SOLVING ROUTING AND WAVELENGTH ASSIGNMENT PROBLEM WITH MAXIMUM EDGE-DISJOINT PATHS

CHIA-CHUN HSU*
Department of Transportation and Logistics Management
National Chiao Tung University
Hsinchu 300, Taiwan

HSUN-JUNG CHO AND SHU-CHERNG FANG
Department of Transportation and Logistics Management
National Chiao Tung University
Hsinchu, 300, Taiwan
and
Department of Industrial and Systems Engineering
North Carolina State University
Raleigh, NC 27606, USA

(Communicated by Louis Caccetta)

Abstract. The routing and wavelength assignment (RWA) problem in wavelength-division multiplexing optical networks is critically important for increasing the efficiency of wavelength-routed all-optical networks. Given the physical network structure and a set of connection requests, the problem aims to establishing routes for the requests and assigning a wavelength to each of them, subject to the wavelength continuous and wavelength clash constraints. The objective is to minimize the number of required wavelengths to satisfy all requests. The RWA problem was proven to be NP-hard and has been investigated for decades. In this paper, an algorithm based on the maximum edge-disjoint paths is proposed to tackle the RWA problem. Its performance is verified by experiments on some realistic network topologies. Compared with the state-of-the-art bin-packing based methods and the particle swarm optimization algorithm, the proposed method can find the best solution among all testing instances in reasonable computing time.

1. Introduction. In optical networks, information is transmitted through optical fibers operating at a speed in the order of terabits per second. Wavelength-division multiplexing (WDM) is a technology which multiplexes a number of optical carrier signals onto a single optical fiber by using different wavelengths of laser light, thus further provides users more efficient use of the huge bandwidths. Modern systems can handle up to 370 signals and can thus expand a basic 273 Gbps system over a single fiber to a bit rate over 101Tbps. In such networks, sending data from one node to another requires an establishment of connection, which can be realized by determining a path in the network between the two nodes and allocating a wavelength on all of the edges traversed by the path. The all-optical path is commonly

2010 Mathematics Subject Classification. Primary: 58F15, 58F17; Secondary: 53C35.
Key words and phrases. Edge-disjoint paths, MEDP, routing and wavelength assignment problem, RWA.
* Corresponding author.
called a lightpath and the pair of nodes asking a lightpath for information transmission is a connection request. Given an undirected network \( G = (V, E) \) and a set of connection requests \( T = \{(s_i, t_i) \in V \times V | i = 1, ..., I \text{ and } s_i \neq t_i\} \), the routing and wavelength assignment (RWA) problem attempts to route each request and to assign wavelengths to these routes subject to two constraints: (i) in the absence of wavelength converters, a lightpath must use the same wavelength on all fiber links which it traverses, this is known as the wavelength continuity constraint; (ii) lightpaths that share a common physical link cannot use the same wavelength, which is known as the wavelength clash constraint. A feasible solution of the RWA problem is composed of a set of lightpaths \( \pi = \{\pi_i | i = 1, ..., I\} \) in \( G \), each corresponding to one request \( (s_i, t_i) \in T \); and a set of wavelengths \( \omega = \{\omega_i | i = 1, ..., I\} \) for these paths. Path \( \pi_i \) and \( \pi_j \), \( i \neq j \), cannot use the same wavelength if they share a common edge. The objective is to minimize the number of utilized wavelengths to satisfy all requests in \( T \).

2. Related works on RWA. The RWA problem was proven to be NP-complete \cite{3} in 1992. The first heuristic method was proposed in \cite{3}. Since then, different heuristic methods have been developed. Reference \cite{24} covers different approaches and variants developed in the 1990s for RWA. A functional classification of RWA heuristics can be found in \cite{4}. In the literature, the approaches for solving the RWA problem can be divided into two main categories. The first one decomposes the problem into two subproblems, the routing subproblem and wavelength assignment problem \cite{8, 11, 13, 15, 18, 23} to be solved separately. The other one solves the two subproblems simultaneously \cite{21, 6, 15, 22}. The study in \cite{1} employed a multicommodity flow formulation combined with randomized rounding to calculate the route for each request. Then the wavelength assignment subproblem is solved based on the graph-coloring techniques. In which the graph, called the conflict graph? is built with one node corresponding to each request (and its route) and an edge exists between two nodes if their associated routes share one edge. The work presented in \cite{14} also used the two-phase decomposition strategy to solve the RWA problem. First, one or more candidate routes are determined for each request by the kth-shortest path algorithm. Then the wavelength assignment problem is tackled by solving an instance of the partitioning coloring problem (PCP) defined over a partitioned conflict graph. The authors proved that the decision version of PCP is NP-complete, and proposed six heuristic methods for solving PCP. In \cite{18}, the same decomposition scheme was employed, but new algorithms for each phase were proposed. In the routing phase, candidate routes are precomputed by an edge-disjoint-paths based approach. That is, several edge disjoint paths are precomputed as path candidates for each request. Next, a Tabu-search for solving PCP was proposed to solve the wavelength assignment problem. The initial feasible solution of PCP is provided by one of the six methods provided in \cite{14}, then a Tabu-search attempts to improve the solution by removing one color. The computational results indicated that the proposed Tabu search outperforms the best heuristic for PCP. Generally speaking, the routing problem may be solved by using a shortest-path algorithm, an EDP-based algorithm, or a combinatorial optimization algorithm \cite{4}. The first two types are sequential algorithms, while the last one takes a combinatorial approach. Consequently, the wavelength assignment problem can be handled by a sequential or combinatorial approach. The sequential approach sorts routes according to different schemes. For example, routes can be sorted in descending order of their lengths.
Then a wavelength is assigned to the sorted routes. For the combinatorial approach, a number of heuristic methods based on well-known graph coloring methods have been proposed. Although dividing RWA into two subproblems allows the use of existing algorithms, good solutions for each subproblem do not guarantee a good solution to the RWA problem. Hence some algorithms treat the RWA problem as an integral problem. The first such heuristic method called Greedy-EDP-RWA was developed in [15]. It employs the solution technique in [12] to solve the maximum edge-disjoint paths problem. Compared with the one in [1], Greedy-EDP-RWA was reported to run much faster to reach an equally good solution. The state-of-art heuristic for RWA was proposed in [21]. The author adapted the idea from bin-packing heuristics to the RWA problem by considering each connection request as an item and copies of the original graph as bins. The weight of an item is set to be the number of links in routing a request. To say that a bin does not have enough capacity for two items is equivalent to saying that two requests cannot be routed on the same copy of the original graph with edge-disjoint paths. Four bin-packing based heuristics were proposed in [21]: (i) first fit heuristic (FF-RWA), (ii) best fit heuristic (BF-RWA), (iii) first fit decreasing heuristic (FFD-RWA) and (iv) best fit decreasing heuristic (BFD-RWA). Computational results showed that FFD-RWA and BFD-RWA both outperform Greed-EDP-RWA [15]. Detailed descriptions of the bin-packing based algorithms will be provided later. In [17], BFD-RWA is embedded into a biased random-key genetic algorithm. A chromosome is a vector of real numbers in the interval $[0, 1]$. Each gene is associated with a connection request. The requests are sorted in a non-decreasing order in terms of the sum of their lengths and genes before BFD is applied. Computational results indicate that better solutions can be found than those obtained by a multistart variant of BFD in less time on average. In recent years, other soft computing techniques such as the particle swarm optimization (PSO) [8], artificial bee colony (ABC) [11] and memetic algorithm [6] were applied to solve the RWA problem. For the PSO and ABC algorithms, several route candidates are precomputed for each request using the $k$th-shortest paths algorithm. A particle or a population of bees represents a set of $I$ route IDs. Each ID represents a route which connects the corresponding connection request. During the search process, the route IDs are recombined according to different evolutionary scheme. Then a bin-packing based method is used to solve the wavelength assignment problem. Different local search approaches were proposed in [16] and [6]. Both works construct the initial solution by BFD-RWA [21]. In [16], a variable neighborhood descent (VND) and an iterated local search (ILS) were developed. The experimental results showed that VND-ILS is able to improve the solution quality significantly. In [6], a memetic algorithm which includes the ILS, mutation, and recombination operators was proposed. In addition, a multilevel algorithm was applied to address large size instances. The results showed that this method can be considered as the most sophisticated heuristic algorithm known in the literature.

