Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 59 (2015) 106 – 114

Procedia Computer Science

International Conference on Computer Science and Computational Intelligence (ICCSCI 2015)

GreedyZero Algorithms for Conflict-Free Scheduling in Low Stage Interconnection Network

Mehrnaz Moudi[a,*], Mohamed Othman[a]

[a]*Department of Communication Technology and Network, Universiti Putra Malaysia, 43400 UPM, Serdang, Selangor D.E., Malaysia*

Abstract

Low Stage Interconnection Networks are a class of Interconnection Networks. They have been generated from Multistage Interconnection Networks (MINs). Although the conflict in the optical switches, there is the considerable interest to use the optical technology in interconnection networks implementation. To avoid this problem, GreedyZero algorithms has been assigned to the Low Stage Interconnection Networks for improving the network performance by reducing the number of passes. The results marked nearly 50% reduction in the number of passes and proved improvement of scheduling in the Low Stage Interconnection Networks by GreedyZero algorithms.

Peer-review under responsibility of organizing committee of the International Conference on Computer Science and Computational Intelligence (ICCSCI 2015)

Keywords: Low Stage Interconnection Network; GreedyZero Algorithm; Conflict.

1. Introduction

The advances in electro-optic technologies have made significant improvement in the optical technology. The idea of optical implementation of Multistage Interconnection Networks (MINs) meet the increasing requirements of high performance computing communication applications for low communication latency, high channel bandwidth and parallel processing[1,2,3,4]. Among different types of interconnection networks, Optical Multistage Interconnection Network (OMIN) is generally used in communication. Low Stage Interconnection Networks have been generated from MIN and present conflict that related with optical switches, because of undesired coupling two signals in each switch. Hence, scheduling more than one message without any conflict is not possible in the Low Stage Interconnection Networks at the same time. Avoiding conflict is using the minimum number of passes for a permutation and execution time to schedule the input request to reach output[5,6,7]. This research

* Corresponding author. Tel: +603-8946 1707; Fax: +603-8946 6576 / +603-8948 3745
E-mail address: mehrnazmoudi@gmail.com , mothman@upm.edu.my

Peer-review under responsibility of organizing committee of the International Conference on Computer Science and Computational Intelligence (ICCSCI 2015)
doi:10.1016/j.procs.2015.07.343

applies GreedyZero algorithms to show scheduling improvement for Low Stage Interconnection Networks. These networks were proposed in[8] to reduce execution time. Now the GreedyZero algorithms[9] are also developed to reduce the number of passes considerably by free conflict scheduling all the inputs to outputs. The algorithms effect on the network performance by reducing the number of passes.

The reminder of this paper is organized as follows. The problem is presented in Section 2. Description of the Low Stage Interconnection Networks is provided in Section 3. The use of GreeyZero algorithms in Low Stage Interconnection Networks are described in Section 4. In Section 5, experimental results are discussed. Section 6 remarks the conclusion.

2. Problem Outline

An important problem in OMIN is conflict which is at the result of coupling two signal channels with each other in switches. There are four legal passing connections in every switch[10,11,12]. Fig.1 shows four connections upper straight, lower straight, lower to upper cross, and upper to lower cross as a,b,c, and d respectively.

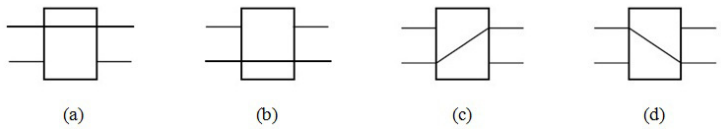


Fig. 1: Legal Passing Connections in a switch at a Time[6]

In our previous work[8], Low Stage Interconnection Networks were proposed as new architecture which were not able to reduce the conflict. Although, the messages were scheduled faster than the OMIN. Under the constraint of avoiding conflict, many researchers have been proposed various scheduling algorithm for avoiding[13,14,15]. In this research, GreedyZero algorithm are used as a scheduling algorithm which had applied with success in OMINs[9] to reduce conflict. We apply this algorithm to separate the messages without conflicts in Low Stage Interconnection Networks to improve networks performance.

