to [37] for details about the problems including planning and design of telecommunication networks, routing, network protection, grooming, restoration, wireless communications, network location and assignment problems, Internet protocol, world wide web, stochastic issues and many others. In this paper, we focused on the application of heuristics/meta-heuristics to the routing and wavelength assignment problem. For a comprehensive review, readers are referred to [6].

Heuristic algorithms have been applied for dynamic RWAP considering different objectives, or based on different protection schemes. For examples, Oki et al. [32] proposed a weighted-SRLG selection algorithm (WSRLG) for finding a given number of disjoint

paths between the source and destination nodes with the smallest cost. Xiao et al. [41] presented a simple heuristic as well for finding the maximum disjoint paths. Lee et al. [26] developed a hierarchical scheme for handling multiple simultaneous failures under SRLG constraints. In the papers of [42,43], the multiple segment protection scheme is proposed for the SRLG protection. Heuristics have also been proposed for the static RWA [33,34,38,40,50].

Additionally to these heuristics, many meta-heuristics including genetic algorithms, particle swarm optimization (PSO), artificial immune systems (AIS), ant colony optimization (ACO), and viral systems (VS) and many others have also been applied for the telecommunication optimization problems.

The artificial immune systems, firstly introduced by Farmer et al. [10] have continued being applied to several network problems. For examples, an AIS developed in [4] is applied to find a set of k -Spanning trees; the AIS in [22,25] is applied to the routing problem. The viral system (VS) [7] has been applied to a library of medium-to-large-sized cases of the Steiner problem.

Particle swarm optimization (PSO), firstly introduced by Kennedy et al. [9,23], has been applied to the static and dynamic routing and wavelength problems in [17,36] and [15,16,18], respectively. An adaptive PSO has been applied to the RWA [5], a discrete PSO has been applied to the multiple destination routing problem [47]. Branch-and-bound and PSO have been hybridized to the peer-to-peer optimization in large unreliable networks [3]. The multiple constrained QoS multicast routing has been addressed by the PSO as in [21]. Algorithms based on other swarm intelligence including ant colony [8] and bee colony [31] have been applied to the network optimization problems, such as dynamic server allocation problem in Internet hosting centers [31], the RWA [27], the multicast Routing problem [20], and so on.

However, few efforts have been devoted to the shared-path protection scheme for the static RWA with SRLG-constraints. Among these efforts, [39] modeled the survivable network problems as an integer linear programming to maximize the revenue and proposed a tabu search heuristic considering different path protection schemes including dedicated, shared and unprotected. Zang and Mukherjee [46] proposed an off-line heuristic to minimize the total number of wavelengths used on links in the network. Zhou and Mouftah [50] proposed a genetic algorithm for long-haul WDM mesh network sparse capacity planning with SRLG constraints to minimize the cost and load balancing. In their algorithm, a solution (configuration) of the static RWA under SRLG is encoded to be a binary string. The classical GA is applied. However, as we stated in Section 1 that even for a small-size network, the length of the binary string will be prohibitively large in their algorithm. It is well-known that for such a large binary encoding, GAs would be very time-consuming and low-efficiency. In our previous work [49], a two-stage EDA-based approach has been proposed for the problem in order to make the application of EAs efficient. The developed algorithm has been experimentally justified to be more effective than the heuristic developed by [46] in terms of solution quality. The proposed algorithm is currently the best evolutionary-based algorithm for the RWA under SRLG constraints.

In this paper, the problem we studied the same problem as in [49]. Also, the same two-stage algorithmic framework is applied. The most important difference to the constructive heuristic developed in [49] is that a simple yet natural heuristic is proposed for the working and protection paths routing. The novel working path construction scheme makes the working path routing process easy to be understood. Moreover, when applying EDA to find the optimal routing and wavelength assignment solution, the solution representation in the EDA will be totally different to the one in [49]. And hereafter new EDA components should be developed to adapt to the new solution representation. Moreover, a new repair heuristic and an intelligent initialization heuristic can be developed thanks

to the new working path routing heuristic. As well known in the evolutionary computation community, the solution representation defines the fitness landscape of the problem to be searched by the EA. Actually, it is the ruggedness of the fitness landscape that affects the performance of an evolutionary algorithm at most. Based on the new routing and wavelength assignment heuristic, the developed EDA is expected to perform better than the EDA in [49] in terms of solution quality, and faster than that in terms of computational time. These claims will be justified in the experimental studies.

3. Problem definition

The static RWA under SRLG constraints can be modeled as an optimization problem on a simple directed graph. The same as in [49], the problem can be described as follows. Given:

1. V : the set of nodes in the graph under consideration.
2. E : the set of directed links (edges) in the graph.
3. W : the number of wavelengths available on each link. The wavelengths are numbered from 1 to W .
4. D : the set of ducts.
5. R : the set of connection requests. $M = |R|$. The requests are numbered from 1 to M . Each connection request has a source node and a destination node.⁴ It requires a working lightpath and a backup lightpath from its source to its destination. In this paper, a lightpath is a directed path in which all the links use the same wavelength.
6. G : the set of SRLGs. If two links are in the same SRLG, we say that they are SRLG-joint. Otherwise, we call them SRLG-disjoint.

The goal is to determine a working lightpath and a backup lightpath for each connection request in R . The constraints are:

- C1 The number of wavelengths used on each link cannot exceed W , or in mathematical form:

$$\sum_{k=1}^{D_{ij}} [W_w(i, j) + W_p(i, j)] \leq W, \quad \forall (i, j) \in E$$

where $W_w(i, j)$ and $W_p(i, j)$ are the wavelength used by the working lightpath and backup lightpath that pass through the link (i, j) .

- C2 The working lightpath and its backup lightpath for each connection request must be SRLG-disjoint. If link e_1 is used in the working lightpath and e_2 in its protect lightpath, then e_1 and e_2 are SRLG-disjoint. Otherwise, we call them SRLG-joint. In mathematical form, the constraints on SRLG can be written as follows:

$$\sum_{1 \leq s, d \leq N} \delta_{i,j,\kappa}^{s,d,w} \leq 1 \quad \forall \kappa \in D, \quad 1 \leq w \leq W$$

where $\delta_{i,j,\kappa}^{s,d,w}$ takes 1 if wavelength w is used on the link (i, j) by some back lightpath from node s to node d when duct κ fails; 0 otherwise.

- C3 Two working lightpaths cannot use the same wavelength on the same link, or in mathematical form:

$$m_{ij}^w + \sum_{1 \leq s, d \leq N} F_{ij}^{s,d,w} \leq 1$$

⁴ Two different requests in R may have the same source and destination.

for all link $\langle i, j \rangle$ where m_{ij}^w takes 1 if wavelength w is used by some backup lightpath that occupies link $\langle i, j \rangle$; 0 otherwise, and $F_{ij}^{s,d,w}$ is 1 if wavelength w on link $\langle i, j \rangle$ is carrying traffic from source s to destination d , 0 otherwise.

C4 A protection lightpath cannot share the same wavelength on the same link with any working lightpaths, or in mathematical form:

$$m_{ij}^w \times t_{ij}^{s,d,w} = 0, \quad \forall \langle i, j \rangle, \langle s, d \rangle \in E$$

where $t_{ij}^{s,d,w}$ takes 1 if wavelength w is used for the working path of the request $\langle s, d \rangle$ that traverse link $\langle i, j \rangle$, 0 otherwise.

