An Efficient Approach for Message Routing in Optical Omega Network

Monir Abdullah, Mohamed Othman and Rozita Johari

Department of Communication Technology and Network, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia monabdullah@hotmail.com and {mothman, rozita}@fsktm.upm.edu.my

Abstract

Optimal routing in Optical Omega Network (OON) is an NP-hard problem. A major problem called crosstalk is introduced by OON, which is caused by coupling two signals within a Switching Element (SE). Many heuristic algorithms were designed by many researchers to perform this routing such as sequential algorithm, degree-descending algorithm. Routing the messages in degree-decreasing of the message conflicts gave best performance among them. The Simulated Annealing (SA) algorithm also was used to improve the performance of solving the problem and gave best result. This paper presents an efficient approach for message routing in OON. The proposed approach that combines of SA algorithm with the best heuristic algorithms gave much better result in a very minimal time.

Keywords: Simulated annealing, Optical omega network, Crosstalk problem.

1. Introduction

OON has been used in telecommunication and parallel computing systems for many years. This network consists of N inputs, N output, and n stages (n=log₂ N). Each stage has N/2 SEs, each SE has two inputs and two outputs connected in a certain pattern [16]. As optical technology advances, there is a considerable interest in using optical technology to implement interconnection networks and switches [13]. A major problem called crosstalk is introduced by OON, which is caused by coupling two signals within a SE. To avoid crosstalk problem, various approaches have been proposed by many researchers to transfer messages from a source to a destination address. These will need the division of the messages into several groups. Then, deliver the messages using one time slot (pass) for each group. In other words, the less passes the transfer has, and the better the performance.

Optimal routing in OON is an NP-hard problem. Previously, four heuristic algorithms [11] were used to simulate the performance in real time. Degree-decreasing algorithm gave best result and the sequential down algorithm is the second. Genetic algorithm [7] was proposed to improve the performance but the drawback of the genetic algorithm is its long running time. Also ant colony was proposed for this problem [17], SA was used for optimization [1]. SA gives the best result among all previous algorithms [2] but in this research, unsuitable SA parameters were used.

The main contribution of this paper is the successful combination of SA algorithm to the best heuristic algorithms to the routing problem. Our approach improved the
performance of the routing in terms of average number of passes and the running time of the algorithm.

The rest of the paper is organized as follows. In section 2 a formal description of OON is presented, whereas, section 3 reviews the different heuristic algorithms available for routing. In section 4, the SA algorithm to the problem is discussed. Section 5 gives a description of the proposed approach to the routing problem, and section 6 provides simulation results to evaluate the performance of the proposed approach. Conclusions as well as future work are made in the last section.

2. Optical Omega Network

OON has a shuffle-exchanged connection pattern [9]. To connect the source address to the destination address, the address is shifted one bit to the left circularly in each connection such as source to the first stage, one stage to the next stage, etc. For example in an 8*8 network each connection between the stages is shuffle-exchanged as shown in Fig. 1

\[
\begin{align*}
000 &\rightarrow 100 \\
001 &\rightarrow 001 \\
010 &\rightarrow 010 \\
011 &\rightarrow 001 \\
100 &\rightarrow 000 \\
101 &\rightarrow 101 \\
110 &\rightarrow 110 \\
111 &\rightarrow 111
\end{align*}
\]

Fig. 1: Shuffle-exchange

To route a permutation in OON, we consider the shuffle exchange connection pattern. Sending all the 8 inputs to 8 outputs in one time slot (pass), can be done as shown in Fig. 2.

Fig. 2: Permutation in an 8*8 Omega network

From Fig. 2, we can see that there exists crosstalk in all the SE if we want to route all the inputs to the outputs in one pass.

2.1. Crosstalk in OON

Crosstalk [4] occurs when two signals channels interact with each other. When a crosstalk happens, a small fraction of the input signal power may be detected at another output although the main signal is injected at the right output. For this reason, the input signal will be distorted at the output due to the loss and crosstalk introduced on the path [13]. The channels carrying the signals could cross each other in order to embed a particular topology. This is shown as Fig. 3:
Fig. 3: Two type of switching connections

Each SE can be in two connecting schemes as shown in Fig. 3. Since these two ways will cause crosstalk, what we need to do is to avoid these situations to happen in all the SEs.