3. Low Stage Interconnection Network

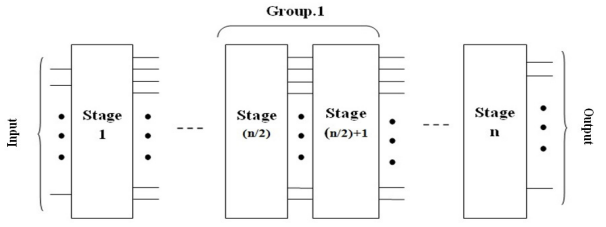
The Low Stage represents two switches in each row of the OMIN as one switch. In these architectures the number of switches is less than the original number which leads to having the number of stages new. Table 1 presents the number of stages of three different architectures in Low Stage Interconnection Network.

Table 1: Number of Stages in Low Stage Interconnection Network.

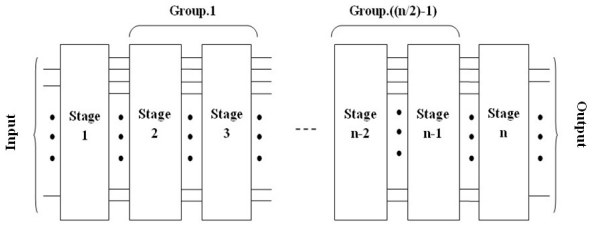
Architectures	Number of Stages
First Low Stage	$(n = log_2N) - 1$
Second Low Stage	$(n = log_2N)/2 + 1$
Third Low Stage	$(n = log_2N) - 1$

The Low Stage strategy is done for three different architectures which are shown in Fig.2. According to the configuration of each architecture, we start to group switches in the network. For the First Low Stage (FLS), the switches of two middle stages in each row are considered as one group. The group is applied as one switch. In the Second Low Stage (SLS), all the switches for each row are grouped except in the first and last stages. The grouping starts from the switch in second stage with the switch

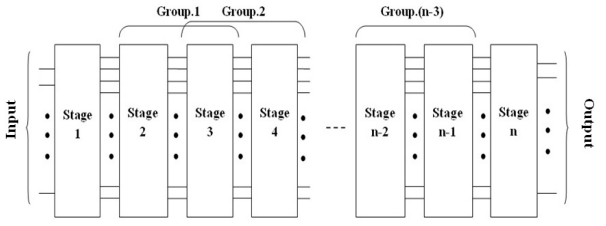
in the third stage; the switch in the fourth stage is grouped with the switch in the fifth stage and it continues for each row except the switches in the last stage. The total number of groups is (n/2) - 1 in each row. In the Third Low Stage (TLS), the switches are grouped together in each row from left to right. The used strategy is grouping the switch in the second stage with switch in the third stage, switch in the third stage with switch in the fourth stage and it continues until using the switches in each row except the switches in the first and last stage. It provides (n - 3) groups of switches. This architecture is efficient for big size networks where $N \geq 64$.



(a) First Low Stage



(b) Second Low Stage



(c) Third Low Stage

Fig. 2: Three Architectures of Low Stage Interconnection Network

The detailed description for each of these architectures have been clarified in[8]. To show the performance improvement in Low Stage Interconnection Networks, messages with conflicting path between them are listed for scheduling. The description of scheduling algorithm is presented in the following section.

4. Scheduling Algorithm

To avoid conflict, the first step of the scheduling is generating random permutations with the network size $N \times N$ consisting of N source and N destination which are addresses to build the combination matrix. Then, by using a pattern checking method, Bitwise Window Method (BWM)[16], conflict matrix is set. Conflict matrix is a $N \times N$ matrix with assigning value 1 for having a conflict and value 0 if

there is not conflict. Finally, the GreedyZero algorithm is applied on conflict matrix as scheduling to find number of passes.

Algorithm 1 GreedyZero Algorithm

```
1: procedure BITWISE WINDOW METHOD
2:
3:     n: the new number of stages;
4:     CM: the N×n Combination Matrix;
5:     Conflict Matrix: N×N Matrix;
6:     i = 0 , j = 0 , k = j + 1 ;
7:
8:     for each integer i to n − 1 do
9:         for each integer j to N − 1 do
10:            for each integer k to N − 1 do
11:                if CM(j)(i) = CM(k)(i) then
12:                    Conflict Matrix [j][k] = 0 ;
13:                else
14:                    Conflict Matrix [j][k] = 1 ;
15:                end if
16:            end for
17:        end for
18:    end for
19:
20:    Return Conflict Matrix;
21:
22: end procedure
23:
24: procedure SCHEDULING
25:    i = 1;
26:    loop:
27:    ArrayM ⟵ Sum of each columns/rows from Conflict Matrix in descending order;
28:    Group G_i ⟵ Biggest entry in ArrayM;
29:
30:    for each node do
31:        if intersection of each node with all the entries in group G_i = 0 then
32:            Add node to the group G_i;
33:        end if
34:    end for
35:    Initialise all columns/rows corresponding to all entries in group G_i to 0;
36:    i = i + 1;
37:    goto loop and repeat until the alternation of Conflict Matrix becomes zero and all messages
       are clustered in different groups.
38:
39: end procedure
```