C5 If two working lightpaths are SRLG-joint, their protection lightpaths cannot use the same wavelength on the same link, or in mathematical form:

$$G_{ij}^{s,d,w} \times G_{ij}^{u,v,w} = 0, \quad \forall \langle i, j \rangle \in E \quad \text{and} \quad WP_{sd} \cap WP_{uv} = \emptyset$$

where $G_{ij}^{s,d,w}$ takes 1 if wavelength w on link $\langle i, j \rangle$ is used as protection for connection request from source s to destination d , and WP_{sd} is the working path for connection request from s to d .

The objective is to minimize the cost:

$$\sum_{e \in E} (F_e + S_e) \quad (1)$$

where F_e is the number of wavelengths on link e used in working lightpaths and S_e the number of wavelengths on link e used in backup lightpaths.

4. Greedy heuristic

Since the objective of this problem is to minimize the total number of wavelengths used in a routing scheme, a heuristic for the above problem should have the following properties:

1. The wavelength sharing in the backup lightpaths should be maximized.
2. SRLG-disjointness among the working lightpaths should be encouraged such that backup lightpaths have a good chance to share wavelengths.
3. The lightpaths should be as short as possible, since a wavelength needs to be assigned to each link traversed by these lightpaths.

The proposed greedy heuristic consists of four phases taking the constraints C1–C5 into account. The four phases are the working path routing, the protection path routing, the repair heuristic, and the wavelength assignment heuristic. In the following sections, we will use the example described in Section 1 to demonstrate the greedy heuristic. In the example, we assume that two connection requests from node 2 to node 3 and from node 2 to node 4 need to be routed and protected.

First of all, to route the working paths for each connection request $i \in R$, we construct a set of paths using Yen's K-shortest path routing algorithm [44] as the candidate paths without taking the SRLG constraints into consideration, and denote the set of candidate paths as $PathList_i = \{P_{i1}, \dots, P_{iu_i}\}$, $1 \leq i \leq M$, where u_i is the number of the available paths for connection request i . The u_i 's, $1 \leq i \leq M$ are not necessary to be equal. It is worth pointing out that these candidate paths need to be computed only once. The underlying rationale is that no matter what routing algorithms are to be applied, the working paths have to be among these candidate paths.

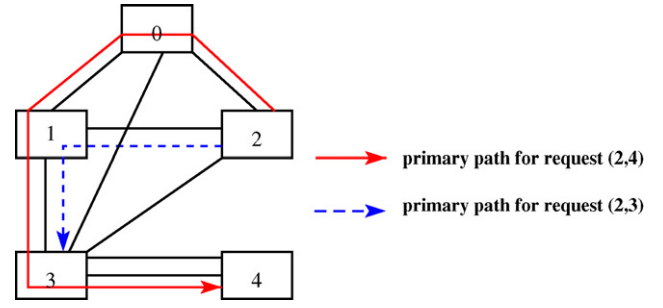


Fig. 2. The routing paths of the connection requests of the example network.

Example: Recall the example network, we see that the candidate paths for connection request $\{2, 3\}$ are $P_{11} = 2 \rightarrow 3$, $P_{12} = 2 \rightarrow 1 \rightarrow 3$, $P_{14} = 2 \rightarrow 0 \rightarrow 3$, and $P_{14} = 2 \rightarrow 0 \rightarrow 1 \rightarrow 3$, while for $\{2, 4\}$, the paths are $P_{21} = 2 \rightarrow 3 \rightarrow 4^{(1)}$, $P_{22} = 2 \rightarrow 3 \rightarrow 4^{(2)}$, ${}^5P_{23} = 2 \rightarrow 1 \rightarrow 3 \rightarrow 4$, $P_{24} = 2 \rightarrow 0 \rightarrow 3 \rightarrow 4$ and $P_{25} = 2 \rightarrow 0 \rightarrow 1 \rightarrow 3 \rightarrow 4$. It can be noted that $u_1 = 4$ and $u_2 = 5$.

In the first phase of our greedy heuristic, we choose a combination of the candidate paths $PathList_i$, $1 \leq i \leq M$ as the working paths for the connection requests in R . In the second phase, a heuristic is presented to compute backup paths for all the working paths established in the first stage. Since there is no guarantee that the constructed solution to the RWP under SRLG constraints in the first two stages is feasible, a heuristic is proposed in the third phase to repair the infeasible solutions if necessary. In the fourth phase, we adopted the same heuristic used in [49] to assign a wavelength to each path constructed in the previous three phases and returns a network cost.

4.1. The working and protection path routing scheme

The working path routing procedure is fairly easy. We only need to choose a combination of the constructed candidate working paths. Specifically, given a combination $\mathbf{x} = (x_1, \dots, x_M)$ where $x_i \in \{1, 2, \dots, u_i\}$, we can interpret the combination to be a set of working paths. That is, for the i th connection request, its working path is selected as the x_i th path in $PathList_i$. The working path routing procedure returns a set of working paths WP . It can be seen that the set of working paths WP is uniquely defined given \mathbf{x} . Moreover, it is guaranteed that all the connection requests are routed so long as $\forall i, u_i \neq 0$.

Example: For the example network, if a solution is represented as $\mathbf{x} = (2, 5)$, the working path routed for the connection requests are then P_{12} and P_{25} , respectively. The working paths set WP is then $\{2 \rightarrow 1 \rightarrow 3; 2 \rightarrow 0 \rightarrow 1 \rightarrow 3 \rightarrow 4\}$ as shown in Fig. 2.

Given a set of working lightpaths $WP = \{wp_1, wp_2, \dots, wp_M\}$ for all the connection requests in R (where wp_r is for request r). The goal of the protection path routing is to route bp_r for each connection request wp_r . The routing is based on a permutation $\sigma = (\sigma_1, \dots, \sigma_M)$ of the working paths. Suppose that for the first K working paths, a set of protection paths $BP^K = \{bp_{\sigma_1}, \dots, bp_{\sigma_K}\}$ have been routed (in the case of $K = 0$, no backup lightpath has been established). The protection path σ_{K+1} for the corresponding working path is as follows:

⁵ Since there are two links between node 3 and node 4, we use superscripts (1) and (2) to describe the resultant different paths. In the experiments, we introduce a new node between the nodes 3 and 4 to differentiate the paths.

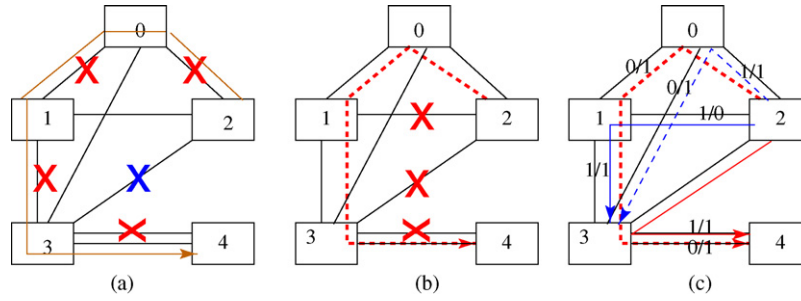


Fig. 3. The routing paths of the connection requests of the example network.

Step 1 Let H be the set of links that

- (a) have been used by working lightpath $wp_{\sigma_{K+1}}$, or
- (b) are SRLG-joint with links in working path $wp_{\sigma_{K+1}}$.

Step 2 Applying Dijkstra algorithm to the graph $(V, E \setminus H)$ with the equal link length to find the shortest path from the source to the destination in connection request π_{K+1} as backup lightpath $bp_{\sigma_{K+1}}$. If Dijkstra algorithm fails in finding a path, set $bp_{\sigma_{K+1}} = \emptyset$.