2.1. Bin-packing based methods for RWA. The bin packing problem is a classical combinatorial optimization problem that has been widely studied in the literature. Given is a list of $I$ items of various sizes, and identical bins with a limited capacity. The problem is to pack these items into the minimum number of bins, without violating the capacity constraints. Four classical algorithms for the bin packing problem are the First Fit (FF), Best Fit (BF), First Fit Decreasing (FFD) and Best Fit Decreasing (BFD) algorithms. The FF algorithm packs each item into
the bin with the lowest index. On the other hand, the BF algorithm packs each item into the bin which leaves the least room left over after packing the item. The FFD and BFD algorithm first place larger items into bins and then fill up remaining space with smaller items.

Algorithm 1 FF (FFD) algorithm

Input:
\( G = (V, E); \)
\( T = \{(s_i, t_i)|i = 1, \ldots, l\}; \)
\( H; \)

Begin:
1. (ONLY FOR FFD, RWA: sort demands \( T \) in a non-increasing order in terms of the lengths of their shortest paths in \( G \))
2. \( \pi = \emptyset; \)
3. Create \( G_1 := G; \)
4. BINS := \{G_1\};
5. For \( i = 1 \) to \( l \)
6. \( \pi_i = \emptyset; \)
7. For \( j = 1 \) to \(|\text{BINS}|\)
8. Find shortest path \( SP_j^i \) for \( (s_i, t_i) \) in \( G_j; \)
9. If \( |SP_j^i| \leq H \) then
10. \( \pi_i = SP_j^i; \)
11. \( w_i = j; \)
12. Remove edges in \( \pi_i \) from \( G_j; \)
13. Break;
14. End if
15. End for
16. If \( \pi_i = \emptyset \) then
17. NEW = \(|\text{BINS}| + 1; \)
18. Create \( G_{\text{NEW}} := G; \)
19. BINS = BINS \cup \{G_{\text{NEW}}\};
20. Find shortest path \( SP_{\text{NEW}}^i \) for \( (s_i, t_i) \) in \( G_{\text{NEW}}; \)
21. \( \pi_i = SP_{\text{NEW}}^i; \)
22. \( w_i = \text{NEW}; \)
23. Remove edges in \( \pi_i \) from \( G_{\text{NEW}}; \)
24. End if
25. \( \pi = \pi \cup \pi_i; \)
26. \( T = T \setminus (s_i, t_i); \)
27. End for
End

Output: \( \pi \) and \( w \)

Figure 1. Pseudocode of FD(FFD) Algorithm

To apply bin-packing methods to solve the RWA problem, we must define bins, items, and their corresponding size in terms of optical networks. Skorin-Kapov of [21] considered using lightpath requests to represent items and using duplicates of graph \( G \) to represent bins. Each copy of \( G \), i.e., \( G_j, j=1,2,\ldots, \) corresponds to one wavelength. Let the size of each lightpath \( \pi_i \) be represented by the length of its shortest path \( SP_i \) in graph \( G_j \). To solve the RWA problem, we wish to pack as many items (lightpaths) into a minimum number of bins (copies of \( G \)), and hence the number of used wavelengths is minimized. The FF algorithm runs as follows. First, only one copy of \( G \), bin \( G_1 \), is created. Higher indexed bins are created as needed. Lightpath requests are selected and routed on the lowest indexed copy of \( G \) if the
length of the shortest path on such graph is less than the threshold $H$, which is set to be $\max \left\{ \text{dia}(G), \sqrt{|E|} \right\}$ as suggested in [15]. If a lightpath is routed in bin $G_j$, the lightpath is assigned wavelength $j$ and the edges along such path are removed from $G_j$. A new bin is created if no existing bin can accommodate the request. On the other hand, the FFD (and BFD) sorts the requests in a nonincreasing order in terms of the lengths of their shortest paths in $G$. In a sense, the connection requests with the longer shortest paths are usually harder to route. Thus a better solution might be obtained if these requests are considered first and then fill up the remaining space by the requests with the shorter shortest routes. The pseudocode of FF and FFD algorithms are shown in Figure 1.

