Research Article

Genetic Algorithm for Finding Minimal Cost Light Forest of Multicast Routing on WDM Networks

Der-Rong Din

Department of Computer Science and Information Engineering, National Changhua University of Education,
Changhua City 500, Taiwan

Correspondence should be addressed to Der-Rong Din, deron@ms45.hinet.net

Received 12 April 2007; Revised 6 September 2007; Accepted 13 January 2008

Recommended by Shengxiang Yang

Wavelength division multiplexing (WDM) is an important technique to make use of the large amount of bandwidth in optical fibers to meet the bandwidth requirements of applications. Multicast is the transmission of information from one source to multiple destinations simultaneously. Given a multicast request in a WDM network, the goal is to find a set of light trees, the assigned wavelengths of light trees, and construct a *light forest*. In this paper, the *minimal cost multicast routing problem (MCMRP)* on WDM networks with *tap-and-continue (TaC)* nodes is defined and studied. A new cost model which consists of the wavelength usage and communication cost is defined. The objective is to minimize the sum of the cost of used wavelengths and the communication cost of the *light forest*. Specifically, the formulation for the WDM multicast routing problem is given. Because the MCMRP is NP-hard, two *genetic algorithms (GAs)* are proposed to solve this problem. In the proposed GAs, a *path-oriented encoding* chromosome is used to represent the routing paths. These routing paths are used to construct source-based light forests to represent a feasible solution to the multicast request. Moreover, to speed up the convergence of GAs, a *farthest-first* greedy heuristic algorithm is proposed and used to generate one of the initial chromosomes. Simulation results demonstrate that the proposed GAs can run efficiently.

Copyright © 2008 Der-Rong Din. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

1.1. WDM

Wavelength division multiplexing (WDM) is an important technique to make use of the large amount of bandwidths in optical fibers to meet the bandwidth requirements of applications. There is a growing consensus that the next generation Internet will employ an IP-over-WDM architecture [1]. In this architecture, IP routers are attached to the optical cross-connects (OXC) and IP links are realized by *light paths* in a WDM optical network. Communication requests are transmitted by sending packets on a dedicated wavelength of the light path between the source node's transmitter and the destination node's receiver [2, 3]. Because there is no electrooptic (E/O) or optic-electronic (O/E) conversion in OXC, all-optical networks greatly increase the throughput capacity [4]. In addition, because the network requires not only transmission line capacity enhancement, but also cross-connect

node processing capability enhancement. The WDM should be used in combination with wavelength routing [5, 6].

In wavelength routing, data signals are carried on a unique wavelength from a source node to a destination node passing through nodes where the signals are optically routed and switched without regeneration in the electrical domain. When a physical network is given and connections among the nodes in the network are required, an optical path (*light path*) with a dedicated wavelength for each required connection should be established. The *routing and wavelength assignment (RWA)* problem is a problem to select suitable paths and wavelengths among the many possible choices for the required connections. To avoid collision, no two paths using the same wavelength pass through the same link called a *wavelength continuity constraint* [5]. By practical limitations on transmission technology, the number of available wavelengths on a fiber is restricted. So, a good solution to the RWA problem is important to increase the efficiency of the WDM network.

FIGURE 1: Architecture of a TaC node.

In order to solve various applications on WDM networks, mechanisms must be developed to handle not only point-to-point communications but also *multicast*. Multicast is the transmission of information from one source to multiple destinations simultaneously, for example, a one-to-many communication technique. Many broadband services such as video conferences, distance learning, and web casting employ multicast for data delivery. The support of multicast in future WDM networks is essential for these applications. Thus, issues concerning supporting multicast on WDM networks need to be studied.

1.2. Multicast

Recently, several researchers studied the multicast problem over WDM networks [7–12]. A comprehensive survey has been done by Ding and Poo [12]. Several researchers studied the multicast problem on WDM with all or sparse *multicast capable* (MC) [7, 8]. But the drawback of the problem studied in [7, 8] is that the MC node is expensive. To overcome this problem, Ali and Deogun have proposed a low-cost novel architecture called *Tap-and-Continue* (TaC), shown in Figure 1, for realizing multicast [9]. This architecture provides a natural evolution from current unicast cross-connects and is based on *tapping devices*. The proposed device can reduce the cost of MC cross-connects at the expense of more fiber links used in the routing structure. In the TaC cross-connect, optical signals are passed through a set of *Tap-and-Continue Modules* (TCMs). In a TCM, an extremely small

fraction of the input signal is tapped and forwarded to the local station. The remaining power on the order of 99.9% is switched to any one of the other outputs [9].

In [9], the route from source to destination is constructed as a *trial*, but this is impractical. For example, consider the network shown in Figure 2(a), where node s is the source and nodes d_1 , d_2 , and d_3 are the destinations. The trial $s \rightarrow d_2 \rightarrow d_3 \rightarrow d_2 \rightarrow d_1$ goes through d_3 by way of passing directed link (d_2, d_3) and immediately forwards the light back to d_2 through the reverse link (d_3, d_2) , which is impractical. This is because the delay between source and the farthest destination may be too large. In the WDM with TaC nodes, the trial can be replaced by a *light tree* which consists of light paths $s \rightarrow d_2 \rightarrow d_3$ and $s \rightarrow d_2 \rightarrow d_1$, as shown in the Figure 2(b). After this replacement, the delay between the source and the farthest destination can be reduced. But on the other hand, the wavelength usage is greater than the original trial. From this, there is a tradeoff between cost and wavelength usage.

In [10, 11], another version of multicast problems on the WDM network is studied. Nodes in the WDM network are further subdivided into four types [10, 11]: *Drop and Continue* node (DaC-node), *Wavelength Conversion* node (WC-node), *Light-Splitting* node (*split*-node), and *virtual source* (VS). A DaC-node, which is the same as the TaC node in [9], is capable of tapping a small amount of optical power from the wavelength channel and transmitting the remainder. The VS is capable of both *split*-node and WC-node. The VS plays a key role in the construction of a *multicast tree* [11].

```

BEGIN
  generate initial population;
  evaluate the fitness and penalty function of each individual in the population.
  WHILE (not terminated) DO
    select individuals form parent to reproduce;
    generate offspring through crossover and mutation;
    evaluate the individual fitness of the offspring;
    replace the worst ranked part of the population with offspring.
  END of WHILE
END

```

ALGORITHM 1: Genetic algorithm.

FIGURE 2: (a) Example of trial, (b) light trees.

In this paper, the multicast routing problem on WDM networks, whose nodes are TaCs, is studied. Given the WDM network and a multicast request, the goal is to find a *light forest* which consists of one or more light trees rooted at the multicast source and to their destinations, in such a way that the objective can be minimized. The usual goals to the multicast routing optimization problem [13] in traditional networks are the minimization of: (1) the path delay [14] due to blocking and (2) the cost of the path tree to reach the destinations. Often, the total cost minimization of the routing tree of networks with data replication is equivalent to the classical NP-Hard problem: the *minimum Steiner tree* [15]. Because both the routing paths and the wavelength assignments are considered in the multicast problem on WDM networks, the multicast routing optimization problem in general networks is a special case. This means that the multicast routing problem on WDM networks is a *NP-hard* problem.

1.3. Genetic algorithm

For a NP-hard problem, designing an algorithm to find an optimal solution is impractical, due to the exponential growth in execution time. *Genetic algorithms* (GAs) proposed by Holland [16] have been trusted as a class of general-purpose search strategies that strike a reasonable balance between exploration and exploitation. GAs have been constructed as robust stochastic search algorithms for various optimization problems. GAs are implemented as computer simulations to gain better solutions in a population of *chromosomes*

of candidate solutions to an optimization problem. The evolution usually starts from a population of randomly generated individuals and this occurs in generations. In each generation, the fitness of each individual in the population is evaluated; multiple individuals are stochastically selected from the current population, based on their fitness. And individuals are modified, recombined, and possibly randomly mutated to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either the maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. If the algorithm has terminated due to the maximum number of generations, a satisfactory solution may or may not have been reached. GAs find applications in computer science, engineering, and other fields [16]. The structure of the GA is shown in Algorithm 1.

GAs search by exploiting information sampled from different regions of the solution space. The combination of crossover and mutation helps GA escape from local optima. These properties of GA provide a good global search methodology for the multicast problem. In this paper, two GAs are proposed to find the light forest for the multicast request on WDM networks so as to minimize the objective cost. The performance of the proposed algorithms is evaluated in terms of the costs associated with the light forest and the number of wavelengths they construct.

1.4. Contributions and outline

The major contributions of this research are briefly described here. First, a new cost model of the WDM multicast problem is introduced and a new problem is defined. Second, a smart coding method of GAs for the multicast problem on WDM networks is constructed. Two GAs are proposed to solve the WDM multicast problem. Then, a *Farthest-first* greedy algorithm is used to generate the initial population and speed up the convergence of the GA.

The rest of this paper is organized as follows. In Section 2, the assumptions and definitions of the multicast problem are described. In Sections 3 and 4, a simple genetic algorithm (SGA) and an extended genetic algorithm (TLGA)

are proposed to solve this problem. Experimental results are given in Section 5. Finally, conclusions are given in Section 6.