The algorithmic parameters $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_M)$ and the set of working lightpaths WP can be regarded as the input to the protection path routing scheme, and the backup lightpaths $BP = \{bp_1, bp_2, \dots, bp_M\}$ are its output.

Remark 1. If lightpath $bp_{\sigma_{K+1}}$ uses any links in H defined in Step 1, it will lead to an infeasible routing. For this reason, the protection path routing scheme prevents $bp_{\sigma_{K+1}}$ from using these links.

Remark 2. If there is no path from the source node to the destination node for request σ_{K+1} in graph $(V, E \setminus H)$ or $wp_{\sigma_{K+1}}$, the protection path routing scheme will set $bp_{\sigma_{K+1}} = \emptyset$. In this case, the set of source–destination pairs with no protection-paths assigned is returned as I , the set of the corresponding indices of the working paths is returned as J .

Example: For the example network, if the order $\sigma = (2, 1)$, according to the protection routing heuristic, firstly we need to route for the second connection request, i.e. the request from node 2 to node 4, and then the first request from node 2 to node 3. Fig. 3 shows the routing for the second request. In the figure, the working path is denoted by the arrow solid line. Moreover, the links that are contained in the set H are shown by red cross according to Step 1 in the heuristic. The red crosses show the links specified by Step 1(b), i.e. the links that have been used by the working path WP_2 , while the blue cross shows the link specified by Step 1(c). It can be seen that when the Dijkstra algorithm is applied, no path can be found. This means that $bp_{\sigma_2} = \emptyset$. In other words, the routing defined by (\mathbf{x}, σ) is infeasible.

4.2. The repair heuristic

It can be seen that a solution generated by means of the protection path routing procedure may be infeasible. We propose the following heuristic to repair the infeasible solution. Suppose a solution is returned with a working path set WP , a protection path set BP , and an unassigned set of source–destination pairs I and corresponding indices J . The repair heuristic for the k th source–destination pair in I ($\ell = I(k)$ the index of the connection request) is shown as follows.

- Step 1 Set $j = 0, i := J(k)$ (i.e. i is the index of the candidate working path in $PathList_\ell$) and $PI := \emptyset$.
- Step 2 If $P_{\ell i} \neq WP_\ell$, perform the protection path routing scheme for the connection request ℓ (cf. Section 4.1) to find a protection path p ; $PI := PI \cup p$; Set $j := j + 1$.

Step 3 If j is larger than the number of available path in $PathList_\ell$, stop. Calculate the lengths of the paths in PI (if some path p is \emptyset , simply set its length to be ∞) and take the path with minimum length as the protection path for the connection request ℓ . Otherwise, go to Step 2.

Remark 1. To make the infeasible solution feasible, the repair heuristic works on the source–destination pairs whose protection path is not assigned. The heuristic replaces the working path of a source–destination pair with another potential path in the path list. The path with the minimal length is chosen to be the protection path. This can most likely decrease the wavelength used.

Remark 2. It should be pointed out that the repair heuristic can still not guarantee that all the protection paths can be routed since the found protection path p in Step 2 could be \emptyset . However, since we iterate over all the candidate working paths, if no protection path can be found at all, the only reason is that there are no two SRLG-disjoint paths in the network topology. In our experiment, we will ignore these kind of network instances.

Example: In the example network, the protection path for the request σ_1 cannot be routed. When applying the repair operation, we need to check one by one the paths in $PathList_{\sigma_1}$ other than the used working path to find the working path that (1) its corresponding protection path can be routed; and (2) its protection path has the minimum number of length. In the example, from Fig. 3(b), we see that if we choose $P_{11} = 2 \rightarrow 3 \rightarrow 4$ as the working path, its protection path can be routed as $2 \rightarrow 0 \rightarrow 1 \rightarrow 3 \rightarrow 4$. For request σ_2 , it can be routed as $2 \rightarrow 0 \rightarrow 3$. Fig. 3(c) shows the working and protection path routing results.

4.3. The wavelength assignment scheme

Given the set of working lightpath WP and the set of backup lightpath BP for all the connection requests in R , the task of wavelength assignment scheme is to assign wavelengths to each working lightpath and backup lightpath such that the total number of wavelengths used is minimized under the following constraints:

- CW1 two working lightpaths must be assigned different wavelengths if they traverse the same link,
- CW2 a working path and a backup lightpath must be assigned different wavelengths if they traverse the same link, and
- CW3 two backup lightpaths have to be assigned different wavelengths, if they traverse the same link and their corresponding working lightpaths are SRLG-joint.

Viewing each lightpath as a node in a graph and each wavelength as a color, two nodes (lightpaths) are defined to be adjacent if they must be assigned different wavelengths. Then, the above problem becomes the well-known \mathcal{NP} -hard graph coloring problem. We use a first-fit heuristic for this problem. We assume in the heuristic that the number of the wavelengths available are infinite,

the wavelengths are indexed by 1, 2, ... The heuristic works as follows:

- Step 1 Set $P = BP \cup WP$. Remove all the empty paths from P .
 Step 2 Assign wavelength.
 Step 2.1 Remove a lightpath p from P .
 Step 2.2 Assign the allowable wavelength with the smallest index to p . A wavelength is allowable if assigning it to p does not violate **CW1–CW3**.
 Step 3 If $P = \emptyset$, stop. Otherwise, go to Step 1.

There are several ways in Step 1 to select from P a lightpath to remove. In our implementation, we randomly pick a lightpath from P in Step 1. WP and BP are inputs to the wavelength assignment heuristic while the output is a wavelength assignment to each lightpath in $P = BP \cup WP$, $a : P \rightarrow \{1, 2, \dots\}$.

Remark 1. The wavelength assignment a generated in the working path procedure will satisfy the constraints **CW1–CW3**. However, if the number of wavelengths used in a is more than W , it may violate the constraint **C1** (i.e., the number of wavelengths used on some links may be more than W). This is prohibited in the following search procedure, we will set the network cost of these violated solutions as ∞ .

Example: Fig. 3(c) shows the wavelength assignment on these links used for the working and protection paths. In the figure, the numbers in each link is of the ' a/b ' form, where ' a ' is the number of wavelengths used for the working path, and ' b ' is the number for the protection path.

4.4. Brief summary of the greedy heuristic

According to the previous subsections, the proposed greedy heuristic for the routing and wavelength assignment can be summarized as follows.

- For each connection request $r \in R$ with source node s and destination node d , Yen's algorithm is applied to find all the paths from s to d , denote the paths as $PathList_r$. We associate each path in the list an integer number from 1 to u_i .
- To route the working paths for the connection requests, we generate a vector of M integers $\mathbf{x} = (x_1, \dots, x_M)$, while x_i variable is randomly generated from $\{1, \dots, u_i\}$. The solution \mathbf{x} can be easily interpreted as a set of working paths for the connection requests.

- The protection path routing for a connection request is carried out by using the Dijkstra algorithm to find the shortest path of a graph induced from the original graph by discarding the prohibited links as defined in Section 4.1. The routing order is a permutation $\sigma = (\sigma_1, \dots, \sigma_M)$ of the connection requests.
- If there are some protection paths that are \emptyset , the repair heuristic is applied to make the routing feasible.
- If all the working and protection paths are feasible, the wavelengths can be assigned on the used links as described in Section 4.3.

It can be seen that the routing is solely determined by a categorical vector \mathbf{x} and a permutation σ . After the routing and wavelength assignment procedure, if the routings are successful, the number of wavelengths assigned on the links for the working paths and protection paths will be returned. The cost of the routed network, or the objective value of the decision variables (\mathbf{x}, σ) , can be computed according to Eq. (1).

