ISSN 1392 – 124X INFORMATION TECHNOLOGY AND CONTROL, 2011, Vol. 40, No. 4

A HYBRID GENETIC ALGORITHM FOR PARTITIONING OF DATA MODEL IN DISTRIBUTION MANAGEMENT SYSTEMS

Darko Capko, Aleksandar Erdeljan, Srdjan Vukmirovic, Imre Lendak

Faculty of Technical Sciences , Trg Dositeja Obradovica 6, Novi Sad, Serbia
email: dcapko@uns.ac.rs, ftn_erdeljan@uns.ac.rs, srdjanvu@uns.c.rs, lendak@uns.ac.rs

crossref http://dx.doi.org/10.5755/j01.itc.40.4.981

Abstract. In this paper, we propose a Hybrid Genetic Algorithm for data model partitioning of power distribution network. Analytical functions are the core of Distribution Management Systems (DMSs). Efficient calculation of the functions is of the utmost importance for the DMS users; the necessary preconditions for the efficient calculation are optimal load balancing of processors and data model partitioning among processors. The proposed algorithm is applied to different real models of power distribution systems. It obtains better results than classical evolutionary algorithms (Genetic Algorithm and Particle Swarm Optimization). The Hybrid Genetic Algorithm also achieves better results than multilevel algorithm (METIS) in cases of small graphs.

Keywords: graph partitioning, genetic algorithm, Distributed Management System, Common Information Model.

1. Introduction

In order to effectively manage modern adaptive applications within the multiprocessor system, large amounts of data need to be optimally processed in parallel. For optimal calculations to be carried out on certain groups of data, the following conditions must be satisfied: the data are necessarily partitioned across the processors and load balancing of the processors must be achieved. It is necessary to make an optimal distribution of data across the processors in accordance with pre-defined criteria that determine the calculation functions so that the calculations and the overall system functioning is efficient.

The Integrated Smart Grid power system largely increases the amount of data in each of its subsystems. It integrates systems such as [1]: Geographic Information System, Energy Management System, Distribution Management System (DMS), Outage Management System, etc. We treat parallel calculations in the DMS as one subsystem of Smart Grid system. DMSs have the supervision, management, planning and visualization of the distribution network in two modes: on-line (real-time) and simulation. Analytical DMS functions are the central component of the DMS. They allow monitoring and manage the power distribution network.

The most important DMS functons are: Load Flow (LF), State Estimation (SE), Fault Calculation, Performance Indices, Volt/Var Control, etc.

This paper describes the optimal partitioning of power distribution network data model by using the most common DMS functions – Load Flow and State Estimation. Also, the real power distribution networks and the corresponding functions are analyzed. The LF [2] and SE function algorithms are based on the weakly meshed radial data network. SE requires multiple LF execution and will thus not be discussed further. The network architecture makes the data models suitable for partitioning and parallel calculations [3, 4].

Depending on an application context of the DMS, dynamic data can be monitored either *on-line* by the SCADA system (*real-time context*), or *off-line* – defined and analyzed by the user (within *the planning, simulation or testing contexts*). Calculation time of the functions above becomes critical with increasing amounts of data involved in calculations (the present network contains tens of millions of data). For DMS functions optimization, only real-time (RT) context is critical for time calculation.

The initial partitioning of the DMS data model is studied. The partitioning is based on the initial values of objects' attributes and the relationships between the data. As a result, groups of data necessary for calculation are made. Dynamic changes of certain attributes (usually the relations among the data objects) can affect the data; and consequently, calculations. Thus we introduce the optimization criterion by which a potential need of using the data stored in other partitions is lowered. The structure of the power distribution network enables the transformation of data into a graph that can be successfully partitioned. In other words, the problem of DMS data model partitioning can be reduced to the problem of graph partitioning.

This paper reports the results of experiments which show that hybrid Genetic Algorithm (HGA) can be successfully applied for initial partitioning and that it always provides better results than simple Genetic Algorithm (GA)[5] and Particle Swarm Optimization (PSO)[6] algorithm.

The paper is organized in the following way: next section describes related work on algorithms for partitioning graphs. Section 3 describes the problem and defines terms used in the paper. It also includes the details of data model and definitions of the optimization problem. The HGA for initial partitioning is presented in Section 4. Section 5 describes experimental setup, presents and discusses the results. Section 6 is a conclusion.