The BF bin packing algorithm routes requests in the bin which they fit best. The best bin is considered to be the one in which the request can be routed on the shortest path. In other words, assume there are $B$ existing bins, bin $J$ is the best bin for lightpath request $i$ if and only if $J = \arg\min_{j=1,\ldots,B} \left\{ |SP_j^i| \right\}$, where $SP_j^i$ denotes the shortest path of request $i$ in $G_j$. The pseudocode of BF and BFD algorithm are shown in Figure 2.

2.2. Particle swarm optimization for RWA. A PSO technique for solving RWA was proposed in [8]. In order to apply PSO for solving the RWA problem, the general PSO equations are modified so that PSO can be mapped for RWA. The velocity of movement is either influenced according to the global best or local best position, but not both at the same time. The equations are as follow:

$$v_{i}^{k+1} = \alpha \times C_1 (P_{gb} - x_{i}^{k}) + (1 - \alpha) \times C_2 \times (P_{lb} - x_{i}^{k})$$

where $i = 1, \ldots, N$, and $N$ is the population size. The parameter $\alpha$ is either 0 or 1; $C_1$ and $C_2$ are social learning parameters (here we let $C_1 = C_2$), $P_{gb}$ and $P_{lb}$ represent the global best position and local best position, respectively. In the PSO for RWA, the position and velocity of a particle is represented as vectors of route ids. Before the PSO starts, a k-shortest paths algorithm is used to produce several path candidates for each of the connection requests. Each path is identified by a given unique route-id. The particle and its velocity are represented as 1-by-$I$ vectors of route ids. Each particle is attached with an edge usage table which shows the edge usage in terms of routes traversing over an edge in the network. This table helps determine which edges of the network will be overloaded if the routes of the current particle are chosen. In the PSO for RWA, velocity is also a vector of route-ids that will be replaced in the current particle according to the global or local best particle. The minus operator is redefined as follows: $(P_{gb} - x_{i}^{k})$ denotes the different routes between the gbest and the current particle. Similarly, $(P_{lb} - x_{i}^{k})$ represents the routes that are different in the lbest and the current particle. Two social learning parameters $C_1$ and $C_2$ determine the number of routes that will be replaced. The parameter $\alpha$ is used to determine whether the new velocity is affected by the global best or local best particle, but not both in a single iteration for a particle. The add operator in equation (1) is redefined: the application of velocity $v_{i}^{k+1}$ to the particle $x_{i}^{k}$ means the routes in $v_{i}^{k+1}$ will replace the corresponding routes in $x_{i}^{k}$. The particle will move to the next position $x_{i}^{k+1}$ which represents a new candidate solution to the problem. Equation (3) is used to quantify the quality of the solution.
represented by each particle of the swarm in terms of their fitness value.

\[ f(x) = \frac{1}{APL + NWL} \]  

where \( APL \) is the average path length and \( NWL \) is the number of wavelengths required to satisfy all requests. The value of \( NWL \) is obtained by calculating the number of edge disjoint path sets among the predetermined paths. The pseudocode of PSO algorithm for RWA is given in Figure 3. In addition, some strategies are proposed to improve searching ability: First, while applying velocity to a current particle, the routes that traverse the most congested edges of the network should be selected. Second, instead of replacing the route by the route of gbest (lbest), replace it with an alternative route from gbest (lbest). We refer readers to [8] for the detailed descriptions about the strategies.
Figure 3. Pseudocode of PSO Algorithm

3. Solving RWA using edge-disjoint paths. As mentioned before, since a set of EDPs can be assigned the same wavelength, maximizing the number of EDPs may lead to fewer wavelengths required to satisfy all requests. Thereby a good method for solving the maximum edge-disjoint paths problem is useful for tackling the RWA. In [9], we have developed a GA-based method to solve the MEDP problem. Compared with the multi-start greedy algorithm and ant colony optimization method presented in [2], the proposed method performs better in most instances in terms of solution quality and time. The subroutine is called $GA_{MEDP}$, which has two inputs $(G, T)$ and two outputs $(R, S)$, where $R$ denotes the set of accepted requests and $S$ is the edge-disjoint paths set. An intuitive idea of solving the RWA problem can be developed as follows. Solve the MEDP problem with given $(G, T)$ by $GA_{MEDP}$ and assign a wavelength to the accepted requests. After removing these requests from $T$, solve another MEDP in $G$ with the rest of $T$ and assign another wavelength to these accepted requests. The procedure is repeated until $T$ is empty. The pseudocode of this basic algorithm is given in Figure 4.

Figure 4. Pseudocode of a basic MEDP-RWA algorithm

The idea of the basic algorithm is used to solve the RWA problem in [15]. It’s straightforward and easy to implement. However, taking all requests in $T$ into
consideration at each step causes low efficiency, especially when \(|T|\) is large. In our proposed method, instead of dealing with the whole request set, \(T\) is divided into several batches which are solved sequentially by \(GA_{MEDP}\). The batch size denoted by \(B\) is a user-defined value. Similar to the preprocessing on the order of \(T\) in bin-packing algorithm (FFD and BFD), the shortest path of each request in \(G\) is precomputed, then \(T\) is sorted in a non-increasing order of the shortest path distances. Although these paths are unlikely to be the final routes, they still provide good information of the minimum units of resources (edges) they occupy in \(G\). Thus a better solution could be secured if the requests with longer shortest path distance are considered first and then the remaining space is filled up with the request with shorter shortest path length. After \(T\) being sorted, the first \(B\) requests in \(T\), denoted by \(T'\), are selected as a batch. The current wavelength is denoted by the variable \(wl\) initialized to be 1. Then \(GA_{MEDP}\) is executed to find the maximum number of edge disjoint paths among \(T'\) in \(G\). The current wavelength \(wl\) is assigned to the accepted requests and each of the obtained edge-disjoint paths is assigned to the corresponding lightpaths. The residual graph, where all edges used by the paths are removed, is stored in the variable \(G_{wl}\). The rejected requests at this stage remain in \(T\) and will be included in the next batch. Before starting \(GA_{MEDP}\) with the next batch, the algorithm scans all the remaining requests in \(T\) in a backward manner. Starting from the last request, which has the shortest distance of shortest path in \(G\), the algorithm tries to find a shortest path to route the request in \(G_{wl}\). If such path exists, the request is assigned wavelength \(wl\) and removed from \(T\). After the backward-scanning process is done, \(wl\) is increased by 1. Another batch of requests is selected and \(GA_{MEDP}\) is executed again. The algorithm halts when \(T\) becomes empty. We call the proposed method the \(GA_{MEDP}_{-RWA}\) algorithm (GMR). Its pseudocode is shown in Figure 5.

