БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ • BULGARIAN ACADEMY OF SCIENCES

ПРОБЛЕМИ НА ТЕХНИЧЕСКАТА КИБЕРНЕТИКА И РОБОТИКАТА, **66**
PROBLEMS OF ENGINEERING CYBERNETICS AND ROBOTICS, **66**

София • 2015 • Sofia

Comparative Analysis of a Class of Algorithms for Traffic Management in a Crossbar Commutator with Respect to Complex Performance, Speed and Memory

K. Kolchakov, V.Monov

Institute of Information and Communication Technologies, Bulgarian Academy of Sciences
Email kkolchakov@abv.bg, vmonov@iit.bas.

Abstract: *In this paper we study fourteen algorithms for obtaining of non-conflict schedules in the switching nodes of type Crossbar. Our comparative analysis of the algorithms gives an overview of their potentiality related to the complex performance, speed and required memory as a function of the size N of the input connectivity matrix T.*

Key words: *network nodes, node traffic, crossbar switch, conflict elimination, packet messages.*

INTRODUCTION

The paper discusses fourteen algorithms developed to obtain conflict-free schedule in the switching nodes of type Crossbar. Conflicts in the node arise 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 algorithms under consideration solve the problem of avoiding conflicts as long as it is directly related to the switching node performance.

The status of the switch in the switching node is represented by 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 member $T_{ij} = 1$ if a 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].

In this work we present a comparative analysis of fourteen algorithms in terms of speed and required memory in the cases of different sizes of the connection matrix by using fourteen software models corresponding to the algorithms.

ALGORITHMS FOR CONFLICT PROBLEM SOLVING

Studying the properties of various algorithms for non-conflict scheduling in crossbar switching nodes is of essential importance in solving the conflict issue problem. Our comparative analysis of fourteen algorithms enables us to determine the most appropriate algorithm with respect to the studied parameters in the cases of connection matrixes with different sizes. The following algorithms for obtaining non-conflict schedules are considered:

1. Classic algorithm with masks matrixes (**CMA**), [12].
2. Algorithm with joint mask matrixes (**JMA**), [12].
3. Classic algorithm without masks matrixes (**CWA**), [13].
4. Algorithm considering the message direction (**DAA**), [16].
5. An algorithm by diagonal connectivity matrix activation (**ADA**), [6].
6. Algorithm with joint diagonals activations (**AJDA**), [4].
7. Algorithm with diagonal activations of joint sub-switching matrices (**ADAJS**), [2].
8. Classic algorithm with sparse mask matrixes (**CSM**), described and examined in [7].
9. Algorithm with joint sparse mask matrixes (**JSM**), described and examined in [7].
10. Adaptive algorithm for management by weight coefficient of the traffic in Crossbar commutator (**AAM**), [1].
11. Optimum adaptive algorithm for management by weight coefficient of the traffic in Crossbar commutator (**AAMO**), [1].
12. An algorithm by diagonal connectivity matrix activation by finite automat (**ADAFA**), [3].
13. Algorithm with joint diagonals activations by finite automat (**AJDAFA**), [3].

14. Algorithm with diagonal activations of joint sub-switching matrices by finite automat (**ADAJSFA**), [3].

For the algorithms investigation, we use appropriate software models developed and examined in the cited references as it is shown in the following table 1.

Table 1. Software models

Software model	Reference
SMAAM (software model based on the Adaptive algorithm for management by weight coefficient of the traffic in Crossbar commutator) **SMAAMO** (software model based on the Optimum adaptive algorithm for management by weight coefficient of the traffic in Crossbar commutator).	[1]
SMADAJS (software model based on the algorithm with diagonal activations of joint sub-switching matrices)	[2]
SMADAFA (software model based on the algorithm by diagonal connectivity matrix activation by finite automat) **SMAJDAFA** (software model based on the algorithm with joint diagonals activations by finite automat) **SMADAJSFA** (software model based on the algorithm with diagonal activations of joint sub-switching matrices by finite automats)	[3]
SMAJDA (software model based on the algorithm with joint diagonals activations)	[5]
SMADA (software model based on the algorithm by diagonal connectivity matrix activation)	[6]
SMCSM (software model based on a classic algorithm with sparse mask matrixes) **SMJSM** (software model based on the algorithm with joint sparse mask matrixes).	[8]
SMCMA (software model based on the classic algorithm with masks matrixes) **SMJMA** (software model based on the algorithm with joint mask matrixes) **SMCWA** (software model based on a classic algorithm without mask matrixes).	[11]
SMDAA (software model based on an algorithm considering the message direction).	[16]