2. Related Works

Graph partitioning is classified as an NP hard optimization problem [7]. Many suboptimal graph partitioning solutions have been proposed. Two commonly used classes of algorithms are multilevel and evolutionary algorithms.

Multilevel algorithms [8, 9] consist of three phases: *coarsening* – the matching of vertices per levels, *partitioning* – the partition graph consisting of matched vertices at a certain level and *refinement* – improvement of the partitioning at various levels. Most of these algorithms use KL [10] or its variant FM [11] algorithms, which are specialized for local improvement of solutions by exchanging vertices between the individual partitions. In addition, the FM algorithm moves a vertex from one partition to another in each iteration, and the KL algorithm exchanges vertices between partitions. Most commonly used multilevel algorithm is METIS algorithm [9].

In our studies [5, 6, 12], evolutionary algorithms are used for the initial partitioning. This allows the use of somewhat slower algorithms since it is done prior to startup. However, before the algorithms are applied, the coarsening of data is required in order to reduce the problem dimensionality and make a smaller number of data groups. The aim of the study reported in this paper is to improve previously obtained results and determine the applicability of evolutionary algorithms to large data model partitioning in power distribution systems.

Evolutionary algorithms have been successfully used for graph partitioning. The most commonly used algorithms are PSO[6] and GA[5, 13-15].

GA is used for graph partitioning: various size graphs or unweighted ones [13]. Furthermore, parallel GA [14, 15] was developed to speed up the algorithm. If the accuracy of a simple GA is not satisfied, then variants of hybrid GA are recommended [16-19]. GA finds a solution close to the optimum and some other algorithm (often KL/FM) is used for local refinement [13].

This paper compares hybrid GA with previously used GA[5, 12] and PSO[6, 12] algorithms for initial graph partitioning.

3. DMS Data Model Partitioning

3.1. DMS data model

Data models in power distribution utilities, based on Common Information Model (CIM) connectivity/topology model (Figure 1) are studied. CIM [20] is a well-known standard established by the International Electrotechnical Commission (IEC). It defines an object-oriented model of electric power systems, represents resources as classes and associations between them.

The distribution network is a weakly meshed radial network. It begins in high-voltage substations (*Substation*) of the power sources (*EnergySource*), representing the point of supply from energy transmission networks. The connectivity model is composed of transformer substations (represented by *PowerSystemResource*) connected with power lines (*ACLineSegment*), which are used to supply groups of consumers with energy (*EnergyConsumer*). Substations contain various equipment (*ConductingEquipment* and *PowerTransformer*) that is connected by various nodes (*ConnectivityNode*). *ConductingEquipment* objects are modelled with single- or double-ended (*SingleEndedCondEq and DoubleEndedCondEq*) conductors, and these are always connected with *ConnectivityNode*(s). Some of the typical single-ended conductors are *EnergyConsumer*, *EnergySource,* and *BusbarSection*, and double-ended ones are *Switchgear* (*Breaker*, *Fuse*) and *ACLineSegment*. In essence, the connectivity model is an edge-vertex model suitable for a graph presentation, where edges and vertex are instances of *ConductingEquipment* and *ConnectivityNode*, respectively.

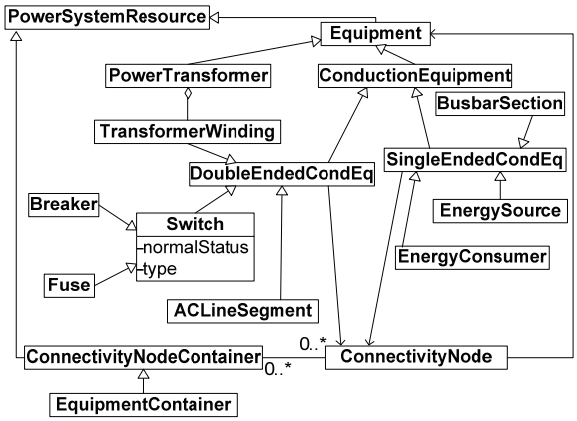

Figure 1. Connectivity CIM based DMS data model

The DMS data model partitioning is described by analyzing the target DMS functions. It is assumed that only one function ξ is running in the system. This is done in order to optimally partition datasets and finish data processing as quickly as possible.

