Network Decomposition using Evolutionary Algorithms in Power Systems

G. Angeline Ezhilarasi, *Member, IACSIT* and K Shanti Swarup, *Senior Member, IEEE*

***Abstract*—Power system has a highly interconnected network that requires intense computational effort and resources for centralised control. Distributed computing is a solution to this and needs the systems to be partitioned optimally into clusters. The network partitioning is an optimization problem whose objective is to minimise the number of nodes in a cluster and the tie lines between the clusters. Evolutionary Algorithms like Discrete Particle Swarm Optimization (DPSO) and Harmony Search (HS) Algorithm is proposed to solve this combinatorial optimization problem. Connectivity of the partitioned networks is done using the conventional graph traversing techniques. Simulation is done on IEEE Standard Test Systems and IEEE 118 bus system case study is presented in this paper. The algorithms are found to be very efficeint in partitioning the system hierarchically and obtain a near optimal sub networks without having any isolated nodes.**

***Index Terms*—Discrete Particle Swarm Optimization, Harmony Search Algorithm, Network Decomposition, Optimal Partitioning.**

I. Introduction

Power system is a complex network with a large number of interconnections. The ever growing size and complexity of the system and the advances in computing technology have given an insight into parallel processing and distributed computing for power system computations. Network decomposition is an essential tool for parallel processing. The large interconnected network be optimally divided into clusters. In order to reduce the overall execution time there should be a balance between the size of the cluster and the interconnection between the clusters.

Over the past decades a number of algorithms have been proposed in literature for optimal network tearing. The techniques include matrix decomposition, successive approximation, dynamic programming and heuristic clustering approaches. These methods tend to form the clusters with less interconnections but fail to balance with the size of the clusters. Some of the evolutionary optimization techniques such as tabu search [1], genetic algorithm [2], simulated annealing [3], ant colony optimization [4] have also been applied to optimal partitioning of the network. These methods tries to minimize an objective function that reflects a balance between the two, so the objective function is formed such that it reflects the features of parallel and distributed processing. Network partitioning based on voltage variation at the load buses relative to other buses is proposed for voltage margin calculations in [5]. A decomposition algorithm is used for partitioning the network for distributed reactive power optimization in power system in [6]. Meta heuristic algorithms have been used for clustering web documents also [7]. However these methods are computation intensive and involves procedures based on natural selection crossover and mutation. It also requires a large population size and occupies more memory. To simulate and implement distributed computing in power system the large interconnected network must be torn into sub networks in an optimal way as shown in Fig. 1.

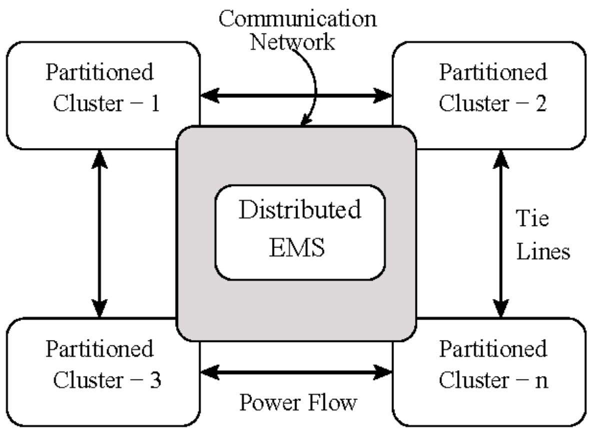


Fig. 1. Power System Network Decomposition

The partitioning of the interconnected network should balance between the size of the sub networks and the interconnecting tie lines in order to reduce the overall parallel execution time. This paper models the network decomposition problem as an optimization problem and proposes to solve by two evolutionary algorithms namely Discrete Particle Swarm Optimization (DPSO) and Harmony Search (HS) Algorithm.

Particle swarm optimization (PSO) is a population based stochastic optimization technique inspired by social behavior of bird flocking or fish schooling. The main advantages of the PSO algorithm are summarized as: simple concept, easy implementation, robustness to control parameters, and computational efficiency when compared with mathematical algorithm and other heuristic optimization techniques. Recently, PSO have been successfully applied to various fields of power system optimization such as economic dispatch and optimal power flow [8] power system stabilizer design, reactive power and voltage control, dynamic security border identification. Improved and modified versions of PSO are given in [9],[10]. Discrete PSO is used for similar Travelling Salesman Problem (TSP) in [11] and

Manuscript received March 15, 2011.

