Comparison of Sequential Methods for Getting Separations of Parallel Logic Control Algorithms Using Volunteer Computing

Eduard Vatutin, Sergey Valyaev, and Vitaly Titov

Southwest State University, Kursk, Russia
evatutin@rambler.ru, serval@boinc.ru, titov-kstu@rambler.ru

Abstract. The article deals with the problem of getting suboptimal separations of graph-schemes of parallel algorithms which arises in the design of logical control systems in the basis of logic multicontrollers. The article brief describes the sequential heuristic methods used to solve the problem. The methodic of experimental comparison for estimation the quality of solutions based on the performing of computational experiments using the BOINC platform is considered. It also shows description of obtained experimental results which allows to identify the areas of the preferable use of sequential heuristic methods depending on the size of the problem and power of constraints. Given results also allows to formulate recommendations for structural-parametric optimization of logical control systems.

Keywords: logic control systems, logic multicontrollers, structural-parametric optimization, graph-schemes of parallel algorithms, separations, heuristic methods, discrete combinatorial optimization, BOINC.

1 Introduction

One of the promising approaches for design of logic control systems (LCS) working with parallel logic control algorithms based on its implementation at the logic multicontrollers (LMC) basis represented by a multi-module homogeneous multisystems with regular topology [1, 2, 3]. The initial problem of logic control for corresponding operational part is solved through finding the separation of parallel graph-scheme of logical control algorithm to blocks of limited complexity and their subsequent placement at the modules of multisystem. Quality of decision directly influences the hardware complexity of designed LCS and determines its speed characteristics.

Precise decision of the problem for graph-schemes with practical important size (at least some hundreds of vertices) can't be possible because the problem corresponds to NP-full class (the number of possible separations is bounded above by the Bell number). Therefore, in practice, for getting solutions different heuristic methods are used to produce quite good quality of solutions in a reasonable time. In various conditions of application (different size of the problem and power of constraints) they demonstrate the preparation of solutions with significantly different quality, therefore, the

actual problem is to compare them in different conditions in order to make recommendations on areas of their preferable application. This problem is computationally difficult and weakly coupled, which allows effective organization of corresponding computational experiments using grid systems on a voluntary basis that was implemented within Gerasim@Home project based on BOINC platform.

2 Statement of the Problem

A formal presentation of the problem of getting separation has the following form. It is required to obtain a separation $Sep\left(A^0\right)=\left\{A_1, A_2, ..., A_H\right\}$ of the set of vertices A^0 for source graph-scheme of parallel logic control algorithm $G^0=\left\langle A^0, V^0\right\rangle$ satisfying the following conditions:

$$\begin{gathered}\bigcup_{i=1}^{H} A_i = A^0, \quad A_i \neq \varnothing, \quad A_i \cap A_j = \varnothing, \quad i, j = \overline{1,H}, \quad i \neq j, \\ \neg\left(a_i \omega a_j\right) \forall a_i, a_j \in A_k, \quad i \neq j, \quad k = \overline{1,H}, \\ W\left(A_i\right) \leq W_{\max}, \quad \left|X\left(A_i\right)\right| \leq X_{\max}, \quad \left|Y\left(A_i\right)\right| \leq Y_{\max}, \quad i = \overline{1,H},\end{gathered} \tag{1}$$

where $W\left(A_i\right)=\sum_{a_j \in A_i} W\left(a_j\right)$ – summary "weight" of vertices within the *i*-th block (size in controller memory measured at microcommands); $X\left(A_i\right)=\bigcup_{a_j \in A_i} X\left(a_j\right)$ – a set of logical conditions included in the vertices within the *i*-th block; $Y\left(A_i\right)=\bigcup_{a_j \in A_i} Y\left(a_j\right)$ – a set of microoperations included in the vertices within the *i*-th block, $W_{\max}$ – constraint in the memory capacity of the controller as part of LCS, $X_{\max}$ – constraint in the number of signals of logic conditions received by the controller, $Y_{\max}$ – constraint in the number of microoperation signals issued by the controller, such that

$$\begin{gathered}Z_H = H\left(Sep\left(A^0\right)\right) \rightarrow \min, \\ Z_\alpha = \sum_{i=1}^{H} \sum_{j=1, j \neq i}^{H} \alpha\left(A_i, A_j\right) \rightarrow \min, \\ Z_\delta = \delta\left(Sep\left(A^0\right)\right) \rightarrow \min, \\ Z_X = \sum_{i=1}^{H} \left|X\left(A_i\right)\right| - \left|X\left(A^0\right)\right| \rightarrow \min, \\ Z_Y = \sum_{i=1}^{H} \left|Y\left(A_i\right)\right| - \left|Y\left(A^0\right)\right| \rightarrow \min,\end{gathered} \tag{2}$$