4. Benchmark instances. In order to evaluate the performance of the proposed method for solving RWA, numerical testing is conducted in comparison with the state-of-art bin-packing method \([21]\) and the PSO method reported in \([8]\). An instance of RWA consists of an undirected network \(G\) and a set of connection requests \(T\). For the network topology, we use 15 benchmark networks provided in \([13, 6, 11]\) which assume the patterns and sizes of some real-life telecommunication networks. The topology of NSFNET and EON are the most studied realistic networks in the literature. Network CHNNET and ARPANET are provided by \([13]\), USAnet is given by \([11]\), and other instances are taken from \([6]\). The main quantitative characteristics of these networks are shown in Table 1. Regarding the connection requests, for each network, different numbers of connection requests are randomly generated according to a given probability \(p\), i.e., the probability that there is a request between a pair of nodes. Four instances are generated for each network with \(p\) equals to 0.2, 0.4, 0.6 and 0.8, respectively. For the networks of smaller size, namely, CHNNET, NSFNET, NewYork, ARPANET, EON, France and Norway, one more instance with \(p=1\) is generated. That means every pair of different nodes of the network requests a connection between them. There are 67 testing instances in total. The last two numbers of the instances name indicate the value \(p\) that is used to generate the instance. For example, instance CHNNET_02 is generated on network CHNNET with \(p=0.2\) and CHNNET_1 is generated with \(p=1\).
Algorithm 5 GA_MEDP_RWA(GMR) algorithm

**Input:** $G = (V,E)$, $T = \{(s_i, t_i)|i = 1, ..., l\}$ and batch size $B$

**Begin:**
1. $w = \emptyset, \pi = \emptyset, \sigma = \{\sigma_i = i| i = 1, ..., l\}$;
2. $l = 1$;
3. Sort $T$ in non-increasing order of their shortest paths distance in $G$;
4. While $T \neq \emptyset$
   5. $T' = \text{Get Req}(T, B)$;
   6. $(R, S, G_i) = \text{GA_MEDP}(G, T')$;
   7. $w_j = l, \forall j \in \{k| (s_k, t_k) \in R\}$;
   8. $\pi_j = p_j, \forall j \in \{k| (s_k, t_k) \in R\}$;
   9. $T = T \setminus \{(s_j, t_j)|\forall j \in \{k| (s_k, t_k) \in R\}\}$;
   10. $\sigma = \sigma \setminus \{\sigma_m\}$;
   11. for $m = |T|$ to 1
      12. Find shortest path $SP_{sm}$ for $(s_{sm}, t_{sm})$ on $G_i$;
      13. If $|SP_{sm}| < \infty$ then
         14. $w_{sm} = l$;
         15. $\pi_{sm} = SP_{sm}$;
         16. $T = T \setminus \{(s_{sm}, t_{sm})\}$;
         17. $\sigma = \sigma \setminus \sigma_m$;
      18. end if
   19. end for
   20. $l = l + 1$;
21. end while

**End**

**Output:** $\pi$ and $w$

Figure 5. Pseudocode of GMR algorithm

Table 1. Main quantitative characteristics of testing networks

| Graph   | $|V|$ | $|E|$ | Min. | Avg. | Max. | Diameter |
|---------|------|------|------|------|------|----------|
| CHNNET  | 15   | 27   | 3    | 3.6  | 5    | 5        |
| NSF     | 16   | 25   | 2    | 3.1  | 4    | 4        |
| New York | 16  | 49   | 2    | 6.1  | 11   | 3        |
| APPANET | 20   | 32   | 3    | 3.2  | 4    | 6        |
| EON     | 20   | 39   | 2    | 3.9  | 7    | 5        |
| France  | 25   | 45   | 2    | 3.6  | 10   | 5        |
| Norway  | 27   | 51   | 2    | 3.8  | 6    | 7        |
| cost266 | 37   | 57   | 2    | 3.1  | 5    | 8        |
| janos-us-ca | 39  | 67   | 2    | 3.1  | 5    | 10       |
| giul    | 39   | 86   | 3    | 4.4  | 8    | 6        |
| piro40  | 40   | 89   | 4    | 4.5  | 5    | 7        |
| USAnet  | 46   | 75   | 2    | 3.3  | 5    | 11       |
| Germany50 | 50  | 88   | 2    | 3.5  | 5    | 9        |
| zib54   | 54   | 80   | 1    | 3.0  | 10   | 8        |
| ta2     | 65   | 108  | 1    | 3.3  | 10   | 8        |

(Min., Avg. and Max. denote the minimum, average and maximum degree, respectively)
5. Computational experiments. The computational results and comparisons with the bin-packing based heuristic and PSO are reported in Sections 5.1 and 5.2, respectively. The three algorithms are implemented in MATLAB and the experiments are conducted on a PC with Intel Core i7 CPU @1.6GHz and 4 Gb of memory running the Windows 7 operating system. For PSO and the proposed algorithm, 30 runs are conducted on each instance to obtain the best, worst and average of the objective values. Individual computational time is also recorded to give the average time. Regarding the four bin-packing methods, each method is executed once and the results and computational time are recorded.