Any solutions generated by GH will satisfy constraints **C2–C5**. An inappropriate setting of \mathbf{x} and σ may generate a solution which is incomplete (there are no backup lightpaths for some requests) or/and violates constraint **C1**. In these cases, the network cost of the parameter settings is set to be ∞ .

It is not easy to develop a simple mathematical model for the relationship between the algorithmic parameter settings and the quality of the solution obtained. Finding an optimal parameter settings turns out to be a black box optimization problem. In this paper, we propose to use estimation of distribution algorithm (EDA) as a tool for tuning the algorithmic parameters. The proposed EDA and the whole algorithm will be described in the following.

5. The algorithm

5.1. Estimation of distribution algorithm

EDAs are a type of model-based evolutionary algorithm (EA). Like general EAs [13], EDAs also maintain and evolve a population of solutions until stop criteria are met. Different from general EAs, EDAs generate offspring by sampling from a previously constructed probabilistic model instead of by applying the genetic operators such as crossover and mutation operators. A number of EDAs have been proposed both for global continuous optimization problems and combinatorial optimization problems and achieved great success [14]. The framework of the EDA can be summarized in Alg. EDA.

Algorithm EDA(Θ_1)

Input: parameters Θ_1 .

Output: The best solution found \mathbf{x}^* .

1. **Parameter Settings.** Set appropriate algorithmic parameters, such as population size, selection size, initial probability model $\mathcal{P}(0)$, etc.
2. **while** computational budget has not been exceeded,
3. **Initialization.** Set $t := 0$. Generate the initial population $S(0)$. Evaluate the fitness of the population.
 Selection. Select promising solutions from $S(t)$.
4. **Modeling.** Construct a probability model $\mathcal{P}(t)$ based on the selected solutions.
5. **Sampling.** Sample N offspring from $\mathcal{P}(t)$. Evaluate the sampled offspring.
6. **Replacement.** Partially or fully replace $S(t)$ by the sampled solutions to form $S(t+1)$. Set $t := t+1$.
7. **end while;**
8. Find the best solution \mathbf{x}^* with the lowest objective value from $S(t)$ and return it.

From the framework, one can see that the generation of new offspring is through *sampling* from a probability model which is constructed from some selected promising solutions. On the contrary, in traditional genetic algorithms (GAs), new offspring is created by using crossover and mutation operations. No explicit model is applied in traditional GAs. The advantage of EDA is that the evolutionary search is explicitly (rather than implicitly as in GAs) guided by the probability model. The most important components of the EDAs include the model construction, and the sampling procedure.

For the considered static RWA with SRLG-constraints, EDA works on top of the proposed greedy heuristic to tune the control parameters in respective two stages. In the following, we first describe the main components, including initialization, selection, probability model construction and new offspring sampling, of the EDA with category (for working path routing) and permutation (for protection path routing) representations. Then the whole two-stage evolutionary approach for the optimization of the RWA is proposed.

5.2. Initialization of population and probability model

To apply EDA to an optimization problem with permutation representation, we use a probability matrix $\mathcal{P} = (p_{ij})$, $1 \leq i, j \leq M$ to represent the probability model, where p_{ij} denotes the probability of assigning element i to location j . It is initialized to be

$$\mathcal{P}_0 = \begin{pmatrix} \frac{1}{n} & \cdots & \frac{1}{n} \\ \vdots & \ddots & \vdots \\ \frac{1}{n} & \cdots & \frac{1}{n} \end{pmatrix} \quad (2)$$

The search space is the set of all possible permutations Π . EDA randomly chooses N permutations from Π to constitute the initial population $S(0)$.

To apply EDA to problem with a categorical representation, a probability matrix $\mathcal{Q} = (q_{ij})$ is adopted, where q_{ij} represents the probability of using the candidate path j to route the i th connection request. It is initialized to be

$$\mathcal{Q}_0 = \begin{pmatrix} \frac{1}{u_1} & \cdots & \frac{1}{u_M} \\ \vdots & \ddots & \vdots \\ \frac{1}{u_1} & \cdots & \frac{1}{u_M} \end{pmatrix} \quad (3)$$

The search space is the set of all the combinations of the constructed candidate paths.

To initialize the population for problems with categorical representation, we could generate a set of random solutions. In this paper, we initialize the population based on the principles mentioned in Section 4 for routing the working paths. Specifically, each combination is constructed as follows. The selection of potential working paths for the connection requests follows a random order $\pi = (\pi_1, \dots, \pi_M)$ of the connection requests. Suppose that for the connection requests π_1, \dots, π_K ($1 \leq K < M$), a set of working paths $WP^K = \{wp_{\pi_1}, \dots, wp_{\pi_K}\}$ have been selected from the candidate paths (in the case of $K = 0$, a path is randomly selected from $PathList_{\pi_1}$). The working path $wp_{\pi_{K+1}}$ for connection request π_{K+1} is as follows:

Step 1 For each potential path $p_i \in PathList_{\pi_{K+1}}$, calculate its similarity degree as follows: first set the degree value s_i to zero, then for each link ℓ in p_i , if ℓ exists in any one of P_{π_k} , $1 \leq k \leq K$, set $s_i := s_i + 1$;

Step 2 Select the path with the minimal similarity degree, say $p_{\min} := \operatorname{argmin}_{1 \leq i \leq u_{K+1}} s_i$. $P := P \cup p_{\min}$;

The similarity degree of a path p to a path set P in Step 1 is computed as the sum of the number of links in p which are the same as to the links in the path set P . To increase the SRLG-disjoint degree of the set of working paths, the working path routing procedure encourages the use of the candidate paths with small similarity degree to previously routed paths. The reason is that a path with small similarity degree is very likely to be SRLG-disjoint with the other paths. The working path routing procedure tries to establish a set of working paths which are as small similarity degree as possible.

The purpose of the construction is to make the initial population as informative as possible. The high-quality initial population can make the following search for optimal solutions more effective than random initial population.

5.3. Selection and update of probability matrix

Assume that the population at generation t is $S(t) = \{\pi^1, \pi^2, \dots, \pi^N\}$ with cost $C = \{c_1, c_2, \dots, c_N\}$ and the probability matrix at generation $t - 1$ is \mathcal{P}_{t-1} (or \mathcal{Q}_{t-1}). The best $N/2$ individuals are selected as $\mathcal{H}(t) = \{\sigma^1, \dots, \sigma^{N/2}\}$ for the aim of probability matrix construction. That is, the probability matrix $\mathcal{P}_t = (p_{ij}(t))$ (or \mathcal{Q}_t) is updated by PBIL [2] as follows:

$$p_{ij}(t) \text{ (or } q_{ij}(t)) = (1 - \beta) \cdot \frac{2}{N} \sum_{k=1}^{N/2} I_{ij}(\sigma^k) + \beta \cdot p_{ij}(t-1) \text{ (or } q_{ij}(t-1)),$$

$$1 \leq i, j \leq M. \quad (4)$$

where $I_{ij}(\sigma^k)$ is the indicator function and defined as follows:

$$I_{ij}(\sigma^k) = \begin{cases} 1, & \text{if } \sigma^k(i) = j; \\ 0, & \text{otherwise;} \end{cases}$$

and $0 \leq \beta \leq 1$ is the learning rate, which controls the contribution of the present promising solutions to the probability matrix $\mathcal{P}(t)$ (or $\mathcal{Q}(t)$). The larger β value indicates more contribution of the information learned from the history.

