Scheme Partitioning by Means of Evolutional Procedures Using Symbolic Solution Representation

Oleg B. Lebedev(✉), Svetlana V. Kirilchik, and Eugeniy Y. Kosenko

Southern Federal University, Rostov-on-Don, Russia
{lebedev.ob,kirilchik}@mail.ru, eykosenko@gmail.com

Abstract. The paper provides a methodology of symbolic representation for solving the partitioning problem. The approach is based on adjacency matrix of a graph, adaptive mechanisms for adjacency matrix modification. Also the structure of adjacency matrix evolutionary modification for solving the problem of finding a partition is considered.

Keywords: Partitioning · Graph · Hyper-graph · Evolutionary procedures · Adaptation · Topology · Matrix · Solution symbolic representation

1 Introduction

One of the most relevant integer programming tasks is partitioning, considered in combinatorial part of the graph theory. Modern VLSI contains tens of millions of transistors, so due to the limited possibilities of the computing resources (memory, speed) the entire scheme topology can not be designed. Normal scheme partitioning is implemented by grouping components into the blocks. The partitioning results in variety of blocks and interconnections between the blocks [1].

A hierarchical partitioning structure is applied to large schemes. By now most of the developed partitioning algorithms use graph or hypergraph as a scheme model. Graph partitioning procedure is part of a large number of algorithms solving various problems. Often, this procedure is implemented in iterative structures [2]. This requires high quality and time standards to the problem of finding the maximal matching.

Existing to date a greater number of partitioning algorithms provide acceptable results obtained when solving the problems of low and medium complexity [3]. The needs for the problems of large and very large dimension appear to be the motivation to research and develop new effective algorithms. Analysis of the literature shows that the most successful in these circumstances are the methods based on simulation of evolutionary processes [4].

2 Main Provisions

The partitioning of the hypergraph with the weighted vertices and edges is formulated the following way [5].

R. Silhavy et al. (eds.), *Artificial Intelligence Trends in Intelligent Systems*,
Advances in Intelligent Systems and Computing 573, DOI 10.1007/978-3-319-57261-1_10

Let $H = (X, E)$ be given a *hypergraph*, where $X = \{x_i \mid i = 1, 2, ..., n\}$ is the set of vertices and $E = \{e_j \mid e_j \subset X, j = 1, 2, ..., m\}$ is a set of edges (each edge links a subset of vertices). Let $\Phi = \{\varphi_i \mid i = 1, 2, ..., n\}$ be given a set of vertice weights and $\Psi = \{\psi_i \mid i = 1, 2, ..., n\}$ be given a set of edge weights. It is necessary to form K - nodes, that is the set X divided into K non-empty and disjoint subsets X_v, X_v, $X = \cup X_v$, $(\forall i,j)\ [X_i \cap X_j = \varnothing]$, $X_v \neq \varnothing$.

Restrictions are applied to the formed nodes. Vector $P = \{p_v \mid v = 1, 2, ..., k\}$ defines maximal total weight of vertices assigned to node v, and vector $N = \{n_v \mid v = 1, 2, ..., k\}$ designates maximal number of vertices assigned to node v.

The capacity restrictions are:

$$\sum\nolimits_{i \in I} \varphi_i \le p_v, I = \{i | x_i \in X_v\}, v = 1, 2, \ldots, k \tag{1}$$

$$|X_v| \le n_v, v = 1, 2, \ldots, k \tag{2}$$

Equation (1) is a restriction on the maximal weight of the node, and the Eq. (2) - on the maximal number of vertices in the node [6].

Sometimes the number of outputs γ_{max} for the nodes is given. The restriction takes the form of:

$$\begin{gathered} \gamma_v \le \gamma_{max}, v = 1, 2, \ldots, k \\ \gamma_v = |E_v|, E_v = \{e_j | (e_j \cap X_v \neq \emptyset) \& (e_j \cap X_v \neq e_j)\} \end{gathered} \tag{3}$$

E_v is the set of edges connecting set of vertices X_v with the vertices of the rest nodes.