2. ASSUMPTION, FORMULATION, AND RELATED WORKS

2.1. Network model

In this paper, all nodes in a WDM network are *TaC* nodes. The functions of the *TaC* nodes are listed as follows [17]:

- (i) *drop only*: when the locally attached router is a destination and there is no need to forward a copy to any downstream node;
- (ii) *continues only*: when the locally attached router is not a destination and there is a down-stream destination;
- (iii) *drop and continue*: when the locally attached router is a destination and there is a down-stream destination.

For the given WDM network, the unique source node is able to send multiple “copies” to the same output using different wavelengths along different paths. That is, the source node of the multicast request has multiple transmitters, and therefore packets can be transmitted to as many children as needed when constructing a multicast tree, rooted at itself. Similarly, a source can transmit to its children on different wavelengths using different transmitters.

2.2. Problem definition

A WDM network is represented by an undirected and connected graph $G(V, E, c, w)$, where V is the node-set of G which represents the set of nodes in the network with $|V| = n$, and E is the edge-set of G with $|E| = m$ corresponding to fiber links between nodes in the network. Each link carries two opposite-directed fibers in two directions. Each edge e in E is associated with a cost $c(e)$ which represents the communication cost of edge e . Let w be the number of wavelengths provided by the WDM network. Function $c(P(u, v))$ is additive over the edges of a light path $e \in P(u, v)$ between two nodes u and v , as shown in (1):

$$c(P(u, v)) = \sum_{e \in P(u, v)} c(e). \quad (1)$$

In this paper, a multicast request $r(s, D)$ is given, the request is for setting up a multicast channel from the source s to a group of destinations $D = \{d_1, d_2, \dots, d_{|D|}\}$, where $s \in V$, $s \notin D$, and $D \subseteq V$. The problem studied in this paper is called the *minimal cost multicast routing problem (MCMRP)*. The goal of MCMRP is to find the routing trees (or tree) and the assigned wavelengths for the multicast request such that the total cost, consisting of routing cost and wavelength cost, can be minimized.

Let $T_k(s, D_k)$ be the *light tree* for the request $r(s, D_k)$ on the k th wavelength network, where k is in $\{1, 2, \dots, w\}$ and $D_k \subseteq D$. That is, the set D of destinations can be partitioned into several subsets, $D = \bigcup_{k=1,2,\dots,w} D_k$ and $D_i \cap D_j = \emptyset$, for $i \neq j$; $i, j \in \{1, 2, \dots, w\}$. Let $T = \bigcup_{k=1,2,\dots,w} T_k$ denote the *light forest* for the request $r(s, D)$. The cost of a light tree $T_k(s, D_k)$ on the k th wavelength network is defined as the

sum of the cost of all edges in the light tree $T_k(s, D_k)$ as shown in (2):

$$c(T_k(s, D_k)) = \sum_{e \in T_k(s, D_k)} c(e). \quad (2)$$

Similarly, the cost of the light forest T is defined in (3):

$$c(T) = \sum_{k=1}^w c(T_k(s, D_k)). \quad (3)$$

In this paper, the input optical signal of a *TaC* node can only be forwarded to an output port leading to its child. A node in the WDM network can perform a “drop and continue” function to transmit the signal to its child until all nodes in D_k receive it. That is, no *light splitting* function is enabled on node. Therefore the degree of nodes, except for source node, of light tree on each wavelength network is less than or equal to 2.

Another important goal of the model is to minimize the number of wavelengths to serve the multicast request without causing wavelength conflicts. Since wavelength is one of the most important resources in the WDM network, if the number of used wavelengths can be reduced, the consequent multicast requests can find the light tree or light forest easily in the WDM network. Thus, the blocking probability of the multicast request can be reduced. Otherwise, the multicast request will be blocked or rejected. In the equation described above, if the number of wavelengths used to route the multicast request is not restricted, the routing path may tend to consume more wavelengths.

In MCMRP, given a multicast request $r(s, D)$ and a WDM network $G(V, E, c, w)$, the objective cost has two components. The first one is the cost of the light forest $c(T)$ and the other is the number of used wavelengths. In the WDM network, two paths must be assigned different wavelengths if their routes share a common link. Define binary variables y_k , for wavelength $k = 1, 2, \dots, w$; $y_k = 1$, if wavelength k is used by the light forest T ; $y_k = 0$, otherwise. Thus, the total objective cost of the MCMRP can be defined in (4):

$$c(r(s, D)) = c(T) + \alpha \times \sum_{k=1}^w y_k, \quad (4)$$

where α is the ratio of the cost of light forest to that of the wavelengths used. To reduce the routing cost, rerouting the path to destination on a whole new wavelength network is a good approach, but this may increase the number of used wavelengths. Thus, there is a tradeoff between the choices of a routing path on used or new wavelengths.

2.3. Light-splitting constraint

A key observation is that, due to the *light-splitting* constraint of the WDM node, a single light tree may not be sufficient to transmit the multicast message to all destinations in a multicast session. Consider the example shown in Figure 3, a WDM network with 18 nodes and a multicast request $r(s, D) = r(s, \{1, 2, 3, 4, 5, 6\})$ are given. The number near

FIGURE 3: Example of MCMRP and the shortest path tree P .

the edge represents the cost of edge. When the shortest path heuristic algorithm is used to find the shortest tree from s to D , node 7 is used to forward data to nodes 1, 2, and 3 as the bold lines show in Figure 3. This shortest path tree violates the *light-splitting constraint*. To solve this problem, path $P(s, 3)$ or $P(s, 2)$ may be reassigned another wavelength. Once destinations 2 and 3 are reassigned to other wavelengths, source node s needs to send out three “copies” on link $(s, 7)$, and three wavelengths are needed on link $(s, 7)$ in a wavelength-routed network. Alternately, paths $P(s, 2)$ and $P(s, 3)$ may be rerouted to satisfy the *light-splitting constraint*.

2.4. GA-based approaches in multicast routing

For the multicast problem on general networks, there were several multicast routing algorithms [18–24]. For example, KMB algorithm (proposed by Kou et al.) [18], minimum path cost heuristic (MPH) [19], dynamic center-based heuristic (DCBH) [20], and best effort residual delay (BERD) [20] belong to traditional heuristic algorithms. In [21], the minimum cost Steiner tree is constructed based on GAs to form the QoS-driven multicast routing tree with the lowest cost of network resources. In [22], a multicast routing algorithm based on chaotic neural networks was proposed. In [23], a hybrid approach based on neural networks and GAs was proposed to solve the QoS multicast routing problem. In [24], a QoS multicast routing algorithm based on the tabu search was proposed, speeding up the convergence to the optimal solution. In [8], a QoS multicast routing algorithm based on a tabu-hierarchy GA (THGA) in IP/DWDM optical Internet was designed. This constructs a QoS, and cost optimized or suboptimized multicast routing tree with network load balance was supported.

Recently, GAs were used to find the multicast routing so as to reduce the total cost from the multicast delivery tree. Examples are the multimedia multicast routing [25, 26], WDM multicast network routing [27, 28], QoS-based multicast routing [28–30], WDM ring multicast [31], WDM any-

cast [32], group multicast [33], and others related to Steiner trees. In [28], authors considered the problem of minimizing the number of split-capable nodes in the network for a given set of multicast requests. A GA that exploits the combination of alternative shortest paths for the given multicast request, in order to minimize the number of required split-capable nodes, was proposed. Several GA-based algorithms have been proposed to solve the Steiner tree problem with QoS constraints. In [34], the authors considered the framework of multicast routing in wireless ad hoc networks. An unconstrained problem was also addressed in [35], where the authors apply GAs to the calculation of optimal link weights. These are then used to automatically generate multicast routes. Another genotype representation scheme was proposed in [36]. In [14], the authors considered the bandwidth-delay-constrained least-cost multicast routing problem by proposing several novel solutions based on GAs.

3. SIMPLE GENETIC ALGORITHM (SGA) FOR WDM MULTICAST ROUTING

In this section, a simple genetic algorithm (SGA) is proposed to solve MCMRP. The development of the SGA requires: (1) a chromosomal coding scheme, (2) crossover operators, (3) mutation operators, (4) a fitness definition, (5) a replacement strategy, and (6) termination rules. The outline of the SGA is shown in Algorithm 2.

3.1. Chromosomal coding

A pair of source-destination nodes can be connected by a *path*. There are usually many possible paths between any pair of nodes in a given network. For example, consider the network shown in Figure 3; the possible paths between the node s and the node 1 include $s \rightarrow 7 \rightarrow 1$, $s \rightarrow 9 \rightarrow 1$, $s \rightarrow 7 \rightarrow 14 \rightarrow 1$, ..., and so on. For each destination d_i in D , there is a path table which consists of R_i possible paths associated with it. These paths in the path table are sorted and indexed in the nondescending order according to the cost of the path. The encoding method, denoted as *path-oriented encoding*, was first proposed in [37] for the point-to-point routing problem and used by Hwang et al. [38] to solve the multicast routing problem in a general network. In the MCMRP, because both routing paths and assigning wavelengths are considered simultaneously, the encoding method proposed in [38] cannot be applied directly. Thus, an adjustment algorithm and/or extended coding method should be developed to generate a feasible solution.

