UDC 519.854

LISTROVOY S.V., Doctor of Technical Sciences, Full Professor,
MOTSNYI S.V., graduate student (Ukrainian State University of Railway Transport)

A heuristic approach to solving the minimum vertex cover problem using guaranteed predictions

This paper presents a heuristic approach to solving the minimum vertex cover problem with guaranteed predictions, which can be effectively implemented on the multi-core platforms because of the high degree of the instruction-level parallelism. The C++ program to compute and display the figures of the test results for each experiment was written. According to the results this approach is optimized for the very dense graphs.
***Key words**: guaranteed predictions, nonlinear equations, leaf vertices.*

Introduction

The problem of finding the minimum vertex cover for a random graph was one of the first tasks which were called NP-complete and denoted by NPC [1]. There were many attempts of developing the exact algorithms which would allow the problem be solved in polynomial time. However, both in theory and practice, it is not yet known a fast method which uses a reasonable amount of time for computing the solution. There are only approximation algorithms which are optimal up to a constant factor [2]. In other words, they return a vertex cover which has a number of vertices no more than k times bigger in comparison with the minimum cover possible (k is a constant factor of the particular approximation algorithm).

Over the last few decades this problem has been studied with great attention. It is connected with the fact that the minimum vertex cover problem is used in many important and contemporary fields of science and technology. In particular, it is widely used in a telecommunication system monitoring [3] by means of which the areas with slow performance and/or damaged parts of a network can be detected. The minimum vertex cover algorithms, which provide mechanisms and means of detection and analysis of the regions of similarity inside a DNA and relationships between the complete genome sequences [4], play an important role in the biological sequence alignment (protein, DNA, RNA etc.). Such algorithms are also crucial in resolving conflicts of the many problems of computational biology [5].

The importance of this research can be easily seen by examining the coordination of the shared resources in the heterogeneous high performance computing systems, where the choice of the effective and efficient method of solving the minimum vertex cover problem plays crucial role in providing stability in the high intensity task management environment and obtaining the most cost-efficient level of performance in the distributed systems. It is much easier to expand and manage such systems when dealing with an algorithm that has an improved time complexity.

Review of recent research and publications

In the recent years the efforts of finding an asymptotically optimal algorithm for solving the minimum vertex cover problem were done taking into account its parameterized complexity [6]. The main idea behind the parameterized complexity is that it is possible to change the structure of the input parameters to get the practical tractability. Hence, on the one hand there is a big set of the input values and on the other hand there is a wide variety of parameters which can affect the overall computational complexity of the algorithm being analyzed. This approach makes it possible to form the more flexible classification of the NP-hard problems in comparison with the classical methodology when complexity is measured in terms of the input size only.

If it is true that $P \neq NP$, there must exist many natural problems that require exponential or worse running time. However, using a parameterized algorithm allows us to solve such problems efficiently for any input set of values provided that some parameter k is fixed. In other words, if there exist some function f(k) that affects the algorithm complexity and there is a k-parameter that has a relatively small value, there is an algorithm which solves the problems in $O(f(k) \times n^{O(1)})$ time, where n is a number of the input values.

Problems that have fixed k-parameter are called parameterized problems and belonged to the complexity class FPT (fixed-parameter tractable). The vertex cover problem is said to be in this class too. Quite a long time the optimized parameterized algorithms are developed and investigated. At the present time one of the quickest known algorithms solves this problem in $O(kn \times 1.2738^k)$ time [7], where n is a number of vertices of a random graph and k is the size of the vertex cover.

The approximation algorithms are often used for solving the optimization problems. One of the most important properties of the approximation algorithm is an approximation factor. It is also called a relative performance guarantee and can be defined as:

$$p(i) \geq max\ (f(i) / OPT;\ OPT/ f(i)), \quad (1)$$

where OPT is an optimal solution for the problem instance i and f(i) is the cost of the solution of an approximation algorithm.

The most advanced method up to this day was discussed in the work [8] where it was proven that there exist an algorithm with the approximation factor equal to $(2 - \Theta(\frac{1}{\sqrt{\log n}}))$. Such algorithm is said to be an p(i)-approximation algorithm.

Among the weakest sides of the approaches to solving the minimum vertex cover problem is the lack of attention to the problem of parallelization of operations by means of which the efficiency of execution in a distributed environment could be increased. Many of the known algorithms have too high value of the fixed parameter which reduces performance of the system.