The main criterion is F_1 that denotes the total cost of edges in the cut.

$$F_1 = \sum\nolimits_{j=J} \varphi_j, J = \{j | e_j \in \mathrm{C}\} \tag{4}$$

$C = \{e_j \mid (\forall v)\ [e_j \cap X_v \neq e_j]\}$ – the set of edges in the cut.

Second frequently used criterion is F_2 - the total number of outputs [7].

$$F_2 = \sum\nolimits_{v=1}^{v} \gamma_v$$

The criterion F can be also used, it is an additive convolution of criteria F_1 and F_2.

$$F = k_1 \cdot F_1 + k_2 \cdot F_2$$

Here is an example. Let $G = (X, U)$ be given a graph, shown in Fig. 1.

The task is to form three nodes. Considered legal number of vertices assigned to nodes is $n_1 = 3$, $n_2 = 4$, $n_3 = 3$. Initial partitioning $X_1 = \{x_1, x_2, x_3\}$, $X_2 = \{x_4, x_{10}, x_7, x_6\}$, $X_3 = \{x_8, x_5, x_9,\}$ is given [8]. The number of connections between nodes set to 4. The state of object to be optimized is estimated by vector $S = \{s_i \mid i = 1, 2, ..., n\}$. The value s_i appears to be the serial number of a vertice x_i. Elements s_i that satisfy $1 \le i \le n_i$ correspond to the first node X_1. Elements s_i that satisfy $n_1 + 1 \le i \le n_1 + n_2$

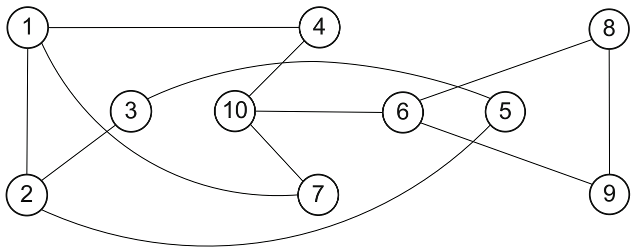

Fig. 1. Initial partitioning of graph G

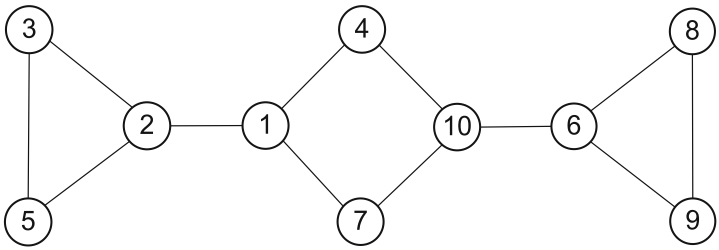

Fig. 2. Optimal partitioning of graph G

correspond to the second node X_2 and so on. In our case, $S_н = \{x_1, x_{2,} x_{3,}, x_4, x_{10,} x_7, x_6, x_8, x_{5,} x_9\}$ [9].

Optimal partitioning of graph G, described by the vector $S_o = \{x_2, x_{3,} x_{5,}, x_1, x_{4,} x_7, x_{10}, x_6, x_{8,} x_9\}$ is shown in Fig. 2

Each decision (each vector S) corresponds to adjacency matrix R which rows and columns are labeled with nodes of graph G in the same order as in corresponding vertices are located in S. Adjacency matrixes for vectors $S_н$ and S_o are shown in Figs. 3 and 4. Since the adjacency matrix is symmetrical about the main diagonal, in the figures the top part of the matrix is not filled in and is not considered [10]. Thus, the rows and columns from 1 to n_1 are designated by serial numbers of vertices belonging to the first node, and the rows and columns from $((n_1 + n_2 +\ldots + n_i) + 1)$ to $((n_1 + n_2 + \ldots + n_i) + n_{i+1}$ + ni + 1) - belonging to (n_{i+1}) node [11].