EXAMINATION OF SOFTWARE MODELS

Table 2 and Table 3 show the results from the software models investigation with respect to speed of execution **S[Sec.]** and memory resources **M[KB].** Figure 1 and Figure 2 illustrate the results from Table 2 and Table 3 graphically.

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].

R=R(v)+R(w) **(1)**

P=(R(w)/R).100[%] **(2)**

From formula 2 it is seen that when the nil solutions **R(v)** vanish to nil, then the performance P vanish to 100%[1].

Table 2. Speed.

N	SMCWA S[Sec.]	SMDAA S[Sec.]	SMJMA S[Sec.]	SMCMA S[Sec.]	SMCSM S[Sec.]	SMADA S[Sec.]	SMAJDA S[Sec.]	SMJSM S[Sec.]	SMAAM S[Sec.]	SMAAMO S[Sec.]
10	0,61	0,67	0,18	0,33	-	-	-	-	0,68	1,01
20	1,87	0,671	3,25	1,23	-	-	-	-	4,13	6, 48
30	5,56	5,54	11,86	3,82	-	-	-	-	21,98	32,42
40	12,46	17,29	70,93	5,16	-	-	-	-	59,70	93,75
50	21,67	30,43	91,93	7,91	0,256	0,174	-	-	147,86	195,82
60	39,36	55,30	-	13,11	-	-	-	-	332,85	429,29
70	59,57	82,58	-	20,81	-	-	-	-	659,57	763,48
80	87,92	-	-	31,82	-	-	-	-	-	-
90	-	-	-	44,90	-	-	-	-	-	-
100	-	-	-	61,33	0,88	0,96	1,81	1,67	-	-
150	-	-	-	-	2,79	1,61	3,53	3,67	-	-
200	-	-	-	-	4,40	3,22	6,89	5,81	-	-
250	-	-	-	-	8,30	4,78	7,45	8,09	-	-
300	-	-	-	-	14,97	7,43	12,42	11,17	-	-
N	**SMADAFA S[Sec.]**	**SMAJDAFA S[Sec.]**	**SMADAJSFA S[Sec.]**	**SMADAJS S[Sec.]**						
100	38,95	39,69	0,85	0,38						
150	122,59	122,15	1,29	0,53						
200	292,99	291,4	2,10	0,59						
250	605,89	581,23	2,99	0,75						
300	985,8	902,52	3,22	0,83						

Table 3. Needed memory.

N	SMCWA M[KB]	SMDAA M[KB]	SMJMA M[KB]	SMCMA M[KB]	SMCSM M[KB]	SMADA M[KB]	SMAJDA M[KB]	SMJSM M[KB]	SMAAM M[KB]	SMAAMO M[KB]
10	8,816	20,016	38,4	46,4	-	-	-	-	4,336	9,424
20	35,216	80,016	313,6	377,6	-	-	-	-	17,004	37,400
30	79,216	180,016	1065,6	1274,4	-	-	-	-	38,240	83,440
40	140,816	320,016	2534,4	3033,6	-	-	-	-	67,492	149,488
50	200,016	500,016	4960	5960	-	-	-	-	106,032	230,864
60	316,816	720,016		10310,4	-	-	-	-	151,784	332,968
70	431,216	980,016		16385,6	-	-	-	-	207,844	452,752
80	563,216			24473,6	-	-	-	-	-	-
90				34862,4	-	-	-	-	-	-
100				47840	4030	401	401	4010	-	-
150	-	-	-	-	32080	902	902	32030	-	-
200	-	-	-	-	108140	1630	1630	108140	-	-
250	-	-	-	-	256230	2540	2540	256060	-	-
300	-	-	-	-	500340	3604	3604	500080	-	-
N	**SMADAFA M[KB]**	**SMAJDAFA M[KB]**	**SMADAJSFA M[KB]**	**SMADAJS M[KB]**						
100	640	640	1,048	0,760						
150	1440	1440	1,048	0,760						
200	2560	2560	1,048	0,760						
250	4000	4000	1,048	0,760						
300	5760	5760	1,048	0,760						

