Message Routing and Scheduling in Optical Multistage Networks using Bayesian Inference method on AI algorithms

Ajay K Katangur[1], Somasheker Akkaladevi[2]

[1]Texas A&M University - Corpus Christi
Department of Computing Sciences
Corpus Christi, Texas 78412, USA
ajay.katangur@tamucc.edu

[2]Virginia State University
Department of Computer Information Systems
Petersburg, Virginia 23806, USA
sakkaladevi@vsu.edu

Abstract

Optical Multistage Interconnection Networks (MINs) suffer from optical-loss during switching and crosstalk problem in the switches. The crosstalk problem is solved by routing messages using time division multiplexing (TDM) approach. This paper focuses on minimizing the number of groups (time slots) required to realize a permutation. Many researchers concentrated on this NP-hard problem and concluded that AI algorithms perform better than the heuristic algorithms. They also showed that majority of the times the performance of Genetic Algorithm (GA) was better than Simulated Annealing Algorithm (SAA). In this research, we implement a new approach to minimize the number of passes required for scheduling a given permutation. A combinational method is developed which comprises the use of Bayesian inference method on GA and SAA to always guarantee the best solution, instead of only using either GA or SAA. Simulations are performed in java using multiple threads to run SA and GAA in parallel and to evaluate the performance of the new method. The results are then compared to those obtained from GA and SAA.

1. Introduction

The most commonly used networks for switching and communication applications are the Multistage Interconnection Networks (MINs), and these are used in the telecommunications industry and also in parallel computing systems for many years. The network has N inputs, and N outputs [15]. This network consists of n stages (n = $\log_2 N$) composed of N/2 switching elements (SEs) in each stage. Each SE has four terminals; two for the inputs and two for the outputs connected in a certain pattern. The most widely used MINs are the electronic MINs, where the signals are transmitted by varying the voltage for logic 1 and logic 0 [10]. With ever growing networks, the demand for bandwidth is increasing, and optical technology is used to implement interconnection networks [14] and switches as this technology supports high bandwidth and speed [9, 12]. In optical MINs (OMIN) light is used to transmit the messages through fiber-optic cable [14].

The electronic MINs and the optical MINs have many similarities [10, 15], but there are some fundamental differences between them such as the optical-loss during switching [10] and the crosstalk problem in the switches [11].

In this research, we consider the Omega network pattern for the MIN, which has a shuffle-exchange connection pattern [12]. It has N inputs, N outputs and n stages where $n = log_2N$. To transfer messages from a source address to a destination address on an optical Omega network without crosstalk, we need to divide the messages into several groups and then, deliver the messages using one time slot (pass) for each group, which is called the time division multiplexing (TDM) [11]. In each group, the paths of the messages going through the network are crosstalk free. So, from the performance aspect, we want to separate the messages into groups such that no message conflicts arise with each other in the same group as well as we want to reduce the total number of the groups [12]. In other words, the fewer passes the transfer has, the better the performance of the optical network. Many researchers concentrated on this problem of routing the messages using the minimum number of passes, and came up with many different ways of scheduling the messages. The most popular methods are the heuristic algorithms such as the sequential increasing algorithm (SIA), sequential decreasing algorithm (SDA), degree increasing algorithm (DIA), degree descending

1-4244-0910-1/07/$20.00 ©2007 IEEE.

algorithm (DDA), and artificial intelligence (AI) methods such as the genetic algorithm (GA) [3, 8]and simulated annealing algorithm (SAA) [2, 3, 7]. The research results proved that out of the heuristic algorithms the DDA performance was better than the other algorithms [6]. The results concluded that the performance of both the GA and SAA was better than the heuristic algorithms. Out of the AI algorithms GA performed slightly better than the SAA in terms of the number of passes calculated, where as GA took long time than SAA to schedule the messages in groups without conflicts [6]. The researchers concluded that SAA is better than GA as it takes lesser time; although GA beat it in terms of the average number of passes obtained [6].