2.2. Approaches to avoid crosstalk

To reduce the negative effect of crosstalk, many approaches have been proposed. One way to solve crosstalk is to use a 2N*2N regular OON to provide the N*N connection [12], which is space domain approach. But half of the inputs and outputs are wasted in this approach. Another more efficient solution, is to route traffic through an N*N optical network to avoid coupling two signal within each SE. This idea can be implemented using time domain approach [7]. This means, in every SE, there is only one of the following legal passing ways as shown in Fig. 4.

2.3. Time domain Approach

Time domain approach [14] is used to route all the inputs in several groups (passes), such that no crosstalk will be caused in each pass. Since we can only pass a message in a SE in one of the four ways in Fig. 4, we can see that there is no way to realize a permutation in a single pass through a optical network without crosstalk. The reason is at least the two input links on an input switch or the two output links on an output switch cannot be active in the same pass [13]. So, we need to use at least two passes to realize a permutation [5]. For example, we can route the 8 messages shown in Fig. 1 in two passes as shown in Fig. 5. In this research, we deal with randomly generated permutations, which have random source addresses and random destination addresses. The goal of our research is to minimize the number of passes to route all the inputs to the outputs without crosstalk.
3. Different Heuristic Routing Algorithms

3.1. The Window Method

Window Method [9] is used to find the conflicts among the messages to be sent. It can be described roughly as follows. Given a permutation, we combine each source address and its corresponding destination address to produce a matrix. The optical window size is the m-1 where m=log2 N and N is the size of the network. We use this window on the produced matrix from left to right except the first column and last column. If two messages have the same bit pattern in any of the optical window, they will cause conflict in the network. That means they cannot be in the same group, hence, they have to be routed in different passes. To see how the Window Method works, let's look at the following example as in Fig. 6. Suppose the network size is 8.

<table>
<thead>
<tr>
<th>Step 1 (window 0):</th>
<th>Step 2 (window 1):</th>
<th>Step 3 (window 2):</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 1 0 0 0 0</td>
<td>0 0 1 0 0 0 0 0</td>
<td>0 0 0 1 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 1 0 0 0 0</td>
<td>0 0 1 0 0 0 0 0</td>
<td>0 0 0 1 0 0 0 0</td>
</tr>
<tr>
<td>0 1 0 0 1 0 0 0</td>
<td>0 1 0 0 1 0 0 0</td>
<td>0 1 0 0 1 0 0 0</td>
</tr>
<tr>
<td>0 1 0 0 1 0 0 0</td>
<td>0 1 0 0 1 0 0 0</td>
<td>0 1 0 0 1 0 0 0</td>
</tr>
<tr>
<td>1 0 0 0 0 0 0 0</td>
<td>1 0 0 0 0 0 0 0</td>
<td>1 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 0 1 0 1 0 0 0</td>
<td>1 0 1 0 1 0 0 0</td>
<td>1 0 1 0 1 0 0 0</td>
</tr>
<tr>
<td>1 1 0 1 1 0 1 0</td>
<td>1 1 0 1 1 0 1 0</td>
<td>1 1 0 1 1 0 1 0</td>
</tr>
<tr>
<td>1 1 1 1 1 1 1 1</td>
<td>1 1 1 1 1 1 1 1</td>
<td>1 1 1 1 1 1 1 1</td>
</tr>
</tbody>
</table>

Fig. 6: Window Method

3.2. Routing Algorithms

The basic idea of the routing algorithm is as follows:

While (not end of messages list)
   Select one of the left messages;
   Schedule the message in a time slot with no conflict with other messages that have been already scheduled;

3.3. Four Heuristic Routing Algorithms

There are many ways to decide the order of the scheduling. The four heuristic algorithms choose the message in the following way:

1. Choose a message sequentially in increasing order of the message source address.
2. Choose a message sequentially in decreasing order of the message source address.
3. Choose a message based on the order of increasing degrees in the conflict graph.
4. Choose a message based on the order of decreasing degrees in the conflict graph.

The purpose of these routing algorithms is to schedule the messages in different passes in order to avoid the path conflicts in the network [11]. The degree of each message in the conflict graph is the number of conflicts to other messages it has in the conflict graph. Scheduling the messages in decreasing
degrees of the message conflicts gave the best performance among these four algorithms.