Let R_i be an integer which represents the size of the path table from s to destination d_i , for $i = 1, 2, \dots, |D|$. The value of R_i is a parameter of the algorithm and is initially set to a small number ($R = R_1 = R_2 = \dots = R_{|D|} = 16$ or 64). For a given source node s and destination set $D = \{d_1, d_2, d_3, \dots, d_{|D|}\}$, the chromosome of SGA can be represented by an array of integers with length $|D|$. A *path gene* p_i , $i = 1, 2, \dots, |D|$, of the chromosome is an integer in $\{1, 2, \dots, R_i\}$, which represents a possible path between s and d_i . For example, Figure 4 shows the path table for the

```

BEGIN
  Initialize population randomly;
  generate randomly the routing paths for destinations;
  call Light-forest construct algorithm to generate feasible light forests;
  compute the fitness of light forests.
  WHILE (not terminated) DO
    choose parents from population; /* Selection */
    generate offspring through crossover and mutation;
    call Light-forest construct algorithm to generate feasible light forests;
    compute the fitness of light forest;
    replace the worst ranked part of population with offspring.
  END of WHILE
END

```

ALGORITHM 2: Simple genetic algorithm.

p_1	p_2	p_3	p_4		p_i		$p_{ D }$
-------	-------	-------	-------	--	-------	--	-----------

Routing table of $s \rightarrow 1$		
Path number	Path lists	Cost
1	$s \rightarrow 7 \rightarrow 1$	10
2	$s \rightarrow 7 \rightarrow 14 \rightarrow 1$	11
3	$s \rightarrow 9 \rightarrow 1$	13
4	$s \rightarrow 8 \rightarrow 9 \rightarrow 1$	22
...	...	
R		

FIGURE 4: Example of the path-oriented encoding.

source-destination pair $(s, 1)$. The relationship between the path gene and path table is explained in Figure 4.

Obviously, a chromosome represents a candidate solution for the multicast routing problem, because it guarantees a path between the source node and any of the destination nodes. However, a chromosome does not necessarily represent a light tree, because a chromosome may violate the light-splitting constraint. Thus, a decoding method called *Light-Forest Constructing Algorithm (LFCA)* is proposed to transform the chromosome to the corresponding light forest. Given a path-oriented encoding chromosome, after performing the LFCA, the corresponding light forest on the WDM network $G(V, E, c, w)$ is constructed. Moreover, the cost of the light forest is also computed. Let $A = \{p_1, p_2, \dots, p_{|D|}\}$ be the set of paths represented by the chromosome, which are extracted from respective path tables. For each path p , let $\text{Bypass}(p)$ be the set of paths contained in A whose destination nodes are passed by path p . The details of the *Light-Forest Construct Algorithm (LFCA)* are shown as in Algorithm 3.

Assume the paths extracted from path tables of all destinations of the chromosome are shown in Figure 5(a). After performing LFCA, the final light trees on first and second wavelength networks are shown in Figure 5(b). Observe the result shown in Figure 5; destination 5 is reached by the

routing path $P(s, 6)$. Two wavelengths are used to transmit the multicast request. The cost of the light forest is equal to $62 + 2 \times \alpha$. It is noteworthy that the tests of “Is there any conflict between path p_k and tree T_j ?” and “Does adding path p_k to T_j form a cycle?” can provide knowledge by adding edge of path p_k into T_j edge-by-edge and test “Whether the degree of the endvertices of edge is greater than 2?” If no, path p_k is added into light tree T_j to form a new light forest, such as $T_j = T_j \cup p_k$. Otherwise, p_k is in conflict with light tree T_j and p_k tries to be assigned to the next light forest T_{j+1} if possible. This process is repeated until a wavelength is found for the path.

3.2. Crossover operator

The crossover operator used in the development of SGA is two point crossover (TPC). The TPC randomly selects two chromosomes for crossover from previous generation and then uses a random number generator, in which two integers i_1, i_2 ($i_1 \neq i_2$) are generated in the interval $[1, |D|]$. These integers are used as the crossover sites.

It is worth noting that after performing the TPC crossover, LFCA should be used to construct the new light forest and the cost of the new light forest should be recomputed.

3.3. Mutation

Two types of mutations are used to develop this algorithm; they are *single routing path mutation (SRPM)* and *multiple routing paths mutation (MRPM)*.

- (i) *Single routing path mutation (SRPM)*: first, a destination d_i ($i = 1, 2, \dots, |D|$) is selected randomly from a chromosome. Then the SRPM changes the value of path gene p_i to an integer z which is selected randomly from $[1, R_i]$. After performing SRPM, the light forest may be changed. Thus, the LFCA should be used to construct the new light forest and evaluate the cost of the new chromosomes.

FIGURE 5: (a) The path represented by the chromosome, (b) the routing paths and wavelength assignment of example 1 after performing LFCA.

- (ii) *Multiple routing paths mutation (MRPM)*: first, select an integer z between 1 and $|D|$. A set Z of z destinations is randomly generated from a chromosome. Then, MRPM changes the value of the destination node d_i in Z to a random integer, which is randomly generated between 1 and R_i . After performing the MRPM, the light forest may be changed. Thus, the LFCA should be used to construct the new light forest and evaluate the cost of the new chromosome.

3.4. Fitness definition

Generally, GAs use fitness functions to map objectives to costs and to achieve the goal of a minimal cost light forest. An objective function value is associated with each chromosome and can be represented by the equation as follows:

$$\text{minimize } \sum_{j=1}^w \text{cost}(T_j) + \alpha \times \sum_{j=1}^w y_j. \quad (5)$$

3.5. Replacement strategy and termination rules

Initially, assume that $N_{\text{population}}$ is the number of chromosomes to be generated and N_{parent} connection chromosomes are randomly constructed. In the process of selection, $N_{\text{population}}/2$ pairs of connections are randomly selected for crossover to generate the new generation of chromosomes. After crossover, chromosomes are sorted according to the fitness in the increasing order; N_{parent} chromosomes with the smallest fitness are selected to construct the new generation. The values of $N_{\text{population}}$ and N_{parent} will be determined through experiments. Execution of SGA can be terminated when the number of generations exceeds an upper bound (n_g), specified by the user.

4. TWO-LEVEL GENETIC ALGORITHM (TLGA)

In this section, another genetic algorithm, named *two-level genetic algorithm* (TLGA), is proposed to solve the MCMRP. In the TLGA, the encoding method of the SGA is extended by adding an *assigning gene* which represents the assigned

wavelengths of the paths. Because this extension may cause the chromosome generated by the TLGA to be an infeasible solution, some heuristics are developed to modify the chromosome into a constraint-satisfied solution. The outline of TLGA is shown in Algorithm 3 and the details of TLGA are described in the following subsections.

4.1. Chromosome encoding

Because the problem involves representing routing paths of multicast and assigning wavelengths, a coding scheme that uses positive integral number is employed. A chromosome of one-dimensional array with size $2 \times |D|$ is introduced to represent the routing paths and assigned wavelengths of paths.

The structure of the chromosome is divided into two parts, as shown in Figure 6(a). The first part denoted as the PG (*path gene*), $\text{PG} = \{\text{pg}_1, \text{pg}_2, \dots, \text{pg}_{|D|}\}$, is the same as the PG of SGA described in Section 3.1. The second part of the chromosome is used to represent the assigned wavelength of the respective path, denoted as the AG (*Assigning Gene*), $\text{AG} = \{\text{ag}_1, \text{ag}_2, \dots, \text{ag}_{|D|}\}$, where each element ag_i ($i = 1, 2, \dots, |D|$) falls in $\{1, 2, \dots, w\}$.

Let $\text{Edge}(P(s, d_i))$ be the set of edges that compose the path $P(s, d_i)$ and all the edges with at least one of its endpoints, not be s , be on $P(s, d_i)$. Let $G' = G \setminus \text{Edge}(P(s, d_i))$ be the remaining graph by removing those edges and nodes in $\text{Edge}(P(s, d_i))$. For example, assume that G is the network shown in Figure 3. If $P(s, 2) = s \rightarrow 7 \rightarrow 14 \rightarrow 2$, then $\text{Edge}(P(s, 2)) = \{(s, 7), (7, 14), (14, 2), (1, 7), (1, 14), (3, 14), (2, 11), (2, 15), (2, 16)\}$. To generate the initial values of the chromosomes, first the content of the chromosome is selected randomly. Then, the content is modified by performing the *chromosome adjustment algorithm* (CAA) Algorithm 4 to form a feasible solution. There are several algorithms called by the CAA; they are (1) the *wavelength mapping algorithm*, (2) the *wavelength packing algorithm*, and (3) the *path rerouting algorithm*. The details of these algorithms are described as follows.