G. Angeline Ezhilarasi is with Department of Electrical Engineering, Indian Institute of Technology Madras, Chennai, 600036 INDIA (phone: +91-98407 28451; fax: +91 44 2257 4402; e-mail: angel.ezhil@gmail.com).

Dr. K. S. Swarup is with Department of Electrical Engineering, Indian Institute of Technology Madras, Chennai, 600036 INDIA

Transmission Network Expansion and Planning (TNEP) in [12]. Evolutionary Tristate PSO is proposed in [13].

The recently evolving heuristic algorithm called Harmony Search (HS) Algorithm is based on the improvisations done by the musician to obtain a better harmony. The harmony search algorithm is applied for solving structural optimization problems [14], an improved harmony search is applied to optimal economic power dispatch [15]. HS algorithm has been used for transmission network planning [16]. In software engineering the HS algorithm has been used for the task assignment problem [17]. Self adaptive harmony search is proposed in [18] for expert system applications. A novel derivative of harmony search for discrete optimization problems has been proposed in [19]. In [20] a hybrid method has been proposed combining the harmony search method with the sequential quadratic programming method and a global harmony search algorithm is proposed for unconstrained optimisation problems as well. Hence the literature shows the applicability of the HS algorithm to a wide range of optimization problems.

The following sections are organized as follows. Section 2 deals with the formulation of the objective function for the network decomposition problem. The aim of this optimization is to minimize the cost function, which is a measure of the execution time of the applications in the torn network. An overview of the DPSO and its implementation to NP problem is done is section 3. The HS algorithm and its implementation is detailed in section 4. The algorithm is tested on IEEE standard test systems and simulation results are discussed in section 5. The case study demonstrate validity of the application of these algorithms by attaining a near optimal solution with less computational effort.

II. Network Decomposition

Power system by itself is a highly interconnected network with large number of nodes geographically distributed connected through tie lines. The Network Decomposition (ND) problem in power system has two main objectives namely (1) To minimize the number of nodes in a partition. (2) To minimize the number of lines in a partition. The number of nodes and the number of clusters represent the computational load on each cluster and the tielines indicate the communication between the clusters. Hence the network decomposition can be formulated as the combinatorial optimization problem with the following objective function.

$$Min \quad C(N, L) = \alpha N + \beta L \tag{1}$$

where,

C	Cost function of the partition
N	Maximum number of nodes in a partition
L	Maximum number of lines in a partition
α, β	Weight parameters for each term

This optimization problem is subject to the constraint that the nodes in each cluster must form a connected graph. This constraint checks the observability at the instant of decomposition of the network. This problem is solved using two evolutionary algorithms namely discrete particle swarm optimization and Harmony search algorithm in this paper.

III. Discrete Particle Swarm Optimization

A. Overview of the DPSO

Particle Swarm Optimization (PSO) was developed by Kennedy and Eberhart through simulation of bird flocking in a two dimensional space. In this search space, every feasible solution is called a particle and several such particles in the search space form a group. The particles tend to optimize an objective function with the knowledge of its own best position attained so far and the best position of the entire group. Hence the particles in a group share information among them leading to increased efficiency of the group. The original PSO treats non linear optimization problem with continuous variables. However the practical engineering problems are often combinatorial optimization problems for which Discrete Particle Swarm Optimization (DPSO) can be used.

In a physical n-dimensional search space, the position of a particle ‘p’ is represented as a vector X and the velocity of a particle as V . Let $Pbest_p$ and $Gbest_p$ be the best position of the particle p and its neighbors reached so far. In DPSO, the particles are initially set to binary values randomly. The probability of the particle making a decision is a function of the current particle, velocity, $Pbest$ and $Gbest$. The velocity of the particle given in (2) determines a probability threshold. The sigmoid function shown in (3) imposes limits on the velocity updates. The threshold is constrained within the range [0, 1] such that higher velocity likely chooses 1 and lower velocities chooses 0. Hence the position update is done using the velocity as shown in equation (4).