4. Simulated Annealing

Simulated Annealing originated in the annealing processes found in thermodynamics and metallurgy [2,7]. Simulated Annealing was introduced by [3] and is used to approximate the solution of very large combinatorial optimization problems [8] (e.g. NP-hard problems). It is based upon the analogy between the annealing of solids and solving optimization problems. When SA was first proposed [6], it was mostly known for its effectiveness in finding near optimal solutions for large-scale combinatorial optimization problems [9], such as the traveling salesperson problem, buffer allocation in production lines [15], and chip placement problems in circuits [10].

Simulated Annealing was derived from physical characteristics of spin glasses [6]. The principle behind simulated annealing is analogous to what happens when metals are cooled at a controlled rate. The slowly falling temperature [3] allows the atoms in the molten metal to line them up and form a regular crystalline structure that has high density and low energy. But if the temperature goes down too quickly, the atoms do not have time to orient themselves into a regular structure and the result is a more amorphous metal with higher energy. See Fig. 8 and Fig. 9 below.

Perfect= all atoms lined up on crystal lattice sites, no defects.
Perfect= this is the lowest energy “state” for this set of atoms: Fig. 8

4.1. Performance of SA over four Algorithms

SA algorithm improved the performance in term of average number of passes [1]. SA gave the best results. As initial solution for SA algorithm, sequential algorithm was used to generate the next solution. Also to generate the new solution per iteration, sequential algorithm was used [1].

Move sets are the most important operators in SA algorithm. There are three types of move sets for SA Inversion, Transition and Switching. Inversion move set is mostly suitable for application in SA to generate a new solution [1].
An Efficient Approach for Message Routing in Optical Omega Network

1. **Sequential** algorithm to compute current gain as initial solution
2. Initialize the Current Energy $E$, to be equal the current gain
3. While ($t_{stop}$>0 and $T$>$T_{final}$)
4. For $i$=0 to $M$ do step 5 through 9.
5. Inversion move set
6. **Sequential** Algorithm to compute new gain.
7. If the new gain $\leq$ current gain then the solution is accepted.
   
   Else
   
   $R$=Random Number(0<$R$<1),
   
   $Y$=exp($-\Delta E/T$),
   
   If($R$<$Y$), then accept the solution,
   
   Else reject it.
8. If we accept the solution then Current energy will be set to the new energy.
   
   Else the old solution will kept
9. If we accept the new solution then $t_{stop}$=$t_s$.
   
   Else decrement $t_{stop}$ by 1 ($t_{stop}$=$t_{stop}$-1).
10. Change the temperature by a factor $T$=$T^\alpha$.
11. **End of while**

---

**Fig. 10:** SA algorithm to the problem

---

5. **The Proposed Approach**

All previous researchers proved that degree descending algorithm is the best algorithm among four heuristic algorithms. Also they proved that sequential down algorithm is better than sequential algorithm. In our approach SA and best heuristic algorithms will be combined. A good result will be taken by using degree descending as initial solution to SA. In another word, the result will be optimized before using SA. Because of these points, our approach is combination successfully SA with these two heuristic algorithms. The sorted vertice in degree descending order is taken in this research as the initial solution to the problem. Sequential down will be used to proceed the SA every iterations. Our approach is shown in Fig. 11.
1. **Descending** algorithm to compute current gain as initial solution
2. Initialize the Current Energy $E_c$ to be equal the current gain
3. While ($t_{stop} > 0$ and $T > T_{final}$)
   4. For $i=0$ to $M$ do step 5 through 9.
5. Inversion move set
6. **Sequential down** Algorithm to compute new gain.
   7. If the new gain $\leq$ current gain then the solution is accepted.
   Else
   
   $R = \text{Random Number}(0<R<1)$,
   $Y = \exp(-\Delta E/T)$,
   If($R<Y$), then accept the solution,
   Else reject it.
8. If we accept the solution then Current energy will be set to the new energy,
   Else the old solution will kept
9. If we accept the new solution then $t_{stop}=t_r$,
   Else decrement $t_{stop}$ by 1 ($t_{stop}=t_{stop}-1$).
10. Change the temperature by a factor $T=T^\alpha$.
11. End of while

---

Fig. 11: The Proposed Approach

The number of passes is calculated using degree descending and then assigns it to the current gain. From this point the best result is taken (the minimum number of passes) before entering the SA algorithm. It's better to start with a good solution that has been heuristically built.