- (i) The *wavelength mapping algorithm* (WMA) is used to map the set of used wavelengths to the new set $\{1, 2, \dots, |W_{\text{old}}|\}$, where $W_{\text{old}} = \{w(p(s, d_i)) \mid \text{with } d_i$

```

Sort paths which are represented by the chromosomes in the descending order according to the cost
of paths and reindex;
let  $A = \{p_1, p_2, \dots, p_{|D|}\}$  be the set of paths and  $c(p_1) \geq c(p_2) \geq \dots \geq c(p_{|D|})$ ;
 $k = 1$ ;
 $T_1 = T_2 = \dots = T_w = \emptyset$ ;
select and remove a path  $p_k$  from  $A$ ;
WHILE ( $A \neq \emptyset$ ) DO
     $flag = FALSE$ ;
     $j = 1$ ;
    WHILE ( $flag \neq TRUE$ ) DO
        IF ( $T_j \cup p_k$  satisfies light-splitting constraint and does not form a cycle) THEN
             $T_j = T_j \cup p_k$ ;
             $flag = TRUE$ ;
            path  $p_k$  is assigned wavelength  $j$ ;
            compute Bypass( $p_k$ );
             $A = A \setminus \text{Bypass}(p_k)$ ;
        ELSE
             $j++$ ;
            IF ( $j > w$ ) THEN
                print("not enough wavelengths");
                exit();
            END IF
        END IF
    WHILE ( $p_k \notin A$ )
         $k++$ ;
        select and remove a path  $p_k$  from  $A$ ;
    END of WHILE
     $k++$ ;
END of WHILE
END of WHILE

```

ALGORITHM 3: Light-forest construct algorithm.

being a leaf node of the light tree} is the set of wavelengths used by the routing paths. To do the mapping, only a bijective function should be found.

- (ii) The *wavelength packing algorithm* (WPA) is used to reduce the number of used wavelengths of the light forest. To do this, first for the given light forest, which consists of several light trees on different wavelength network, a bipartite graph $BG(U, V, E)$ is constructed. Then node in the set (U and V) of BG is the light tree; edge in BG is established for two nodes if the respective light trees can be merged without violating the light-splitting constraint. Second, the *maximal matching finding algorithm* is performed to find the maximal matching. Finally, two light trees connected by the matching edge are merged to form a new light tree, the two original light trees are deleted, and the wavelength mapping algorithm is performed. This process is repeated until no matching edge is found.
- (iii) Given a light tree T and a source-destination pair (s, d), the *path rerouting algorithm* tries to find the new minimal cost path $p(s, d)$ such that $T \cup p(s, d)$ satisfies the light-splitting constraint.

Because the routing paths and the assigned wavelengths in the TLGA are generated randomly, this may generate an

infeasible solution. Thus, the chromosome should be adjusted to a constraint-satisfied solution by performing the CAA. The details of CAA are described in Algorithm 5.

4.2. Crossover operator

Four types of crossover operators are used to develop this TLGA. They are (1) the *single point crossover* (SPC), (2) the *single point wavelength crossover* (SPWC), (3) the *single point routing path crossover* (SPRPC), and (4) the *wavelength exchanging operator* (WEO). The details of the crossover operators are described as follows.

- (i) *Single point crossover* (SPC): randomly select a crossover site integer i from interval $[1, |D|]$. Two chromosomes are mated as shown in Figure 6(b). In SPC, the operator is applied on both the path gene and the wavelength gene.
- (ii) *Single point wavelength crossover* (SPWC): randomly select a crossover site integer i from interval $[1, |D|]$. Two chromosomes are mated as shown in Figure 6(c). In SPWC, the operator is only applied on the wavelength gene.
- (iii) *Single point routing path crossover* (SPRPC): randomly select a crossover site integer i from interval

(a)

(b)

(c)

(d)

FIGURE 6: (a) Encoding method, (b) SPC, (c) SPWC, and (d) SPRPC.

```

BEGIN
  Perform wavelength mapping algorithm;
  let  $B$  be the set of wavelength networks which violate the light-splitting constraint;
  let  $j$  be the minimal index of new wavelengths.
  WHILE ( $B \neq \emptyset$ ) DO
     $T = \emptyset$ ;
    randomly select and remove a wavelength network from  $B$ ;
    let  $P$  be the set of paths assigned to the wavelength network;
    sorted paths in  $P$  in the increasing order according to the cost of paths;
    WHILE ( $P \neq \emptyset$ ) DO
      select and remove a path  $p(s, d)$  from  $P$ ;
      If ( $p(s, d) \cup T$  violates the light-splitting constraint or forms a cycle), then
        perform path rerouting ( $T, p(s, d)$ ) to find path  $p'(s, d)$ ;
        If (path  $p'(s, d)$  is found), then
          if ( $p'(s, d)$  is a new path), then
            insert path  $p'(s, d)$  into the path table;
            increasing the table size of respective destination  $d$  by 1;
          else
            path  $p(s, d)$  is assigned wavelength  $j$ ;
            increase  $j$  by 1;
        END IF
      END IF
    ELSE
       $T_j = P \cup T$ ;
    END IF
  END of WHILE
  Perform wavelength packing algorithm;
END

```

ALGORITHM 4: Chromosome adjustment algorithm.

$[1, |D|]$. Two chromosomes are mated as shown in Figure 6(d). In SPRPC, the operator is only applied on the path gene.

and ag_j , $1 \leq i, j \leq |D|$ ($i \neq j$); and the assigned wavelengths of these two paths are exchanged.

4.3. Mutation

Three types of mutations are used to develop the TLGA. They are (1) the *single routing path mutation* (SRPM), (2) the *single*

- (iv) *Wavelength exchanging operator* (WEO): randomly select two wavelength assignments of assigning genes ag_i

```

BEGIN
  Initialize population randomly;
  generate routing paths for destinations randomly;
  generate assigning wavelengths of routing paths randomly;
  call chromosome adjustment algorithm to generate feasible light forests;
  compute the fitness of light forests.
  WHILE (not terminated) DO
    choose parents from population; /* Selection */
    generate offspring through crossover and mutation;
    call chromosome adjustment algorithm to generate feasible light forests;
    compute the fitness of light forests;
    replace the worst ranked part of the population with offspring.
  END of WHILE
END

```

ALGORITHM 5: Two-level genetic algorithm.

```

BEGIN
  let TS be the set of light trees on wavelengths networks;
  perform wavelength mapping algorithm;
   $mm = |W_{old}|$ ; // maximal matching initial value
  WHILE ( $mm > 0$ ) DO
    Construct a bipartite graph  $BG(U, V, E)$  of the set of the TS;
    nodes in  $U$  and  $V$  represent the light tree in TS;
    let  $E$  be the set of edges  $e = (u_i, v_j) \in E$ , where  $u_i \in U$  and  $v_j \in V$ , and  $T_i \cup T_j$ 
      satisfies the light-splitting constraint;
    perform maximal matching algorithm to find the maximal matching  $mm$  of BG;
    for each matching edge, merge two light trees  $T_i$  and  $T_j$  which are connected by it;
    perform wavelength mapping algorithm.
  END of WHILE
END

```

ALGORITHM 6: Wavelength packing algorithm.

wavelength mutation (SWM), and (3) the *wavelength packing mutation* (WPM).

- (i) *Single routing path mutation* (SRPM): randomly select an integer i in the interval $[1, |D|]$, the single routing path mutation (SRPM) changes the value of pg_i to an integer randomly selected from $\{1, 2, \dots, R_i\}$.
- (ii) *Single wavelength mutation* (SWM): randomly select an integer i in the interval $[1, |D|]$, the single wavelength mutation (SWM) changes the value of ag_i to an integer randomly selected from $\{1, 2, \dots, w\}$.
- (iii) *Wavelength packing mutation* (WPM): WPM changes the assigned wavelengths of paths in the selected chromosome by performing the wavelength packing algorithm (see Algorithm 7).

5. HEURISTIC INITIALIZATION

Because finding the optimal solution for the MCMRP with a large problem size is impractical, GAs can be used to find the near optimal solution. But, only using random initialization

the GA may take a lot of time to converge. Therefore, heuristic algorithms are used to find the better initial solution, instead of random initialization, to speed up the convergence of GAs. In this section, a *shortest path-based* heuristic algorithm is proposed to find the minimal cost path from source to destinations (see Algorithm 6).

Assume that $P(s, d_i)$, $i = 1, 2, \dots, |D|$, is the minimal cost path from the source node s to the destination $d_i \in D$. Paths $P(s, d_i)$, $i = 1, 2, \dots, |D|$ are combined to form P , such that $P = \bigcup_{i=1}^{|D|} \{P(s, d_i)\}$. It is worth noting that P should be a tree. If P satisfies the *light-splitting constraint*, then P is the *minimal cost light tree* of the multicast request. Otherwise, a heuristic algorithm is needed to modify it to a light tree or light forest which satisfies the light-splitting constraint.