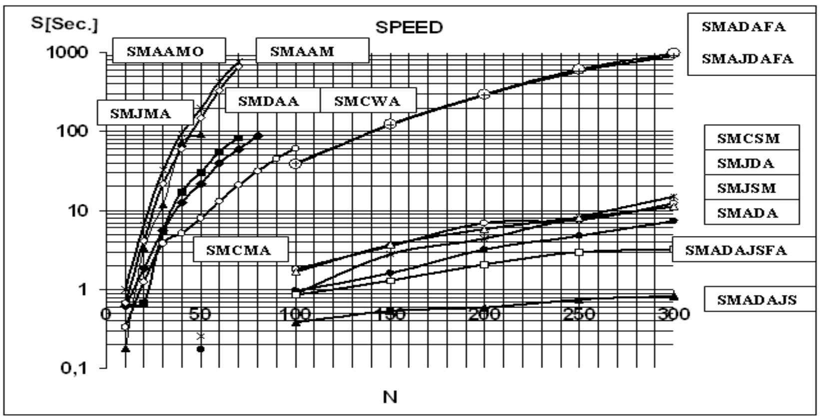

Figure 1. Speed

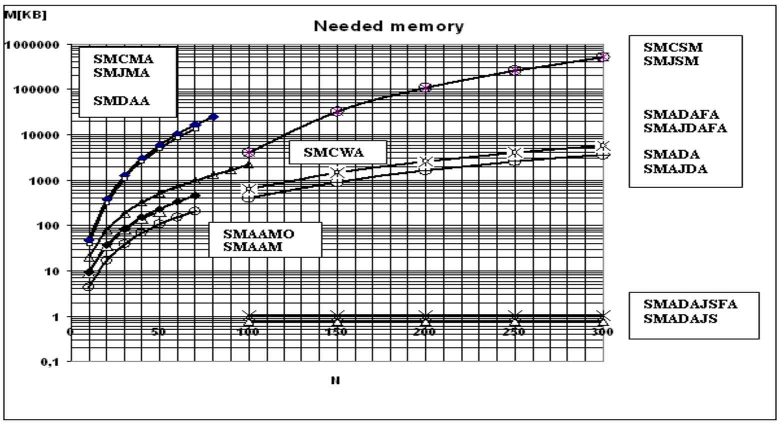

Figure 2. Needed memory.

To facilitate the performance examination, 5 kinds of matrixes for simulation of the input connectivity matrix **T** are chosen. The special input matrixes **2A, 2B, 2C, 2D** and **2E**[1]are represented on **Figure 3**. **Table 4** represents the investigation results related to the performance **P** of the software models.

The results of the study of algorithms with respect to the performance P from Table 4 are presented graphically in Figure 4.

1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1

2A

1	1	0	0	0
1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	0	1	1

2B

0	0	1	1	1
0	0	0	1	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0

2C

0	0	0	1	1
1	0	1	0	1
0	1	1	0	0
1	1	1	1	0
1	0	1	1	1

2D

1	1	1	1	1
0	1	1	1	1
0	0	1	1	1
0	0	0	1	1
0	0	0	0	1

2E

Figure 3. Five special input matrices.