$$\begin{aligned} V_p^{k+1} = \omega V_p^k &+ C_1 rand_1 \left(Pbest_p^k - X_p^k \right) \\ &+ C_2 rand_2 \left(Gbest - X_p^k \right) \end{aligned} \tag{2}$$

$$S\left(V_p^{new}\right) = \frac{1}{1 + e^{-V_p^{new}}} \tag{3}$$

$$\begin{aligned} &If \; \left(rand < S\left(V_p^{new}\right) \right) \; then \quad X_p^{new} = 1 \\ &\qquad\qquad\qquad\qquad else \quad X_p^{new} = 0 \end{aligned} \tag{4}$$

where,

ω	Weight parameter
C_1, C_2	Weight factors
$rand$	Random number between 0 and 1
X_p^{k+1}, X_p^k	Position of the particle at the (k+1)th and kth iteration
V_p^{k+1}, V_p^k	Velocity of the particles at the (k+1)th and kth iteration
$Pbest_p^k$	Best position of the particle p until the kth iteration
$Gbest$	Best position of the group until the kth iteration

The PSO gains self adapting properties by incorporating any one of the evolutionary techniques such as replication, mutation and reproduction. In order to improve the convergence in PSO mutation is generally done on the weight factors. Also if there is no significant change in the Gbest for a considerable amount of time then mutation can be applied. In this work mutation is used to update the particles as they are constituted by binary values only. The entire position update process of the particles is done based on the mutation probability normally above 0.85[6]. This ensures that the particles are not trapped in their local optimum and do no deviate far off from the current position as well. The process of the algorithm and its implementation aspects are described in detail in the following sections.

B. Generation of the Particles

The objective of the network partition problem is to allocate every node of power system network to a sub network, such that the nodes are equally distributed and number of lines linking them is minimum. Hence the structure of the particle is framed as matrix of dimension (nc x nn), where 'nc' is the number of clusters or sub networks and 'nn' is the total number of nodes in the power system network. It is ensured that the each node is assigned to one cluster only .i.e, the sum of the columns of the particle array is 1 always. The velocity of the particles corresponds to a threshold probability. Initially all velocities are set to zero vectors and all solutions including the Pbest and Gbest are undefined. The position of particles 'p' in the search space is created as follows:

1) Set j = nc, the number of sub networks
2) Generate a random number R1 in the range [1, nn/nc].
3) Set the number of nodes to 1 from N1 to R1.
4) Set j = j -1, if j = 0 go to step 2, otherwise go to step 5
5) Repeat steps 1 to 4 for all the particles.
6) Stop the initialization process.

Clusters/Nodes	N1	N2		Nn-1	Nn
C1	1	0	0	0	0
C2	0	1	0	0	0
	0	0	1	0	0
Cnc-1	0	0	0	1	0
Cnc	0	0	0	0	1

Fig. 2. Structure of a particle in the generation

The particle structure of the network partition problem is shown in Fig. 2 for the system with 'nc' clusters or sub networks and 'nn' nodes.

C. Evaluation of Particles

The particles in the solution space are evaluated by means of the fitness function given by equation (1). The fitness function is such that the results of the optimization problem would balance the computational load on the processors and reduce the communication overhead as well. The choice of the exponents of M and L determines the order of solution times required for sub network solution and full solution of the interconnected network in a typical parallel processor solution. Once the particles are evaluated the Pbest and Gbest are selected from the swarm in that iteration as follows:

1) Set j = 1, p = the number of particles.
2) If F (Xp) > Pbestp) then Pbest = F (Xp)
3) If max (F (Xp)) > Gbest) then Gbest = F (Xp)
4) Set j = j +1, if j = p go to step 2, otherwise go to step 5
5) Stop the evaluation process.

D. Modification of Particles

To modify the particles in the solution space for the next iteration, the velocity of the particles are obtained from equation (2). In this process of updating the velocity the weight factors must be known a priori. It has been shown that the irrespective of the problem the following parameters are appropriate. $C_1 = C_2 = 2.0,\ \omega_{\max} = 0.9,\ \omega_{\min} = 0.4$. In this paper the weighting function is kept constant for all iterations and is taken as the average of its range. Once the velocities are updated for the next iteration the particles are updated based on the sigmoid function given by (3). Since this is a discrete optimization problem and there exists some constraints on the redundancy of the nodes in the sub networks. The particles as depicted in Fig. 3. are modified using the following procedure based on a high mutation probability

1) Set N = number of Nodes and M = number of clusters.
2) Select a column in random from 1 to N
3) Find the row index whose element is 1
4) Select a row in random from 1 to M whose element is 0.
5) Flip the elements using the condition given by (4).
6) Repeat the above steps for all the particles.