where $Z_H = H\left(Sep\left(A^0\right)\right)$ – the number of blocks in the separation; Z_α – complexity of the interconnect network for separation $Sep\left(A^0\right)$; $\alpha\left(A_i, A_j\right)$ – coupling coefficient of blocks pair (it is equal to 1 if the blocks are connected by the control in direction from A_i to A_j, which requires an additional command for transfer of control between controllers, and 0 otherwise); $Z_\delta = \delta\left(Sep\left(A^0\right)\right)$ – the total number of interblock interactions (summary interblock traffic); Z_X – the extent of duplication of logical conditions signals; Z_Y – the extent of duplication of microoperations signals. Minimizing these partial quality criteria can decrease hardware complexity of LCS (criteria Z_H, Z_X, Z_Y, Z_α) by reducing the required number of controllers, complexity of interconnect network and hardware complexity of separate controllers, and improve performance of LCS (criterion Z_δ) by reducing control traffic and communication subsystem loading. These partial quality criteria (2) can be brought together as part of the integral criterion, which is a weighted sum of the normalized values of partial criteria [4]:

$$J\left(Sep\left(A^0\right)\right) = \frac{K_H}{\omega_{\max}} H + \frac{K_X}{\left|X\left(A^0\right)\right|}\left(\sum_{i=1}^{H}\left|X\left(A_i\right)\right| - \left|X\left(A^0\right)\right|\right) +$$
$$+ \frac{K_Y}{\left|Y\left(A^0\right)\right|}\left(\sum_{i=1}^{H}\left|Y\left(A_i\right)\right| - \left|Y\left(A^0\right)\right|\right) + \frac{K_\delta}{\delta\left(A^0\right)}\delta\left(Sep\left(A^0\right)\right) + \qquad (3)$$
$$+ \frac{K_\alpha}{\omega_{\max}\left(\omega_{\max} - 1\right)}\sum_{i=1}^{H}\sum_{j=1,\ i\neq j}^{H}\alpha\left(A_i, A_j\right),$$

where $K_H, K_X, K_Y, K_\alpha, K_\delta$ – weights determined by an expert, $\omega_{\max}$ – degree of parallelism of graph-scheme [5], $\delta\left(A^0\right)$ – the maximum possible intensity of interblock interactions for initial graph-scheme determined by assuming that each vertex forms a separate block.

Comparison of decisions quality of various heuristic methods carried out by generating the samples of graph-schemes $\Lambda = \left\{G_1^0, G_2^0, ..., G_K^0\right\}$ (appropriate generator of graph-schemes with random structure [6] is used in the experiments for this purpose), building separations $Sep_{F_i}\left(A_j^0\right)$, $j = \overline{1, K}$ using different methods F_i followed by evaluation of average values of quality criteria $\gamma_z\left(F_i\right)$, $z \in \left\{H, X, Y, \alpha, \delta, J\right\}$ and probabilities $\rho_z\left(F_i\right)$ of obtaining the quasi-optimal solutions with a minimum value of selected quality criteria among the solutions of other methods (conditional optimum). These operations are performed using the developed program system PAE [7, 8] and corresponding computing unit [9] based on working as a part of the grid under BOINC Manager control.

In a series of computational experiments it has been shown [10, 11, 12, 13] that quality of solutions obtained using different heuristic methods essentially depends on the size of the problem N and power of constraints $X_{\max}$, $Y_{\max}$ и $W_{\max}$, and to make recommendations on the applicability of various heuristic methods it is necessary to study the space formed by the values $\left(N, X_{\max}, W_{\max}\right)$ and shown in Fig. 1. Constraint $Y_{\max}$ in this case is not critical and can be excluded from consideration by duplicating the corresponding controller in LCS [1].

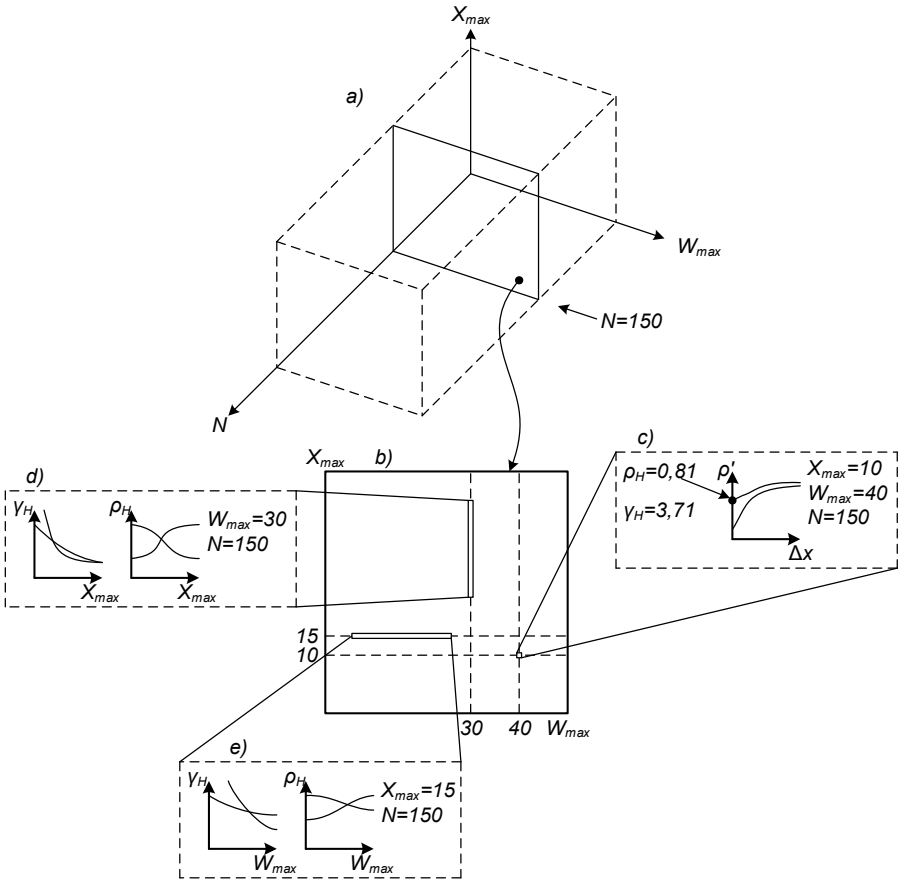

Fig. 1. Space of parameters (a), map of the slice $\left(X_{\max}, W_{\max}\right)$ (b) and early computational experiments (c, d, e)