In this research, we implement a new approach to improve the performance on the number of passes required for scheduling a given permutation. For some permutation if GA leads to a better result than SAA, and if we only use SAA to schedule the messages, we are going to miss the best solution. Instead it will be better if we start running both the algorithms in parallel using multiple threads. Using the Bayesian inference criterion [13], after a specific point during the algorithms execution we decide on which algorithm should be continued to run for scheduling the messages, and the other algorithm will be terminated. Based on Bayesian inference, at that instant of time if SAA is continued to run till the end we save time on the scheduling and still get a better result on the number of passes required, or otherwise we proceed with running GA by terminating SA. With this approach we can always guarantee the best solution with a fairly acceptable run time. We developed a combinational method which comprises the use of Bayesian inference method [13] on the AI algorithms (GA & SAA) to always guarantee the best solution, instead of only using either GA or SAA. Simulations are performed to evaluate the performance of the new method, and the results are compared to those obtained from GA and SAA.

2. Optical Multistage Interconnection Network

The new method is implemented on an Omega network, as this was the network model used by other researchers. A typical illustration of the Omega network with 8 inputs, 8 outputs, 3 stages and 4 switching elements per stage is shown in Figure 1 [12, 14].

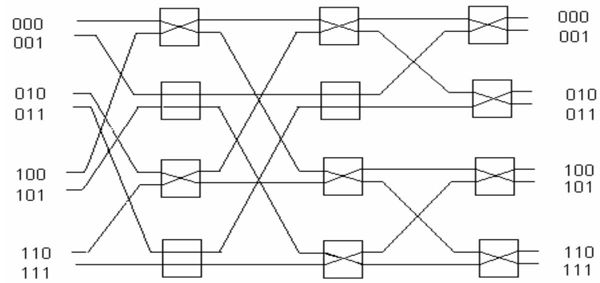

Figure 1. An 8 × 8 Omega network

2.1 Crosstalk in OMIN

Crosstalk [11] occurs when two signal channels interact with each other. When crosstalk occurs, 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 [12]. The channels carrying the signals could cross each other in order to embed a particular topology. Each switching element can be in one of the two connecting schemes as shown in Figure 2 [11].

Straight connection **Cross connection**

Figure 2. Two types of switching connections

Since the two ways shown in Figure 2 will cause crosstalk, what we need to do is to avoid these situations to happen in all the switching elements.

2.2 Approaches to avoid Crosstalk

To reduce the negative effect of crosstalk, we can use a $2N \times 2N$ regular MIN to provide the $N \times N$ connection [10, 12]. But half the inputs and outputs are wasted in this approach. Another more efficient solution is to route the traffic through an $N \times N$ optical network to avoid coupling two signals within each switching element. This can be implemented using the TDM approach [12]. Figure 3 shows the different combinations a switch can allow data to avoid crosstalk.

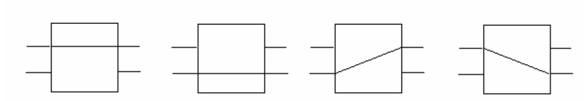

Figure 3. Different ways of passing data through a Switch to avoid crosstalk

To avoid crosstalk, we use the TDM approach [11], which is to partition the set of messages into several groups [6] such that the messages in each group can be sent simultaneously through the network without any crosstalk. That is, to route all the inputs [4] in several groups (passes), such that no crosstalk will be caused in each pass. Since we can only pass a message in a switching element in one of the four ways shown in Figure 3, 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 [10]. So, we need to use at least two or more passes to realize a permutation [5].

For this research the permutations are generated randomly, which have random source addresses and random destination addresses. The goal of this research is to minimize the number of passes required to route all the inputs to the outputs without crosstalk by using the Bayesian inference method on the GA & SAA.

3. Existing Routing Algorithms

The purpose of the routing algorithms is to schedule the messages in different passes in order to avoid the switch conflicts in the network [5, 6]. The more efficient the algorithm is, the less passes it will generate. The order of the messages to be picked for scheduling is an essential cause for generating the different results [6].

The heuristic routing algorithm SIA selects a message sequentially in increasing order of the message source address. The heuristic routing algorithm SDA selects a message sequentially in decreasing order of the message source address. The heuristic routing algorithm DIA selects a message based on the order of the increasing degree of conflicts with other messages. The heuristic routing algorithm DDA selects a message based on the order of the decreasing degree of conflicts with other messages [6]. The DDA performs better out of these four heuristic algorithms.

The AI algorithms GA and SAA outperform the heuristic algorithms. Out of the AI algorithms the GA performs slightly better than the SAA in terms of the number of passes, where as the SAA performance in terms of the time is much better than the GA. Hence previously researchers have concluded keeping time in mind; SAA would be a better option for scheduling the messages for the routing problem [6].