DMS data model (Q) contains all elements α_i that are included in a calculation. Conducting equipment and transformers are elements usually used for the DMS functions.

If two elements α_i and α_j are in a relation $N(\alpha_i, \alpha_j)$ then it is assumed that the calculation function ξ uses them together. The elements are ***connected*** and they are called ***neighbors***.

The relation between neighbors that depends on the state of an element (u_{ij}) could be temporarily inactive. We refer to it as ***potential connection Pot(αi, αj, u_{ij})***. When a potential connection is activated, two elements become neighbors

$$(\forall \alpha_i, \alpha_j \in Q)\ Pot(\alpha_i, \alpha_j, u_{ij} = active) \Rightarrow N(\alpha_i, \alpha_j) \qquad (1)$$

For example, state of a switch between pieces of equipment could be active (closed) or inactive (open).

The set of mutually connected elements is called ***calculation region R*** (or just ***region***)

$$(\forall \alpha_i \in R_k)\ N(\alpha_i, \alpha_j) \Leftrightarrow \alpha_j \in R_k, \quad k \in \{1,2,...,n\} \qquad (2)$$

and it is the smallest data unit that can be processed by calculation function.

All regions create a ***calculation domain*** D ($D = R_1 \cup R_2 \cup ... \cup R_n$) and function $Y_i = \xi(R_i)$ is applied to each region R_i to produce output result set $Y = Y_1 \cup Y_2 \cup ... \cup Y_n$.

3.2. Model Partitioning

The process of partitioning large datasets consists of the following phases: 1) formation of the initial graph, 2) topological analysis and creation of the graph for calculations (coarsened graph), and 3) partitioning of the coarsened graph.

1. Formation of the initial graph

The initial graph is derived from the conduction elements and other equipment involved in the calculations. The edges of the initial graph are: (i) double ended equipment, (ii) single ended equipment with one fictive vertex. The transformers with two windings are presented with two edges (primary + derived), and transformers with three windings are presented with three edges (primary + two derived). The vertices are connectivity nodes.

2. Topological analysis and creating of the graph for calculations

Topological analysis is one of the DMS functions. It depends on the status of conducting equipment; for instance, the state of the switches (open / closed). The analysis determines the basic group of elements (calculation regions) necessary for the calculations. The result of the analysis is a calculation domain D that could be represented by undirected weighted graph G = (V, E) with vertices (V) and edges (E).

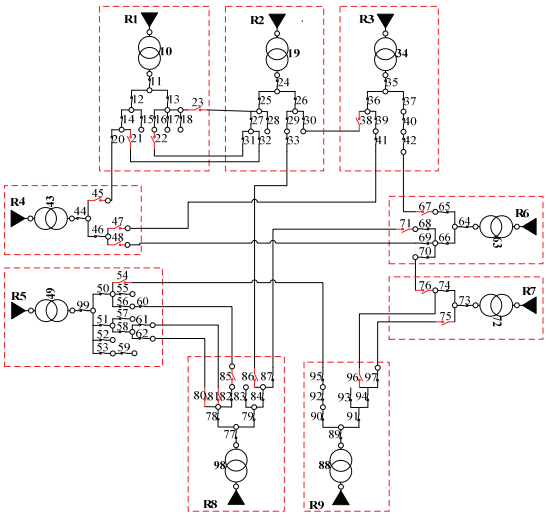

Figure 2. An example of initial data model in power distribution system

The weight of vertex v_i ($w(v_i)$) depends on the complexity of the calculation for the appropriate region R_i. The example of determining the complexity of the calculation for different types of electrical components in power distribution systems is given in [21]. If Load Flow (LF) function is considered, it can be inferred that the complexity of the LF calculation is linearly proportional to the total number of elements in the region. Therefore the weight of vertex v_i is equal to the number of elements in the region R_i. The weight of edge $e_{i,j} = (v_i, v_j)$ ($w(e_{i,j})$) is equal to the number of the potential connections between elements of two regions (for example there are 3 open switches between regions R_1 and R_2, therefore $w(e_{1,2})=3$). Thus, for example, for LF calculations, the initial graph model from Figure 2 would be transformed into the coarsened graph shown in Figure 3.