Zero algorithm is based on the grouping of messages that have summations equal to zero in the conflict matrix while for the GreedyZero algorithm, the grouping of messages is based on the idea of the greedy graph colouring[17]. In this method, the messages are sorted according having bigger summations in the conflict matrix. Then, they are scheduled in such a way that are efficient in terms of both execution time and number of passes[18,19]. In Algorithm 1, the pseudo code of the GreedyZero algo-

rithm is recommended. GreedyZero algorithm consists of three subalgorithms, namely, GreedyZeroX, GreedyZeroY and GreedyZeroXY.

4.1. GreedyZeroX Algorithm

The GreedyZeroX algorithm use the greedy graph colouring method in the ZeroX algorithm. The algorithm procedure starts by checking the BWM for all possible path conflicts between messages and return conflict matrix. For scheduling, the sum of each column of conflict matrix is stored in an array which is initialized as follows:

ArrayM ← sum of each columns from Conflict Matrix in descending order;

Then, the highest value element from ArrayM is selected and the related message scheduled in the first group of messages as first pass. For the rest of the messages, the process is repeated until all the messages have been scheduled according algorithm without conflict. The analysis of the GreedyZeroX algorithm shows its advantage over the ZeroX algorithm in the way of message grouping without conflict.

4.2. GreedyZeroY Algorithm

The GreedyZeroY algorithm use the greedy graph colouring method in the ZeroY algorithm. Similar to GreedyZeroX algorithm, after checking the conflicts between messages by BWM, the messages are scheduled without conflict in different groups. The difference of between GreedyZeroX and GreedyZeroY algorithms is in the initializing ArrayM. The initializing of this array is shown in the following statement:

ArrayM ← sum of each rows from Conflict Matrix in descending order;

The benefit of the GreedyZeroY algorithm over the ZeroY algorithm concerns the way of message grouping, which makes this algorithm simpler.

4.3. GreedyZeroXY Algorithm

A combination of the GreedyZeroX and GreedyZeroY algorithms is the basis for the GreedyZeroXY algorithm. For this algorithm, GreedyZeroX algorithm is implemented and then GreedyZeroY algorithm is implemented. Finally, the obtained result for the GreedyZeroXY algorithm is the minimum result between GreedyZeroX algorithm and GreedyZeroY algorithm. The primary purpose of the GreedyZeroXY algorithm is to reduce the average number of passes compared to the individual GreedyZeroX and GreedyZeroY algorithms. Overall, the GreedyZero algorithms advantage compared to the Zero algorithms is the way of message grouping which is based on greedy graph colouring method and makes it simpler.

5. Experimental Results

In this research, the scheduling algorithms have been applied in three different architectures of Low Stage Interconnection Networks for scheduling request and reducing conflict. The used scheduling algorithm in previous work[8] was the Sequential Decreasing (SeqDec) algorithm. As the purpose of the scheduling is arranging and selecting the messages to avoid the path conflicts, SeqDec algorithm had been chosen to schedule the messages in decreasing order. We evaluate the performance of Low Stage Interconnection Networks by using GreedyZero algorithms and SeqDec algorithm. The algorithms are simulated in different network sizes covering from the smallest size, 4 to the largest size, 256.

A network size $N \times N$ means that this network has N source nodes and N destination nodes. Two parameters, number of passes and execution time have been evaluated to show the development. They are compared between four algorithms when is applied for Low Stage Interconnection Networks. Furthermore, there are trade-offs in the performance between the execution time and the number of passes[20]. The followings display the use of GreedyZero algorithms in Low Stage Interconnection Networks and discuss the obtained results from these algorithms compared with SeqDec algorithm in terms of performance metrics.

5.1. Number of Passes

In Fig.3, All algorithms are applied in First Low Stage Interconnection Network. The similarity among the number of passes in three GreedyZero algorithms can be observed clearly.