5.4. Sampling and replacement

An offspring π is sampling from the probability matrix $\mathcal{P}(t)$ (or $\mathcal{Q}(t)$) at generation t iteratively. Here we use the guided mutation operator [49] to sample new offspring. The guided mutation constructs a new offspring based on the probability matrix and the best solution found so far. Its underlying assumption is the so-called proximate optimality principle (POP), which is proposed by [12]. The POP states that good solutions have similar structure. It is widely adopted, implicitly or explicitly, in most, if not all, meta-heuristics. Previous study on the guided mutation [48,49] has shown its superiority over the other recombination operators.

The procedure to sample a permutation π by guided mutation can be described as follows. Suppose that the best solution found so far is \mathbf{x} , and current probability matrix is \mathcal{P} . To sample a new offspring, first a set $I = \{i_1, \dots, i_s\}$, where $s = \lceil \alpha M \rceil$ is randomly selected, α is a parameter of the guided mutation and $\lceil x \rceil$ rounds the element of x to the nearest integer towards infinity. The elements for the new offspring σ in I are copied from \mathbf{x} , that is, $\sigma_i = \mathbf{x}_i$, $i \in I$. The selection for elements in the rest unfilled components, i.e. $\{1, 2, \dots, M\} \setminus I$ is proportional to the probability matrix \mathcal{P} .

Specifically, the following procedure can be used to select an element for a location k in $\{1, 2, \dots, M\} \setminus I$. To begin with, we denote

$U = I$ the already-assigned components, and $V = \{\mathbf{x}_{i_1}, \dots, \mathbf{x}_{i_s}\}$ the already-used elements.

- Step 1 Reset the k th column of the probability matrix $\mathcal{P}(t)$ by setting $p_{i,k}(t)$, $i \in V$ as zero, since these elements are forbidden to be σ_k for constructing a feasible permutation.
- Step 2 Sum the k th column of the probability matrix $\mathcal{P}(t)$,

$$h = \sum_{i=1}^M p_{i,k}$$

$$\text{set } \mathbf{h}_k(t) = \left(\frac{p_{11}(t)}{h}, \dots, \frac{p_{iM}(t)}{h} \right)^T;$$

The solution construction procedure terminates when all unfilled components are filled.

The replacement is performed after sampling. To perform replacement, the generated offspring are partially replaced with solutions in the current population. The best N individuals are selected from the combined set of the current population and the sampled offspring. Note that in order to preserve the diversity of the population, duplicates are not allowed in the new population.

5.5. Structure of the EDA with guided mutation

The following Alg. *EDAP* (*EDAC*) summarizes the algorithmic framework of the proposed EDA with guided mutation for permutation-based solution representation, called *EDAP*, and category-based solution representation (*EDAC*).

Algorithm *EDAP*(*EDAC*)(N, α, β)

Input: population size N , the control parameter in the guided mutation operator α and the learning rate β .

Output: The best solution found \mathbf{x}^* .

1. **while** computational budget has not been exceeded,
2. **Initialization.** Set $t := 0$. Generate the initial population $S(0)$ for *EDAP* randomly, and for *EDAC* heuristically (cf. subsection 5.2). Evaluate the fitness of the population by the greedy heuristic. Set the initial probability matrix $\mathcal{P}(0)$ using Eq. 2 for *EDAP* and Eq. 3 for *EDAC*.
3. **Selection.** Select half the best solutions in $S(t)$.
4. **Modeling.** Construct $\mathcal{P}(t)$, the probability model, using Eq. (4) based on the selected solutions.
5. **Sampling.** Sample N offspring $\{v^1, \dots, v^N\}$ by guided mutation as described in Section 5.4. Evaluate the sampled offspring by the greedy heuristic.
6. **Replacement.** Choose the best N solutions from $\{v^1, \dots, v^M\} \cup S(t)$ to form $S(t+1)$ without duplication. Set $t := t+1$.
7. **end while;**
8. Return the best solution \mathbf{x}^* found in $S(t)$.

Step 3 Select an element, e.g. v , by using the roulette wheel method based on $\mathbf{h}_k(t)$, set $\sigma_k := v$.

Step 4 Set $U = U \cup \{k\}$ and $V = V \cup \{v\}$.

The iteration continues until all components in $\{1, 2, \dots, M\} \setminus I$ are filled, i.e. $U = \{1, 2, \dots, M\}$ or $V = \emptyset$. In our algorithm, we sample N offsprings from the probability matrix $\mathcal{P}(t)$.

To sample a category solution $\mathbf{y} = (y_1, \dots, y_M)$ by using the guided mutation based on the probability matrix $\mathcal{Q}(t)$ and the best solution found so far \mathbf{x} . The same as the above sampling of permutation, we select a set of indices I in which the elements in those locations are copied to \mathbf{y} , that is $y_i = x_i$, $i \in I$. The rest of the components are filled according to $\mathcal{Q}(t)$. Specifically, to fill the j th component, $j \in \{1, 2, \dots, M\} \setminus I$, the following procedure is given:

Step 1 Normalize the probability element q_{ij} , $1 \leq i \leq u_j$, that is,

$$q_{ij} = \frac{q_{ij}}{u_j} \cdot \sum_{k=1}^{u_j} q_{kj}$$

Step 2 Select an element, say v , by roulette wheel method based on the normalized q_{ij} , $1 \leq i \leq u_j$. Set $y_j = v$.

In the paper, the above EDA with guided mutation terminates when the algorithms cannot find a better solution within 30 generations, or the execution time excesses over a given time limit.

5.6. Tuning the control parameters of the greedy heuristic

The proposed procedure for tuning these parameters works in the following with *EDAP* (*EDAC*) as input.

Step 1 *Tuning x for the working path routing procedure.*

Step 1.1 Randomly generate a permutation $\tilde{\sigma}$ as the input parameter for the protection path routing procedure.

Step 1.2 Use *EDAC*(N, α_1, β_1) to tune the working path routing, where the cost of a working path routing vector \mathbf{x} is set as $f(\mathbf{x}, \tilde{\sigma})$. Set \mathbf{x}^* to be the found best working path routing vector.

Step 2 *Tuning σ for the protection path routing procedure.*

Step 2.1 Use *EDAP* (α_2, β_2) to tune σ where the cost of σ is set as $f(\mathbf{x}^*, \sigma)$. Set σ^* to be the best setting of σ found.

Return the best solution found to the static RWAP under SRLG-constraints is the solution generated by GH with parameter setting (\mathbf{x}^*, σ^*) .

Table 1
The characteristics of test network instances.

Settings	Nodes	Links	SRLGs	Requests	Wavelengths
1	19	62	31	100	32
2	19	62	29	100	32
3	19	62	27	100	32
4	24	86	43	100	32
5	24	86	41	100	32
6	24	86	40	100	32
7	24	86	39	100	32
8	31	94	47	150	64
9	50	200	85	250	64
10	50	200	85	275	64
11	50	200	85	300	64
12	60	200	90	250	64
13	65	200	100	250	64
14	65	200	100	300	64
15	100	340	150	350	64

6. Computational results

6.1. Test problems

The network topology used is similar to the network topology in [46]. Parameters of a certain network topology include the number of nodes $n = |V|$, the number of SRLGs $S = |G|$, the number of links $L = |E|$, the maximum number of wavelength W and the number of demands to be routed $M = |R|$.