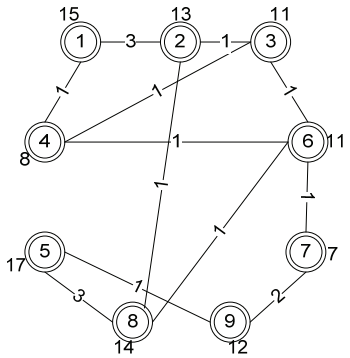

Figure 3. Coarsened graph for Load Flow calculation

3. Partitioning of the coarsened graph - definition of optimization criteria

The problem of graph partitioning is based on partitioning of the undirected graph with vertices and edges that have a certain weight. The result of p-way partitioning is a set of p partitions $\Pi=\{\pi_1, \pi_2, ..., \pi_p\}$,

where partition π_i contains an optimal set of regions (vertices).

3.3. Optimization criterion

In order to define optimization criterion for a partition π_k, we need to define the partition weight $W_{\pi k}$:

$$W_{\pi_k} = \sum_{R_j \in \pi_k} w_{R_j} \quad (3)$$

as the sum of all contained regions' weights. We also need to define a function ϕ_k as:

$$\phi_k = \sum_{Rp, Rq \in \pi_k} Pot_{R_p, R_q} \quad (4)$$

This function is an indicator of "good connectivity" between the regions in the partition.

First, it is necessary to group regions into a defined number of partitions (p), so that the weights of these partitions are approximately the same. They can never be greater than the *maximal partition weight M* defined as:

$$M = (1+\varepsilon) \cdot \frac{1}{p} \sum_{k=1}^{p} W_{\pi_k} \quad (5)$$

where W_{π_k} is weight of partition π_k, p is the number of partitions, and $\varepsilon \in [0, (p-1)/p]$ is the tolerance.

The optimization criterion should obtain the maximum connection inside a partition:

$$\Psi = \sum_{k=1}^{p} \phi_k \,, \quad (6)$$

$$F = \max(\Psi) \,, \quad (7)$$

where function ϕ_k is given in (4), and all partition weights are constrained by:

$$W_{\pi_k} \le M, \forall k \in \{1, 2, ..., p\} \,. \quad (8)$$

It should be noted that the maximal number of connections inside each of the partitions means a minimal number of connections among partitions.

4. Algorithms

We apply the HGA to find the best approximate solutions for the optimization problem (7) with respect to the constraints given in (8). HGA consists of the simple genetic algorithm. Its solutions are improved by using FM algorithm.

It is assumed that a graph with the n regions (vertices) should be divided into the p partitions. The solution is represented by the vector of the n elements whose values are indices of the given partitions (integers between 0 and p-1).

4.1. Genetic Algorithm (GA)

GA starts from randomly constructed initial solutions, which are considered individuals in a *population*. The initial partition weight can affect the quality of the solution. After coarsening the graph, the DMS data model partitioning problem becomes the weighted graph partitioning problem. The weighted graph has less than 1000 vertices (realistic estimations are around 500 vertices). Experimental results show that good solutions for this kind of graph partitioning can be obtained when the number of individuals is similar to the number of vertices (regions). The GA execution time is not considered since the algorithm execution time is not critical. The algorithm is executed prior to the start of the system.

GA is an iterative algorithm that consists of selection, crossover and mutation.

Different types of *selection* (random, tournament and roulette-wheel) are studied. Experimental results show that the best selection type is roulette-wheel.

In addition, different types of crossovers can be used: in one point, in the k-points, etc. For the purposes of our research, we chose the one point crossover. It randomly selects a cut point which is the same on both parent individuals. The cut point divides the individuals into two disjoint parts. Two offsprings are created from opposite parts of parents. If the constraint for partition weights (given by (8)) is not satisfied then a high penalty factor is used to decrease function F. Besides the one point crossover (without preprocessing), we investigate the following crossovers: (i) a crossover with unique partition representation (normalization) in all individuals [22], and (ii) a crossover with the favorite best partition [5]. Our experiments show that the best results are obtained with the crossover without preprocessing ([22], [5]). The probability that a certain individual is involved in a crossover is called crossover rate. Mutation is performed with a probability that is called mutation rate. The choice of mutation rate significantly influences the solution quality.