This article treats and summarizes an approach to solving the minimum vertex cover problem for the random graphs that is optimal for using in the distributed environments under high load conditions. The main purpose is to create an algorithm with improved complexity bounds in comparison with the existing methods.

An algorithm with the guaranteed predictions

The term “arbitrary undirected graph” is used here for the sake of the problem formalization. It implies an ordered pair G(V, E) where V is a set of vertices and E is a set of edges or links. An edge in the undirected graph is represented by the unordered pair (u, v) ∈ E. The edges in such a graph have no orientation.

The vertex covers of the arbitrary undirected graph are the subsets of vertices $V' \subseteq V$ such that each edge (u, v) ∈ G meets the following requirements: $u \in V', v \in V'$. The minimum vertex cover problem is to find a vertex cover of smallest possible size. The exact algorithms for solving the minimum vertex cover have the time complexity that is generally increased with the number of vertices in a graph. In this paper we focus on the effective approximation algorithm with the guaranteed predictions which uses heuristic guaranteed predictions, has improved local searching ability and gives near to optimal solution. The term "prediction" is used to refer to a set of equations by means of which it is possible to choose the most optimal direction in the algorithm pipeline.

The proposed algorithm consists of two different parts: the main procedure A that has 11 basic steps and an additional procedure B that checks a given graph for the presence of the leaf vertices (the vertices with degree one, i.e. they are the endpoints of exactly one edge).

- **Procedure A**. Basic steps for solving the minimum vertex cover problem:

Step 1. Given a graph G(V, E), an initial nonlinear equation is formed as:

$$f_z(x_i x_j) = 0, \quad (2)$$

where $x_i x_j$ are such pairs of the vertices that form the full cover of the graph.

Step 2. The equation (2) is then processed by the procedure B. If this procedure returns the second possible solution (all possible solutions are stated in the procedure B description), then the minimum vertex cover will be found and it will be represented by the full set of values $R_z^{ї}$ which will be returned by both procedures. Therefore, procedure A will be finished. However, if procedure B yields the first or the third possible result, we must go to the next step.

Step 3. Depending on the results that had been obtained on the previous step, in the equation (2) or its derivative $f_z^{|}(x_i x_j) = 0$ which contain the partial set of the vertex cover the term $S_p^* = x_l x_m$ with the maximum frequency $h_l + h_m$ (maximum degree of the graph’s vertices) is formed along with such three variables: $x_l = 0, x_m = 0$; $x_l = 1, x_m = 0$; $x_l = 0, x_m = 1$. Then move on to the next step.

Step 4. The variable z is then assigned the value of 1. The first pair of variables $x_l = 0, x_m = 0$ is substituted into the current equation that is now defined as $f_1(x_i x_j) = 0$. The variables x_l and x_m are added into the partial result $R_z^{\times}$ and the new equation is processed by the procedure B. If we get the second possible solution from the procedure B, then the minimum vertex cover will be found - it will be represented by the full set of values $R_z^{ї}$. After saving this result, move on to the next step.

Step 5. Depending on the result obtained from the procedure B on the previous step, an equation $f_1(x_i x_j) = 0$ (or $f_1^{|}(x_i x_j) = 0$) along with its partial vertex cover and all full sets of values $R_z^{ї}$ are added to the set M. Then move on to the next step.

Step 6. The variable z is assigned the value of 2. The second pair of variables $x_l = 1, x_m = 0$ is substituted into the current equation that has the form of $f_2(x_i x_j) = 0$. All terms of the equation which contain just one variable $\{x_t\}$ are set to null. After that the equation will be defined as $f_2^{\downarrow}(x_i x_j) = 0$. All variables $\{x_j\}$ adjacent to $\{x_t\}$ and the variable x_m are added into the partial solution $R_z^{\times}$. The new equation is processed by the procedure B. If we get the second possible solution from the procedure B, then the minimum vertex cover will be found and it will be represented by the full set of values $R_z^{\ddot{I}}$. After saving the result, move on to the next step.

Step 7. Depending on the result obtained from the procedure B on the previous step, an equation $f_2^{\downarrow}(x_i x_j) = 0$ (or $f_2^{\|}(x_i x_j) = 0$) along with its partial vertex cover and all sets $R_z^{\ddot{I}}$ from the previous steps are added to the set M. Then move on to the next step.