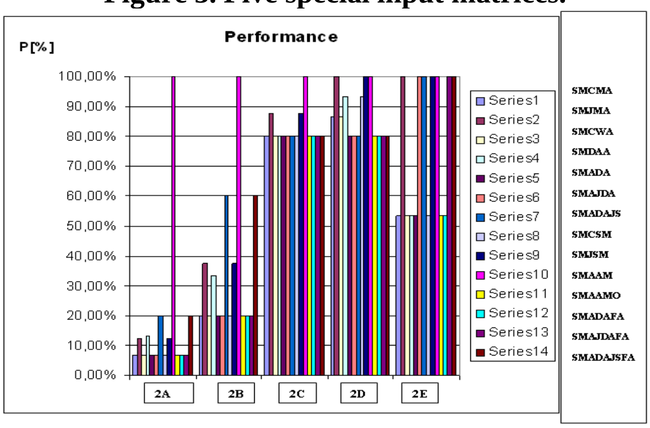

Figure 4. Performance

Table 4. Performance

P[%]	2A	2B	2C	2D	2E
SMCMA	6,66%	20%	80%	86,6%	53,3%
SMJMA	12,5%	37,5%	87,5%	100%	100%
SMCWA	6,66%	20%	80%	86,6%	53,3%
SMDAA	13,3%	33,3%	80%	93,3%	53,3%
SMADA	6,66%	20%	80%	80%	53,3%
SMAJDA	6,66%	20%	80%	80%	100%
SMADAJS	20%	60%	80%	80%	100%
SMCSM	6,66%	20%	80%	93,3%	53,3%
SMJSM	12,5%	37,5%	87,5%	100%	100%
SMAAM	100%	100%	100%	100%	100%
SMAAMO	6,66%	20%	80%	80%	53,3%
SMADAFA	6,66%	20%	80%	80%	53,3%
SMAJDAFA	6,66%	20%	80%	80%	100%
SMADAJSFA	20%	60%	80%	80%	100%

From the results in Table 4 it is seen that there is a programming model with 100% performance with different special input matrices. In SMAAMT there is a detector of requests that locate them so that the programming model processes only them by assigning individual weight coefficients without generating zero solutions (R (v) = 0). In the optimized version SMAAMO weight coefficients are assigned to requests located in diagonals parallel to the main diagonal in the connection matrix and the requests in a given diagonal have the same weight. This approach has been adopted in order to increase the speed, but a disadvantage is the appearance of zero solutions.

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 talso taken into account as follows:

CP = P.t, for N = const. , t = 1/S **(3)**

In formula (3) we choose the value of N to be N = 100, because we have data for the speed S at N = 100 for nine of the fourteen algorithms and this will make the study representative. For software models SMJMA, SMCWA, SMDAA, SMAAM and SMAAMO the value of N is N=50 and for all other models we have N = 100. Table 5 shows the results, and Figure 5 gives a graphical representation.

Table 5. Complex Performance.

CP	**2A**	**2B**	**2C**	**2D**	**2E**
SMCMA	0,0010	0,0032	0,0130	1,4115	0,0086
SMJMA	0,0013	0,0040	0,0094	1.0800	0,0108
SMCWA	0,0030	0,0092	0,0036	3,9920	0,0245
SMDAA	0,0043	0,0109	0,0262	3,0602	0,0174
SMADA	0,0693	0,2083	0,8332	83,328	0,5551
SMAJDA	0,0367	0,1104	0,4419	44,1920	0,5524
SMADAJS	0,5263	1,5789	2,1050	210,5200	2,6315
SMCSM	0,0756	0,2272	0,9090	106,0167	0,6056
SMJSM	0,0748	0,2245	0,5239	59,8800	0,5988
SMAAM	0,6700	0,6700	0,0067	0,6700	0,0067
SMAAMO	0,0003	0,0010	0,0040	0,4080	0,0027
SMADAFA	0,0017	0,0051	0,0204	2,0480	0,0136
SMAJDAFA	0,0016	0,0050	0,0200	2,0080	0,0251
SMADAJSFA	0,235	0,7058	0,9411	94,1120	1,1764

