DE GRUYTER OPEN

Print ISSN: 1312-2622; Online ISSN: 2367-5357
DOI: 10.1515/itc-2018-0010

Optimal Performance Algorithm for Non-conflict Scheduleof Sub Matrices in a Crossbar Node

K. Kolchakov, V. Monov

Key Words: *Network nodes; node traffic; crossbar switch; conflict elimination; packet messages.*

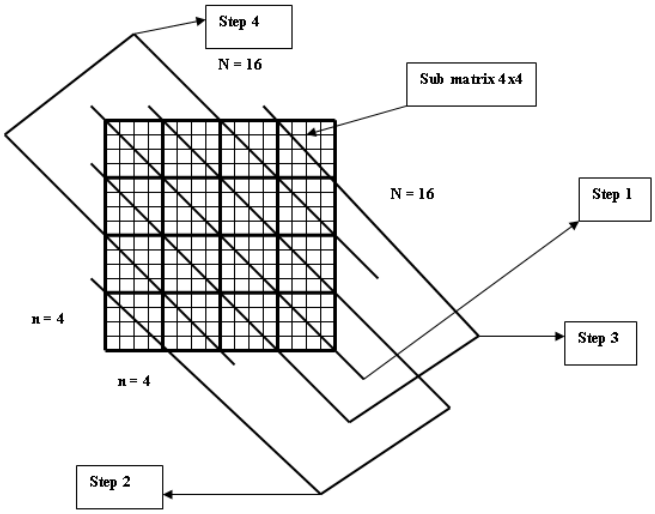
Abstract. *In this paper, we present the synthesis of an optimal conflict-free schedule in terms of performance of the sub matrices in the connection matrix of a scheduling algorithm with diagonal activations of joint sub-switching matrices for a crossbar switch node. The algorithm was recently proved to be optimal with respect performance. Here, we have made a comparison of this algorithm with two other algorithms in terms of needed memory and complex performance.*

Introduction

The traffic via Crossbar switching nodes is casual and depends on the users. The formulation of a conflict issue during operation of the switching nodes is as follows. The dimensions of switches in the switching nodes are N x N, where N sources of packet massages are connected to N receivers via the switches of the switching node. The traffic is random by nature and conflicts are available in the following two cases:

- When one source of message requests communication to two or more message receivers.
- When one message receiver receives communication requests from two or more message sources.

The evasion of conflicts is directly related to the switching node performance. The status of the switch of the switching node is represented with the so called connection matrix. For N x N dimensional switch the dimension of the connection matrix T is N x N also, where every element $T_{ij} = 1$ if the connection request from i- source to j- receiver exists. In the opposite case $T_{ij} = 0$.



A conflict situation arises if any row of the connection matrix has more than a single 1, which corresponds to the case when one source requests a connection with more than one receiver. The presence of more than a single 1 in any column of the matrix T also indicates a conflict situation, it means that two or more sources have requested a connection with the same receiver.[1,7,8,9].

The scheduling algorithm with diagonal activations of joint sub-switching matrices (ADAJS) was examined in [1] and proved to be optimal with respect to the overall performance and necessary memory by means of analyzing and comparing fourteen different algorithms for design of a non-conflict schedule.

In this study, our aim is to synthesize a conflict-free schedule which is optimal with respect to the performance of the sub matrices entering into the connections matrix used in the algorithm ADAJS.

Description of the Algorithm

The connections matrix T with N x N size, where N is an integer being a degree of two, is divided into sub-matrices (S) with dimension n x n, (n also is a degree of two), i.e:

$$T = [S_{ij}], \quad i = 1\ n, j = 1, n$$

The sets of sub matrices located along the main diagonal are processed simultaneously in each of the diagonals. For sub matrices in diagonals parallel to the main one, the principle of reconciliation is used [2].

The idea of synthesis of the algorithm ADAJS is based on the knowledge that the diagonal sub matrices with requests for service in the matrix T are non-conflict in the diagonal where they are located. There are diagonals with sub matrices of requests that are non-conflict to one another. *Figure 1* shows joint couples of non-conflict diagonals with sub matrices of requests for service and the main diagonal of sub matrices that can not be jointed with anyone else [2].The whole process of the implementation of ADAJS algorithm for obtaining a non-conflict schedule is divided into steps. The first step refers to the main diagonal sub matrices processed simultaneously and without conflict. The next steps are related to the reconciliation of the diagonals parallel to the main diagonal by pairs (*figure 1*) [3].