Let the columns and rows of the matrix with the numbers from l to $l + m$ are designated by elements belonging to the one X_i node. Now we'll consider area Q_i of matrix R formed by the intersection of columns and rows with the numbers from l to $(l + m)$ symmetrically about the main diagonal [12]. Elements of the range Q_i of matrix R reflect the connections between the corresponding vertices of the node X_i. The number h_i of nonzero elements of the area Q_i of matrix R equals the number of internal connections between the vertices of node X_i.

The number F of non-zero elements of matrix R that do not belong to any range Q_i equals the number of external connections between the nodes X_i. Figures 3 and 4 show areas Q_1, Q_2, Q_3 formed respectively at the intersection of 1–3, 4–7 and 8–10 rows and columns of the adjacency matrix. Let *NQ* be the elements set of matrix *R* not belonging to any area Q_i.

Suppose that we call *NQ* external relations range. For a matrix shown in Fig. 3 $Q_1 = 2$, $Q_2 = 3$, $Q_3 = 1$, $F = 6$, for matrix in Fig. 4 $Q_1 = 3$, $Q_2 = 4$, $Q_3 = 3$, $F = 2$. The graph partitioning into nodes is made for the purpose of minimization of the number of connections between the nodes [13]. The problem is form an area of external

	1	2	3	4	10	7	6	8	9	5
1	⊗									
2	1	⊗								
3		1	⊗							
4	1			⊗						
10				1	⊗					
7	1				1	⊗				
6					1		⊗			
8							1	⊗		
9							1	1	⊗	
5		1	1							⊗

Fig. 3. Initial state of adjacency matrix

	2	3	5	1	4	7	10	6	8	9
2	⊗									
3	1	⊗								
5	1	1	⊗							
1	1			⊗						
4				1	⊗					
7				1		⊗				
10					1	1	⊗			
6							1	⊗		
8								1	⊗	
9								1	1	⊗

Fig. 4. Final state of adjacency matrix

connections *NQ* with a minimal value *F* by means of rearranging rows and columns in the adjacency matrix of the graph [15].

3 Evolutionary Mechanisms to Form a Minimal Area of External Relations

Area of external relations *NQ* with the minimal value F in matrix R is formed within its evolutionary modifications. The evolutionary modification of the matrix *R* is carried out by means of selective group permutations of adjacent rows and columns. It provides a directional consistent movement of elements of the matrix *R* with nonzero values from the area of external relations *NQ* to areas *Qi*. Adaptive process consists of repeated steps, each of them is a transition from one solution (state of matrix *R*) to the best one [17].

At each step pairs of adjacent rows and columns $(i,\ i+1)$ are analyzed. The analysis is carried out in two strokes. In the first stroke all pairs $(i,\ i+1)$ where the first element i is an odd number are analyzed. In the second stroke all pairs where the first element i is even are analyzed.

For example, let n = 10, then in the first stroke pairs of rows and columns {(1,2), (3,4), (5,6), (7,8), (9,10)} are considered. In the second stroke pairs {(2,3), (4,5), (6,7), (8,9)} are considered.

Pairs of rows and columns are analyzed independently. According to the analysis the decision to interchange adjacent pair of rows and columns is made.

Local permutations goal is to move the non-zero elements of the matrix from bottom to up and from the right to the left. The global objective is to form an area of external connections *NQ* with the minimal value *F*, that is a partitioning of graph *G* with the minimal number of connections between the nodes [17].

The pair of rows and columns $(l,\ l+1)$ in $n*n$ matrix $R = \|r_{ij}\|$ is selected for analysis. And let the rows and columns intersect the area Q_k, formed at the intersection of columns and rows with numbers from v to w. According to the following formulas parameters S_1, S_2, S_3, S_4 are calculated.

$$\sum_{j=1}^{v-1} r_{lj} = S_1; \quad \sum_{j=1}^{v-1} r_{l+1,j} = S_2.$$

$$\sum_{i=w+1}^{n} r_{il} = S_3; \quad \cdot \quad \sum_{i=w+1}^{n} r_{i,l+1} = S_4.$$

S_1 and S_2 are the sums of *l-th* and $l+1$*-th* raws in matrix *R*, the elements do not belong to the area Q_k.