3 Brief Classification of Heuristic Methods for Getting Separations

Heuristic methods for the synthesis of separations can be divided into two broad classes: sequential and iterative [1, 2, 3]. Methods of first class provides a single solution using various heuristic rules. They are characterized by low asymptotic time and space complexity (usually not more than $O\left(N^2\right)-O\left(N^3\right)$) and, as a consequence, a relatively low cost of computation time during obtaining solutions. Methods of the second class produce limited exhaustion of subset of solutions, evaluation of their quality and then selecting the best of them. They are characterized by significantly high cost of computing time (at least by 1–2 orders higher that sequential methods). As part of the computational experiments the following methods were used:

- S.I. Baranov method [14, 15] that implements greedy approach based on weighting function that shows optimality of including selected vertex a_i to current block A_j and taking into account constraints (1);
- modification of S.I. Baranov method [16, 17] that implements greedy approach with sequential formation of blocks and additional constraint on the consideration of vertices only from adjacent neighborhood of current block;
- method of parallel-sequential decomposition [18, 19] which uses parallel-sequential strategy during building the separation and based on performing a series of equivalent transformations of source graph-scheme such as breaking of loops [20], uniting linear ways [21], building matrix of relations [22], building a set of sections with using mathematical apparatus of *R*-expressions [23, 24], building blocks of separation in parallel [25], with a number of modifications that improve the quality of the solutions [26] and decrease computing time costs [27, 28, 29].

For each of them the appropriate software implementation [15, 17, 19] was developed which is made in the form of a DLL library and used in the process of getting separations using late binding mechanism.

4 Parallelization Strategy Using BOINC Grid

Analyzing the entire space shown in Fig. 1 at the moment it is not possible even with a grid because it would require hundreds of years of calculations with real performance of the project in the region of several teraflops followed by several tens of terabytes of experimental data, so experiments were organized to analyze two-dimensional slices (maps) with the following parameters:

- $3 \le W_{\max} \le 200$, $1 \le N \le 600$ with $X_{\max} = +\infty$ and $Y_{\max} = +\infty$;
- $4 \le X_{\max} \le 150$, $1 \le N \le 700$ with $W_{\max} = +\infty$ and $Y_{\max} = +\infty$.

The upper boundary for the range of variation in the experiments constraints determined by the location of borders of bend and further weakening of constraints is

impractical for selected upper limit for size of the problem. The construction of such maps is quite resource intensive (especially with growth of *N*) and requires hundreds of years of CPU time, which is impossible without parallelization of the computational process. This problem, as mentioned above, is weakly coupled and, therefore, can be effectively parallelized by following strategies:

1. The individual elements of the parameter space (points of the map) are input parameters for the generator of graph-schemes of control algorithms with pseudo-random structure and for methods of synthesis of partitions can be processed in parallel.
2. The individual separations $Sep_{F_i}\left(A_j^0\right),\ j=\overline{1,K}$ as part of the sample Λ of graph-schemes of algorithms corresponding to each of the points on the map can be processed in parallel.

When targeting computations oriented to BOINC grid computing time of workunits should not be very small and preferably should be from a few minutes to several hours, which is implemented as outlined above using the following strategies: according to the first strategy samples of graph-schemes of algorithms for different points of the parameter space are formed in parallel on different machines within the grid; constructing of individual separations, each of which requires no more than a few seconds of CPU time, within the sample produced sequentially (the second strategy of parallelization is not realized, which is useful for areas with low value of size of the problem).

As tools for parallel computing technical capabilities of the volunteer distributed computing platform BOINC [30] have been used, which has been successfully used in several projects, including the Russian projects SAT@Home, Optima@Home and NetMax@Home [33, 34, 35]. The server part of the code, implemented in the distributed computing project Gerasim@Home [31], serves as a control issue of workunits (source parameter file in XML format for calculating at volunteer machine sized by a few hundred bytes) on remote client machines upon request from the BOINC Manager software and receiving the obtained results (the resulting binary file with sample of quality criteria values sized by about 200 KB without compression). To protect against potential errors of different character each task was processed in duplicate on different machines (quorum 2), followed by byte-to-byte collation of the results at server side (validation). Costs of computing time to conduct experiments using the grid infrastructure totaled about 267 years (81,2 EFLOPs) of Intel Core 2 Duo E6300 processor time, achieved a gain in processing time as compared with implementation of the calculations made on a single machine is 558 times. Calculations were attended by more than 1,300 volunteers (900 PCs) from 69 countries of the world that provide average real performance of the project by [32] at the level of 2–2,5 TFLOP/s (3 TFLOP/s during command challenges). The size of received "raw" experiment data was 409 GB.