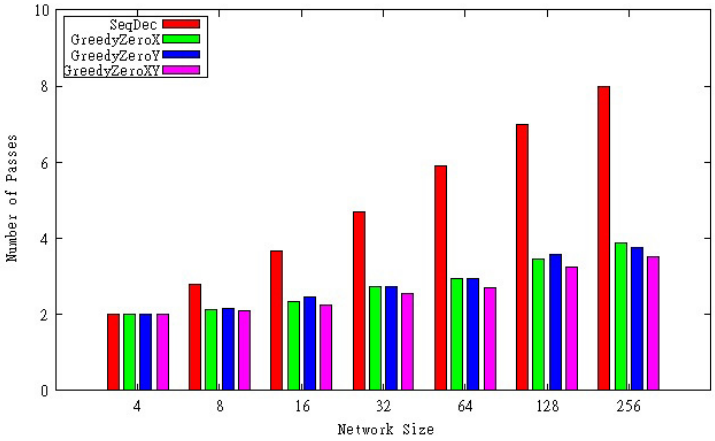


Fig. 3: Average Number of Passes in First Low Stage Interconnection Network

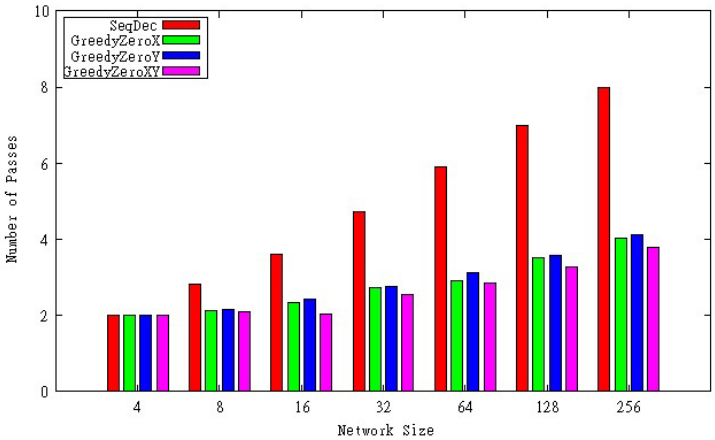


Fig. 4: Average Number of Passes in Second Low Stage Interconnection Network

The results demonstrate that there is a considerable difference in the number of passes between GreedyZero algorithms and SeqDec algorithm. In Fig.4, the average number of passes from SeqDec algorithm is considerably worse compared to the GreedyZero algorithms in Second Low Stage Interconnection Network. The number of passes in Third Low Stage Interconnection Network is reduced

with GreedyZero algorithms in compare with SeqDec algorithm which is shown in Fig.5. It is obvious from these figures, the less number of passes between scheduling algorithms in Low stage Interconnection Network is for GreedyZero algorithms. These algorithms group the messages in less number of groups and reduce conflict because of their method to nd the possible passes between messages. The least number of passes is for GreedyZeroXY algorithm, as this algorithm finds the minimum number of passes between GreedyZeroX algorithm and GreedyZeroY algorithm. From the algorithm grouping results in three architectures, we found the number of passes is not improved for different architectures of Low Stage Interconnection Networks.

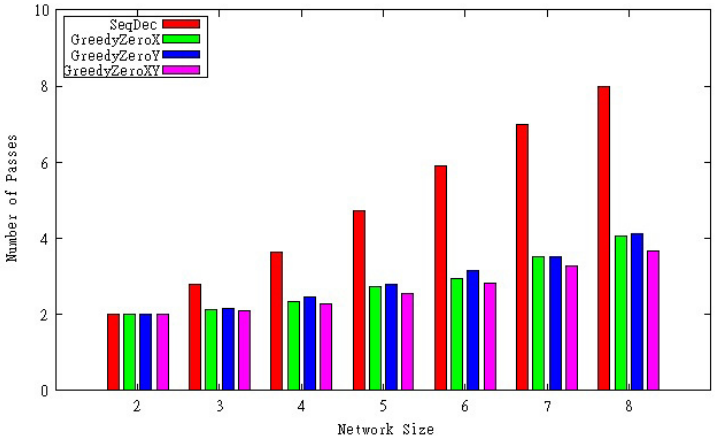


Fig. 5: Average Number of Passes in Third Low Stage Interconnection Network

5.2. Execution Time

The comparison of execution time between the GreedyZero Algorithms and SeqDec algorithm proves a similar performance except in bigger network size.