Also, the results gained from our experiments show good convergence of GA (i.e. good global search). However, inadequately fine solutions indicate the need to combine GA with algorithms for local improvements. FM algorithm is chosen for local improvement.

4.2. Fiduccia-Mattheyses (FM) algorithm

FM algorithm [11] is a group migration heuristic which starts with a partitioned graph and iteratively moves vertices to improve solution. It selects the vertex that should be moved from one partition to another. The criterion for moving vertex v_i from partition π_S to partition π_J is the maximum value of gain parameter (g). The parameter is calculated as:

$g(i, J)=FS(i, J)-TE(i)$

where *FS(i, J)* is the number of edges connecting the vertex v_i with vertices in the partition π_J, and *TE(i)* is the number of edges incident to vertex v_i and another vertex in the same partition (π_S) as vertex v_i.

In the case of weighted graphs, it is the sum of weights of the corresponging edges that is calculated instead of the number of edges. Fiduccia and Mattheyses [11] propose a method of FM algorithm realization. The method speeds up the execution time of the algorithm by using data structure (*bucket list*) that reduces computation time (an iteration can be done in O(|E|) time). For multiple-way partitioning, all possible transfers of border vertices (neighbors with vertices from other partitions) are checked.

Pseudo code of a simplified FM algorithm is given below:

```
Input: partitioned graph G=(V,E) ->
                    Π=π1∪π2∪...∪πp
Output: improved partitions of graph G ->
                    Π`=π1`∪π2`∪...∪πp`
Repeat
    for each border vertex vi in graph G
        for each partition πJ in Π
            g(i,J) ← FS(i,J)-TE(i)
        end for
    end for
    find vb∈πS and πD ← (max(g)=g(b,D))
    if(w(πD∪{vb})< M)
        πD` =πD∪{vb}
        πS` =πS\{vb}
    end if
    recalculate g(i,J)(for neighbors of vb)
until (max(g) ≤ 0)
```

In the HGA, we first use the GA to partition a graph; then, we use the FM algorithm to improve this graph.

4.3. Hybrid Genetic Algorithm

The HGA utilizes the results of FM algorithm to improve the local optimum. The initial population is generated randomly (for each gene in each individual the partition number is randomly set). Since the FM algorithm moves a vertex from one partition to another (in the GA, this movement presents a value change of a particular gene), hybridization of the genetic algorithm is carried out in a mutation phase. HGA uses mutations from the classical genetic algorithm; but it also introduces FM mutations periodically (every *freq* iterations).

Pseudo code of the HGA is given below:

```
Input: unpartitioned graph G=(V,E)
Output: partitioned graph Π=π1∪π2∪...∪πp
create initial population X={x1,x2,...,xm}
it ← 0
repeat
   select pairs (xi,xj) for crossover
   it ← it+1
   for each pair (xi,xj)
        offspringi,j ← Crossover(xi,xj)
        X = X ∪ {offspringi,j}
   end for
   Xmut ← select individuals for mutation
   for each individual xk∈Xmut
        xnk ← Mutation(xk)
        X = X ∪ {xnk}
   end for
   if it%freq = 0
        XFM ← select individuals for FM
        for each individual xq∈PFM
           xnq ← FM_mutation(xq)
           X = X ∪ {xnq}
        end for
   end if
  X←choose the best m individuals from X
until (no progress) or it > maxit
```

The algorithm is finished either if there is no improvement or, after a predefined number of iterations (*maxit*) are executed.

5. Experimental Results

We tested the models of the real electric power distribution network. The network characteristics are shown in Table 1.

Table 1. Test data models

Graph name	Total number of elements	Size of initial graph (uncoarsened)		Size of coarsened graph	
		Number of vertices	Number of edges	Number of regions	Number of edges
bg54	1126254	295225	299131	54	44
it206	1787939	431102	434486	206	286
pec106	2284322	762411	766755	106	52
bg63	1196078	300503	304637	63	52
bg5x	5980390	1502505	1523180	315	260

Models *bg54* and *bg63* are different versions of Belgrade (Serbia) power distribution network. *Pec* is a part of North Carolina (United States) power distribution network model; *it206* is a network model of Milano (Italy); and *bg5x* is Belgrade network model multiplied five times.