5.1. GA_MEDP_RWA vs. bin-packing based methods. The results of the proposed method GA_MEDP_RWA (GMR) and the four bin-packing based (BPB) methods are shown in Table 2 and Table 3. The first column gives the name of the testing instance. The first three columns under GMR show the maximum, average, and minimum objective values acquired among 30 runs. The fourth column is the mean computation time. The best objective value among the five is underlined and in boldface. The first three columns under GMR show the maximum, average, and minimum objective values acquired among 30 runs. The fourth column is the mean computation time. For the FF, FFD, BF and BFD methods, let \( # w_l \) denote the objective value and \( t_{FFD}, t_{BF} \) and \( t_{BFD} \) represent the the computation time of the four methods, respectively. Since four bin-packing based methods are deterministic, only one-time execution is required for each method on each instance. Finally, for each testing instance, the best objective value is underlined and the worst one is in gray background.

Among the five solution methods, one is said to achieve the best objective value on an instance if its acquired value is less than or equal to those obtained by other methods. The numbers of times that each method achieved the best objective values among the 67 instances are summarized in Figure 6. Observe that, on all testing instances, the proposed GMR method can achieve the best objective value at least once among 30 runs. In particular, GMR can find solutions that almost dominate all other methods at all instances on graph NSF, Norway and Germany50. Regarding the GMRs individual result (of 30 runs), it reaches the best value in all 30 runs on some small testing instances, e.g., CHNNET \(_02\), NSF \(_02\), NewYork \(_02\), France \(_02\), janos-us-ca \(_02\), etc. With the growth of the instance size, GMR is still comparable to BPB methods even if it does not always find the same solution in each run. Although on four instances, namely, NewYork \(_{06}\), ARPANET \(_{02}\), ARPANET \(_{04}\) and giul \(_{02}\), the worst result (max value) is inferior to the value acquired by other methods, it happens rarely since the average value on these instances is much closer to the minimum value. To have further understanding of the individual runs on larger instances, the histograms of performance frequencies on three testing networks zib54, ta2 and Germany50, are provided in Figure 7-9 respectively. Observe that, on all instances, the difference between the upper and lower boundaries is quite small compared to that between the four BPB methods. For example, on ta \(_{08}\), GMRs solutions have objective values between 128 and 131 but the results of BPB are between 128 and 135; on zib54 \(_{08}\), GMR is between 117 and 122 but BPB is between 117 and 130, etc. Secondly, for each of the 12 instances shown in Figure 7-9 the distribution is centralized. The mode, median and mean of the distribution is quite close. Above observations confirm the effectiveness of the proposed method.
Figure 6. Number of times that the best value is achieved by different methods

Figure 7. Frequency graphs of testing results on zib54
Bin-packing based methods have clearly advantages with respect to computation time, especially on those relatively small instances. For example, the average computational time of GMR on network CHNNET and NSF can be ten to twenty times that of bin-packing based methods. However, the difference of computation time becomes smaller as the problem size grows. We take the instances on the four networks: Norway, giul, Germany50, and ta2, whose sizes are in an ascending order, to demonstrate how the computation time changes with the problem size. Define the relative differences to be \((\bar{t} - t_{bin}) / \bar{t}\), where \(\bar{t}\) is the average computation time of GMR and \(t_{bin}\) is the computation time of one of the four bin-packing based methods (bin can be FF, FFD, BF or BFD, etc.). The relative differences for all the instances on the four networks are shown in Figure 10-13 respectively.

In Figure 10 observe that the relative differences go down as the number of connection requests increases. The trend also exists in Figure 11-13 and most of other instances. In addition, the relative difference becomes smaller as the network size grows. In Figure 7 the proposed method spent less time on Germany50_08 than BFD did. In Figure 8 the same situation happened on ta2_06 and ta2_08. Since the four networks of Norway, giul, Germany50 and ta2 are in an ascending order of the network size, Figure 10-13 clearly show the phenomenon that the relative differences drop as the network size grows.
Figure 8. Frequency graphs of testing results on ta2

Figure 9. Frequency graphs of testing results on Germany50
### Table 3. Results of GMR and bin-packing based methods (cont’d)