4. Bayesian Inference method

Bayesian inference is statistical inference in which evidence or observations are used to update or to newly infer the probability that a hypothesis may be true [1]. Bayesian statisticians believe that Bayesian inference uses aspects of the scientific method, which involves collecting evidence that is meant to be consistent or inconsistent with a given hypothesis. They say that as evidence accumulates, the degree of belief in a hypothesis changes. With enough evidence, it will often become very high or very low. Bayesian statisticians also believe that Bayesian inference is a suitable logical basis to discriminate between conflicting hypotheses. Hypotheses with a very high degree of belief should be accepted as true; those with a very low degree of belief should be rejected as false [13].

An example of Bayesian inference is: For billions of years, the sun has risen after it has set. The sun has set tonight. With very high probability (or I strongly believe that or it is true that) the sun will rise tomorrow. With very low probability (or I do not at all believe that or it is false that) the sun will not rise tomorrow [13].

Bayesian inference usually relies on degrees of belief, or subjective probabilities, in the induction process and does not necessarily claim to provide an objective method of induction [13]. Bayes' theorem adjusts probabilities given new evidence in the following way:

$$P(H_0 \mid E) = \frac{P(E \mid H_0)P(H_0)}{P(E)}$$

where H_0 represents a hypothesis, called a null hypothesis, which was inferred before new evidence, E, became available.

$P(H_0)$ is called the *prior probability* of H_0.

$P(E \mid H_0)$ is called the *conditional probability* of seeing the evidence E given that the hypothesis H_0 is true. It is also called the *likelihood function* when it is expressed as a function of H_0 given E.

$P(E)$ is called the *marginal probability* of E: the probability of witnessing the new evidence E under all mutually exclusive hypotheses. It can be calculated as the sum of the product of all probabilities of mutually exclusive hypotheses and corresponding conditional probabilities [$\sum P(E \mid H_i)P(H_i)$]. $P(H_0 \mid E)$ is called the *posterior probability* of H_0 given E. The factor $P(E \mid H_0)$ / $P(E)$ represents the impact that the evidence has on the belief in the hypothesis. If it is likely that the evidence will be observed when the hypothesis under consideration is true, then this factor will be large. Multiplying the prior probability of the hypothesis by this factor would result in a large posterior probability of the hypothesis given the evidence. Under Bayesian inference, Bayes' theorem therefore measures how much new evidence should alter a belief in a hypothesis. Multiplying the prior probability $P(H_0)$ by the factor $P(E \mid H_0) / P(E)$ will never yield a probability that is greater than 1. Since $P(E)$ is at least as great as $P(E \cap H_0)$, which equals $P(E \mid H_0).P(H_0)$, replacing $P(E)$ with $P(E \cap H_0)$ in the factor $P(E \mid H_0) / P(E)$ will yield a posterior probability of 1. Therefore, the posterior probability could yield a probability greater than 1 only if $P(E)$ were less than $P(E \cap H_0)$ which is never true [13].

The marginal probability, $P(E)$, can also be represented as the sum of the product of all probabilities of mutually exclusive hypotheses and corresponding conditional probabilities: $P(E \mid H_0)P(H_0) + P(E \mid \text{not } H_0)P(\text{not } H_0)$. As a result, we can rewrite Bayes' theorem as:

$$P(H_0 \mid E) = \frac{P(E \mid H_0)P(H_0)}{P(E \mid H_0)P(H_0) + P(E \mid notH_0)P(notH_0)}$$

With two independent pieces of evidence E_1 and E_2, Bayesian inference can be applied iteratively. We could use the first piece of evidence to calculate an initial posterior probability, and then use that posterior probability as a new prior probability to calculate a second posterior probability given the second piece of evidence. Bayes' theorem applied iteratively implies:

$$P(H_0 \mid E_1, E_2) = \frac{P(E_1 \mid H_0) \times P(E_2 \mid H_0)P(H_0)}{P(E_1) \times P(E_2)}$$

This iteration of Bayesian inference could be extended with more independent pieces of evidence. Bayesian inference is used to calculate probabilities for decision making under uncertainty [13].