The analytical description of the steps shown in *figure 1* is as follows:

Step1: Activation of sub matrices S_{11}, S_{22}, S_{33}, S_{44}

Step3: Activation of sub matrices S_{21}, S_{32}, S_{43}, S_{14}

Step2: Activation of sub matrices S_{41}, S_{12}, S_{23}, S_{34}

Step4: Activation of sub matrices S_{31}, S_{42}, S_{13}, S_{24}

$T = [S_{ij}]$, $i = 1 \div 4$, $j = 1 \div 4$

The size (n) of the sub matrix determines the number of steps (I) as follows:

(1) $I = N/n$

for **N** = const., I = f (n), where $1<n \leq N / 2$.

Optimal Performance Algorithm for Non-conflict Scheduling in Sub Matrices

The optimal size of the sub matrices for different sizes N is determined by the minimum processing time. A clear minimum of the operating time (**TW**) is seen when n = 4 for N = 128, 256, 512, 1024 and 2048. It can be concluded that for size N up to 2048, the optimal sub matrix size is $n_{opt} = 4$ [4].

Two algorithms are used to obtain conflict-free schedule in the sub matrices in the ADAJS algorithm:

Algorithm with joint diagonals activations (**AJDA**) [5].

Algorithm by diagonal connectivity matrix activation (**ADA**) [6].

Typical for both algorithms is that they admit zero solutions, which reduces their performance.

In the new synthesized algorithm AOP (Algorithm for Optimal Performance) the joint diagonals are used again but with a periodical check is performed for depletion of requests. Furthermore, an initial check is done whether the sub matrix is not zero.

Another characteristic of the new algorithm is that it is checked for the presence of requests for service in the diagonal perpendicular to the main diagonal. Thus the check for requests includes joint diagonals parallel and perpendicular to the main one.

On *figure 2* we have illustrated the main diagonal, the perpendicular to the main diagonal and the jointed pairs of diagonals parallel to them.

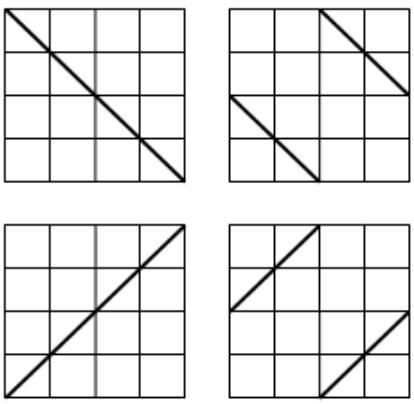


Figure 2

The new algorithm AOP was implemented using the MATLAB software system. The software model SMAOP, describing the AOP algorithm in the MATLAB environment is given on *figure 4*.

Software Models Performance

A software models performance (P) is defined as a ratio of the non- nil resolutions to the total number of the solutions. R(v) is the set of the nil solutions, R(w) is the set of the non-nil solutions, and R is a set of the all solutions[1].

(2) $R=R(v)+R(w);$

(3) $P=(R(w)/R).100[\%].$

From formula 3 it is seen that when the nil solutions R(v) vanish to nil, then the performance P tends to 100% [1].

Figure 3 presents typical input matrices where for completeness we have added the zero and unit matrices.

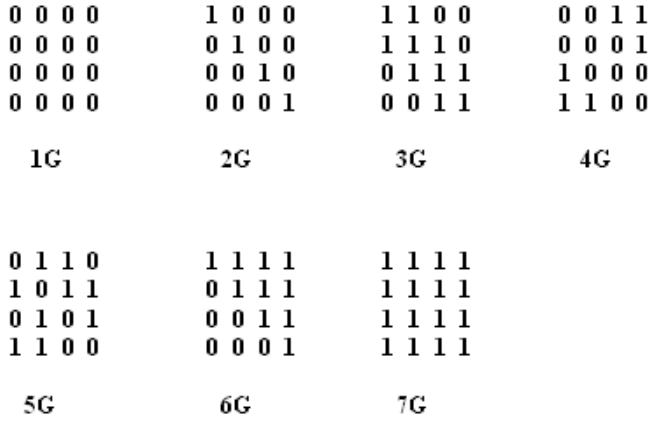


Figure 3

Table 1 presents the results of studying the performance of three algorithms: the algorithm AJDA implemented by its software model SMAJDA, the algorithm ADA implemented by its software model SMADA and the algorithm AOP implemented by its software model SMAOP.

In the software model of *figure 4,* we first check that the input submatrix T is not zero. The main diagonal of queries is selected and the available ones are activated. It is currently checking that the total number of requests is exhausted. If there are unrealized queries, the perpendicular to the main diagonal of queries is selected. After activation, they are checked again if there are unrealized queries. The presence of unrealized queries results in the activation of the query diagonals parallel to the main diagonal and perpendicular diagonal. Verification for unrealized queries is ongoing until the queries are fully exhausted.