Step 8. The variable z is assigned the value of 3. The third pair of variables $x_l = 0, x_m = 1$ is substituted into the current equation that is got a new form as $f_3(x_i x_j) = 0$. All terms of the equation which contain just one variable $\{x_t\}$ are set to null. The equation will be defined now as $f_3^{\downarrow}(x_i x_j) = 0$. All variables $\{x_j\}$ adjacent to $\{x_t\}$ and the variable x_l are added into the partial solution $R_z^{\times}$. The equation is then processed by the procedure B. If we get the second possible solution from the procedure B, then the minimum vertex cover will be found - it will be represented by the full set of values $R_z^{\ddot{I}}$. Save it and move on to the next step.

Step 9. Depending on the result obtained from the procedure B on the previous step, an equation $f_3^{\downarrow}(x_i x_j) = 0$ (or $f_3^{\|}(x_i x_j) = 0$) along with its partial vertex cover and all sets $R_z^{\ddot{I}}$ are added to the set M. Then move on to the next step.

Step 10. Check if all the equations in the M have got the form of identity 0=0, if true – choose among all sets $\{R_z^{\ddot{I}}\}$ the minimum one, it will be the minimum vertex cover of the given graph. Otherwise, go to the next step.

Step 11. Among the equations $f_1(x_i x_j) = 0$, $f_2(x_i x_j) = 0$, $f_3(x_i x_j) = 0$ which haven't got the form of identity 0=0 choose the equation $f_i^{*}(x_i x_j) = 0$ with the most number of the terms. Then the equation (2) that was used on the previous step is substituted by the equation $f_i^{*}(x_i x_j) = 0$. Then move on to the step 2 and repeat all the steps until the minimum vertex cover is found.

Let's describe the additional procedure B that is often executed inside the main operations loop. It is required for proper handling of the leaf or pendant vertices of the graph. When such a vertex is found it is removed from the graph and its adjacent vertex is put into the cover. Putting the vertex into the cover implies removing the vertex and all its adjacent edges from the graph and moving to the next step of the algorithm.

- **Additional procedure B**. Leaf vertices checking:

Step 1. Check if the equation (2) has the terms $x_q x_j \in f(x_i x_j)$ with the variables $\{x_q\}$ which occur only once. If it is true, then all the variables $\{x_j\}$ which are neighbors of the $\{x_q\}$ variables are set to null and added to the partial solution $R_i^{\times}$ while the equation (2) is transformed into $f_z^{\downarrow}(x_i x_j) = 0$ with smaller number of variables. Then move on to the next step. Otherwise, the procedure B is finished.

Step 2. Check if the equation $f_z^{\downarrow}(x_i x_j) = 0$ has got the form of identity 0=0. If it is true, then the partial solution $R_z^{\times}$ is transformed into the full solution $R_z^{\ddot{I}}$, i.e. the vertex cover of the graph is defined by the variables from the $R_z^{\ddot{I}}$, therefore, the procedure is finished. Otherwise, the equation (2) is transformed into $f_z^{\downarrow}(x_i x_j) = 0$ and then we must go to the first step again.

The additional procedure B can yield such three possible results:

1. The equation (2) is not changed.
2. The equation (2) has got the form of identity 0=0 and there is the full solution $R_z^{\ddot{I}}$.
3. The equation (2) is transformed into the equation $f_z^{\downarrow}(x_i x_j) = 0$ with smaller number of the variables and some partial solution $R_z^{\times}$.

Let's look at an example to see how we apply our heuristic algorithm for solving the minimum vertex cover problem. Table 1 contains all connections of random graph vertices.

Table 1

Connections of the vertices of the graph

Vertex 1: 3 6 7 8 12	Vertex 7: 1 3 5 6 8
Vertex 2: 5 6 9 11 12	Vertex 8: 1 3 4 5 6 7 9
Vertex 3: 1 4 5 7 8 9 10 11	Vertex 9: 2 3 4 6 8 10 12
Vertex 4: 3 6 8 9 10	Vertex 10: 3 4 9 11
Vertex 5: 2 3 7 8 11 12	Vertex 11: 2 3 5 10
Vertex 6: 1 2 4 7 8 9	Vertex 12: 1 2 5 9

Table 2 contains the list of all the vertex covers and independent sets (cover - set). An independent set of a graph is such a set of vertices no two of which are adjacent.