Two kinds of network topology T_1 and T_2 are tested in the paper. In T_1 , links in a same SRLG have the same input or output node. In T_2 , links in a SRLG are set randomly. Parameters of the used test network topologies are listed in Table 1. The first eight networks have the same parameter settings as used in [46]. In the following simulation, 9 networks from each network topology are taken as examples. The example instances are named with its topology name plus an index. For example, $T_1 - 01$ means the instance is sampled from topology T_1 with parameters from *Network1*.

Table 2
The comparison of EDA/GH, BH/EA/G and R/GH on test network instances.

Instances	EDA/GH				Time	BH/EA/G				Time	R/GH				t-test	
	Worst	Best	Avg.	Std.		Worst	Best	Avg.	Std.		Worst	Best	Avg.	Std.	t 1	t 2
$T_1 - 01$	363	348	353.7	4.64	0.9	381	365	372.5	10.3	564	538	556.4	7.26	1.5	5.22	74.39
$T_1 - 02$	371	359	365.1	3.31	0.8	381	368	374.5	9.7	588	566	575.3	6.67	1.3	2.90	89.26
$T_1 - 03$	358	348	353.9	2.84	1.0	378	364	368.7	8.6	521	493	503.80	8.40	2.1	5.17	53.45
$T_1 - 04$	392	384	386.9	2.28	1.3	390	377	384.6	17.8	707	688	696.8	6.30	2.3	0.41	146.29
$T_1 - 05$	398	383	387.7	5.51	2.1	401	384	390.7	10.4	619	594	605.6	8.38	4.9	0.81	68.70
$T_1 - 06$	390	378	384.6	3.89	2.2	391	377	384.9	15.9	664	631	650.10	9.06	4.8	0.06	85.15
$T_1 - 07$	392	384	386.9	2.28	3.0	425	384	394.0	11.1	707	688	696.80	6.30	5.7	2.98	146.27
$T_1 - 08$	670	646	655.9	7.49	5.3	651	636	649.2	10.7	1101	1070	1083.1	13.67	14.7	2.62	86.67
$T_1 - 09$	1210	1195	1202.7	5.33	12.7	1249	1221	1239.1	20.1	1739	1700	1727.60	12.08	24.6	5.53	125.71
$T_1 - 10$	1358	1339	1346.6	5.39	12.9	1381	1348	1372.5	19.7	1885	1845	1870.0	12.75	25.0	4.01	119.57
$T_1 - 11$	1397	1385	1390.6	4.14	13.0	1445	1426	1439.2	20.6	2010	1960	1984.30	19.21	25.6	7.31	95.54
$T_1 - 12$	1404	1381	1391.2	7.78	23.4	1430	1414	1424.8	18.5	2163	2113	2145.3	15.41	47.2	5.29	138.15
$T_1 - 13$	1497	1471	1488.9	7.57	34.2	1560	1526	1551.6	23.8	2428	2379	2406.40	18.88	50.3	7.94	142.64
$T_1 - 14$	1756	1730	1743.1	8.32	34.8	1836	1809	1825.9	10.7	2811	2731	2783.3	22.97	52.3	19.32	134.64
$T_1 - 15$	2177	2140	2153.9	10.58	55.7	2287	2195	2215.0	20.1	3436	3354	3403.6	30.61	128.1	8.51	122.02
$T_2 - 01$	357	349	354.2	3.81	0.8	371	355	364.7	11.7	549	510	537.60	12.21	1.4	2.70	45.34
$T_2 - 02$	364	348	356.0	4.49	1.2	371	355	363.4	9.5	570	543	556.60	6.98	2.1	3.22	76.43
$T_2 - 03$	374	364	369.5	3.53	1.3	409	399	404.5	8.1	546	480	530.80	19.09	2.0	12.52	26.27
$T_2 - 04$	421	405	414.9	5.02	2.3	412	399	406.3	11.5	673	646	660.60	7.83	5.6	2.67	83.53
$T_2 - 05$	381	369	374.9	4.14	3.1	381	366	376.3	9.1	559	538	551.5	6.24	5.3	0.44	74.57
$T_2 - 06$	404	387	396.4	4.45	3.0	395	380	388.3	9.3	628	598	611.90	8.29	5.1	2.48	72.43
$T_2 - 07$	415	399	407.3	5.50	2.7	438	419	428.9	8.5	648	609	624.90	12.32	4.8	6.75	51.00
$T_2 - 08$	680	661	670.9	6.10	5.8	730	715	727.8	9.7	1018	962	1000.40	20.94	12.1	15.70	47.77
$T_2 - 09$	1195	1164	1178.3	9.50	14.3	1209	1192	1203.4	8.6	2059	1990	2023.40	20.65	20.0	6.19	117.57
$T_2 - 10$	1298	1256	1275.1	12.77	14.0	1234	1213	1225.7	12.6	1885	1845	1870.00	12.55	20.3	8.70	105.07
$T_2 - 11$	1435	1405	1422.5	10.69	15.1	1409	1397	1401.8	9.1	2512	2443	2452.00	26.25	23.0	4.66	114.86
$T_2 - 12$	1430	1410	1419.9	7.03	20.1	1487	1451	1476.9	15.9	2195	2114	2160.00	24.48	34.0	10.37	91.89
$T_2 - 13$	1508	1489	1500.4	5.05	30.0	1590	1567	1582.7	21.5	2483	2433	2452.00	14.96	45.7	11.78	190.58
$T_2 - 14$	1804	1771	1790.6	8.99	33.1	1910	1866	1889.7	10.5	3080	3019	3051.00	18.41	53.2	22.67	194.54
$T_2 - 15$	2165	2126	2142.4	10.29	50.9	2207	2155	2183.3	9.81	3454	3334	3387.40	30.89	150.4	9.10	120.92

Note that in a generated test networks, there maybe no two link- and SRLG-disjoint paths between some source–destination pairs. In order to avoid such cases, the source–destination pairs are selected from all of the source–destination pairs who have at least two link- and SRLG-disjoint paths. For fair comparison, firstly the heuristic algorithm proposed in [46] is performed on a certain test network, then M source–destination pairs which can be routed are picked and used as the test network for the new proposed algorithms.

6.2. Comparison results

To test the performance of the proposed algorithm, three algorithms are employed including a random algorithm, the EDA-based approach proposed in [49] and the proposed algorithm. The resultant algorithms are called R/GH, BH/EA/G and EDA/GH, respectively.

R/GA searches the search space by running GH on randomly generated control parameters. In the proposed EDA/GH, we need to set the algorithmic parameters, including the population size, the sampling population size, the guided mutation parameter α , and probability updating parameter β . In all the experiments, the population size is set to 100, the size of selected solutions is 50, and the size of the sampled solutions is 100.

To decide the proper parameters for α and β , we used $T_1 - 01$ as an example instance. For each algorithmic parameter combination, i.e. (α_1, β_1) and (α_2, β_2) , the proposed algorithm was carried out 10 times on $T_1 - 01$, where $\alpha_1, \alpha_2 = 0.0, 0.1, 0.2, \dots, 0.9$ and $\beta_1, \beta_2 = 0.0, 0.1, 0.2, \dots, 0.9$. The combination of the algorithmic parameters with the best average fitness value is picked as the proper parameter for the proposed algorithm. As a result, the parameter settings for EDA/GH are $\alpha_1 = 0.2, \beta_1 = 0.2, \alpha_2 = 0.3, \beta_2 = 0.2$ though these settings may be not suitable for the other network instances.