Before introducing the proposed algorithm, some notations are introduced and stated here. Let $\text{degree}(s)$ be the degree of the source node s on tree P , then tree P can be divided into $\text{degree}(s)$ subtrees. Let $v_1, v_2, \dots, v_{\text{degree}(s)}$ be the nodes in P which are directly adjacent to the source node s ; the subtree rooted at v_i is represented by $\text{PT}(v_i)$, for $i = 1, 2, \dots, v_{\text{degree}(s)}$. This is due to the fact that the routing tree

```

BEGIN
  let  $w(p(s, d_i))$  be the assigned wavelength of the path  $p(s, d_i)$ ;
  let  $W_{old} = \{w(p(s, d_i)) \mid d_i \text{ is a leaf node of the light tree}\}$  be the set of wavelengths used by
  the routing paths; ( $|W_{old}| \leq w$ )
  find a one-to-one and onto (bijective) mapping function  $wm : W_{old} \rightarrow \{1, 2, \dots, |W_{old}|\}$ ;
  for each path  $p(s, d_i)$ , the assigned wavelength  $w(p(s, d_i))$  is changed to  $wm(w(p(s, d_i)))$ ;
END

```

ALGORITHM 7: Wavelength mapping algorithm.

```

Input: tree  $T$ , path  $p(s, d)$ 
Output: return path or NULL
BEGIN
  For each leaf node  $d_i$  in  $T$ , find the set  $\text{Edge}(P(s, d_i))$ ;
  for each leaf node  $d_i$  in  $T$ , remove  $\text{Edge}(P(s, d_i))$  from  $G$ , let the remaining graph be  $G'$ ;
  find a minimal cost path  $p'(s, d)$  from source  $s$  to destination  $d$  on graph  $G'$ .
  IF (path  $p'(s, d)$  is found), then
    return path  $p'(s, d)$ ;
  else
    return NULL.
  END IF
END

```

ALGORITHM 8: Path rerouting algorithm.

in a wavelength network cannot contain cycle(s) and must satisfy the light-splitting constraint. Thus, it is easy to find that if there is more than one destination on the leaves in one of the subtrees, only one of these destinations can be selected to route by this subtree and the other destinations should be rerouted. To determine the rerouting paths of the destinations in the leaves, a heuristic method is developed and used. For each subtree $PT(v_i)$, $i = 1, 2, \dots, v_{\text{degree}(s)}$, only the farthest destination v with $c(P(s, v)) \geq \max_{u \in PT(v_i)} \{c(P(s, u))\}$ is routed by the path $P(s, v)$ on $PT(v_i)$ and the others should be rerouted. This is denoted as the *farthest-first greedy* (FG).

Consider the network shown in Figure 3, there are three destinations in the subtree $PT(7)$ with nodes 1, 2, and 3. The farthest (maximal cost) destination is node 2, thus the routing paths to destination nodes 1 and 3 should be rerouted. The farthest destinations in the subtrees $PT(8)$ and $PT(10)$ are nodes 6 and 4, respectively. Now, the problem is “how to find the reroute paths?”, because the routed paths and nodes used by the farthest path in each subtree cannot be used twice in the same wavelength network. Moreover, the rerouting paths together with the existing paths cannot form a cycle (or cycles) or violate the light-splitting constraint. Thus, the path $P(s, v)$ to the farthest destinations and $\text{Edge}(P(s, v))$ in each subtrees should be removed. Let *UNREACH* be the set of all destinations that have not been routed and which are sorted in the descending order according to the minimal cost of path on G .

According to the above discussions, the rerouting path can be determined to be more complex. Let $G_1, G_2, \dots, G_w$ be the w wavelength graphs of a WDM network, the script

index i of G_i means that the routing path is on the i th wavelength network. Let k be an integer which indicates the maximal index of the currently used wavelength network. To find a rerouting path of a destination v , $k + 1 (\leq w)$ wavelength graphs $(G_1, G_2, \dots, G_k, G_{k+1})$ are considered in the proposed algorithm. Let $P^i(s, v)$ be the minimal cost path from source s to destination v in the wavelength graph G_i , the cost of the path $P^i(s, v)$ is defined as $c(P^i(s, v))$, $i = 1, 2, \dots, k + 1 (\leq w)$ and is determined as follows.

- (i) If $P^i(s, v)$ passes destinations, $\{v_1, v_2, \dots, v_k\}$ in *UNREACH* on G_i , then the costs of $P(s, v_1), P(s, v_2), \dots, P(s, v_k)$ on G should be subtracted from the cost of path $P^i(s, v)$. That is, $c(P^i(s, v)) = c(P^i(s, v)) - c(P(s, v_1)) - c(P(s, v_2)) - \dots - c(P(s, v_k))$.
- (ii) If $P^i(s, v)$ uses the wavelength graph the $(k + 1)$ th ($\leq w$), where i is equal to $k + 1$, then the cost of path $P^i(s, v)$ should be increased by α .
- (iii) If path from s to v cannot be found in the wavelength network G_i , then $c(P^i(s, v)) = \infty$.

After finding and computing the paths and costs of the possible paths, the path with the minimal cost is selected to be the rerouting path. If i is equal to $k + 1$, then increase k by 1. Find the set $\text{Edge}(P^i(s, v))$ and remove those edges in $\text{Edge}(P^i(s, v))$ from G_i . Let $\text{dest}(v_i)$ be the farthest destination node in subtree $PT(v_i)$. Let $\text{Live}(\text{dest}(v_i))$ be the set of edges that have one endpoint $\text{dest}(v_i)$ and another endpoint in G' . For each subtree $PT(v_i)$, a subgraph is constructed by $G' \cup \text{Live}(\text{dest}(v_i))$. It is worth noting that the cost of the routing path, which is called the *extended path*,

Step 1. Let $k = 1$, $G'_1 = G'_2 = \dots = G'_w = G$. For each destination d_i in D , find the minimal cost path $P(s, d_i)$ on G_1 . Combine the paths to form a routing tree P . If P satisfies the light-splitting constraint, then done. Otherwise perform *Step 2*.

Step 2. For each subtree $PT(v_i)$ on G_1 , rooted at v_i , $i = 1, 2, \dots, \text{degree}(s)$, find the farthest destination node $\text{dest}(v_i)$, the routing path $P^1(s, \text{dest}(v_i))$ and the $\text{Edge}(P^1(s, \text{dest}(v_i)))$. Modify G'_1 by removing $\text{Edge}(P^1(s, \text{dest}(v_i)))$, $i = 1, 2, \dots, \text{degree}(s)$ from G'_1 and construct *UNREACH*. While *UNREACH* is nonempty, perform *Steps 3 and 4*.

Step 3. Take the farthest destination in *UNREACH*, say v , find the minimal cost paths in the set of sub-graphs $SG = \{G'_z, G'_z \cup \text{Live}(\text{dest}(v_i))\}$, for all $v_i, z = 1, 2, \dots, k + 1\}$.

Step 4. Find the minimal cost path in the path set SG . If the path is found, assign path $P(s, v)$ to the corresponding wavelength z ; subtract $\text{Edge}P(s, v)$ from G'_z ; remove the destination v and other destinations passed by path $P(s, v)$ from *UNREACH*; and update $\text{Live}(\text{dest}(v_i))$. Otherwise, increasing k by one, if k is greater than w , will return false.

ALGORITHM 9: Farthest-first greedy (FG).

routed through node $\text{dest}(v_i)$ should be included in computing the cost of the path $P(s, \text{dest}(v_i))$. This rerouting process is continually executed until *UNREACH* is empty.

The details of the farthest-first greedy (FG) algorithm are described as in Algorithm 9.

Assume $\alpha = 4$; consider the example shown in Figure 3. After performing Steps 1 and 2 of the farthest-first greedy algorithm, the result is shown in Figure 7(a). Because destination 3 is the farthest node in *UNREACH*, Step 3 tries to find the minimal cost path from s to 3. In G_1 the path is $s \rightarrow 9 \rightarrow 13 \rightarrow 3$ with the cost of 15. In $G'_1 \cup \text{Live}(\text{dest}(7))$, the extended path is $2 \rightarrow 11 \rightarrow 3$ with the cost of 9. Similarly, in G'_2 the path is $s \rightarrow 7 \rightarrow 14 \rightarrow 3$ with the cost of 11 ($7 + \alpha$). It is clear that the extended path $c(P^1(2, 3))$ is the minimal cost path, thus, destination 3 is routed in the wavelength graph G_2 .

Then, to find the minimal cost routing path for destination 1, those paths on G_1 , $G_1 \cup \text{Live}(\text{dest}(7))$, and G_2 are considered. They are path $P^2(s, 1) = s \rightarrow 9 \rightarrow 1$ with a cost of 13, extended path $P^1(3, 1) = 3 \rightarrow 13 \rightarrow 1$ with a cost of 3, and path $P^3(s, 1) = s \rightarrow 7 \rightarrow 1$ with a cost of 10 ($6 + \alpha$). Obviously, the minimal cost path is the extended path $P^1(3, 1)$. Thus, the final result by applying the farthest-first greedy algorithm to Figure 3 is shown in Figure 7(c).

6. EXPERIMENTAL RESULTS

In order to evaluate the performance of the proposed GAs, they were implemented and applied to solve problems that were randomly generated. The results of these experiments are reported below. In all the experiments, the algorithms were conducted in C++, and all the experiments were run on a personal computer (PC) with a Pentium III 1 GHZ CPU and 512 MB RAM.