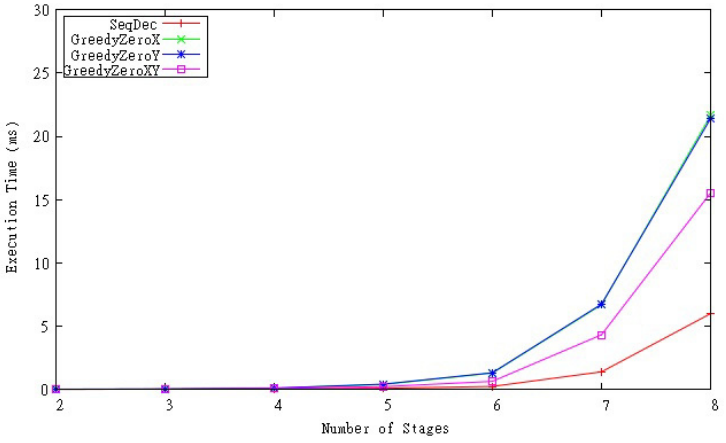


Fig. 6: Average Execution Time in First Low Stage Interconnection Network

In Fig.6, the execution time of GreedyZero algorithms and SeqDec algorithm in the First Low Stage Interconnection Network is compared in different number of stages ($n = log_2 N$, N is Network Size). SeqDec algorithm is faster than GreedyZero algorithms for big network size. Fig.7 and Fig.8 present

the comparison of GreedyZero algorithms and SeqDec algorithm in the Second and Third Low Stage Interconnection Networks according the execution time for different number of stages, n. The findings indicate that the execution time is similar for GreedyZeroX and GreedyZeroY algorithms in all three architectures while there is a definite decrease for GreedyZeroXY algorithm. The execution time for the GreedyZeroXY is less than GreedyZeroX and GreedyZeroY algorithms, because it nds the minimum execution time between GreedyZeroX and GreedyZeroY algorithm. The results of the execution time comparison between different scheduling algorithms are approximately the same. Although, the SeqDec algorithm is faster than the other algorithms. As SeqDec algorithm is simple, this algorithm take less execution time than the GreedyZero algorithms and is not time-consuming for big network size.

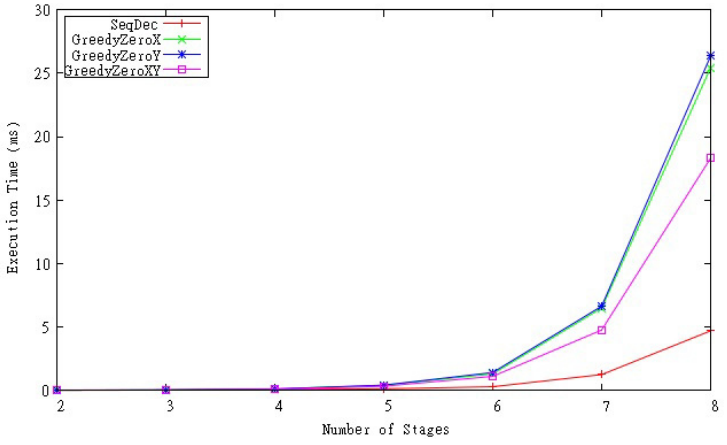


Fig. 7: Average Execution Time in Second Low Stage Interconnection Network

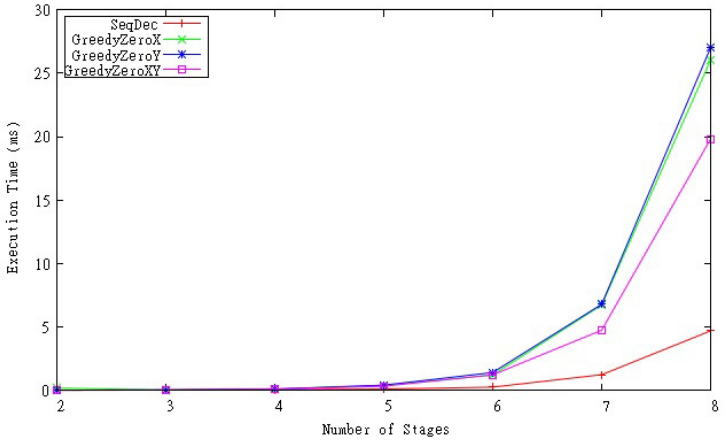


Fig. 8: Average Execution Time in Third Low Stage Interconnection Network

6. Conclusion

In this research, the GreedyZero algorithms have been presented in the Low Stage Interconnection Networks. The algorithms have been developed to achieve better performance for Low Stage Interconnection Networks in terms of 50% reduction in the number of passes. Three architectures yielded

the better result in the number of passes by applying GreedyZero algorithms compared with SeqDec algorithm. Although, for GreedyZero algorithms, the execution time will have a definite increase in big network size in the Low Stage Interconnection Networks.