| instance     | GA_MEDP_RWA (GMR) | FF | FFD | BF | BFD |
|--------------|-------------------|----|-----|----|-----|
| Norway_02    | 9                 | 8  | 9.26| 10 | 4.63| 9   | 4.48| S   | 2.47|
| Norway_04    | 15                | 13 | 3.57| 15 | 4.12| 15  | 4.56| S   | 5.44|
| Norway_06    | 22                | 21 | 7.87| 22 | 8.20| 22  | 8.73| S   | 12.92|
| Norway_08    | 30                | 29 | 46.41| 31 | 45.90| 41  | 19.79| S   | 24.59|
| Norway_10    | 37                | 36 | 63.77| 37 | 62.74| 38  | 28.58| S   | 36.13|
| cost26_d04   | 37                | 36 | 46.38| 37 | 47.44| 19  | 10.74| S   | 12.40|
| cost26_d06   | 35                | 34 | 159.69| 35 | 157.28| 36  | 36.35| S   | 47.28|
| cost26_d08   | 63                | 62 | 217.29| 67 | 210.73| 68  | 144.22| S   | 274.40|
| janes-ca_02   | 25                | 25 | 54.75| 25 | 56.51| 27  | 21.40| S   | 21.48|
| janes-ca_04   | 30                | 30 | 91.42| 32 | 95.82| 33  | 59.49| S   | 68.45|
| janes-ca_06   | 63                | 62 | 210.36| 71 | 119.90| 68  | 124.10| S   | 200.32|
| janes-ca_08   | 89                | 88 | 326.39| 92 | 199.09| 92  | 212.16| S   | 357.57|
| jgiu_02      | 18                | 17 | 26.95| 18 | 17.40| 19  | 8.94| S   | 11.24|
| jgiu_06      | 21                | 20 | 29.66| 19 | 29.10| 19  | 31.70| S   | 46.16|
| jgiu_08      | 36                | 34 | 226.85| 35 | 113.55| 34  | 129.20| S   | 184.70|
| jgiu_12      | 17                | 15 | 119.50| 28 | 119.98| 28  | 38.00| S   | 54.14|
| jgiu_16      | 40                | 38 | 208.81| 46 | 61.98| 46  | 68.00| S   | 84.34| 46  | 109.16|
| jgiu_20      | 60                | 59 | 339.60| 60 | 105.97| 60  | 119.35| S   | 181.52|
| jgiu_24      | 89                | 88 | 339.60| 86 | 146.64| 86  | 186.73| S   | 306.99|
| usaNet_02    | 17                | 16 | 355.64| 47 | 130.44| 45  | 149.76| S   | 225.56|
| usaNet_06    | 67                | 66 | 422.63| 68 | 244.00| 65  | 287.58| S   | 446.66| 66  | 332.89|
| usaNet_08    | 88                | 85 | 490.20| 85 | 456.69| 85  | 544.98| S   | 757.86| S   | 791.38|
| Germany_02   | 21                | 20 | 108.28| 21 | 65.86| 21  | 68.16| S   | 225.58|
| Germany_06   | 27                | 25 | 127.24| 24 | 192.70| 44  | 181.80| S   | 247.70|
| Germany_08   | 61                | 60 | 354.92| 63 | 347.70| 61  | 419.64| S   | 540.59| 65  | 544.74|
| Germany_10   | 76                | 74 | 921.59| 77 | 522.50| 75  | 725.90| 79  | 843.70| S   | 1098.40|
| zib0_02      | 32                | 31 | 149.44| 33 | 32.28| 31  | 50.04| S   | 91.61|
| zib0_06      | 67                | 65 | 436.50| 67 | 299.80| 65  | 295.60| S   | 394.65| 74  | 317.55|
| zib0_08      | 92                | 89 | 741.95| 90 | 522.85| 89  | 606.78| 98  | 898.98|
| zib0_12      | 122               | 117| 1446.41| 117 | 994.00| 117 | 1299.60| 130 | 1437.22|
| tu2_02       | 35                | 34 | 411.59| 35 | 148.93| 34  | 163.00| S   | 236.99|
| tu2_06       | 72                | 69 | 662.79| 74 | 471.30| 69  | 732.54| S   | 832.04|
| tu2_08       | 99                | 97 | 1650.20| 98 | 1411.70| 98  | 1484.00| 100 | 1871.40| 97  | 2071.40|
| tu2_10       | 131               | 128| 2835.57| 130 | 1935.40| 129 | 2577.90| 135 | 3713.46|

![Figure 10. Relative difference of computational time on Norway](image.png)
Figure 11. Relative difference of computational time on giul

Figure 12. Relative difference of computational time on graph Germany
5.2. **GA_MEDP_RWA vs. PSO.** The comparison of the proposed method and PSO is shown in Table 4 and Table 5. The best minimum objective value is underlined. Observe that PSO achieved the best values on 2 relatively small instances, while GMR achieved the best objective values on all instances. On bigger instances, the performances of PSO are not comparable at all to that of GMR. The obtained objective values of PSO are even twice more than that obtained by GMR. Regarding computation time, since a weak termination condition is set to let PSO explore more possible solutions, PSO always spent more computation time.

The difference in the path construction methods of PSO, GMR and bin-packing based methods is crucial for distinct performances. GA_MEDP takes different network conditions and request topologies into consideration and reroute some requests adaptively during the evolutionary process. For bin-packing based methods, the shortest paths are established iteratively to fit into the residual graphs to minimize the number of network resources (edges) to route each request. On the contrary, in PSO, recombining route ids from predetermined path candidates is lacking of ability to accommodate different situations. The deficiency becomes more significant on the instances of bigger network or with more requests.
### Table 4. Results of GMR and PSO (time unit: sec)

| instance   | GA_MEDP | RWA | PSO     |
|------------|---------|-----|---------|
|            | max     | min | avg     | max     | min | avg     |
| CHNNET_02  | 4       | 4   | 2.07    | 7       | 5.9 | 5       | 12.97  |
| CHNNET_04  | 5       | 4   | 3.02    | 9       | 7.5 | 7       | 18.60  |
| CHNNET_06  | 11      | 11  | 10.66   | 18      | 15.5| 13      | 30.39  |
| CHNNET_08  | 14      | 14  | 13.50   | 24      | 20.4| 18      | 48.95  |
| CHNNET_1   | 16      | 15  | 16.31   | 25      | 21.5| 18      | 78.96  |
| NSF_02     | 5       | 5   | 2.41    | 7       | 6.0 | 5       | 6.75   |
| NSF_04     | 8       | 7.3 | 5.80    | 10      | 9.0 | 8       | 13.75  |
| NSF_06     | 9       | 8   | 7.97    | 11      | 10.0| 9       | 16.19  |
| NSF_08     | 14      | 13  | 15.11   | 17      | 15.5| 14      | 29.02  |
| NSF_1      | 17      | 16  | 22.89   | 20      | 18.9| 18      | 36.03  |
| NewYork_02 | 2       | 2   | 1.37    | 3       | 2.9 | 2       | 7.37   |
| NewYork_04 | 3       | 3   | 2.65    | 6       | 5.2 | 4       | 18.78  |
| NewYork_06 | 5       | 4   | 3.93    | 8       | 6.8 | 6       | 25.51  |
| NewYork_08 | 7       | 6   | 6.75    | 10      | 9.1 | 8       | 38.35  |
| NewYork_1  | 8       | 8   | 7.03    | 12      | 10.7| 10      | 50.17  |
| ARPANET_02 | 10      | 9   | 10.58   | 13      | 11.3| 10      | 21.30  |
| ARPANET_04 | 13      | 12  | 16.96   | 19      | 16.3| 14      | 62.15  |
| ARPANET_06 | 21      | 21  | 25.98   | 28      | 24.9| 22      | 93.86  |
| ARPANET_08 | 29      | 29  | 32.31   | 40      | 36.5| 33      | 133.70 |
| ARPANET_1  | 33      | 33  | 62.85   | 46      | 41.2| 36      | 172.60 |
| EON_02     | 4       | 3.1 | 3.19    | 5       | 4.3 | 4       | 10.40  |
| EON_04     | 9       | 8   | 14.74   | 12      | 10.6| 10      | 35.34  |
| EON_06     | 11      | 11  | 17.52   | 15      | 13.0| 12      | 46.61  |
| EON_08     | 14      | 13  | 26.63   | 20      | 18.2| 17      | 48.92  |
| EON_1      | 19      | 18  | 38.17   | 25      | 22.9| 22      | 73.32  |
| France_02  | 8       | 8   | 8.87    | 14      | 11.5| 10      | 50.60  |
| France_04  | 13      | 12  | 16.05   | 22      | 19.3| 17      | 86.48  |
| France_06  | 22      | 22  | 39.07   | 34      | 30.4| 27      | 129.16 |
| France_08  | 27      | 26  | 47.09   | 45      | 40.3| 38      | 290.72 |
| France_1   | 34      | 34  | 60.09   | 54      | 48.9| 45      | 236.64 |
Table 5. Results of GMR and PSO (con’t)