Clusters/Nodes	N1	N2		Nn-1	Nn
C1	1	0	0	0	0
C2	0	1	0	0	0
	0	0	1	0	0
Cnc-1	0	0	0	1	0
Cnc	0	0	0	0	1

Clusters/Nodes	N1	N2		Nn-1	Nn
C1	1	0	0	0	0
C2	0	0	0	0	0
	0	0	1	0	0
Cnc-1	0	0	0	1	0
Cnc	0	1	0	0	1

Fig. 3. Modification of Particle Structure

E. Stopping Criteria

Generally for evolutionary algorithms the solution is reached if the fitness function remains constant for a considerable amount of iterations or a maximum number of iterations can be fixed. In this paper the later is followed.

IV. HARMONY SEARCH ALGORITHM

A. Overview of the HS Algorithm

The Harmony Search (HS) algorithm is a meta heuristic algorithm developed recently based on the improvisation of harmony in the music composition. Music improvisation is analogous to the optimization process seeking an optimal solution. Each musician corresponds to each decision variable; musical instrument's pitch range corresponds to decision variable's value range; musical harmony at certain time corresponds to solution vector at certain iteration; and audience's aesthetics corresponds to objective function. Just like musical harmony is improved time after time, solution vector is improved iteration by iteration. There is no need for initial values of decision variables in HSA. Unlike other heuristic optimisation algorithms that uses gradient search, the HS uses the stochastic random search. The search is based on the harmony memory considering rate and the pitch adjusting rate and so the derivative information becomes unnecessary.

The optimization procedure of the HS algorithm is as follows:

1. Initialize the optimization problem and the algorithm parameters.
2. Initialize the Harmony Memory (HM).
3. Improvise a new harmony from the Harmony Memory.
4. Update the Harmony Memory.
5. Repeat Steps 3 and 4 until the termination criterion is reached.

The detailed explanation of the steps involved in the harmony search is as follows:

B. Initialization of the problem and HS parameters

The optimization problem is well defined initially as

$Minimize\ f(x)\ subject\ to\ x_i \in X_i,\ i = 1,2,3,\cdots n$ where, $f(x)$ is the objective function, x_i is the set of decision variables, X_i is the set of possible range of the design variables and n is the number if decision variables. The parameters of the HS algorithm namely, the size of the harmony memory (HMS) representing the number of solution vectors, the harmony memory considering rate (HMCR), the pitch adjusting rate (PAR) and the stopping criteria are set in this step.

C. Initialize the Harmony Memory (HM)

The harmony memory is a matrix in which the sets of decision variables are stored. In this step the HM is filled with randomly generated decision variable with a uniform distributed and represented as follows in (5)

$$HM = \begin{pmatrix} x_1^1 & x_2^1 & \cdots & x_n^1 \\ x_1^2 & x_2^2 & \cdots & x_n^2 \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{HMS} & x_2^{HMS} & \cdots & x_n^{HMS} \end{pmatrix} \tag{5}$$

D. Improvise the new harmony from the HM

Improvisation is the process of creating a new harmony from the existing memory. A new harmony vector $X' = (x_1', x_1', \cdots, x_n')$ is generated based on three rules: (a) memory consideration (b) pitch adjustment and (c) random selection. The value of the first decision variable is chosen from one of the harmony between $x_1^1\ to\ x_1^{HMS}$. The other decision variables in the harmony memory are also chosen in the same manner with the probability of the Harmony Memory Considering Rate (HMCR) that varies between 0 and 1. HMCR is the rate of choosing a variable from the historical data in the HM and (1-HMCR) is the rate of randomly selecting one value from the possible range of values.

$$x_i' = \begin{cases} x_i' \in \{x_i^1, \cdots, x_i^{HMS}\} & with\ prob\ HMCR \\ x_i' \in X_i & with\ prob\ (1 - HMCR) \end{cases} \tag{6}$$