To fairly compare the mentioned three algorithms, we carried out them 30 times with their proper parameter for each test network instance within the given time limit T . The parameter setting

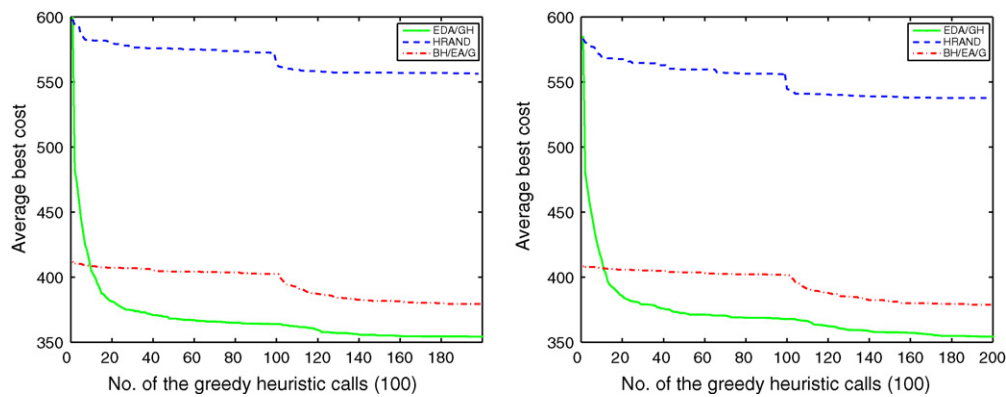


Fig. 4. The evolution procedures of the compared algorithms.

for EDA/GH is copied from [49]. The time limit T for the instances from *Network1* to *Network7* is 120 s, while for *Network8* is 300 s, 1200 s for *Network9–Network11*, 3600 for *Network12–14* and 7200 for *Network15*. Table 2 lists the comparison results among the three algorithms on the test network instances with topologies T_1 and T_2 .

In Table 2, “worst”, “best”, “avg.” and “std.” columns list the average worst, best, average network costs, and the standard deviation over the 30 runs. In the table, the values in “ t -test” column show the t -values between the results of EDA/GH and BH/EA/G (t_2), EDA/GH and R/GH (t_1) by using two-tailed t -test. The null hypothesis used here is that the average cost obtained by BH/EA/G is the same as that obtained by EDA/GH. The values $t_1 < 2.6$ suggests that EDA/GH can find better network routing and wavelength assignment schemes than BH/EA/G. From the tables, we can see that in 26 out of 30 test problems, the solution quality found by the EDA/GH is significantly better than that of BH/EA/G. The t -test between EDA/GH and R/GH indicates that the proposed algorithm performs better than the random algorithm on all the test problems in terms of solution quality.

In Fig. 4, on the x-axis is the number of greedy function calls, on the y-axis is the average cost of the best solutions found over 30 runs. From the figure, we can see that the average quality of the initial population in BH/GA/G is superior to that in EDA/GH. However, the proposed algorithm can find high quality solutions very quick. One possible explanation to this is that the search space of EDA/GH in stage 1 is not as rugged as that of BH/GA/G which is rooted from the different solution representations.

The average times (in seconds) to evaluate per 100 solutions by using EDA/GH and BH/EA/G are shown in columns ‘time’ of Table 2. From the table, one can see that the time used for constructing a solution by the proposed algorithm is at least 1.5 time less than the time used by BH/EA/G. Note that in the heuristic developed in [49], the Dijkstra algorithm has to be applied to route the working path for each connection request. Since it is obvious that the working paths routed for the connection requests can only fall into the candidate path set *PathList*, it is therefore a waste of time to route the connection request at each routing procedure. The experimental results shown in Table 2 also indicate that the solution quality found by the proposed algorithm compensates the overhead of constructing the shortest paths.

From the comparison, we can draw the following conclusions:

1. The usage of the population-based algorithm is beneficial to the search since the population-based algorithm (EDA/GH) clearly outperforms the random algorithm R/GH.
2. The selection of proper solution representation has significant influence to the performance of the population-based algorithm. Since the solution representation reflects the fitness landscape of the optimization problem, a good solution representation will

make the evolutionary search much easier than a bad solution representation.

7. Conclusion

In the paper, a two-stage EDA-based approach was proposed for the static routing and wavelength assignment problem (RWAP) for all optical WDM mesh network survivability under SRLG constraints. A greedy heuristic (GH) with two control parameters was proposed to construct a feasible solution of the considered problem. The estimation of distribution algorithm with guided mutation works on top of the heuristic to find optimal control parameter settings of the heuristic in respective two stages.

The proposed algorithm, called EDA/GH, was empirically compared with the best-known EA-based algorithm for the static RWAP under SRLG constraints, and a pure random algorithm based on the proposed greedy heuristic on a set of 30 test network instances within given time limit. The comparison results favor the proposed algorithm against the best-known EA-based algorithm in 26 out of 30 test network instances, and against the random algorithm in all the instances in terms of solution quality within a given time limit.

The comparison results also indicate that the search for optimal solutions benefits from the population approach. The experiments also show the solution representation has a significant effect to the performance of population-based approach.

Acknowledgements

JS is funded by BBSRC/EPSRC grant BB/D019613/1, UK. The author also would like to thank anonymous reviewers for their constructive comments and suggestions.

References

- [1] M. Alanyali, E. Ayanoglu, Provisioning algorithms for wdm optical networks, *IEEE/ACM Transactions on Networking* 7 (1999) 767–788.
- [2] S. Baluja, Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning, Technical Report CMU-CS-94-163, Carnegie Mellon University, 1994.
- [3] B. Bánhegyi, M. Biazini, A. Montessor, M. Jelasity, Peer-to-peer optimization in large unreliable networks with branch-and-bound and particle swarms, in: *EvoWorkshops*, 2009, pp. 87–92.
- [4] P.C. Berbert, L.J.R.F. Filho, T.A. Almeida, M.B. Carvalho, A. Yamakami, Artificial immune system to find a set of k -spanning trees with low costs and distinct topologies, in: L.N. de Castro, F.J. Von Zuben, H. Knidel (Eds.), *ICRIS 2007*, vol. 4628 of LNCS, 2007, pp. 395–406.
- [5] D.A.R. Chaves, D.O. Aguiar, C.J.A. Bastos-Filho, J.F. Martins-Filho, Novel adaptive routing algorithm for all-optical networks based on power series and particle swarm optimization, in: *BDBComp*, 2009, pp. 609–622.
- [6] D.W. Corne, M.J. Oates, G.D. Smith (Eds.), *Telecommunication Optimization: Heuristic and Adaptive Techniques*, John Wiley & Sons, 2000.
- [7] P. Cortés, J.M. García, J. Munizuri, L. Onieva, Viral systems: a new bio-inspired optimisation approach, *Computers and Operations Research* 35 (9) (2008) 2840–2860.