6.1. Parameters

In this subsection, the effects of the various parameters of GAs are discussed. A WDM network with $n = 100$ nodes, $m = 1208$ edges, and $w = 10$ wavelengths was used to deter-

mine the parameters of GAs. The cost of edge is randomly selected from the set of integers $\{1, 2, \dots, 20\}$. The multicast request $r(s, D)$ with $|D| = 50$ destinations, which are randomly selected from the set of nodes, was used in the experiments. The common parameters of GAs are the crossover probability $p_c = 1.0$, the number of generations $N_{\text{generation}} = 100$, and the ratio $\alpha = 50$. To determine the parameters of GAs, each GA was performed ten times and the average results are used for the comparison.

6.1.1. SGA

For the proposed SGA with heuristic initialization, the parameters of SGA are determined by performing the following experiments.

(1) Effect of the mutation probability: in Figure 8, the effect of the mutation probability on the evolution of SGA is presented. This experiment assumes p_m in $\{0.1, 0.2, 0.3, 0.4, 0.5\}$. It is found that SGA leads to both faster convergence and better result when the mutation probability is equal to 0.2.

(2) Effect of the population size: in Figure 9, the effect of different number of chromosomes in the population (pop) is presented, where pop is in $\{100, 200, 300, 400, 500\}$. It is found that SGA with pop = 500 can get better results.

(3) Effect of the path table size: in Figure 10, the effect of different values of R ($R = R_1 = R_2 = \dots = R_{|D|}$) is presented, where R is $\{5, 10, 15, 20, 25\}$. It is found that SGA with $R = 25$ can get better results.

6.1.2. TLGA

For the proposed TLGA with heuristic initialization, the parameters of TLGA are determined by performing the following experiments.

(1) Effect of the mutation probability: in Figure 11, the effect of the mutation probability on the evolution of TLGA is presented. In this experiment, p_m is $\{0.1, 0.2, 0.3, 0.4, 0.5\}$. It is found that TLGA leads to both faster convergence and to better result when the mutation probability is equal to 0.2.

FIGURE 7: Final result by applying the farthest-first greedy algorithm.

FIGURE 8: The effect of the mutation probability on the performance of SGA.

FIGURE 10: The effect of the path table size on the performance of SGA.

FIGURE 9: The effect of the populations size on the performance of SGA.

FIGURE 11: The effect of the mutation probability on the performance of TLGA.

FIGURE 12: The effect of the population size on the performance of TLGA.

FIGURE 13: The effect of the population size on the performance of TLGA and SGA.

(2) Effect of the population size: in Figure 12, the effect of different number of chromosomes in population (pop) is presented, where pop is $\{100, 200, 300, 400, 500\}$. It is found that TLGA with pop = 500 can get better results.

(3) Effect of the path table size: in Figure 14, the effect of different values of R ($R = R_1 = R_2 = \dots = R_{|D|}$) is presented, where R is $\{5, 10, 15, 20, 25\}$. It is found that TLGA with $R = 25$ can get better results.

6.1.3. Discussions

For the mutation, the purpose is to allow GAs to avoid local minima by preventing the population of chromosomes from becoming too similar to each other, thus slowing or even stopping evolution. If the mutation probability is high, the population of chromosomes may change intensely and be difficult to converge and find the optimal solution. Moreover, increasing the mutation probability will increase the running

FIGURE 14: The effect of the path table size of TLGA.

time of GAs. The experiments show that the mutation probability of both the SGA and the TLGA is set to 0.2. For the population size, in general, the larger the population size, the better the result that can be found by GAs. On the other hand, increasing the size of the population may also increase the time spent for each generation and the storage requirement. To examine the effect of the population size in details, for each experiment with $N_{\text{generation}} = 100$, the average fitness value $\sum_{g=1}^{100} \sum_{\text{for all chromosomes in generation}_g} \text{fitness}_g(i) / (\text{pop})$ is computed, where $\text{fitness}_g(i)$ is the fitness value of the i th chromosome of the g th generation. The results are shown in Figure 13. The experiments show that both the population sizes for the SGA and the TLGA are set to 500. Moreover, in Figure 13, the average fitness value of the TLGA is better than that of the SGA. For the path table size, giving a larger path table size will allow the GA to find a better routing path and the wavelength. It may allow the GA to find a near global optimal solution. On the other hand, as the path table size increases, the time to converge is also increased. The experiments show that SGA and TLGA with the path table size set to 25 can find better results.

6.2. Comparison

In this subsection, the performance of the proposed GAs is examined and compared. For the MCMRP, the proposed algorithms were run on several randomly generated connected networks with different nodes (n is $\{100, 200, 300\}$), and with different numbers of destination nodes $|D|$ is $\{10, 20, 30, 40, 50\}$ and with different values of ratio α in $\{50, 100, 150\}$. Moreover, to examine the performance of the proposed algorithms, two heuristic algorithms are implemented and used for comparisons. They are *minimal trial* (MT) [9] and *shortest path tree* (SPT) algorithms.

The average results of SGA and TLGA and the results of FG, MT, and SPT algorithms for different values of ratio $\alpha = 50, 100, 150$ are shown in Figures 15, 16, and 17, respectively. Observe the results shown in these figures, it can

TABLE 1: Experimental results of algorithms SGA, TLGA, and FG.

$\alpha = 50$												13colspan
n	$ D $	FG		SGA			TLGA				SPT	MT
		FG	average	min.	max.	std.	average	min	max	std.	SPT	MT
100	10	87	83	77	89	3.79	81	77	85	2.53	138	95
100	20	145	144	138	146	2.83	139	134	147	4.22	194	156
100	30	301	298	291	301	3.41	295	286	300	4.60	398	342
100	40	364	361	360	364	1.41	360	355	366	3.49	411	398
100	50	390	381	381	386	2.24	371	371	376	2.24	453	445
200	10	29	25	19	29	3.22	23	18	25	2.41	75	36
200	20	67	67	67	73	2.68	66	60	75	4.84	117	74
200	30	101	101	93	103	3.69	100	92	100	3.58	154	134
200	40	248	244	242	244	0.89	240	233	244	3.61	295	287
200	50	297	290	286	290	1.79	287	286	287	0.45	390	373
300	10	65	60	53	60	3.13	59	54	63	2.86	110	75
300	20	105	105	97	106	3.61	102	94	110	5.06	154	121
300	30	127	126	120	134	4.47	123	121	131	3.69	175	143
300	40	212	212	210	216	2.00	209	203	215	3.79	261	254
300	50	309	305	299	312	4.12	300	296	300	1.79	354	365
$\alpha = 100$												13colspan
n	$ D $	FG		SGA			TLGA				SPT	MT
		FG	average	min.	max.	std.	average	min	max	std.	SPT	MT
100	10	90	87	87	91	1.79	85	79	94	4.84	121	101
100	20	147	147	145	151	2.00	146	143	154	3.82	176	167
100	30	324	312	312	315	1.34	310	306	316	3.22	412	358
100	40	374	374	372	381	3.26	374	372	381	3.26	435	423
100	50	394	384	377	387	3.41	383	381	387	2.00	498	427
200	10	28	28	24	32	2.53	26	17	30	4.40	75	36
200	20	74	70	66	70	1.79	70	69	72	1.00	123	87
200	30	123	110	103	114	3.61	110	107	116	3.00	178	132
200	40	256	246	246	255	4.02	244	241	247	1.90	376	290
200	50	301	295	289	300	3.49	293	291	295	1.26	402	349
300	10	67	61	58	61	1.34	60	56	63	2.24	112	92
300	20	111	110	109	114	1.84	109	109	110	0.45	176	143
300	30	139	130	122	139	5.39	131	124	134	3.41	254	190
300	40	225	216	208	224	5.06	215	213	221	2.83	321	287
300	50	331	318	309	324	4.84	316	312	324	4.00	463	390
$\alpha = 150$												13colspan
n	$ D $	FG		SGA			TLGA				SPT	MT
		FG	average	min.	max.	std.	average	min	max	std.	SPT	MT
100	10	96	92	92	95	1.34	91	82	93	4.12	144	139
100	20	157	152	151	153	0.63	148	143	150	2.41	298	295
100	30	326	312	307	320	4.22	308	307	308	0.45	361	360
100	40	398	375	375	379	1.79	371	367	379	4.00	381	371
100	50	401	398	397	404	2.72	392	388	397	2.86	25	23
200	10	30	28	23	29	2.28	25	23	32	3.26	67	66
200	20	76	73	72	78	2.28	70	61	70	4.02	101	100
200	30	123	110	101	114	4.40	102	102	108	2.68	244	240
200	40	256	250	241	251	4.05	241	241	248	3.13	290	287
200	50	302	300	296	304	2.53	297	292	304	3.85	60	59
300	10	67	65	62	73	3.82	60	57	68	3.82	105	102
300	20	117	114	112	114	0.89	110	108	116	2.83	126	123
300	30	149	138	131	138	3.13	130	125	131	2.28	212	209
300	40	225	218	212	223	3.49	213	208	216	2.61	305	300
300	50	334	319	318	320	0.63	310	307	310	1.34	401	376

TABLE 2: Experimental results of algorithms SGA, TLGA, and FG.