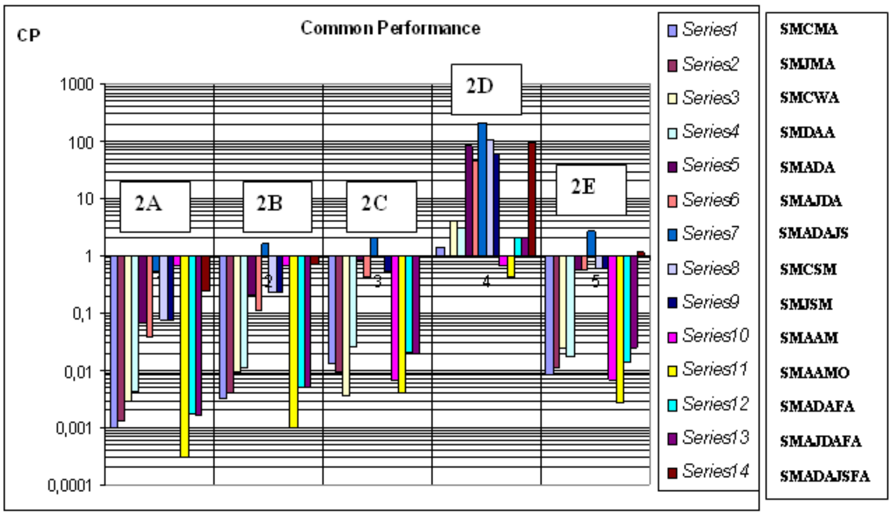

Figure 5. Complex Performance.

CONCLUSION

Software models **SMCMA, SMCWA, SMDAA**, **SMJMA, SMAAM** and **SMAAMO** are times slower than models **SMCSM, SMADA, SMAJDA, SMJSM**, **SMADAJSFA** and **SMADAJS** making them unsuitable for connections matrix sizes greater than one hundred.

Among software models **SMCSM, SMAJDA, SMJSM SMADA, SMADAJSFA** and **SMADAJS,** the fastest is **SMADAJS**. In terms of memory needed software model **SMADAJS** is most economical. The model is based on the algorithm with diagonal activations of joint sub-switching matrices (**ADAJS**).

The traffic is presented usually in the best way by an input matrix of type **2D**. Performance of all software models for **2D** is equal to or greater than 80%. **JMA** and **JSM** algorithms are optimal related to the performance **P** in the case of **2D** input matrix.

Our study of the complex performance CP shows that the optimal model is **SMADAJS**, followed by **SMCSM**, **SMADAJSFA**, **SMADA**, **SMJSM** and **SMAJDA** for input matrix **2D** as closest to the normal traffic.

ACKNOWLEDGEMENTS

The research work reported in the paper is partly supported by the project AComIn "Advanced Computing for Innovation", grant 316087, funded by the FP7 Capacity Programme (Research Potential of Convergence Regions).