Sequential part of the calculations (post-processing) includes determining the average quality criteria values γ_z for each of the samples separately and probabilities ρ_z by comparing samples obtained by different methods of getting separations (generally

on the different machines within the grid at different time) for the same selected point at the parameter space. Post-processing time was of the order of several tens of hours for each computational experiment (slice of space), the main limitation in this case is the bandwidth of random reading large amounts of data from HDD. As a result of post-processing maps (two-dimensional arrays) were formed with size of the order of several tens of megabytes that are used in the future for a detailed comparison of methods for getting separations.

5 Analysis of the Experimental Data

5.1 Analysis of Average Values of the Quality Criteria

During the analysis of the experimental data it was found out that dependences of average values of quality criteria on the size of the problem and force the selected constraint are qualitatively similar and have the form shown in Fig. 2.

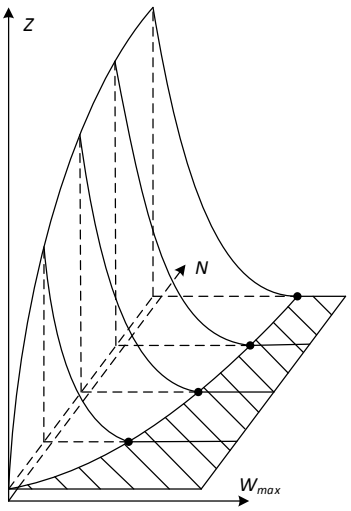

Fig. 2. The general behavior of partial quality criteria *Z* depending on size of the problem *N* and power of constraint ($W_{\max}$ in this example). Shaded area shows the area of insensitivity

In this case, there is an area designated in Fig. 2 hatching, in which the weakening of constraints does not influence the change in value of the quality criteria. With the increasing constraint value (in the example in Fig. 2 – decreasing value $W_{\max}$) there is a monotonic increase in the value of partial quality criteria for graph-schemes of algorithms with the same selected size. In view of these features it is possible to formulate recommendations for developers of hardware part of LCS which has been done in [36, 37, 38], respectively, for the start of the bend (the corresponding values marked in Fig. 2 points) and for a 5% increase in the partial quality criteria that for most practical applications is an acceptable value and can significantly reduce the hardware requirements for LCS. Based on the results obtained during the structural-parametric

optimization it can be concluded that upon the presence of the balance requirements for LCS, LCS structure comprising a large number of relatively simple controllers is more rational in terms of relationship between their functionality and hardware costs. For each specific size of the problem it can be specified the special limit values of restrictions $X'_{\max}$ and $W'_{\max}$, increase of constraint values over these limits leads to excessive growth of the hardware complexity of LCS but does not lead to a accompanying significant increase of its performance. For example, during practical implementation of control algorithms contained $N = 200$ vertices sufficient structure of matrix multisystem contains 5×5 modules and each of them must include memory for storing microcommands with size $W'_{\max} = 18$ command words and $X'_{\max} = 6$ pins for receiving signals of logical conditions from controlling object.

5.2 Analysis of Probabilities for Obtaining a Quasi-optimal Solutions

During the comparison of heuristic methods the most indicative is the probabilities ρ_z of obtaining solutions in which value of selected partial quality criterion has minimal value compared with the solutions obtained by other heuristic methods with the same parameters of the experiment (in other words, selected criterion is quasi-optimal). These values allow to draw conclusions about how this or that method performs minimization of selected quality criterion in the given conditions. The obtained results are analyzed in detail in [36], an example of the experimental dependences for the adjacent greedy strategy are shown in Fig. 3 and 4 [39].

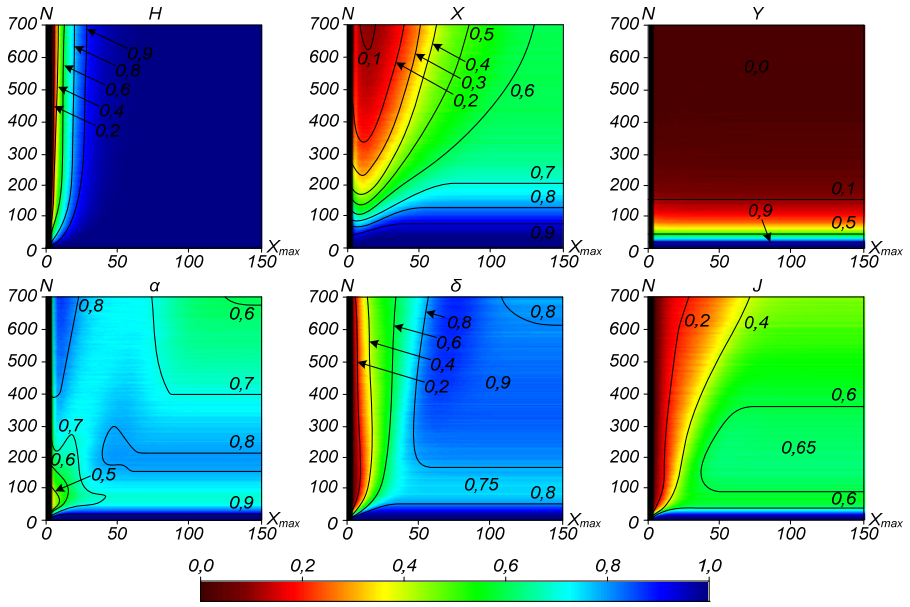

Fig. 3. Dependencies of ρ_z for adjacent greedy strategy in the slice $(X_{\max}, N)$