Acknowledgement

This research was supported by the Fundamental Research Grant Scheme (FRGS), Malaysian Ministry of Education, (Ref: FRGS/1/2014/ICT03/UPM/01/1).

References

1. Lu E, Zheng S.Q. Parallel Routing and Wavelength Assignment for Optical Multistage Interconnection Networks. In Proc. of the International Conference on Parallel Processing (ICPP04), 2004.
2. Tian H, Katangur A.K, Zhong J, Pan Y. A Novel Multistage Network Architecture with Multicast and Broadcast Capability. Journal of Supercomputing, 2006, 35(3): 277-300.
3. Garofalakis J, Stergiou E. Analytical model for performance evaluation of Multilayer Multistage Interconnection Networks servicing unicast and multicast traffic by partial multicast operation. Perform Eval, 2010, 67: 959-976.
4. Pan Y, Qiao C, Yang Y. Optical Multistage Interconnection Networks: New Challenges and Approaches. IEEE Communications Magazine, Feature Topic on Optical Networks, Communication Systems and Devices, 1999, 37(2): 50-56.
5. Othman M, Shahida T.D. The development of crosstalk-free scheduling algorithms for routing in Optical Multistage Interconnection Networks. In: Christos J Bouras (editor). Trends in Telecommunications Technologies. InTech; 2010. ISBN: 978-953-307-072-8.
6. Aggarwal R, Kaur L, Aggarwal H. Multistage interconnection networks: a transition from electronic to optical. Journal of Emerging Technologies in Web Intelligence, 2010, 2(2): 142-147.
7. Bashirov R, Karanller T. On path dependent loss and switch crosstalk reduction in optical networks, Information Sciences, 2010, 180(6): 1040-1050.
8. Moudi M, Othman M. A Cost-Effective Architecture For Optical Multistage Interconnection Network. Journal of Networks, 2013, 8(2): 345-350.
9. Moudi M, Othman M. A GreedyZero algorithm to minimise the conflicts in an Optical Multistage Interconnection Network, Journal of Network and Computer Applications, 2014, 41: 312-318.
10. Vince A. A framework for the greedy algorithm. Discret Appl Math, 2002. 121: 247-260.
11. Katangur A.K, Akkaladevi S, Pan Y. Analyzing the performance of optical multistage Interconnection networks with limited crosstalk, Cluster Computing, 2007, 10: 241-250.
12. Almazyad A.S. Optical omega networks with centralized buffering and wavelength conversion. Journal of King Saud University-Computer and Information Science, 2011, 23(1): 15-28.
13. Shanmugam G, Ganesan P, Vanathi P.T. Meta heuristic algorithms for vehicle routing problem with stochastic demands. Journal of Computer Science, 2011, 7: 533-542.
14. Dorigo M, Sttzle T. AntColony Optimization. A Bradford book. Cambridge, Massachusetts, London, England: The MIT Press; 2004.
15. Moudi M, Othman M. A challenge for routing algorithms in Optical Multistage Interconnection Networks. Journal of Computer Science, 2011, 7(1): 1685-1690.
16. Abed F, Othman M. Efficient window method in optical multistage interconnection networks. In Proc. the IEEE International Conference on Telecommunications and Malaysia International Conference on Communications (ICT- MICC), 2007, pp. 181-185.
17. Al-Omari H, Sabri K. New graph colouring algorithms. American Journal of Mathematics and Statistics, 2006, 2(4): 739-741.
18. Borella A, Cancellieri G, Mantini D. Space Division Architectures for Crosstalk Reduction in Optical Interconnection Networks, QoS-IP, LNCS 2601, 2003, pp. 460-470.
19. Tropp J.A. Greed is good: algorithmic results for sparse approximation. IEEE Transactions On Information Theory, 2004, 52(10): 2231-2242.
20. Al-Shabi M, Othman M. A new algorithm for routing and scheduling in optical omega network International Journal of Advanced Computer Science and Applications, 2008, 16(1): 26-31.