Sequential algorithm was used to proceed the SA every iterations. But in our approach, sequential down will be used because as we mentioned before it's better than sequential algorithm. Inversion move set will be used to generate a new solution.

A good combination of SA with best heuristic algorithms built suitable SA parameters.

6. **Analysis of the Test Result**

Many cases showed that degree-descending algorithm is better than sequential algorithm to be as an initial solution to proceed SA algorithm. All researchers proved that degree-descending algorithm is the best among four heuristic algorithms. SA algorithm was proved that it can improve the performance 0.768 passes in average over the sequential algorithm but only 0.157 passes over the degree-descending algorithm [1]. From this point we proposed degree-descending algorithm to be as an initial solution to proceed SA algorithm.

Many cases also showed that sequential down algorithm is better than sequential algorithm to proceed SA algorithm. This example shows that sequential down algorithm can even reduce two passes than sequential algorithm. It is in an 8*8 network. The original conflict matrix is shown as in Table 1.
The solution of sequential algorithm has four passes: 1 1 1 2 2 3 2 4 but the solution of the sequential down algorithm has two passes: 2 1 2 2 1 2 1 1. Using the SA algorithm directly will take two passes as initial solution. But if we start with sequential algorithm, the SA will start with four passes then will take a lot of time to improve the solution. All researchers proved that sequential down algorithm is better than sequential algorithm.

Also [1] was proved that SA algorithm can improve the performance 0.768 passes in average over the sequential algorithm but 0.617 passes over the sequential down algorithm. From this point also we proposed sequential down to proceed SA algorithm every iterations.

Our approach directly gets the best results. After many careful observations and the testing of many cases the number of rounds was set at 100, starting temperature at 1000, final temperature at 0.05, cooling rate at 0.9, the number of iterations per temperature at only 5, and if there is no improvement the stop condition at only 5. Too low values for these parameters resulted in high values for the average number of passes. When the values of these parameters were made too high, there was no improvement in the average number of passes and the time complexity increased. Therefore, after many careful observations with different network sizes, the preceding values were chosen to be optimum for generating a good average value for the number of passes with a good low-time complexity.

To determine how the number of iterations per temperature affects the results, some other cases are tested with different values of iterations per temperature. A 16*16 network is used, varied the temperature cooling rate, and employed the optimum values of 1000 for the starting temperature, a final temperature of 0.05, and a temperature cooling rate of 0.9. The number of rounds was set at 1000. Also, the average number of passes is decreased when the iterations/temperature values grow larger.

This is because at the same temperature, we iterate for some time to find the best solution. If this value is high, then the chance of obtaining a good solution is also high. The decreasing line shows that the result becomes better with increasing iterations/temperature values. But as we see, after a certain point, even if the iterations/ temperature is increased there is not much if any change in the value of the average number of passes. If we keep trying values less than the optimum values for the parameters, then the average number of passes obtained is not good. Even too greater values too much longer would result in increased running time without much improvement in the value of the average number of passes. The results are shown in Fig. 12.
Thus, from this series of test results, it can be concluded that the our approach gives the best solutions when the starting temperature is around 1000, final temperature is around 0.05, temperature-cooling rate is around 0.9, and the number of iterations per temperature is around 5. The algorithm performance on different size of networks is shown in Fig. 13.

We get that the largest difference between descending algorithm and our approach is 0.275. In other words, our approach can improve 0.275 pass over descending algorithm.

From Fig. 13, we see that our approach performs the best result.
Also our approach decreases the running time as follow:

1. The number of iterations is decreased because our approach gets the optimal at the beginning.

2. The stop condition is decreased. Because of previous points we can say our approach faster than SA algorithm. Fig. 15 shows the running time of SA and our approach.

![Performance of running times of diff. algorithm for different sizes of network](image)

**Fig. 15:** Running Time of SA and our approach

### 7. Conclusion and Future Work

In this paper, we have presented an efficient approach for message routing in optical omega network. Our approach improved the number of performance of the routing. A good combination of SA and the best heuristic algorithms saves a lot of time. Also good parameters of our approach makes the algorithm directly gives the least number of passes for message sending without crosstalk. The most obvious advantage of using our approach over the simulated annealing algorithm is that it improves the performance in both average number of passes and running time. Our future research topic is designing a new parallel algorithm of the proposed approach.

### 8. References