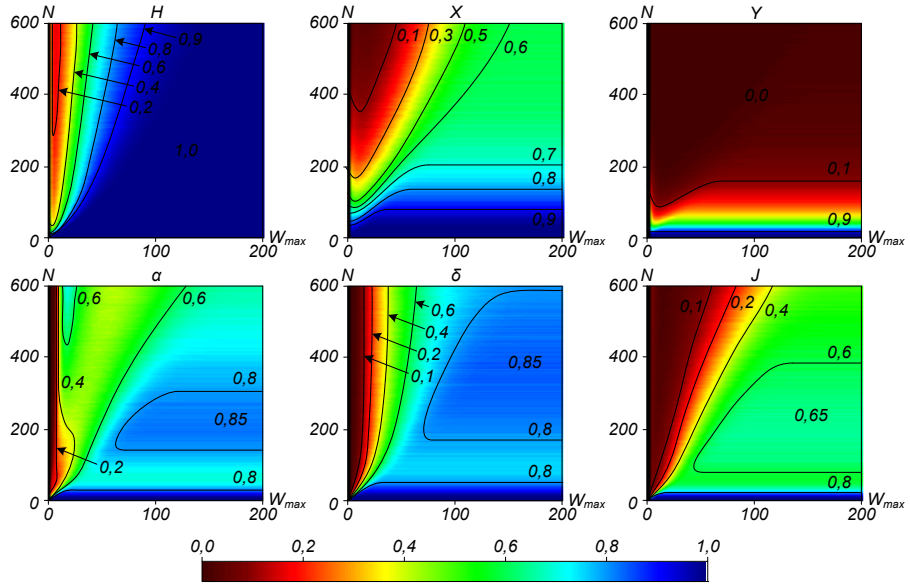

Fig. 4. Dependencies of ρ_z for adjacent greedy strategy in the slice $(W_{\max}, N)$

Analysis of the results confirm earlier made conclusions that in different areas of the parameter space the different methods show significantly different degree of minimization of quality criteria and, consequently, the probabilities of obtaining quasi-optimal solutions.

5.3 Analysis of the Areas of Preferable Use of Methods

By comparing the probabilities of obtaining quasi-optimal solutions for various heuristic methods it is possible to obtain the areas in which the use of the selected method is preferred and corresponding probability ρ_z has maximum value. Corresponding two-dimension maps [40, 41] are shown in Fig. 5 and 6. Analysis of the results allows to make main conclusion, confirming the previously obtained partial results, that the area of the preferable use of the method of S.I. Baranov is a region of weak constraints and low-size of problem, the method of parallel-sequential decomposition demonstrates the preparation of solutions of the highest quality in the field of strong constraints, and the method based on adjacent greedy strategy occupies an intermediate position in the field of medium strength constraints.

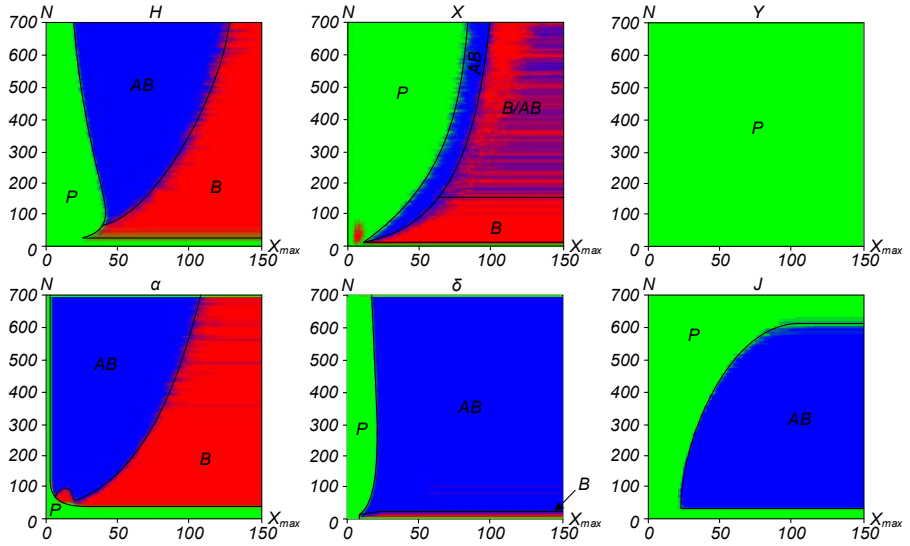

Fig. 5. Areas of preferable use of the methods in the slice $(X_{\max}, N)$. P – method of parallel-sequential decomposition, B – S.I. Baranov method, AB – adjacent greedy strategy