$\alpha = 50$							
n	$ D $	Z-value	$\beta = 95\%$	$\beta = 99\%$	F-value	$\beta = 95\%$	$\beta = 99\%$
100	10	1.388	accept H0	accept H0	14.364	reject H0	reject H0
100	20	3.112	reject H0	reject H0	8.009	reject H0	reject H0
100	30	1.657	accept H0	accept H0	11.628	reject H0	reject H0
100	40	0.840	accept H0	accept H0	1.988	accept H0	accept H0
100	50	9.982	reject H0	reject H0	5.018	reject H0	accept H0
200	10	1.572	accept H0	accept H0	10.368	reject H0	reject H0
200	20	0.572	accept H0	accept H0	7.182	reject H0	reject H0
200	30	0.615	accept H0	accept H0	13.616	reject H0	reject H0
200	40	3.402	reject H0	reject H0	0.792	accept H0	accept H0
200	50	5.140	reject H0	reject H0	3.204	accept H0	accept H0
300	10	0.746	accept H0	accept H0	9.797	reject H0	reject H0
300	20	1.526	accept H0	accept H0	13.032	reject H0	reject H0
300	30	1.637	accept H0	accept H0	19.981	reject H0	reject H0
300	40	2.214	reject H0	accept H0	4.000	accept H0	accept H0
300	50	3.520	reject H0	reject H0	16.974	reject H0	reject H0
$\alpha = 100$							
n	$ D $	Z-value	$\beta = 95\%$	$\beta = 99\%$	F-value	$\beta = 95\%$	$\beta = 99\%$
100	10	1.226	accept H0	accept H0	3.204	accept H0	accept H0
100	20	0.733	accept H0	accept H0	4.000	accept H0	accept H0
100	30	1.813	accept H0	accept H0	1.796	accept H0	accept H0
100	40	0.000	accept H0	accept H0	10.628	reject H0	reject H0
100	50	0.800	accept H0	accept H0	11.628	reject H0	reject H0
200	10	1.246	accept H0	accept H0	6.401	reject H0	accept H0
200	20	0.000	accept H0	accept H0	3.204	accept H0	accept H0
200	30	0.000	accept H0	accept H0	13.032	reject H0	reject H0
200	40	1.422	accept H0	accept H0	16.160	reject H0	reject H0
200	50	1.705	reject H0	accept H0	12.180	reject H0	reject H0
200	10	1.212	accept H0	accept H0	1.796	accept H0	accept H0
300	20	1.669	reject H0	accept H0	3.386	accept H0	accept H0
300	30	-0.496	accept H0	accept H0	29.052	reject H0	reject H0
300	40	0.545	accept H0	accept H0	25.604	reject H0	reject H0
300	50	1.007	accept H0	accept H0	23.426	reject H0	reject H0
$\alpha = 150$							
n	$ D $	Z-value	$\beta = 95\%$	$\beta = 99\%$	F-value	$\beta = 95\%$	$\beta = 99\%$
100	10	Ztest	reject H0	accept H0	1.796	accept H0	accept H0
100	20	5.078	reject H0	reject H0	0.397	accept H0	accept H0
100	30	2.981	reject H0	reject H0	17.808	reject H0	reject H0
100	40	2.886	reject H0	reject H0	3.204	accept H0	accept H0
100	50	4.807	reject H0	reject H0	7.398	reject H0	reject H0
200	10	2.385	reject H0	accept H0	5.198	reject H0	accept H0
200	20	2.053	reject H0	accept H0	5.198	reject H0	reject H0
200	30	4.910	reject H0	reject H0	19.360	accept H0	accept H0
200	40	5.560	reject H0	reject H0	16.403	reject H0	reject H0
200	50	2.059	reject H0	accept H0	6.401	accept H0	accept H0
300	10	2.927	reject H0	reject H0	14.592	reject H0	reject H0
300	20	4.264	reject H0	reject H0	0.792	accept H0	accept H0
300	30	6.533	reject H0	reject H0	9.797	reject H0	reject H0
300	40	3.628	reject H0	reject H0	12.180	reject H0	reject H0
300	50	19.221	reject H0	reject H0	0.397	accept H0	accept H0
Mean test		$\beta = 95\%: Z_{0.025} = -1.96, Z_{0.975} = 1.96$			$\beta = 99\%, Z_{0.005} = -2.575, Z_{0.995} = 2.575$		
Var. test		$\beta = 95\%: F_{0.975}(9, 9) = 4.03, F_{0.025}(9, 9) = 0.248$			$\beta = 99\%, F_{0.995}(9, 9) = 6.541, F_{0.005}(9, 9) = 0.153$

FIGURE 15: Comparison of algorithms on the MCMRP with $\alpha = 50$. FIGURE 16: Comparison of algorithms on the MCMRP with $\alpha = 100$.

be found that TLGA can get the best solution. TLGA has a better performance than FG, MT, and SPT.

Table 1 shows the comparison in details. The average, minimal (min.), and maximal (max.) results of these GAs after running 10 times are compared. Moreover, the standard deviations (std.) of SGA and TLGA are shown. After computing the “ratio” by the formula ((cost of FG/cost of TLGA $\times 100\%$) or (cost of SGA/cost of TLGA $\times 100\%$), the

farthest-first greedy (FG) algorithm gets the ratio 106.27% on average and the SGA gets a 102.45% ratio on average.

As mentioned above, in the TLGA, the encoding method of SGA is extended by adding an AG which represents the assigned wavelengths of the paths. Moreover, the content is modified by performing the CAA to form a feasible solution. In the CAA, three heuristic algorithms: (1) the wavelength mapping algorithm, (2) the wavelength packing algorithm,

FIGURE 17: Comparison of algorithms on the MCMRP with $\alpha = 150$.

and (3) the path rerouting algorithm are called to find better wavelength assignments of the multicast tree. More crossover and mutation operators are used in the TLGA and the heuristic algorithm FG is used to generate one chromosome of the TLGA. All these improvements can help the TLGA to find better results quickly. Thus, the TLGA gets better results than the SGA and FG.

Table 2 shows the result of hypothesis statistical tests of SGA and TLGA. In this table, the difference in means and variances of results obtained by SGA and TLGA is examined. For these cases, we justify the assumptions by constructing two confidence intervals at $\beta = 95\%$ and 99% . First, the difference of means of the results obtained by performing the SGA and TLGA is examined. Z -distribution was used to estimate the difference between two means with two-side hypothesis " H_0 : the means of SGA and TLGA are the same." For confidence interval of $\beta = 95\%$ and 99% , the z -value should be in interval $(-1.96, 1.96)$ and $(-2.575, 2.575)$, respectively. If the z -value is within the interval, the means of SGA and TLGA are the same which accepts H_0 ; otherwise they are different which rejects H_0 . Observe the results shown in Table 2, in confidence interval $\beta = 95\%$ for $\alpha = 50$, two-fifths of the cases are rejected. For $\alpha = 100$, two-fifteenths of the cases are rejected and for $\alpha = 150$, all the cases are rejected. For the confidence interval $\beta = 99\%$ for $\alpha = 50$, one-third of the cases is rejected. For $\alpha = 100$, none of the cases is rejected and for $\alpha = 150$, eleven-fifteenths of the cases are rejected.

Second, in the difference of variances, the results obtained by performing the SGA and TLGA are examined. F -distribution was used to estimate the difference between two variances with two-side hypothesis " H_0 : the variances of SGA and TLGA are the same." For the confidence interval of $\beta = 95\%$ and 99% , the f -value with sampling size (10, 10)

is interval $(0.248, 4.03)$ and $(0.153, 26.541)$, respectively. If the f -value is within the interval, then the variances of SGA and TLGA are the same which accepts H_0 ; otherwise they are different which rejects H_0 . Observe the results shown in Table 2, in confidence interval $\beta = 95\%$ for $\alpha = 50$, eleven-fifteenths of the cases are rejected. For $\alpha = 100$, three-fifths of the cases are rejected and for $\alpha = 150$, eight-fifteenths of the cases are rejected. For the confidence interval of $\beta = 99\%$ for $\alpha = 50$, two-thirds of the cases are rejected. For $\alpha = 100$, eight-fifteenths of the cases are rejected and for $\alpha = 150$, two-thirds of the cases are rejected.

7. CONCLUSIONS

In this paper, the *minimal cost multicast routing problem* (MCMRP) on WDM networks with *Tap-and-Continue* (TaC) nodes is defined and studied. A new cost model which consists of the wavelength usage and communication cost is defined. The objective is to minimize the sum of the cost of used wavelengths and the communication cost of the *light-forest*. Specifically, the formulation for the WDM multicast routing problem is given. Because the MCMRP is NP-hard, two *genetic algorithms*, simple genetic algorithm (SGA) and two-level genetic algorithm (TLGA), are proposed to solve this problem.

In the proposed SGA, a *path-oriented encoding* chromosome is used to represent the routing paths. These routing paths are used to construct source-based light forests to represent a feasible solution to the multicast request. Moreover, to speed up the convergence of GAs, a *farthest-first* greedy heuristic algorithm is proposed and used to generate one of the initial chromosomes.