We ran initial partitioning tests using HGA on datasets from Table 1.

The tests were carried out for 2, 3, 4, 5, 6 and 8 partitions. Each test was repeated 100 times. The HGA is applied with the following parameters:

Population size	*bg54*, *bg63*	100 individuals
	pec106	120 individuals
	it206	220 individuals
	bg5x	330 individuals
Crossover rate		0.5 (50%)
Mutation rate		0.5 (50%)
FM mutation frequency		100 iterations
Balancing coeficient ε		0.1 (10%)
Maximal number of iterations		5000

Our experiments show that the best results are achieved with a mutation rate of 0.5 (50%) and crossover rate of 0.5 (50%).

The obtained results (for function F given by (7)) are shown in Table 2.

Table 2. Partitioning results (the best results are in bold)

p	*Model*	GA	PSO	HGA	METIS
2	*bg54*	397	392	**398**	**398**
	bg63	479	475	**483**	**483**
	pec106	157	149	**163**	**163**
	it206	936	853	**940**	937
	bg5x	2393	2379	2447	**2465**
3	*bg54*	378	385	**389**	388
	bg63	466	455	**474**	**474**
	pec106	152	146	**161**	**161**
	it206	889	746	**912**	907
	bg5x	2327	2190	2411	**2465**
4	*bg54*	359	363	**384**	382
	bg63	461	444	**468**	465
	pec106	149	138	153	**156**
	it206	859	674	881	**883**
	bg5x	2270	2062	2373	**2452**
5	*bg54*	353	336	**377**	**377**
	bg63	437	419	**461**	**461**
	pec106	135	129	148	**152**
	it206	845	603	841	**888**
	bg5x	2262	1949	2351	**2465**
6	*bg54*	335	310	**359**	346
	bg63	444	360	**460**	448
	pec106	135	120	150	**152**
	it206	850	530	821	**870**
	bg5x	2246	1933	2232	**2463**
8	*bg54*	285	255	**361**	360
	bg63	363	237	**443**	399
	pec106	132	119	145	**148**
	it206	480	421	757	**820**
	bg5x	2225	1908	2280	**2385**

The table includes the results obtained from previously developed GA and PSO algorithms [5, 6, 12].

Based on these results, we can conclude that the HGA always achieves significantly better results than simple GA and PSO algorithms. This is particularly evident when partitioning graphs which have over 100 vertices, and when there is a large number of partitions.

The applicability of the HGA is determined by giving a comparison of the results gained from the HGA and the ones gained from the METIS algorithm. The METIS algorithm is much faster but this characteristic is insignificant for the practical applications of the model initial partitioning. Thus we conclude that the HGA yields better results for graphs of up to 200 vertices. We recommend specialized multilevel algorithms for partitioning graphs that are over 200 vertices and if there is a large number of partitions required.

6. Conclusions

In this paper, we developed a new HGA for initial partitioning of large data models. The amount of data that need to be processed in the DMSs is increased. The increase calls for parallel calculations of analytical DMS functions. The calculations are executed in different processors; consequently different data are sent to different processors. The DMS network is such that the problem of data partitioning boils down to the problem of graph partitioning.

The algorithm is successfully applied to large scale electricity power distribution data model. Experiments reported in the paper show that HGA always obtains better results than simple GA and PSO algorithms do. Further, HGA is compared with METIS algorithm. The results show that HGA is good for partitioning graphs with up to 200 vertices.

Acknowledgements

We would like to thank our associates at Telvent DMS Llc, Novi Sad, Serbia for their help during our work on this research. They were the sole source of all real-life electric power system data. This work has been partly supported by the Serbian Ministry of Education&Science, through grant No. 32018, 2011.

References

[1] **E. Santacana, G. Rackliffe, L. Tang, X. Feng.** Getting Smart. *IEEE Power and Energy Magazine*, 2010, *Vol*.8, *No*.2, 41–48.

[2] **D. Shirmohammadi, H.W. Hong, A. Semlyen, G.X. Luo.** A Compensation-Based Power Flow Method For Weakly Meshed Distribution And Transmission Networks. *IEEE Transactions on Power Systems*, 1988, *Vol*.3, *No*.2, 753–762.