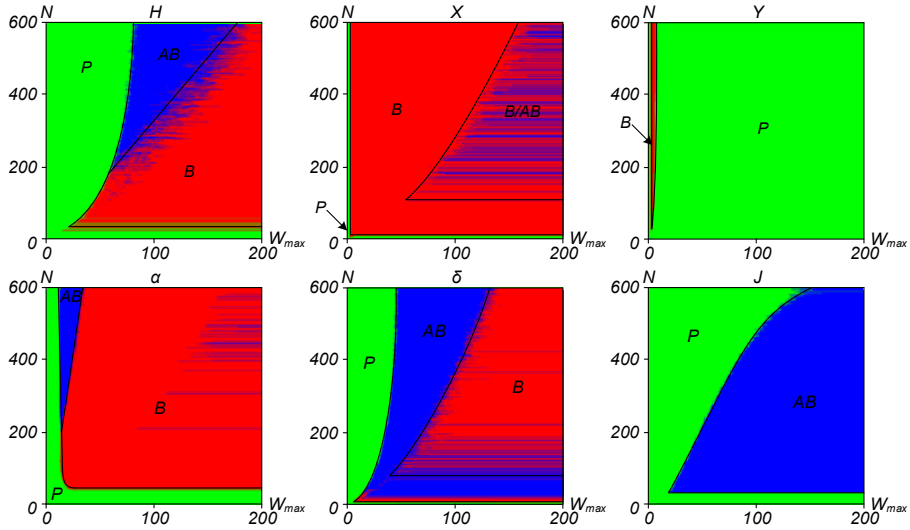

Fig. 6. Areas of preferable use of the methods in the slice $(W_{\max}, N)$

6 Prospects for the Future Investigations

The current series of experiments performed within Gerasim@Home project has several objectives. First, the expansion “up” (in the direction of greater size of the

problem) dependencies in Fig. 3–6 is interesting that will clarify the behavior of used sequential heuristic methods to large graph-schemes of parallel logic control algorithms. In addition, it would be interesting to study the quality of solutions obtained using different iterative methods such as random search [45] and its variations, modifications of limited depth first search, group intelligence approaches [42], modifications of simulated annealing [43] and so on. In order to test this method group the corresponding computing unit [46] has currently been developed and created a subproject, which goal is to analyze the quality of decisions in a more simple test problem of getting shortest path between selected pair of vertices in graph with using selected methods [47, 48, 49] including some additional features, for example, combinatorial returns [50].

7 Acknowledgements

The authors would like to thank all volunteers who took part in the calculation within the distributed computing project Gerasim@Home. The authors also wish to thank Anna Vayzbina for assistance in preparing the English version of the article. The work was performed within the base part of the state assignments for the Southwest State University in the 2014–2017 years and under support of scientific school NSh-2357.2014.8.

8 References

1. Zotov, I.V., Koloskov, V.A., Titov V.S. et al.: Organization and Synthesis of Microprogram Multimicrocontrollers (in Russian). Kursk State Technical University, Kursk (1999)
2. Vatutin, E.I., Zotov, I.V., Titov, V.S. et al.: Combinatorial-logic Problems of Synthesis of Separations of Parallel Logic Control Algorithms in Design of Logic Multicontrollers (in Russian). Kursk State Technical University, Kursk (2010)
3. Vatutin, E.I.: Logic Multicontrollers Design. Getting Separations of Parallel Logic Control Algorithms (in Russian). Lambert Academic Publishing, Saarbrucken (2011)
4. Vatutin, E.I.: Assessment of the Quality of Separations of Parallel Control Algorithms to Sequential Subalgorithms Using the Weighting Function (in Russian). Intellectual and Informational Systems (Intellect – 2005). pp. 29–30. Tula State University, Tula (2005)
5. Vatutin, E.I.: Determining the Parallelism Degree of Graph-scheme of Parallel Algorithm (in Russian). Intellectual and Informational Systems (Intellect – 2009). pp. 24–26. Tula State University, Tula (2009)
6. Vatutin, E.I.: Constructing Random Sample Parallel Logic Control Algorithms. 11th International Student Olympiad on Automatic Control (BOAC'06). pp. 162–166. IFMO, Saint-Petersburg (2006)
7. Vatutin, E.I., Zotov, I.V.: Program System for Getting Separations of Parallel Control Algorithms (in Russian). Systems Identification and Control Problems