4.1. Simple example of Bayesian Inference - From which bowl is the cookie?

To illustrate, suppose there are two bowls full of cookies. Bowl #1 has 10 chocolate chip and 30 plain cookies, while bowl #2 has 20 of each. Our friend Fred picks a bowl at random, and then picks a cookie at random. We may assume there is no reason to believe Fred treats one bowl differently from another, likewise for the cookies. The cookie turns out to be a plain one. How probable is it that Fred picked it out of bowl #1? [13]

Intuitively, it seems clear that the answer should be more than a half, since there are more plain cookies in bowl #1. The precise answer is given by Bayes' theorem. Let H_1 correspond to bowl #1, and H_2 to bowl #2. It is given that the bowls are identical from Fred's point of view, thus $P(H_1) = P(H_2)$, and the two must add up to 1, so both are equal to 0.5. The datum D is the observation of a plain cookie. From the contents of the bowls, we know that $P(D \mid H_1) = 30/40 = 0.75$ and $P(D \mid H_2) = 20/40 = 0.5$. Bayes' formula then yields:

$$P(H_1 \mid D) = \frac{P(H_1).P(D \mid H_1)}{P(H_1).P(D \mid H_1) + P(H_2).P(D \mid H_2)}$$
$$= \frac{0.5 \times 0.75}{0.5 \times 0.75 + 0.5 \times 0.5} = 0.6$$

Before observing the cookie, the probability that Fred chose bowl #1 is the prior probability, $P(H_1)$, which is 0.5. After observing the cookie, we revise the probability to $P(H_1|D)$, which is 0.6.

It's worth noting that our belief that observing the plain cookie should somehow affect the prior probability $P(H_1)$ has formed the posterior probability $P(H_1 \mid D)$, increased from 0.5 to 0.6. This reflects our intuition that the cookie is more likely from the bowl 1, since it has a higher ratio of plain to chocolate cookies than the other [13]. The decision is given as a probability, which is different from classical statistics.

5. Appling Bayesian Inference on GA and SAA

We first start by running GA and SAA in parallel using multiple threads on a given permutation. The optimum parameters for crossover, mutations etc. for GA and starting temperature, final temperature etc. for SAA are chosen from the works performed by other researchers [6]. After a specific point during the algorithms execution we decide on which algorithm should be continued to run for scheduling the messages, and which one should be terminated by applying the rules of Bayesian inference as specified in section 4. The GA works on the basis of generating new populations (for this problem it will be the ordering of messages) by applying crossover and mutation operations [3]. After a new population is generated we evaluate its fitness (number of passes required for routing) and based on the fitness we accept or reject the population. After 50 generations we stop the GA and then calculate the final fitness. For every new permutation we keep repeating this process. In case of SAA we generate a solution and evaluate its fitness at a given temperature. According to the annealing schedule the temperature is decreased for the next iteration and the fitness is evaluated [3, 7]. This process is repeated until we reach the final temperature, at which point the final fitness is calculated. Normally we chose the parameters for SA such that we are going to have 1000 different temperature values between the starting and final temperatures. These are the optimum values for the algorithms based on existing research works [6]. After 5 generations for the GA and 100 temperature values for SA we apply the Bayesian inference rules on the so far obtained results from GA and SAA, and then decide which algorithm is going to run till the end. The value of 5 generations for GA and 100 temperature iterations for SAA is based on the fact that GA runs 20 times slower than SA. This conclusion was obtained on the existing results by previous works [6]. This method of applying Bayesian inference on GA and SAA will be illustrated with an example.

The values for the example are random values, and Table 1 shows the results of the simulations performed. The approach we followed is similar to the approach used in section 4.1. The probability of generating a better solution in terms of GA is little better than that of SAA, but the time taken by GA to find the solution was very high compared to SAA according to the results published [6]. Hence researchers concluded that SAA is better than GA taking all the factors into consideration. In this approach instead of blindly accepting SAA as the best solution we use the Bayesian inference method where by assigning probabilities to the GA and SAA methods. We assign a high probability to SAA compared to that of GA as SAA was very close to GA in terms of the number of passes generated, but with time in play was the best [6].