S_3 and S_4 are the sums of *l-th* and $l+1$*-th* columns in matrix *R*, the elements do not belong to the area Q_k.

If a pair of raws $(l,\ l+1)$ in matrix *R* belong to two adjacent areas Q_k and Q_{k+1}, the parameters S_1, S_2, S_3, S_4 are calculated as follows:

$$\sum_{j=1}^{l-1} r_{lj} = S_1; \quad \sum_{j=1}^{l-1} r_{l+1,j} = S_2$$

$$\sum_{i=l+2}^{n} r_{il} = s_3; \quad \sum_{i=l+2}^{n} r_{i,l+1} = S_4$$

In this case, the sums S_1 and $S_{2,}$ S_3 and S_4 include all the elements of *l-th* and *(l + 1)-th* rows, *l-th* and *(l +1)th* columns of the triangular matrix *R*, except element $r_{l+1,l}$.

The main idea of the analysis is to determine the truth value of the 2 following conditions.

1. $(S_2 - S_1) + (S_3 - S_4) > 0.$
2. $(S_2 - S_1) + (S_3 - S_4) = 0.$

The answer is a qualified "yes", that is - to rearrange, answer is generated if the condition 1 holds. In the case the condition 2 is satisfied the answer "yes" is generated with the a priori probability *P*. Answer "No" is generated in all the other cases [13].

Adaptive search procedure continues until there are pairs, for which conditions 1 and 2 hold. As a result, the area of external connections *NQ* with the minimal value *F* will be formed and the graph partitioning with the minimal number of connections between nodes will be defined [18].

Example. Figure 1 shows graph *G*. Initial partitioning is given. The adjacency matrix of a graph *G* is shown in Fig. 3.

At the first step and first stroke pairs {(1,2), (3,4), (5,6), (7,8), (9,10)} are considered, at the second stroke - pairs {(2 3), (4,5), (6,7), (8,9)}. Pair of rows and columns is rearranged, if one of the above two conditions is fulfilled. In the initial matrix *R* columns and rows are labeled with serial numbers of vertices of the graph G. Swap of adjacent pair of rows and columns *(i, i + 1)* also leads to a rearrangement of its labels. Further rows and columns will be identified by its labels [19].

Step 1, stroke 1: (1.2) - yes; (3.4) - no; (10.7), yes; (6.8) - no; (9.5), yes. Thus, the rearrangement is carried out on pairs (1,2), (10,7), (9.5) at the 1st stroke of the 1st step. The modified matrix *R* shown in Fig. 5.

Step 1, stroke 2: (1.3) - yes; (8.5) - yes. The modified matrix is shown in Fig. 6.
Step 2, stroke 1: (6.5) - yes. The modified matrix *R* is shown in Fig. 7.
Step 2, stroke 2: (10.5) - yes. The modified matrix *R* is shown in Fig. 8.
Step 3, stroke 1: (7.5) - yes. The modified matrix *R* is shown in Fig. 9.
Step 3, stroke 2: (4.5) - yes. The modified matrix *R* is shown in Fig. 10.
Step 4, stroke 1: (1.5) - yes. The modified matrix *R* is shown in Fig. 4.
Step 4, stroke 2: no permitted permutations.

After the four steps in the modified matrix the area of external relations *NQ* with a minimal value *F = 2* is formed [20, 21].

To overcome the local barrier approaches based on a combination of different types of evolution are used.