Table 2

Vertex covers and independent sets of a given graph

1) 1 2 3 5 6 8 9 10 (8) - 4 7 11 12 (4)	11) 1 3 4 5 6 8 9 11 12 (9) - 2 7 10 (3)
2) 1 2 3 4 5 6 7 9 10 (9) - 8 11 12 (3)	12) 1 3 5 6 8 9 10 11 12 (9) - 2 4 7 (3)
3) 1 2 3 4 5 6 7 9 11 (9) - 8 10 12 (3)	13) 2 3 4 5 6 7 8 10 12 (9) - 1 9 11 (3)
4) 1 2 3 4 5 6 8 9 11 (9) - 7 10 12 (3)	14) 2 3 4 6 7 8 9 11 12 (9) - 1 5 10 (3)
5) 1 2 3 4 5 6 8 10 12 (9) - 7 9 11 (3)	15) 2 3 4 6 7 8 10 11 12 (9) - 1 5 9 (3)
6) 1 2 3 4 5 7 8 9 10 (9) - 6 11 12 (3)	16) 2 3 5 6 7 8 9 10 12 (9) - 1 4 11 (3)
7) 1 2 3 4 5 7 8 9 11 (9) - 6 10 12 (3)	17) 2 3 6 7 8 9 10 11 12 (9) - 1 4 5 (3)
8) 1 2 3 4 7 8 9 11 12 (9) - 5 6 10 (3)	18) 3 4 5 6 7 8 9 11 12 (9) - 1 2 10 (3)
9) 1 2 4 5 7 8 9 10 11 (9) - 3 6 12 (3)	19) 3 5 6 7 8 9 10 11 12 (9) - 1 2 4 (3)
10) 1 3 4 5 6 7 9 11 12 (9) - 2 8 10 (3)	20) 1 4 5 6 7 8 9 10 11 12 (10) - 2 3 (2)

The equation of the given graph is defined as:

$$X_1X_3+X_1X_6+X_1X_7+X_1X_8+X_1X_{12}+X_2X_5+X_2X_6+X_2X_9+ +X_2X_{11}+X_3X_4+X_3X_5+X_3X_7+X_3X_8+X_3X_9+X_3X_{10}+ +X_3X_{11}+X_4X_6+X_4X_8+X_4X_9+X_4X_{10}+X_5X_7+X_5X_8+ +X_5X_{11}+X_5X_{12}+X_6X_7+X_6X_8+X_6X_9+X_7X_8+X_8X_9+ +X_9X_{10}+X_9X_{12}+X_{10}X_{11}=0. \quad (3)$$

Table 3 contains the frequencies of the variables presence in the terms of the equation (3).

Table 3

The frequencies for each of the variables of the equation (3)

X_i	1	2	3	4	5	6	7	8	9	10	11	12
h_i^x	5	5	8	5	6	6	5	7	7	4	4	4

Let's choose the term with the maximum frequency of the appropriate variable in the equation (3). In this example X_3X_8 is the best matching term with the frequency of 8+7=15. Taking into account this term the equation (3) is transformed into the system of three equations each of which contains the following values of the variables: (X_3 =0, X_8 =0); (X_3 =0, X_8 =1); (X_3 =1, X_8 =0).

The first equation with variables X_3 =0 and X_8 =0 is defined as:

$$X_1X_6+X_1X_7+X_1X_{12}+X_2X_5+X_2X_6+X_2X_9+X_2X_{11}+X_2X_{12}+ X_4X_6+X_4X_9+X_4X_{10}+X_5X_7+X_5X_{11}+ +X_5X_{12}+X_6X_7+X_6X_9+X_9X_{10}+X_9X_{12}+X_{10}X_{11}=0. \quad (4)$$

The variables X_3 and X_8 are included into the partial cover.

The second equation is formed taking into account that X_3 =0, X_8 =1. As X_8 =1 it follows that X_1 =0, X_4 =0, X_5 =0, X_6 =0, X_7 =0, X_9 =0 and the equation (3) is turned into this form:

$$X_2X_{11}+ X_2X_{12} + X_{10}X_{11}=0. \quad (5)$$

The partial cover now contains X_3 and the variables X_1, X_4, X_5, X_6, X_7, X_9.

In the same way we form the third equation with X_3 =1and X_8 =0. Provided that X_3 =1 we set the following variables to null: X_1 =0, X_4 =0, X_5 =0, X_6 =0, X_7 =0, X_9 =0, X_{10} =0. Hence, the third equation is defined as:

$$X_2X_6+ X_2X_{12} =0. \quad (6)$$

At this stage the partial cover contains X_8 and the X_1, X_4, X_5, X_6, X_7, X_9, X_{10}. Since all the terms of the equations (5) and (6) are included in the equation (4) they are excluded from the further analysis.