We consider *P(H₁)* = *P(GA)* = 0.495 and *P(H₂)* = *P(SAA)* = 0.505, where *P(GA)* and *P(SA)* are the probabilities of accepting the GA or SAA method for the Bayesian inference. From the simulations performed, for a particular run after 5 generations of GA, GA produced an average number of 4.236 passes and SA produced an average number of 4.241 passes after 100 temperature iterations for a network size of 32 (32×32). We can now use the following Bayesian inference equation on these values.

$$P(H_1 \mid D) = \frac{P(H_1).P(D \mid H_1)}{P(H_1).P(D \mid H_1) + P(H_2).P(D \mid H_2)}$$

$P(D|H_1)$ is the probability of GA compared to SAA, and $P(D|H_2)$ is the probability of GA compared to SAA, and can be calculated by using the average number of passes obtained.

$$P(D \mid H_1) = \frac{100 \times 4.241}{4.241 + 4.236} = 50.02949 \text{ and}$$
$$P(D \mid H_2) = \frac{100 \times 4.236}{4.241 + 4.236} = 49.97051$$

$P(H_1|D)$ is the probability of deciding whether GA or SAA should be run until the end, as both the algorithm are running in parallel on different threads. If the calculated probability value is greater than 0.5 SAA will be terminated, and GA will produce the final order for scheduling the messages, otherwise GA will be terminated. To find the value of $P(H_1|D)$ we substitute these values into the above equation:

$$P(H_1 \mid D) = \frac{P(H_1).P(D \mid H_1)}{P(H_1).P(D \mid H_1) + P(H_2).P(D \mid H_2)}$$
$$P(H_1 \mid D) = \frac{0.495 \times 50.02949}{0.495 \times 50.02949 + 0.505 \times 49.97051} = 0.4953$$

In this case the probability value is less than 0.5, so GA will be terminated and SAA will produce the final order for scheduling the messages. In this case GA produced an average number of 4.236 passes, where as SA produced 4.21. We can see that GA is better than SAA in terms of the average number pf passes, but according to the Bayesian inference principle we conclude that SAA will be better in this case to produce the final order for scheduling the messages.

Let's consider one more example. In this case $P(H_1)$ = $P(GA)$ = 0.495 and $P(H_2)$ = $P(SAA)$ = 0.505. After 5 generations of GA, GA produced an average number of 4.162 passes and SA produced an average number of 4.288 passes after 100 temperature iterations for a network size of 32 (32×32).

$$P(D \mid H_1) = \frac{100 \times 4.288}{4.288 + 4.162} = 50.74556 \text{ and}$$
$$P(D \mid H_2) = \frac{100 \times 4.162}{4.288 + 4.162} = 49.25444$$

$$P(H_1 \mid D) = \frac{P(H_1).P(D \mid H_1)}{P(H_1).P(D \mid H_1) + P(H_2).P(D \mid H_2)}$$
$$P(H_1 \mid D) = \frac{0.495 \times 50.74556}{0.495 \times 50.74556 + 0.505 \times 49.25444} = 0.50245$$

In this case the probability value is greater than 0.5, so SAA will be terminated and GA will produce the final order for scheduling the messages. In this example the average number of passes produced by GA was lower than SAA, and also applying the Bayesian inference principle also resulted in GA to be continued to produce the final order for scheduling the messages.

6. Simulation Results

The simulations were performed using code written in JAVA programming language on a 3.4 GHz Intel Pentium

IV PC with hyper-threading running Microsoft Windows XP with 1GB of RAM and a 250GB hard disk. We used the multi-threading approach for running the algorithms in parallel. For this comparison purpose we used single point crossover, with the number of rounds set at 1000, the number of generations set at 50, the population size set at 2, and the mutation probability set at 0.1 for GA. For SAA the number of rounds was set at 1000, starting temperature at 1000, final temperature at 0.325, cooling rate at 0.992 and the number of iterations per temperature at 20. These values were considered to be the optimum values for these algorithms [6]. Since our problem is to apply the Bayesian inference method to decide on the particular algorithm to be used for scheduling we simply considered these values instead of again performing simulations for these values. Table 1 shows the simulation results on different size networks.

From Table 1, we see the decision produced by the Bayesian inference method. Except for network sizes of 32 and 128 Bayesian inference method produced a decision of terminating the GA. It is clearly evident from the tabular results that GA takes a long time to schedule the messages compared to SAA. Also from the tabular results we see that the difference in the number of passes produced by SAA compared to that of GA is minimal where the Bayesian inference method decided to terminate the GA and continued to run SAA for scheduling the messages.

7. Conclusion

In this research we used a combinational method which comprises the use of Bayesian inference method on the AI algorithms (GA & SAA) to always guarantee the best solution for scheduling the messages in an OMIN, instead of only using either GA or SAA. Initially GA and SAA algorithms are executed in parallel using multiple threads. In the middle of the execution, Bayesian inference method decided on which algorithm should be terminated. The simulations results proved that the Bayesian Inference method improved the performance of scheduling the messages instead of only using either GA or SAA. The big advantage of using this approach is that it's does not comprise the advantages provided by either GA or SAA, as we get a better scheduling using GA, otherwise we save time and still get a good scheduling order by using SAA.