The first approach uses the idea of annealing simulation method. In case the analysis shows that conditions 1,2,3 are not met, the rearrangement is performed with

	2	1	3	4	7	10	6	8	5	9
2	⊗									
1	1	⊗								
3	1		⊗							
4		1		⊗						
7		1			⊗					
10				1	1	⊗				
6						1	⊗			
8							1	⊗		
5	1		1						⊗	
9							1	1		⊗

Fig. 5. Step 1, stroke 1

	2	3	1	4	7	10	6	5	8	9
2	⊗									
3	1	⊗								
1	1		⊗							
4			1	⊗						
7			1		⊗					
10				1	1	⊗				
6						1	⊗			
5	1	1						⊗		
8							1		⊗	
9							1		1	⊗

Fig. 6. Step 1, stroke 2

the probability $P = exp(-\Delta F/kT)$, where T denotes the temperature, ΔF - the difference in the sums of the analyzed raws.

	2	3	1	4	7	10	5	6	8	9
2	⊗									
3	1	⊗								
1	1		⊗							
4			1	⊗						
7			1		⊗					
10				1	1	⊗				
5	1	1					⊗			
6						1		⊗		
8								1	⊗	
9								1	1	⊗

Fig. 7. Step 2, stroke 1

	2	3	1	4	7	5	10	6	8	9
2	⊗									
3	1	⊗								
1	1		⊗							
4			1	⊗						
7			1		⊗					
5	1	1				⊗				
10				1	1		⊗			
6							1	⊗		
8								1	⊗	
9								1	1	⊗

Fig. 8. Step 2, stroke 2

The second approach uses one of the genetic search structures. The population appears to be the set of adjacent matrixes (encoded as chromosomes). Decoding, that is obtaining solutions, is carried out by the described above adaptive process.

Time complexity of adaptive procedures in one step is *O(n)*. Comparison with the known algorithms has shown that the betters results are obtained in less time.

	2	3	1	4	5	7	10	6	8	9
2	⊗									
3	1	⊗								
1	1		⊗							
4			1	⊗						
5	1	1			⊗					
7			1			⊗				
10				1		1	⊗			
6							1	⊗		
8								1	⊗	
9								1	1	⊗

Fig. 9. Step 3, stroke 1

	2	3	1	5	4	7	10	6	8	9
2	⊗									
3	1	⊗								
1	1		⊗							
5	1	1		⊗						
4			1		⊗					
7			1			⊗				
10					1	1	⊗			
6							1	⊗		
8								1	⊗	
9								1	1	⊗

Fig. 10. Step 3, stroke 2

4 Conclusion

Due to the fact that the partitioning problem considered in the scope of combinatorial graph theory, is one of the most relevant integer programming tasks, an algorithm was developed to solve this problem.

It is known that a hierarchical partitioning structure is applied to large schemes.

To set partitioning problem hypergraph was used as a scheme model.

Graph partitioning procedure is part of a large number of algorithms to solve various problems. This procedure is quite often implemented in iterative structures as the high demands are made to the quality and the time to solve the problem of finding the maximal matching.

Paper shows as the state of the optimization object is estimated by vector S, which is associated with the adjacency matrix *R*.

The developed procedure has allowed to simplify the problem of graph partitioning into nodes with the minimal number of interconnections. The goal is to form the external connections area with the minimal criterion value F by means of rows and columns permutations in the adjacency matrix.

The evolutionary modification of the formation process of external connections with the minimal criterion value in the matrix *R*. The modification point is to permutate adjacent rows and columns of the matrix *R*, which provides a consistent directional movement of the elements within the matrix.

Implemented adaptive process consists of repeated steps, each of that is a transition from one solution to the better one and is performed in two strokes.

Experimental studies have been carried out to prove the high efficiency of the proposed evolutionary procedures.

Acknowledgements. This research is supported by grants of the Russian Foundation for Basic Research of the Russian Federation, the project № 15-01-05297.