| instance         | GA | MEDP | RWA | PSO | max | avg | min | avg time | max | avg | min | avg time |
|------------------|----|------|-----|-----|-----|-----|-----|---------|-----|-----|-----|---------|
| Norway_02        | 9  | 8.6  | 8   | 9.26| 15  | 13.1| 11  | 61.56   | 25  | 22.5| 20  | 101.40  |
| Norway_04        | 15 | 14.6 | 14  | 18.47| 25  | 22.5| 20  | 101.11  |
| Norway_06        | 22 | 21.4 | 21  | 32.84| 37  | 32.6| 29  | 202.11  |
| Norway_08        | 30 | 29.5 | 29  | 46.41| 48  | 43.6| 40  | 254.75  |
| Norway_1         | 37 | 36.6 | 36  | 63.77| 59  | 54.1| 49  | 323.64  |
| cost266_02       | 19 | 18.2 | 18  | 46.68| 26  | 23.8| 21  | 105.22  |
| cost266_04       | 35 | 34.1 | 33  | 159.69| 54  | 50.9| 47  | 239.95  |
| cost266_06       | 54 | 53.1 | 53  | 237.55| 79  | 73.2| 70  | 436.38  |
| cost266_08       | 68 | 67.2 | 67  | 275.29| 99  | 93.4| 88  | 703.53  |
| janos-us-ca_02   | 26 | 26.0 | 26  | 54.75| 40  | 36.1| 31  | 288.18  |
| janos-us-ca_04   | 40 | 39.6 | 39  | 97.42| 66  | 60.5| 56  | 529.74  |
| janos-us-ca_06   | 68 | 67.8 | 67  | 210.36| 111 | 102.7| 94  | 959.93  |
| janos-us-ca_08   | 89 | 88.2 | 88  | 326.39| 144 | 134.1| 122 | 1239.46 |
| giul_02          | 11 | 10.3 | 10  | 26.93| 16  | 14.5| 13  | 294.44  |
| giul_04          | 21 | 20.2 | 19  | 82.66| 32  | 28.8| 26  | 451.61  |
| giul_06          | 25 | 24.6 | 24  | 150.73| 42  | 38.3| 34  | 738.95  |
| giul_08          | 36 | 34.9 | 34  | 226.85| 59  | 53.5| 49  | 990.63  |
| piro40_02        | 17 | 17   | 17  | 55.51| 22  | 20.0| 19  | 159.88  |
| piro40_04        | 28 | 28   | 28  | 118.50| 36  | 33.7| 32  | 300.88  |
| piro40_06        | 46 | 46   | 46  | 208.81| 66  | 58.7| 52  | 474.73  |
| piro40_08        | 60 | 60   | 60  | 338.61| 86  | 79.4| 73  | 684.40  |
| USAnet_02        | 25 | 24.0 | 23  | 86.13| 40  | 36.4| 32  | 446.56  |
| USAnet_04        | 47 | 45.9 | 45  | 355.64| 76  | 69.5| 65  | 983.93  |
| USAnet_06        | 67 | 66.1 | 65  | 422.63| 106 | 98.1| 92  | 1418.73 |
| USAnet_08        | 88 | 86.5 | 85  | 490.20| 139 | 126.5| 120 | 1715.20 |
| Germany50_02     | 21 | 20.7 | 20  | 109.28| 37  | 33.5| 31  | 379.56  |
| Germany50_04     | 45 | 44.2 | 43  | 350.24| 74  | 69.4| 66  | 683.95  |
| Germany50_06     | 61 | 60.5 | 60  | 584.92| 102 | 95.7| 91  | 969.87  |
| Germany50_08     | 76 | 74.7 | 73  | 921.59| 126 | 120.5| 115 | 1271.90 |
| zib54_02         | 32 | 31.1 | 30  | 149.44| 75  | 64.1| 58  | 619.63  |
| zib54_04         | 67 | 65.7 | 65  | 406.90| 159 | 141.6| 133 | 1462.19 |
| zib54_06         | 92 | 90.0 | 88  | 762.56| 208 | 195.2| 187 | 2955.29 |
| zib54_08         | 122 | 119.2 | 117 | 1446.41| 289 | 266.3| 230 | 4147.09 |
| ta2_02           | 35 | 34.6 | 34  | 411.59| 75  | 70.3| 66  | 1453.68 |
| ta2_04           | 72 | 70.1 | 69  | 945.79| 152 | 141.8| 133 | 2852.21 |
| ta2_06           | 99 | 97.4 | 96  | 1865.20| 208 | 198.7| 181 | 3867.39 |
| ta2_08           | 131 | 129.7 | 128 | 2835.57| 284 | 271.0| 258 | 5953.52 |
6. Conclusions and future work. This paper has presented an algorithm for solving the RWA problem. The algorithm combines the ideas of the bin-packing method and edge-disjoint paths together. It considers only a number of requests at a time and solves the corresponding MEDP problem. The remaining requests are then scanned in a backward manner such that the one with shorter shortest paths tries to fit into the existing wavelengths first. Then a new batch of requests is selected and the MEDP problems is solved again. In such manner, the algorithm solves RWA back and forth until a solution is constructed. The performance of the proposed GMR method has been verified by several experiments on realistic network topologies. The experimental results have shown that, compared with the bin-packing based methods and PSO, GMR can find the best solution in all testing instances. Although it takes longer computational time than the bin-packing methods for small instances (e.g., ten to twenty times that of bin-packing based methods on network CHNET and NSF), the relative difference of computational time goes down as the problem size grows. In this paper, the proposed algorithm is capable of solving small and medium size instances in reasonable time. However, for very large instances (for example, |T| > 1,000,000), finding a solution is extremely time-consuming. A divide-and-conquer approach called the multilevel algorithm may be worth studying to enhance the proposed method for solving large-size problems. Finding different ways to identify good permutations of the request set and developing local search methods are also needed to further improve the algorithm.