REFERENCES

1. Kolchakov K., Monov V. Adaptive Algorithm for Management by Weight Coefficients of the Traffic in Crossbar Commutator , International Journal " Information Models and Analyses" Vol.4, № 1, pp. 53-60. http://www.foibg.com/ijima/vol04/ijima04-01-p05.pdf
2. Kolchakov K, Monov V. An Algorithm for Non – Conflict Schedule with Diagonal Activation of Joint Sub Matrices, 17-th International Conference on "DISTRIBUTED COMPUTER AND COMMUNICATION NETWORKS (DCCN-2013): "CONTROL, COMPUTATION, COMMUNICATIONS, Moscow, October 07-10, 2013, pp. 180-187, ISBN 978-5-94836-366-0.
3. Kolchakov K. Non- conflict schedule through finite automats in switching nodes. International Conference Automatics and Informatics `13 03-07.10.2013 Sofia, Bulgaria. Proceedings: ISSN 1313 – 1850, Proceedings CD: ISSN 1313 – 1869. pp. I-191 - I-194.
4. Kolchakov K., Taschev T. "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, VOL.630, pp.245-250, ISBN 978-1-4503-1193-9.
5. Kolchakov K. Examination on Algorithms for Non Conflict Schedule with Diagonal Activation in Case of a Large Size Switching Matrix. International Conference Automatics and Informatics `12 03-05.10.2012 Sofia, Bulgaria. Proceedings CD: ISSN 1313 – 1869. pp.341-344.
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 03.10-07.10.2011., pp. B-247 – B251, Proceedings ISSN 1313-1850, CD ISSN 1313-1869
7. K. Kolchakov, "An Approach for Performance Improvement of Class of Algorithms for Synthesis of Non-conflict Schedule in the Switch Nodes" , Proceedings of the 11th International Conference on Computer Systems and Technologies, CompSysTech'10, 17-18 June 2010, Sofia, Bulgaria, pp.235-239, ISBN 978-1-4503-0243-2.
8. Kolchakov K., Software Models with Sparse Mask Matrixes for Non-Conflict Schedule obtaining in a Switching Node, Доклад в трудове на международна конференция (DCCN – 2009), "Distributed Computer and communication networks DCCN 2009", Moscow: R&D Company "Information and Networking technologies" ISBN 978-5-9901871-1-5 София, България, Октомври 5 – 9, 2009, стр. 121-126.
9. K.H.Kolchakov, An approach for non-conflict schedule synthesis –modeling and optimization, Доклад в Сборник научных трудов том 10 Интелектуальные системы и технологии УДК 004.896(06), Научная сесия МИФИ – 2008, Москва 2008, стр. 100 – 101, ISBN 978-5-7262-0883-1.
10. Kolchakov K.H., Non-conflict Schedule Synthesis Based on Communication Matrix Scanning, Доклад в Сборник научных трудов том 3 Интелектуальные системы и технологии, Научная сесия МИФИ – 2007, Москва 2007, стр.132 - 133.
11. Kolchakov K., A Modeling Approach for Obtaining of a Non – Conflict Schedule, "Distributed Computer and communication networks DCCN 2007", Moscow, September 10-12, 2007, pp.168-173.
12. К.Колчаков "Моделиране на подход с матрици-маски за получаване на безконфликтно разписание в комутационен възел".Работна статия в поредицата на ИИТ-БАН, IIT/WP-238B, Ноември 2007.
13. K.Kolchakov, Algorithm for Synthesis of Non-conflict Schedule in a Communication Node, Доклад на Българо-руския семинар (DCCN – 2006), Sofia, Bulgaria, Oktober 30 - November 2, 2006, pp. 149 – 155.
14. K.H.Kolchakov, Synthesis approach of optimized non-conflict schedule in communication node, Доклад в Сборник научных трудов том 3 Интелектуальные системы и технологии, Научная сесия МИФИ – 2005, Москва 2005, стр. 162-163.
15. P. Gupta, N. McKeown, "Designing and Implementing a Fast Crossbar Scheduler", IEEE *Micro*, Jan.-Feb. 1999, pp.20-28.
16. Kolchakov, K., Synthesis of an Optimized Non-conflict Schedule Accounting the Direction of Messages Transfer in a Communication Node. Cybernetics and Information Technologies, Bulgarian Academy of Scienses, Vol. 4, No 2, 2004, pp. 88-95, Sofia.

17. D. Serpanos, P. Antoniadis:"FIRM: A Class of Distributed Scheduling Algorithms for High-Speed ATM Switches with Multiple Input Queues", *IEEE Infocom 2000 Conference,* Tel Aviv, Israel, March 2000.

Сравнительный анализ класса алгоритмов для управления трафика в кросбар переключателе в отношении комплексного производительности, быстродействий и необходимой памяти

Кирил Колчаков, Владимир Монов

Институт информационных и коммуникационных технологии

(Резюме)

Исследованы четырнадцать алгоритм для получения бесконфликтного расписания в кросбар переключателе. Были исследованы возможности алгоритмов в отношении к комплексной производительност, быстродействий и необходимой памяти в зависимости размера входной матрицы.