[3] **M. Popovic, I. Basicevic, V. Vrtunski.** A Task Tree Executor: New Runtime for Parallelized Legacy Software. 16*th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems, ECBS* 2009, *San Francisco, USA*, 2009.

[4] **A. Igumenov, T. Petkus.** Analysis of Parallel Calculations in Computer Network, *Information Technology and Control*, 2008, *Vol*.37, *No*.1, 57–62.

[5] **D. Capko, A. Erdeljan, S. Vukmirovic, I. Lendak.** Using Genetic Algorithm for Power Distribution Network Partitioning. 2*nd Regional Conference, Industrial Energy and Environmental Protection in South Eastern European Countries*, *Zlatibor, Serbia*, 2010.

[6] **D. Capko, A. Erdeljan, I. Lendak.** PSO algorithm for Graph Partitioning (in Serbian). 17*th Telecommunication* Forum 2009, *Belgrade, Ser*bia, 2009.

[7] **M.R. Garey, D.S. Johnson.** Computers and Intractability: A Guide to the Theory of NP-Completeness. *W.H. Freeman, San Francisco*, 1979.

[8] **B. Henderson, R. Leland.** A Multilevel Algorithm for Partitioning Graphs. *Proceedings of ACM/IEEE conference on Supercomputing*, *San Diego*, 1995.

[9] **G. Karypis, V. Kumar.** A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 1998, *Vol*.20, *No*.1, 359–392.

[10] **B.W. Kernighan, S. Lin.** An efficient heuristic procedure for partitioning graphs, *The Bell System Technical Journal*, 1970, *Vol*.49, *No*.2, 291–307.

[11] **C.M. Fiduccia, R.M. Mattheyses.** A linear time heuristic for improving network partitions. *In: Proc.* 19*th IEEE Design Automation Conference*, 1982, 175–181

[12] **D. Capko, A. Erdeljan, M. Popovic, G. Svenda.** An Optimal Relationship-Based Partitioning of Large Datasets. 14*th East-European Conference on Advances in Databases and Information Systems*, *Novi Sad, Serbia*, 2010.

[13] **T.N. Bui, B.R. Moon.** Genetic Algorithm and Graph Partitioning. *IEEE Transactions on Computers*, 1996, *Vol*.45, *No*.7, 841–855.

[14] **E.G. Talbi, P. Bessiere.** A Parallel Genetic Algorithm for the Graph Partitioning Problem. *In: Proc. of the International Conference on Supercomputing*, Cologne, 1991.

[15] **E. Cantu-Paz.** A Survey of Parallel Genetic Algorithms, Technical Report. *Illinois Genetic Algorithms Laboratory, Department of General Engineering*, *Illinois*, 1997.

[16] **T.A. El-Mihoub, A.A. Hopgood, L. Nolle, A. Battersby.** Hybrid Genetic Algorithm: A Review. *Engineering Letters*, 2006, *Vol*.13, *No*.2, 124–137.

[17] **S. Kang, B. Moon.** A Hybrid Genetic Algorithm for Multiway Graph Partitioning. *In Proceedings of the Genetic and Evolutionary Computation Conference*, 2000.

[18] **A. Misevičius.** An Extension of Hybrid Genetic Algorithm for the Quadratic Assignment Problem, *Information Technology and Control*, 2004, *Vol*.4, *No*.33, 53–60.

[19] **A. Misevičius, D. Rubliauskas.** Performance of Hybrid Genetic Algorithm for The Grey Pattern Problem, *Information Technology and Control*, 2005, *Vol*.34, *No*.1, 15–24.

[20] IEC 61970 Energy management system application program interface (EMS-API) – Part 301: Common Information Model (CIM) Base. *IEC, Edition 2.0*, 2007.

[21] **S.S. Choi, B.R. Moon.** Normalization in genetic algorithms. *In Proceedings of the Genetic and Evolutionary Computation Conference*, 2003, 862–873.

[22] **P. Zhang, J.R. Marti, H.W. Dommel.** Network Partitioning for Real-Time Power System Simulation. *International Conference on Power System Transients*, *Montreal, Canada*, 2005.

Received March 2011.