References

1. Lebedev, B.K.: Adaptation in CAD. In: Monograph. TRTU Publishing House, Taganrog (1999)
2. Lebedev, B.K.: Methods of search adaptation for VLSI CAD. In: Monograph. TRTU Publishing House, Taganrog (2000)
3. Lebedev, B.K.: Ant partitioning algorithms using non-canonical task representation. Reporter of the Rostov State University of Railway Connections, no. 3(63), pp. 42–47. RSURC Publishing House (2016)
4. Lebedev, B.K., Lebedeva, E.M.: Partitioning into classes by means of alternative collective adaptation. Izvestiya of SFU. Eng. Sci. **7**(180), 89–101 (2016). SFU publishing house, Rostov-on-Don
5. Lebedev B.K., Lebedev V.B. Adaptive bee colony behavior model-based program for solving graph problems. Certificate of state registration of the computer program no. 2014663152 issued on 02 April 2015
6. Lebedev, B.K., Kovalenko, M.S.: Solution for partitioning problem based on search engine adaptation. In: Proceedings of the Congress on Intelligent Systems and Information Technology, vol. 3, pp. 72–83. SFU Publishing House, Taganrog (2015). Scientific edition in 3 volumes
7. Lebedev, B.K.: Nature-inspired VLSI design methods. In: Monograph. LAP LAMBERT Academic Publishing GmbH & Co. KG Heinrich – Bocking- Str. 6–8, 66121 Saarbrucken, Deutschland (2014)

8. Lebedeva, E.M.: Scheme partitioning based on ant colony method. Electron. J. Inform. Comput. Sci. Eng. Educ. **2**(13), 20–26 (2013). TTI SFU Publishing House, Taganrog
9. Models of adaptive ant colony behavior in the design tasks. In: Monograph. TTI SFU Publishing House, Taganrog (2013)
10. Gladkov, L.A., Kureichik, V.V., Kureichik, V.M., Lebedev, B.K., Lebedev, V.B., Nuzhnov, E.V., Rodzin, S.I.: Elements of the evolutionary optimization and decision-making theory based on nature-inspired methods. In: Monograph. SFU Publishing House, Taganrog (2013)
11. Zhilenkov, M.: EVA scheme partitioning by mean of ant colony method. In: Proceedings of the 59th Student Conference, pp. 17–18. TTI SFU Publishing House, Taganrog (2012)
12. Kureichik, V.M., Lebedev, B.K.: Hybrid partitioning algorithm based on natural decision-making mechanisms. In: Artificial Intelligence and Decision-Making, pp. 3–15. Publishing House of the Institute of System Analysis, RAS, Moscow (2012)
13. Kureichik, V.M., Lebedev, B.K.: Adaptation applied to topology design problems. In: Monograph. LAP LAMBERT Academic Publishing Gmbh & Co. KG, Saarbrucken, Germany (2012)
14. Gladkov, L.A., Kureichik, V.V., Kureichik, V.M., Lebedev, B.K., Lebedev, V.B.: New approaches and technologies to build decision-making algorithms for optimization problems. In: Collective Monograph. TTI SFU Publishing House, Taganrog (2011)
15. Lebedev, B.K.: Intelligent VLSI topology synthesis procedures. TTI SFU Publishing House, Taganrog (2003)
16. Dorigo, M., Stützle, T.: Ant colony optimization: overview and recent advances. In: Gendreau, M., Potvin, Y. (eds.) Handbook of Metaheuristics. International Series in Operations Research and Management Science, vol. 146, pp. 227–263. Springer, New York (2010). 2nd edn.
17. Dorigo, M., Maniezzo, V., Colorni, A.: The ant system: optimization by a colony of cooperating agents. IEEE Trans. Syst. Man Cybern.-Part B **26**(1), 29–41 (1996)
18. Cong, J., Wu, C.: Global clustering-based performance-driven circuit partitioning. In: Proceedings of ISPD (2002)
19. Engelbrecht, A.P.: Fundamentals of Computational Swarm Intelligence. Wiley, Chichester (2005)
20. Mazumder, P., Rudnick, E.: Genetic Algorithm For VLSI Design, Layout & Test Automation. Pearson Education, Bengaluru (2003)
21. Poli, R.: Analysis of the publications on the applications of particle swarm optimisation. J. Artif. Evol. Appl. 10 p. Article ID 685175 (2008)