In the proposed TLGA, the encoding method of SGA is extended by adding an AG which represents the assigned wavelengths of the paths. Moreover, the content is modified by performing the CAA to form a feasible solution. In the CAA, three heuristic algorithms, (1) the wavelength mapping algorithm, (2) the wavelength packing algorithm, and (3) the path rerouting Algorithm, are called to find better wavelength assignments of the multicast tree. More crossover and mutation operators are used in the TLGA and the heuristic algorithm FG is used to generate one chromosome of the TLGA. All these improvements can help the TLGA to find better results quickly. Thus, the TLGA gets better results than the SGA and FG.

ACKNOWLEDGMENT

The author is very grateful to Associate Editor Dr. Shengxiang Yang and anonymous reviewers for their helpful suggestions and constructive comments.

REFERENCES

- [1] J. Y. Wei, "Advances in the management and control of optical internet," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 4, pp. 768–785, 2002.
- [2] K. H. Liu, *IP Over WDM*, John Wiley & Sons, New York, NY, USA, 2002.
- [3] W. Hu, R. Q. Hu, and Y. Qian, "Wavelength retuning in a WDM mesh network with survivable traffic grooming," in *Proceedings of the 25th IEEE International Performance, Computing, and Communications Conference (IPCCC '06)*, pp. 529–536, Phoenix, Ariz, USA, April 2006.
- [4] B. Mukherjee, *Optical Communication Networks*, McGraw-Hill, New York, NY, USA, 1997.
- [5] T. Lee, K. Lee, and S. Park, "Optimal routing and wavelength assignment in WDM ring networks," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 10, pp. 2146–2154, 2000.
- [6] N. Nagatsu, S. Okamoto, and K. Sato, "Optical path cross-connect system scale evaluation using path accommodation design for restricted wavelength multiplexing," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 5, pp. 893–902, 1996.
- [7] X. Zhang, J. Wei, and C. Qiao, "On fundamental issues in IP over WDM multicast," in *Proceedings of the 8th International Conference on Computer Communications and Networks (ICCCN '99)*, pp. 84–90, Boston, Mass, USA, October 1999.
- [8] X. Wang, M. Hou, X. Yi, and M. Huang, "A QoS multicast routing algorithm based on tabu-hierarchy genetic algorithm in IP/DWDM optical Internet," in *Network Architectures, Management, and Applications III*, K.-W. Cheung, G.-K. Chang, G. Li, and K.-I. Sato, Eds., vol. 6022 of *Proceedings of SPIE*, pp. 802–808, Shanghai, China, November 2005.
- [9] M. Ali and J. S. Deogun, "Cost-effective implementation of multicasting in wavelength-routed networks," *Journal of Light-wave Technology*, vol. 18, no. 12, pp. 1628–1638, 2000.
- [10] N. Sreenath, K. Satheesh, G. Mohan, and C. Siva Ram Murthy, "Virtual source based multicast routing in WDM optical networks," in *Proceedings of the 8th IEEE International Conference on Networks (ICON '00)*, pp. 385–389, Singapore, September 2000.
- [11] N. Sreenath, N. Krishna Mohan Reddy, G. Mohan, and C. Siva Ram Murthy, "Virtual source based multicast routing in WDM networks with sparse light splitting," in *Proceedings of IEEE Workshop on High Performance Switching and Routing (HPSR '01)*, pp. 141–145, Dallas, Tex, USA, May 2001.
- [12] A. Ding and G.-S. Poo, "A survey of optical multicast over WDM networks," *Computer Communications*, vol. 26, no. 2, pp. 193–200, 2003.
- [13] C. A. S. Oliveira and P. M. Pardalos, "A survey of combinatorial optimization problems in multicast routing," *Computers & Operations Research*, vol. 32, no. 8, pp. 1953–1981, 2005.
- [14] A. T. Haghighat, K. Faez, M. Dehghan, A. Mowlaei, and Y. Ghahremani, "GA-based heuristic algorithms for bandwidth-delay-constrained least-cost multicast routing," *Computer Communications*, vol. 27, no. 1, pp. 111–127, 2004.
- [15] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, New York, NY, USA, 1979.
- [16] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, Mich, USA, 1975.
- [17] C. Qiao, X. Zhang, and L. Zhou, "Scheduling all-to-all connections in WDM rings," in *All-Optical Communication Systems: Architecture, Control, and Network Issues II*, vol. 2919 of *Proceedings of SPIE*, pp. 218–229, Boston, Mass, USA, November 1996.
- [18] L. Kou, G. Markowsky, and L. Berman, "A fast algorithm for Steiner trees," *Acta Informatica*, vol. 15, no. 2, pp. 141–145, 1981.
- [19] P. Winter, "Steiner problem in networks: a survey," *Networks*, vol. 17, no. 2, pp. 129–167, 1987.
- [20] N. Narang, G. Kumar, and C. P. Ravikumar, "Efficient algorithms for delay bounded multicast tree generation for multimedia applications," in *Proceedings of the 6th International Conference on High Performance Computing (HiPC '04)*, vol. 1745 of *Lecture Notes in Computer Science*, pp. 169–173, Calcutta, India, December 1999.
- [21] X. Wang, H. Cheng, J. Cao, Z. Wang, and M. Huang, "QoS-driven multicast tree generation using genetic algorithm," in *Proceedings of the 5th International Workshop on Advanced Parallel Processing Technologies (APPT '03)*, vol. 2834 of *Lecture Notes in Computer Science*, pp. 404–413, Xiamen, China, September 2003.
- [22] S. B. Zhang and Z. M. Liu, "A new multicast routing algorithm based on chaotic neural networks," *Chinese Journal of Computers*, vol. 24, no. 12, pp. 1256–1261, 2001.
- [23] D. Pan, M. Du, Y. Wang, and Y. Yuan, "A hybrid neural network and genetic algorithm approach for multicast QoS routing," in *Proceedings of the International Symposium on Neural Networks (ISNN '04)*, vol. 3174 of *Lecture Notes in Computer Science*, pp. 269–274, Dalian, China, August 2004.
- [24] Q. Gao and J. Z. Luo, "A tabu-search-based fast QoS multicast routing optimal algorithm," *Journal of Software*, vol. 15, no. 12, pp. 1877–1884, 2004.
- [25] Q. Zhang and Y.-W. Leung, "An orthogonal genetic algorithm for multimedia multicast routing," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 1, pp. 53–62, 1999.
- [26] C.-F. Tsai, C.-W. Tasi, and C.-P. Chen, "A novel algorithm for multimedia multicast routing in a large scale network," *Journal of Systems and Software*, vol. 72, no. 3, pp. 431–441, 2004.
- [27] M.-T. Chen and S.-S. Tseng, "A genetic algorithm for multicast routing under delay constraint in WDM network with different light splitting," *Journal of Information Science and Engineering*, vol. 21, no. 1, pp. 85–108, 2005.

- [28] J. H. Siregar, Y. Zhang, and H. Takagi, "Optimal multicast routing using genetic algorithm for WDM optical networks," *IEICE Transactions on Communications*, vol. E88-B, no. 1, pp. 219–226, 2005.
- [29] A. T. Haghighat, K. Faez, M. Dehghan, A. Mowlaei, and Y. Ghahremani, "GA-based heuristic algorithms for QoS based multicast routing," *Knowledge-Based Systems*, vol. 16, no. 5-6, pp. 305–312, 2003.
- [30] A. Roy and S. K. Das, "QM²RP: a QoS-based mobile multicast routing protocol using multi-objective genetic algorithm," *Wireless Networks*, vol. 10, no. 3, pp. 271–286, 2004.
- [31] D.-R. Din, "Genetic algorithms for multiple multicasts on WDM ring network," *Computer Communications*, vol. 27, no. 9, pp. 840–856, 2004.
- [32] D.-R. Din, "A hybrid method for solving ARWA problem on WDM network," *Computer Communications*, vol. 30, no. 2, pp. 385–395, 2007.
- [33] L. Sanna Randaccio and L. Atzori, "Group multicast routing problem: a genetic algorithms based approach," *Computer Networks*, vol. 51, no. 14, pp. 3989–4004, 2007.
- [34] T.-C. Chiang, C.-H. Liu, and Y.-M. Huang, "A near-optimal multicast scheme for mobile ad hoc networks using a hybrid genetic algorithm," *Expert Systems with Applications*, vol. 33, no. 3, pp. 734–742, 2007.
- [35] N. Wang and G. Pavlou, "Traffic engineered multicast content delivery without MPLS overlay," *IEEE Transactions on Multimedia*, vol. 9, no. 3, pp. 619–628, 2007.
- [36] R. Bhattacharya, P. Venkateswaran, S. K. Sanyal, and R. Nandi, "Genetic algorithm based efficient routing scheme for multicast networks," in *Proceedings of IEEE International Conference on Personal Wireless Communications (ICPWC '05)*, pp. 500–504, New Delhi, India, January 2005.
- [37] N. Shimamoto, A. Hiramatsu, and K. Yamasaki, "A dynamic routing control based on a genetic algorithm," in *Proceedings of IEEE International Conference on Neural Networks (ICNN '93)*, vol. 2, pp. 1123–1128, San Francisco, Calif, USA, March-April 1993.
- [38] R.-H. Hwang, W.-Y. Do, and S.-C. Yang, "Multicast routing based on genetic algorithms," *Journal of Information Science and Engineering*, vol. 16, no. 4, pp. 885–901, 2000.