Table 4 contains the frequencies of the variables presence in the terms of the equation (4).

Table 4

The frequencies for each of the variables of the equation (4)

X_i	1	2	4	5	6	7	9	10	11	12
h_i^x	3	5	3	4	5	3	4	3	3	4

Again, we choose the term with the maximum frequency of the appropriate variable, now it is located in the equation (4). It's X_2X_6 with the total frequency of 5+5=10. Taking into account this term the equation (4) is transformed into the system of three equations each of which contains the following pairs of the variables: (X_2 =0, X_6 =0); (X_2 =0, X_6 =1); (X_2 =1, X_6 =0).

The first equation with the pair X_2 =0, X_6 =0 is defined as:

$$X_1X_7+X_1X_{12}+X_4X_9+X_4X_{10}+X_5X_7+X_5X_{11}+X_5X_{12}+ +X_9X_{10}+X_9X_{12}+X_{10}X_{11}=0. \quad (7)$$

The partial cover now contains X_2, X_6 and the variables X_3, X_8 from the previous steps. The second equation is formed taking into account that X_2 =0, X_6 =1. As X_6 =1 it follows that X_1 =0, X_4 =0, X_7 =0 X_7 =0, X_9 =0 and the equation (4) is turned into this form:

$$X_5X_7+X_5X_{11}+X_5X_{12}+ X_{10}X_{11}=0. \quad (8)$$

The variable X_2 and the variables X_1, X_4, X_7, X_9 are included into the partial solution.

In the same way we form the third equation with the pair X_2 =1, X_6 =0. Provided that X_2 =1 we set the following variables to null: X_5 =0, X_9 =0, X_{11} =0, X_{12} =0. Hence, the third equation is defined as:

$$X_1X_7+X_1X_{12}+X_2X_5+X_4X_{10}+X_5X_7=0. \quad (9)$$

The partial cover now contains X_6 and the variables X_5, X_9, X_{11}, X_{12}. Since all the terms of the equations (8) and (9) are included in the equation (7) they are excluded from the further analysis.

Table 5 contains the frequencies of the variables presence in the terms of the equation (7).

Table 5

The frequencies for each of the variables of the equation (7)

X_i	1	4	5	7	9	10	11	12
h_i^x	2	2	3	2	3	3	2	3

The term with the maximum frequency of the appropriate variable is chosen in the equation (7) as in the previous steps, now it is X_9X_{10} with the total frequency of 3+3=6. Taking into account this term the equation (7) is transformed into the system of three equations each of which contains the following pairs of the variables: (X_9 =0, X_{10} =0); (X_9 =0, X_{10} =1); (X_9 =1, X_{10} =0).

The first equation with the pair X_9 =0, X_{10} =0 is defined as:

$$X_1X_7+X_1X_{12}+X_5X_7+X_5X_{11}+X_5X_{12}=0. \quad (10)$$

The second equation is formed taking into account that X_9 =0, X_{10} =1. As X_{10}=1 it follows that X_4 =0, X_9 =0 X_{11}=0 and the equation (7) is turned into this form:

$$X_1X_7+X_1X_{12}+X_5X_7+X_5X_{12}=0. \quad (11)$$

The third equation is formed with the pair X_9 =1, X_{10} =0. As X_9 =1 it follows that X_4 =0, X_9 =0, X_{12} =0 and the equation (7) is turned into this form:

$$X_1X_7+X_5X_7+X_5X_{11}=0. \quad (12)$$

Since all the terms of the equations (11) and (12) are included in the equation (10) they are excluded from the further analysis.

The partial cover now contains the variables X_3, X_8, X_2, X_6, X_9, X_{10}.

Let's find again the frequencies of the variables presence in the terms of the equation (10). Table 6 depicts all the required variables along with their frequencies. The variable X_{11} has degree one. Hence, it's removed as a leaf vertex.

Table 6

The frequencies for each of the variables of the equation (10)

X_i	1	5	7	11	12
h_i^x	2	3	2	1	2

After removing X_{11} from the cover X_5 is set to null and included into the partial solution. The equation (10) is defined now as:

$$X_1X_7+X_1X_{12}=0. \quad (13)$$

Table 7 contains the frequencies of the variables presence in the terms of the equation (13).