- [8] M. Dorigo, T. Stützle, The ant colony optimization metaheuristic: algorithms, applications, and advances, in: *Handbook of Metaheuristics*, Kluwer Academic Publisher, Boston, MA, 2003, pp. 251–285.
- [9] R.C. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: *Proceedings of the Sixth International Symposium on Micromachine and Human Science*, Nagoya, Japan, 1995, pp. 39–43.
- [10] J.D. Farmer, N. Packard, A. Perelson, The immune system, adaptation and machine learning, *Physica D* 22 (1986) 187–204.
- [11] D.B. Fogel (Ed.), *Evolutionary Computation: The Fossil Record*, Wiley-IEEE Press, May 1999.
- [12] F. Glover, M. Laguna, *Tabu Search*, Kluwer, 1997.
- [13] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA, 1989.
- [14] W.E. Hart, N. Krasnogor, J.E. Smith (Eds.), *Recent Advances in Memetic Algorithms*, vol. 166, *Studies in Fuzziness and Soft Computing*, Springer, 2005.
- [15] A. Hassan, C. Phillips, Dynamic routing and wavelength assignment using hybrid particle swarm optimization for wdm networks, in: *EPSRC PostGraduate Network Symposium (PGNet)*, 2007.
- [16] A. Hassan, C. Phillips, Chaotic particle swarm optimization for dynamic routing and wavelength assignment in all-optical wdm networks, in: *The 3rd IEEE International Conference on Signal Processing and Communication Systems (ICSPCS 2009)*, Nabraska, 2009.
- [17] A. Hassan, C. Phillips, Improved pso-based static rwa solver avoiding premature convergence, in: *LCS 2009*, University College London, 2009.
- [18] C. Hassan, Phillips, Particle swarm optimization based DRWA for wave-length continuous WDM optical networks using a novel fitness function, *Artificial Intelligence Review Journal* 29 (3–4) (2008) 305–319.
- [19] J.Q. Hu, Diverse routing in optical mesh networks, *IEEE Transactions on Communications* 51 (3) (2003) 489–494.
- [20] X. Hu, J. Zhang, L.-M. Zhang, Swarm intelligence inspired multicast routing: an ant colony optimization approach, in: *EvoWorkshops*, 2009, pp. 51–60.
- [21] X. Jin, L. Bai, Y. Ji, Y. Sun, Probability convergence based particle swarm optimization for multiple constrained qos multicast routing, in: *2008 Fourth International Conference on Semantics, Knowledge and Grid*, 2008, pp. 412–415.
- [22] H. Keko, M. Skok, D. Skrllec, Artificial immune systems in solving routing problems, in: *EUROCON 2003*, 2003, pp. 62–66.
- [23] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: *Proceedings of IEEE International Conference on Neural Networks*, Pisa-taway, NJ, 1995, pp. 1942–1948.
- [24] P. Larrañaga, J.A. Lozano, *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*, Kluwer Academic Publishers, 2002.
- [25] H.Y.K. Lau, E.Y.C. Wong, An ais-based dynamic routing (aisdr) framework, in: C. Jacob, et al. (Eds.), *Proceeding of the 4th International Conference on Artificial Immune Systems*, vol. 3627 of *LNCIS*, 2005, pp. 56–71.
- [26] S. Lee, C. Kim, D. Griffith, Hierarchical restoration scheme for multiple failures in GMPLS networks, in: *Proceedings of IEEE International Conference on Parallel Processing Workshops*, 2002, pp. 177–182.
- [27] G.Z. Marković, D.B. Teodorović, V.S. Ćimović-Raspopović, Routing and wavelength assignment in all-optical networks based on the bee colony optimization, *AI Communications* 20 (4) (2007) 273–285.
- [28] M. Médard, S.G. Finn, R.A. Barry, Redundant trees for preplanned recovery in arbitrary vertex-redundant or edge-redundant graphs, *IEEE/ACM Transactions on Networking* 7 (5) (1999) 641–652.
- [29] G. Mohan, C. Siva Ram Murthy, A.K. Somani, Efficient algorithms for routing dependable connections in wdm optical networks, *IEEE/ACM Transactions on Networking* 9 (5) (2001) 553–566.
- [30] B. Mukherjee, *Optical Communication Networks*, McGraw-Hill, 1997.
- [31] S. Nakrani, C. Tovey, On honey bees and dynamic server allocation in internet hosting centers, *Adaptive Behaviour* 12 (3–4) (2004) 223–240.
- [32] E. Oki, N. Matsuura, K. Shiimoto, N. Yamanaka, A disjoint path selection scheme with shared risk link groups in gmpls networks, *IEEE Communication Letter* 6 (2002) 406–408.
- [33] C. Ou, B. Mukherjee (Ed.), *Survivable Optical WDM Networks*, Optical Networks, Springer, 2005.
- [34] H. Pan, I.Y. Wang, The bandwidth allocation of ATM through genetic algorithm, in: *Proceedings IEEE Global Telecommunication Conference (GLOBEALCOM'91)*, 1991, pp. 125–129.
- [35] S. Ramamurthy, B. Mukherjee, Survivable wdm mesh networks, *Journal of Lightwave Technology* 21 (4) (2003) 870–883.
- [36] T.M. Rao, V. Anand, Particle swarm optimization for routing and wave-length assignment in optical networks, in: *Sarnoff Symposium*, 2006 IEEE, 2006, pp. 1–4.
- [37] M.G.C. Resende, P.M. Pardalos (Eds.), *Handbook of Optimization in Telecommunications*, Springer Science & Business Media, 2006.
- [38] S.H. Shami, I.M.A. Kirkwood, M.C. Sinclair, Evolving simple fault-tolerant routing rules using genetic programming, *Electronics Letters* 33 (17) (1997) 1440–1441.
- [39] L. Shen, X. Yang, B. Ramamurthy, Shared risk link group (SRLG)-diverse path provisioning under hybrid service level agreements in wavelength-routed optical mesh networks: formulation and solution approaches, *IEEE/ACM Transactions on Networking (TON)* 13 (4) (2005) 918–931.
- [40] M.C. Sinclair, Minimum cost routing and wavelength allocation using a genetic algorithm/heuristic hybrid approach, in: *Proceedings of the 6th IEEE Conference on Telecommunications*, Edinburgh, 1998, pp. 67–71.
- [41] G. Xiao, X. Pan, Heuristic for the maximum disjoint paths problem in wavelength-routed networks with shared-risk link groups [invited], *Journal of Optical Networking* 3 (2003) 38–49.
- [42] D. Xu, Y. Xiong, C. Qiao, A new PROMISE algorithm in networks with shared risk link groups, in: *Proceedings of IEEE Global Communications Conference (GlobeCom 2003)*, IEEE, New York, 2003.
- [43] D. Xu, Y. Xiong, C. Qiao, G. Li, Trap avoidance and backup schemes in networks with shared risk link groups, *Journal of Lightwave Technology*, Special issue on Optical Network 21 (2003) 1320–1331.
- [44] J.Y. Yen, Finding the k shortest loopless paths in a network, *Management Science* (1971) 712–716.
- [45] H. Zang, J.P. Jue, B. Mukherjee, A review of routing and wavelength assignment approaches for wavelength-routed optical wdm networks, *Optical Networks Magazine* (2000) 47–60.
- [46] H. Zang, B. Mukherjee, Path-protection routing and wavelength-assignment (RWA) in WDM mesh networks under duct-layer constraints, *IEEE/ACM Transactions on Networking* 11 (April (2)) (2003) 248–258.
- [47] Z.-H. Zhan, J. Zhang, Discrete particle swarm optimization for multiple destination routing problems, in: *EvoWorkshops*, 2009, pp. 117–122.
- [48] Q. Zhang, J. Sun, E.P.K. Tsang, Evolutionary algorithm with the guided mutation for the maximum clique problem, *IEEE Transactions on Evolutionary Computation* 9 (2) (2005) 192–200.
- [49] Q. Zhang, J. Sun, G. Xiao, E. Tsang, Evolutionary algorithms refining a heuristic: hyper-heuristic for shared-path protections in wdm networks under srlg constraints, *IEEE Transactions on Systems, Man and Cybernetics, Part B* 37 (1) (2007) 51–61.
- [50] B. Zhou, H.T. Mouftah, Spare capacity planning using survivable alternate routing for long-haul WDM networks, in: *Proceedings of the Seventh International Symposium on Computers and Communications*, IEEE, New York, 2000, pp. 732–738.