(SICPRO'06). pp. 2239–2250. Institute of Control Problems of RAS, Moscow (2006)
8. Vatutin, E.I., Zotov, I.V.: Visual Environment for Getting Separations of Parallel Logic Control Algorithms (in Russian). Certificate of official registration of the computer software № 2007613222 from 30.07.2007.
9. Vatutin, E.I., Valyaev, S.Yu.: Computing Unit for Getting Separations of Parallel Logic Control Algorithms Using Volunteer Computing (in Russian). Certificate of official registration of the computer software № 2013618013 from 28.08.2013.
10. Vatutin, E.I., Abdel-Jalil, J.N., Najajra, M.H., Zotov, I.V.: Comparison of Methods for Getting Separation of Parallel Logic Control Algorithms. Information and Telecommunication Technologies in Intelligent Systems (ITTIS'06). pp. 92–94. Katania, Italy (2006)
11. Vatutin, E.I., Volobuev, S.V., Zotov, I.V.: A Comprehensive Comparative Evaluation of Methods for Getting Separations During Logic Multicontrollers Design (in Russian). Systems Identification and Control Problems (SICPRO'08). pp. 1917–1940. Institute of Control Problems of RAS, Moscow (2008)
12. Vatutin, E.I.: Parallel Logic Control Algorithm Separation Quality Analysis in the Synthesis of Logic Multicontrollers. 12th International Student Olympiad on Automatic Control (BOAC'08). pp. 95–99. IFMO, Saint-Petersburg (2008)
13. Vatutin, E.I., Volobuev, S.V., Zotov, I.V.: A Comprehensive Comparative Evaluation of Methods for Getting Separations During Logic Multicontrollers Design with Technological Constraints (in Russian). Parallel Computing and Control Problems (PACO'08). pp. 643–685. Institute of Control Problems of RAS, Moscow (2008)
14. Baranov, S.I., Zhuravina, L.N., Peschansky, V.A.: Method for Presentation of Parallel Graph-schemes of Algorithms by a Set of Consecutive Graph-schemes (in Russian). Automatics and Computing. pp. 74–81. Institute of Electronics and Computer Science, Riga (1984)
15. Vatutin, E.I.: Library of Functions for Getting Separations Using S.I. Baranov Method with Greedy Consecutive Forming of Blocks (in Russian). Certificate of official registration of the computer software № 2010612902 from 28.04.2010.
16. Vatutin, E.I., Leonov, M.E.: Using Adjacent Neighborhood in the Greedy Consecutive Forming of Blocks of Separations of Graph-schemes of Parallel Algorithms (in Russian). Proceedings of the Higher Educational Institutions. Instrument Making. Vol. 56. № 6. pp. 30–35. IFMO, Saint-Peterburg (2013)
17. Vatutin, E.I., Titov, V.S.: Library of Functions for Getting Separations with Using Adjacent Greedy Strategy and Consecutive Forming of Blocks (in Russian). Certificate of official registration of the computer software № 2013619395 from 03.10.2013.
18. Vatutin, E.I., Zotov, I.V.: Method for Getting Suboptimal Separations of Parallel Logic Control Algorithms (in Russian). Parallel Computing and Control Problems (PACO'04). pp. 884–917. Institute of Control Problems of RAS, Moscow (2004)
19. Vatutin, E.I., Zotov, I.V.: Parallel-sequential Method for Getting Separations of Parallel Logic Control Algorithms (in Russian). Certificate of official registration of the computer software № 2005613091 from 28.11.2005.

20. Vatutin, E.I., Zotov, I.V.: Identification and Breaking of Sequential Loops in the Problem of Suboptimal Separation of Parallel Logic Control Algorithms (in Russian). Proceedings of Tula State University. Computer Sciences. Information Technologies. Control Systems. Vol. 1. Issue 3. pp. 51–55. Tula State University, Tula (2004)
21. Vatutin, E.I.: Uniting Linear Ways at the Problem of Getting Suboptimal Separations of Parallel Logic Control Algorithms (in Russian). Youth and XXI century. Part 1. pp. 22–23. Kursk State Technical University, Kursk (2004)
22. Vatutin, E.I., Zotov, I.V.: Building the Matrix of Relations in the Problem of Getting Separations of Parallel Logic Control Algorithms (in Russian). Proceedings of Kursk State Technical University. № 2. pp. 85–89. Kursk State Technical University, Kursk (2004)
23. Vatutin, E.I., Zotov, I.V.: Finding of the Base Section at the Problem of Getting Separations of Parallel Algorithms (in Russian). Deposited by VINITI 24.11.2003. № 2036-B2003. Kursk State Technical University, Kursk (2003)
24. Vatutin, E.I., Zotov, I.V., Titov, V.S.: Building of the Set of Sections in the Problem of Optimal Separation of Parallel Logic Control Algorithms (in Russian). Proceedings of Tula State University. Computer Sciences. Information Technologies. Control Systems. Vol. 1. Issue 2. pp. 70–77. Tula State University, Tula (2003)
25. Vatutin, E.I., Zotov, I.V.: Building the Blocks of Separation in the Problem of Decomposition of Parallel Logic Control Algorithms (in Russian). Materials and strengthening technologies. Vol. 2. pp. 38–48. Kursk State Technical University, Kursk (2003)
26. Vatutin, E.I., Zotov, I.V.: Improving the Quality of Separations in the Problem of Synthesis of Logic Multicontrollers Using Parallel-sequential Method (in Russian). Prospects of Development of Weapon Control Systems. pp. 84–92. Bedretdinov and Ko, Moscow (2007)
27. Vatutin, E.I.: Analysis of the Effectiveness and Program Optimization of Methods for Getting Separations of Parallel Logic Control Algorithms in PAE Environment (in Russian). Proceeding of Southwest State University. Series: Control, Computer Science, Informatics. Medical Devices. № 2. Part 1. pp. 191–195. Southwest State University, Kursk (2013)
28. Vatutin, E.I., Titov V.S.: Algorithmic Optimization of Program Implementation of Parallel-sequential Method for Getting Separations of Parallel Logic Control Algorithms (in Russian). Proceedings of the Higher Educational Institutions. Instrument Making. Vol. 56. № 6. pp. 23–29. IFMO, Saint-Peterburg (2013)
29. Vatutin, E.I.: Analysis of the Bottlenecks of Program Implementation of Parallel-sequential Method for Getting Separations of Parallel Logic Control Algorithms (in Russian). Recognition – 2013. pp. 235–237. Southwest State University, Kursk (2013)
30. Anderson, D.P.: BOINC: A System for Public-Resource Computing and Storage. Proc. Fifth IEEE/ACM International Workshop on Grid Computing (GRID'04). pp. 4–10. Pittsburgh (2004)
31. Gerasim@Home: Volunteer Computing Project, http://gerasim.boinc.ru
32. Detailed Users, Hosts and Teams BOINC Statistics, http://boincstats.com