Table 7

The frequencies for each of the variables of the equation (13)

X_i	1	7	12
h_i^x	2	1	1

X_7 is a leaf vertex that must be removed from the cover. Its adjacent vertex X_1 is put into the solution. The equation (13) has got the form of identity 0=0. Therefore, the minimum vertex cover of the given graph consists of the variables X_3, X_8, X_2, X_6, X_9, X_{10}, X_5, X_1.

Experiment results

The C++ program was written to verify validity of the algorithm. It makes it possible to randomly generate the graph instances with a different number of the vertices and a variable degree. The results of our algorithm with guaranteed predictions were compared with that of several other algorithms for the minimum vertex cover on the random graphs (algorithm based on greedy-degree heuristic method and algorithm based on greedy-edge).

We performed 50 different tests. According to our analysis, the algorithm with guaranteed predictions is much more efficient in comparison with others. If the value of average degree gradually increases, other algorithms will have a great disadvantage in many aspects.

Conclusions

A large number of science and technology problems are proved to be NP-hard problems. The main idea behind solving NP-hard problem is to find approximation solution. This paper considers an effective approximation algorithm with guaranteed predictions which, according to experiment results, has an improved approximation degree.

References

1. Richard M. Karp. Reducibility among combinatorial problems (In Complexity of computer computations, IBM Thomas J. Watson Res. Center, Yorktown Heights, N.Y., 1972), pages 85–103, Plenum, 1972.
2. Burkhard Monien and Ewald Speckenmeyer. Ramsey numbers and an approximation algorithm for the vertex cover problem. Acta Inform., 22(1):115–123, 1985.
3. Yong Zhang, Qi Ge, Rudolf Fleisher, Tao Jiang, and Hong Zhu. Approximating the minimum weight weak vertex cover. Theoretical Computer Science, 363(1):99–105, 2006.5. - Volume 7, Issue 4 (Nov., 1992), 404-416.
4. Chantal Roth-Korostensky. Algorithms for building multiple sequence alignments and evolutionary trees. Doctoral thesis, ETH Zurich, Institute of Scientific Computing, 2000.
5. Ulrike Stege. Resolving conflicts in problems from computational biology. Doctoral thesis, ETH Zurich, Institute of Scientific Computing, 2000.
6. Michael R. Fellows. Parameterized complexity: the main ideas and some research frontiers. In Algorithms and computation (Christchurch, Lecture Notes), pages 291–307. Springer, Berlin, 2001.
7. Jianer Chen, Iyad A. Kanj, and Ge Xia. Simplicity is beauty: Improved upper bounds for vertex cover. Technical Report 05-008, Texas A&M University, Utrecht, the Netherlands, April 2005.
8. George Karakostas. A better approximation ratio for the vertex cover problem. In ICALP 2005, volume 3580 of LNCS, pages 1043–1050, Lisboa, Portugal, June 2005. Springer-Verlag Berlin Heidelberg.

Листровой С.В., Моцный С.В. Эвристический подход к решению задачи о наименьшем покрытии с использованием гарантированного прогнозирования. В данной статье описывается эвристический подход к решению задачи о наименьшем покрытии с использованием гарантированного прогнозирования. Благодаря высокой степени распараллеливания операций появляется возможность его эффективной реализации в системах с большим количеством вычислительных ядер. Была написана программа на языке программирования C++ для проведения экспериментального исследования. Согласно результатам, данный подход наиболее оптимизирован для графов с высокой плотностью.

Ключевые слова: гарантированное прогнозирование, нелинейные уравнения, висячие вершины.

Лістровий С.В., Моцний С.В. Евристичний підхід до вирішення задачі про найменше покриття з використанням гарантованого прогнозування. У даній статті описується евристичний підхід до вирішення задачі про найменше покриття з використанням гарантованого прогнозування. Завдяки високому ступеню розпаралелювання операцій з'являється можливість його ефективної реалізації в системах з великою кількістю обчислювальних ядер. Була написана програма на мові програмування C++ для проведення експериментального дослідження. Згідно з результатами, даний підхід найбільш оптимізований для графів з високою щільністю.

Ключові слова: гарантоване прогнозування, нелінійні рівняння, висячі вершини.

Рецензент д.т.н., професор Мойсеенко В.И. (УкрГУЖТ)

Поступила 27.04.2015г.

***Лістровий С.В.**, д.т.н., професор, професор кафедри СКС, Український державний університет залізничного транспорту, Харків, Україна, om1@ yandex.ru*

***Моцний С.В.**, аспірант, Український державний університет залізничного транспорту, Харків, Україна, stanislav.motsnyi@gmail.com.*