Acknowledgements. This research was supported by MOST, Taiwan, R.O.C. under grant MOST 103-2622-E-009-012. We thank our colleagues who provided insight and expertise that greatly assisted the research. We would also like to show our gratitude to the editor and anonymous reviewers for their constructive suggestions during the review process.

REFERENCES

[1] D. Banerjee and B. Mukherjee, [A practical approach for routing and wavelength assignment in large wavelength-routed optical networks] IEEE Journal on Selected Areas in Communications, 14 (1996), 903–908.
[2] M. J. Blesa and C. Blum, Finding edge-disjoint paths in networks: An ant colony optimization algorithm Journal of Mathematical Modeling and Algorithms, 6 (2007), 367–391.
[3] I. Chlamtac, A. Ganz and G. Karmi, Lightpath communications: an approach to high-bandwidth optical WAN’s IEEE Transactions on Communications, 40 (1992), 1171–1182.
[4] J. S. Choi, N. Golmie, F. Lapeyrere, F. Mouveaux and D. Su, A functional classification of routing and wavelength assignment schemes in DWDM networks: Static Case, In Proceedings of the 7th International Conference on Optical Communications and Networks, (2000), 1109–1115.
[5] H. Choo and V. V. Shakhov, [Routing and wavelength assignment in optical WDM networks with maximum quantity of edge disjoint paths] Computational Science - ICCS 2004, 3038 (2004), 1138–1145.
[6] T. Fischer, K. Bauer, P. Merz and K. Bauer, Solving the routing and wavelength assignment problem with a multilevel distributed memetic algorithm Memetic Computing, 1 (2009), 101–123.
[7] X. Guan, S. Guo, W. Gong and C. Qiao, A new method for solving routing and wavelength assignment problems in optical networks Journal of Lightwave Technology, 25 (2007), 1895–1909.
[8] A. Hassan, Particle Swarm Optimization for Routing and Wavelength Assignment in Next Generation WDM Networks, Ph.D thesis, Queen Mary University of London, 2010.
[9] C.-C. Hsu and H.-J. Cho, [A genetic algorithm for the maximum edge-disjoint paths problem] Neurocomputing, 148 (2015), 17–22.
[10] C.-C. Hsu, H.-J. Cho and S.-C. Fang, Routing and wavelength assignment in optical networks from maximum edge-disjoint paths, *Genetic and Evolutionary Computing: Advances in Intelligent Systems and Computing*, 238 (2014), 95–103.

[11] Y. S. Kavian, A. Rashedi, A. Mahani and Z. Ghassemlooy, Routing and wavelength assignment in optical networks using artificial bee colony algorithm, *European Journal of Operational Research*, 124 (2013), 1243–1249.

[12] J. Kleinberg, *Approximation Algorithms for Disjoint Paths Problems*, Ph.D thesis, Massachusetts Institute of Technology, 1996.

[13] P. Leesutthipornchai, C. Charusripinyo and N. Wattanapongsakorn, Solving multi-objective routing and wavelength assignment in WDM network using hybrid evolutionary computation approach, *Computer Communications*, 33 (2010), 2246–2259.

[14] G. Li and R. Shmha, The partition coloring problem and its application to wavelength routing and assignment, in *Proceedings of the First Workshop on Optical Networks*, 2000.

[15] P. Manohar, D. Manjunath and R. K. Shevgaonkar, Routing and wavelength assignment in optical networks from edge disjoint path algorithms, *IEEE Communications Letters*, 6 (2002), 211–213.

[16] A. X. Martins, C. Duhamel, P. Mahey, R. R. Saldanha and M. C. de Souza, Variable neighborhood descent with iterated local search for routing and wavelength assignment, *Computers and Operations Research*, 39 (2012), 2133–2141.

[17] T. F. Noronha, M. G. C. Resende and C. C. Ribeiro, A biased random-key genetic algorithm for routing and wavelength assignment, *Journal of Global Optimization*, 50 (2011), 503–518.

[18] T. F. Noronha and C. C. Ribeiro, Routing and wavelength assignment by partition colouring, *European Journal of Operational Research*, 171 (2006), 797–810.

[19] A. E. Ozdaglar and D. P. Bertsekas, Routing and wavelength assignment in optical networks, *IEEE/ACM Transactions on Networking*, 11 (2003), 259–272.

[20] R. Poli, J. Kennedy and T. Blackwell, Particle swarm optimization - An overview, *Swarm intelligence*, 1 (2007), 33–57.

[21] N. Skorin-Kapov, Routing and wavelength assignment in optical networks using bin packing based algorithms, *European Journal of Operational Research*, 177 (2007), 1167–1179.

[22] Y. Wang, T. H. Cheng and M. H. Lim, A Tabu search algorithm for static routing and wavelength assignment problem, *IEEE Communications Letters*, 9 (2005), 841–843.

[23] W. J. Yoon, D. H. Kim, M. Y. Chung, T. J. Lee and H. Choo, Routing with maximum EDPs and wavelength assignment with path conflict graphs. In *Proceedings of the 2006 international conference on Computational Science and Its Applications*, 3981 (2006), 856–865.

[24] H. Zang, J. P. Jue and B. Mukherjee, A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks, *Optical Networks Magazine*, 1 (2000), 47–60.

Received October 2014; 1st revision August 2015; final revision August 2016.

E-mail address: babamama711025@gmail.com
E-mail address: hjcho001@gmail.com
E-mail address: fang@ncsu.edu