33. Zaikin, O.S., Posypkin, M.A., Semenov, A.A., Khrapov, N.P.: Experience from Organizing Volunteer Computing Projects OPTIMA@Home and SAT@Home (in Russian). Nizhny Novgorod State University Vestnik. № 5(2). pp. 338–346. Nizhny Novgorod State University, Nizhny Novgorod (2012)
34. Zaikin, O.S., Posypkin, M.A., Semenov, A.A., Khrapov, N.P.: Organizing of Volunteer Computing in the BOINC platform at Example of Projects OPTIMA@home и SAT@home. CAD/CAM/CAE Observer. № 3 (71). pp. 87–92 (2012)
35. Posypkin, M.A., Semenov, A.A. Zaikin, O.S.: Using BOINC Desktop Grid to Solve Large Scale SAT Problems. Computer Science. No. 13 (1). pp. 25–34 (2012)
36. Vatutin, E.I., Titov, V.S.: Using Distributed Volunteer Computing Systems for Analysis of Quality of Separations of Parallel Logic Control Algorithms (in Russian). Parallel Computing and Control Problems (PACO'12). pp. 37–54. Institute of Control Problems of RAS, Moscow (2012)
37. Vatutin, E.I., Titov, V.S.: Structural-parametric Optimization of Logic Control Systems Using Volunteer Computing (in Russian). Proceeding of Southwest State University. Series: Control, Computer Science, Informatics. Medical Devices. № 2. Part 1. pp. 12–17. Southwest State University, Kursk (2012)
38. Vatutin, E.I., Titov, V.S.: On the Selection of the Optimal Structure of Logical Multicontroller (in Russian). Automatics and Telemechanics, accepted for publication
39. Vatutin, E.I., Valyaev, S.Yu, Andreev, A.L., Titov, V.S.: Analysis of Probabilities of Getting Suboptimal Decisions Using Greedy Adjacent Strategy of Getting Separations (in Russian). Recognition – 2015, accepted for publication
40. Vatutin, E.I., Titov, V.S.: Comparison of Methods for Getting Separations of Parallel Logic Control Algorithms Using Two-dimension Diagrams (in Russian). Recognition – 2012. pp. 138–140. Southwest State University, Kursk (2012)
41. Vatutin, E.I., Titov, V.S.: Analysis of Areas of Qualitative Superiority of Heuristic Methods of Getting Separations During Logic Multicontrollers Design (in Russian). Proceedings of the Higher Educational Institutions. Instrument Making. Accepted for Publication
42. Dorigo, M.: Optimization, Learning and Natural Algorithms. PhD thesis. Politecnico di Milano, Milano (1992)
43. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by Simulated Annealing. Science. Vol. 220. No. 4598. pp. 671–680 (1983). DOI: 10.1126/science.220.4598.671
44. Dijkstra, E.W.: A note on two problems in connection with graphs. Numerische Mathematik. Vol. 1. pp. 269–271 (1959)
45. Vatutin, E.I., Kolyasnikov, D.V., Martynov, I.A., Titov, V.S.: Random Search Method in the Problem of Getting Separations of Parallel Logic Control Algorithms (in Russian). Multicore Processors, Parallel Programming, FPGA, Signal Processing Systems. pp. 115–125. Altay State University, Barnaul (2014)
46. Vatutin, E.I., Valyaev, S.Yu., Dremov, E.N., Martynov, I.A., Titov, V.S.: Computing Unit for Testing Combinatorial Optimization Algorihtms in the Problem of Getting Shortest Path in Graph Using Volunteer Computing (in Russian). Certifi-

cate of official registration of the computer software № 2014619797 from 22.09.2014.
47. Vatutin, E.I., Dremov, E.N., Martynov, I.A., Titov, V.S.: Weighted Random Search Method for Discrete Combinatorial Problems Solving (in Russian). Proceedings of Volgograd State Technical University. Series: Electronics, Measuring Equipment, Radiotechnics and Communication. № 10 (137). Issue 9. pp. 59–64. Volgograd State Technical University, Volgograd (2014)
48. Vatutin, E.I., Titov, V.S.: Analysis of Results of Ant Colony Optimization Method in the Problem of Getting Shortest Path in Graph with Constraints (in Russian). Proceedings of South Federal University, accepted for publication
49. Vatutin, E.I., Titov, V.S.: Analysis of Results of Limited Depth First Search Method in the Problem of Getting Shortest Path in Graph with Constraints (in Russian). Multicore Processors, Parallel Programming, FPGA, Signal Processing Systems, accepted for publication
50. Vatutin, E.I., Martynov, I.A., Titov, V.S.: Workaround of Deadlocks When Solving Discrete Optimization Problems with Constraints (in Russian). Perspective Information Technologies. pp. 313–317. Samara State Aerospace University, Samara (2014)