Table 1. Results of applying the Bayesian Inference on GA and SAA

Network Size	P(GA)	P(SA)	GA PASSES	SAA PASSES	Final Probability	**Decision**	Time (Sec)
4	0.495	0.505	2	2	0.495	SAA	0.018
8	0.495	0.505	2.538	2.536	0.494802937	SAA	0.022
16	0.495	0.505	3.296	3.311	0.496135072	SAA	0.027
32	0.495	0.505	4.118	4.229	0.501649315	GA	4.591
64	0.495	0.505	5.012	5.109	0.499791997	SAA	0.641
128	0.495	0.505	5.487	5.632	0.501520569	GA	68.846
256	0.495	0.505	6.352	6.418	0.497584057	SAA	29.542

References

[1] G.Anandalingam and L. Chen, "Linear combination of forecasts: a general Bayesian model," *Journal of Forecasting*, vol. 8, pp. 199–214, 1989.

[2] V. Cerny, (1985). Thermodynamically Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm. *Journal of Optimization Theory and Applications*, 45:41-51.

[3] L. Davis, editor (1987). *Genetic Algorithms and Simulated Annealing*. Pitman, London.

[4] L. Giles, Mark W. Goudreau, Routing in Optical Multistage Interconnection Networks: a Neural Network Solution. *Journal of Light wave Technology*, Vol. 13, No.6, 9.1111, 1995.

[5] Q. -P. Gu and S. Peng, "Wavelengths requirement for permutation routing in all-optical multistage interconnection networks," *Proceedings of the 2000 International Parallel and Distributed Processing Symposium*, May 1-5, 2000, Cancun, Mexico, pp.761-768.

[6] A. K. Katangur, Yi Pan, and Martin D. Fraser, "Simulated Annealing Routing and Wavelength Lower Bound Estimation on WDM Optical Multistage Networks," *Special section on Advances in Optical Components and Subsystems for WDM Communications of Journal of Optical Engineering* , Vol. 43, No. 5, pp. 1080-1091, May 2004.

[7] C. Koulamas, S. R. Antony, and R. Jaen, A survey of simulated annealing applications to operations research problems. *Omega International Journal of Management Science*, 22(1): 41-56, 1994.

[8] T. W. Manikas, James T.Cain, *Genetic Algorithms vs. Simulated Annealing: A Comparison of Approaches for*

Solving the Circuit Partioning Problem, May 1996.

[9] K. Padmanabhan and A.N. Netravali, “Dilated networks for photonic switching,” *IEEE Transactions on Communications*, vol. 35, no. 12, pp. 1357-1365, Dec. 1987.

[10] Y. Pan, C. Qiao, and Y. Yang, “Optical Multistage Interconnection Networks: New Challenges and Approaches,” *IEEE Communications Magazine, Feature Topic on Optical Networks, Communication Systems and Devices*, Vol. 37, No. 2, Feb. 1999, pp. 50-56.

[11] C. Qiao, R. Melhem, D. Chiarulli and S. Levitan, “A time domain approach for avoiding crosstalk in optical blocking multistage interconnection networks,” *Journal of Lightwave Technology*, vol. 12. no. 10, pp.1854-1862, Oct. 1994.

[12] X. Shen, F. Yang, and Y. Pan, “Equivalent Permutation Capabilities between Time Division Optical Omega Networks and Non-optical Extra-Stage Omega Network,” *Proceedings of 1999 IEEE International Performance, Computing, and Communications Conference*, Phoenix/Scottsdale, Arizona, USA, pp. 356-361, February, 1999.

[13] J. Stutz, and P. Cheeseman, “A Short Exposition on Bayesian Inference and Probability.” June 1994. *National Aeronautic and Space Administration Ames Research Centre*: Computational Sciences Division, Data Learning Group.

[14] C. Tocci and H.J. Caulfield, *Optical Interconnection – Foundations and Applications*, Artech House Publishers, 1994.

[15] A. Varma and C.S. Raghavendra, Interconnection networks for Multiprocessors and Multicomputers: Theory and Practice, IEEE Computer Society Press, 1994.