From *table 1* it is seen that for zero input matrix (1G)

```
tic;T = randsrc(4,4,[0,1])
G = sparse(T)
%check for zero input sub matrix.
s = 0
for c = T
   s = s + sum(c)
end
if s ~= 0
M = eye(4)
N = sparse(M)
U = G.*N
T1 = T - U
% is there more requests for service.
s1 = 0
for c1 = T1
   s1 = s1 + sum(c1)
end
if s1 ~= 0 % if s1 is nonzero.
Y = [0 0 0 1; 0 0 1 0; 0 1 0 0; 1 0 0 0]
H = sparse(Y)
L = G.*H
T2 = T1 - L
%  is there more requests for service.
s2 = 0
for c2 = T2
   s2 = s2 + sum(c2)
end
if s2 ~= 0  % if s2  is nonzero.
 R1 = [0 1 0 0; 1 0 0 0; 0 0 0 1; 0 0 1 0]
R2 = [0 0 1 0; 0 0 0 1; 1 0 0 0; 0 1 0 0]
J = G.*R1
T3 = T2 - J
%  is there more requests for service.
s3 = 0
for  c3 = T3
   s3 = s3 + sum(c3)
end
if s3 ~= 0 % if s3  is nonzero.
K = G.*R2
end
end
end
end
toc
```

Figure 4. Software model SMAOP

SMAOP has 100% performance and this is because the zero input is not processed. At the same time all the solutions of SMAJDA and SMADA are zero. Also, the results in *table 1* indicate that only SMAOP has 100% performance for all seven input matrices. Thus the algorithm AOP, applied as a processing algorithm of the sub matrices of ADAJS, substantially increases its performance.

Table 1. Performance of algorithms SMAJDA, SMADA and SMAOP

P[%]	1G	2G	3G	4G	5G	6G	7G
SMAJDA	0	6,66	20	80	80	53,3	100
SMADA	0	6,66	20	80	80	100	100
SMAOP	100	100	100	100	100	100	100

A Comparison of SMAOP with SMAJDA and SMADA with Respect to Speed of Execution and Needed Memory

We have established that algorithm SMAOP is optimal with respect to the performance, which is defined as the ratio of the number of nonzero solutions and the number of all solutions (P=(R(w)/R).100[%]). Here, our purpose is to study the behavior of SMAOP with respect to the speed of execution and needed memory as compared to algorithms SMAJDA and SMADA.

It is shown in [3] that the optimal size of sub matrices is n opt. = 4 which implies that the execution time of a program model is less than a second. In order to achieve a higher precision in measuring the speed of execution we applied an approach of multiple cyclic execution of each program model. The obtained results are divided by the number of performed cycles and thus an averaged execution time of the respective program model is obtained.

Table 2 presents the results of the research for 1000 repetitions. The table lists calculated values of the performance in a single execution of the corresponding software model.

It is seen that SMAJDA is the fastest algorithm and SMAOP is the slowest. In addition, SMAOP requires 2.35 times more memory than SMAJDA.

Table 2. Speed and needed memory

Software model	Speed for 1000 time, s	Speed for 1 time[s]	Needed memory, bytes
SMAJDA	20,58	0,02058	768
SMADA	24,68	0,02468	736
SMAOP	135,018	0,13501	1812

Complex Performance

From the study of performance P it is seen that the relationship between the set of non-zero solutions and complete set of solutions is investigated. However, the time factor is not reported in the results, making them incomplete. By introducing the concept of complex performance (CP) the component time is also taken into account as follows:

(4) CP = P.t , for N = const., t = 1/S [1].

The value of CP is a quality indicator in the sense that a smaller value of S (faster algorithm) actually increases the value of CP.

The results for CP from *table 3* show that the software algorithm SMAOP is better than SMAJDA and SMADA only in the cases of input submatrices 1G and 2G. In the cases of all other input matrices, excluding 6G, SMAJDA gives the best results.

Table 3. Complex Performance of algorithms SMAJDA, SMADA and SMAOP

CP	1G	2G	3G	4G	5G	6G	7G
SMAJDA	0	323,61	971,81	3887,26	3887,26	2589,89	4859,08
SMADA	0	269,85	810,37	3241,49	3241,49	4051,86	4051,86
SMAOP	740,68	740,68	740,68	740,68	740,68	740,68	740,68

Conclusion

The main conclusion of this study is that the synthesized algorithm AOP is optimal related to the sub matrices in the connection matrix with respect to the performance. AOP meets the requirement for 100% performance, irrespectively of the type of input sub matrices. With respect to the complex performance CP, AOP shows best results only for input matrices 1G and 2G. With an exception of input matrices 1G, 2G and 6G, SMAJDA is the best algorithm in terms of the complex performance.

References

1. Kolchakov, K., V. Monov. An Approach for Syntesis of Non-conflict Schedule with Optimal Performance of Sub Matrices in a Crossbar Switching Node. Proceedings of the International Conference Automatics and Informatics'2016, Bulgaria, Sofia, 4-5 October 2016, Federation of the Scientific Engineering Unions, John Atanasoff Society of Automatics and Informatics, 2016, ISSN:Proceedings ISSN 1313-1850, CD ISSN 1313-1869, 33-35.

2. Kolchakov, K., V. Monov. Comparative Analysis of a Class of Algorithms for Traffic Management in a Crossbar Commutator with Respect to Complex Performance, Speed and Memory. – *Problems of Engineering Cybernetics and Robotics,* 66, 2015, 53- 62, Sofia, Bulgarian Academy of Sciences, ISSN 0204-9848.

3. Kolchakov, K., V. Monov. A Comparison Study of an Algorithm for Non Conflict Schedule with Diagonal Activation of Joint Sub Matrices in a Large Scale Switching Matrix. – *Information Technologies and Control,* No. 2, 31-35, Printed in Nov. 2015. ISSN1312-2622, On-line ISSN 2367-5357.

4. Kolchakov, K., V. Monov. Hiearchical Two Layers Control Commutator for Implementation of Fully Non-conflict Schedule. – *IJ IMA,* 3, 2014, No. 2, 175-181, ISSN 1314-6416 (Printed), ISSN 1314-6432(On-line).

http://www.foibg.com/ijima/vol03/ijima-fv3-vol03.htm.

http://www.foibg.com/ijima/vol03/ijima03-02-p08.pdf.

5. Kolchakov, K., T. Taschev. An Algorithm of Non-conflict Schedule with Joint Diagonals Activation of Connectivity Matrix. Proceedings of the International Conference on Computer Systems – CompSysTech'12, 22-23 June 2012, Ruse, Bulgaria, ACM PRESS, ICPS, 630, 245-250, ISBN 978-1-4503-1193-9.

6. Kolchakov, K. An Algorithm Synthesis of Non-Conflict Schedule by Diagonal Connectivity Matrix Activation. Proceedings of the International Conference Automatics and Informatics'11, John Atanasoff Society of Automatics and Informatics, Bulgaria, Sofia, 3-7 October 2011, B-247–B251, Proceedings ISSN 1313-1850, CD ISSN 1313-1869.

7. Wanjari, P., A. Chonhari. Implementation of 4x4 Crossbar Switch for Network Processor. – *International Journal of Emerging Technology and Advanced Engineering,* 1, December 2011, No. 2, Website: www.ijetae.com (ISSN 2250–2459).

8. Tashev, T., T. Atanasova. Computer Simulation of MIMA Algorithm for Input Buffered Crossbar Switch. – *International Journal Information Technologies & Knowledge,* 5, 2011, No. 2, 183-189, ITHEA®, Sofia, Bulgaria.

9. Kim, D., K. Lee and H. Yoo. A Reconfigurable Crossbar Switch with Adaptive Bandwidth Control for Networks-on-Chip. IEEE International Symposium on Circuits and Systems, 2005.

Manuscript received on 11.11.2018

Kiril Kolchakov *is Assistant Professor at Modelling and Optimization Department, Institute of Information and Communication Technologies – Bulgarian Academy of Sciences. Major fields of scientific research: distributed information systems design, methods and tools for net models researches. Educations: Technical University – Sofia, speciality Computer Science – Computer Design, 1977.*

Contacts:
Institute of Information and Communication Technologies
Bulgarian Academy of Sciences
e-mail: kkolchakov@iit.bas.bg

Assoc. Prof. Dr Vladimir Monov *received his M.Sc. degree in Electrical Engineering with qualification in Control Engineering and Automation from Technical University, Sofia and Ph.D. degree in Technical Sciences, Bulgarian Academy of Sciences. Since 2010, Dr V. Monov has been the Head of Modeling and Optimization Department at the Institute of Information and Communication Technologies, Bulgarian Academy of Sciences. His professional interests and research activity are in the areas of systems and control theory, information and communication systems, business management systems, applied mathematics and operations research.*

Contacts:
Institute of Information and Communication Technologies
Bulgarian Academy of Sciences
e-mail: vmonov@iit.bas.bg