After all the components are obtained they are pitch adjusted using the Pitch Adjustment Rate (PAR). The decision of pitch adjustment is yes with the probability PAR and No with the probability (1-PAR). It is done as follows

$$x_i' = \begin{cases} x_i' + r.bw & with\ probability\ PAR \\ x_i' & otherwise \end{cases} \tag{7}$$

where,

bw	is a arbitary distance bandwidth
r	is a uniform random number between 0 and 1

E. Update the Harmony Memory

If the new harmony vector $X' = (x_1', x_1', \cdots, x_n')$ is better than the worst harmony in the HM, then the new harmony memory is included in the HM and the existing worst harmony is excluded from the HM. This is decided based on the fitness value of the harmony.

F. Check the Stopping Criteria

The HS algorithm is stopped till the number of improvisations is met and there is no further improvement in the harmony. Steps 3 and 4 are continued until the stopping criteria is met.

G. Implementation of HS Algorithm to ND Problem

The HS algorithm is proposed to optimally partition the power networks into individual clusters, such that the clusters are connected networks. In this clustering problem, the variables in the harmony memory are the nodes or the buses in the network. The initial system data such as nodes and lines and their connectivity is taken from the original network. A set of random vectors are generated varying between 1 to n without any redundancy to create the initial harmony memory. The random memory is partitioned into the desired number of clusters.

The connectivity of the derived cluster is determined using graph theory. The number of tie lines linking the clusters, the number of nodes in a cluster and the number of lines in a cluster is obtained from the partitioned memory. If the cluster has no isolated nodes then based on the fitness value the improvisations are done as explained in the previous section.

V. Connectivity of the Partitions

The graph traversing is mainly used in two major applications in power system namely, topological problems and network flow problems. The topological or structural problems deals with finding the parts of the graph connected and defines one specific or all spanning trees. It also determines how strongly the graph components are connected and how to colour different parts of the network.The network flow problems include solving the shortest path problem, finding feasible optimal flow pattern and recognising the loop flows and wheeling problems.

Breadth First Search (BFS) and Depth First Search (DFS) are the two techniques in graph theory having fundamentally different traversal techniques. While in BFS a node is fully explored before the exploration of any other node begins, in DFS the leaves of the spanning trees are reached in the fastest possible way. Depth first search requires less memory since only the nodes on the current path are stored, whereas in breadth first search all the nodes that have been generated so far has to be stored. If there is a complete traversal possible, BFS guarantees to find it, by the fact that longer paths are never explored until all shorter ones have been examined. This contradicts with DFS, which finds a long path to a solution from one part of the tree, when a shorter path exists in some other unexplored part of the tree.

In this paper BFS is used for the graph traversal and the visited nodes are removed from the initial open list. If the open list is empty at the end the cluster is connected, other wise there are isolated nodes in the cluster. This ensures that the partitioned cluster is completely observable.

VI. Simulation Results

To study the effectiveness of the proposed method of clustering, simulation was done using the IEEE 118 bus test system. The DPSO was tested with Particle size = 100, Number of iterations =100, number of trials = 50 and the mutation probability = 0.5. Similarly parameters of the HS algorithm were set as follows: Harmony Memory Size = 100, Number of Improvisations = 100, Number of trials = 50, Harmony Memory Considering Rate = 0.8

Table I. Partitioning of IEEE 118 Bus System

Solution Methods	Nodes in Clusters	Tie Lines	Fitness Value	Time(s)
DPSO	64, 55	12	76	100.05
HAS	63, 55	9	72	150.95

Simulation was done on Intel Core 2 Duo Processor with 3.00 GHz speed and 2 GB RAM. The convergence characteristics of the DPSO and the HS Algorithm are shown in Fig.4 and Fig.5 respectively. It can be observed that the HS algorithm gives a better solution in terms of fitness value and explores the search space completely. However the HS algorithm is time consuming than the DPSO, which is tolerable since the partitioning is done offline.

The performance of the HS algorithm is depicted by its memory structure. The initial and final harmony memory is shown in Fig.6 and Fig.7 respectively for the 118 bus system. This indicates the randomness in the initial harmony and the hierarchical clustering in the final harmony memory. The connectivity of the clusters is verified by the artificial intelligence techniques such as breadth first search and depth first search and the clusters are found to contain no isolated nodes in the decomposed sub networks.

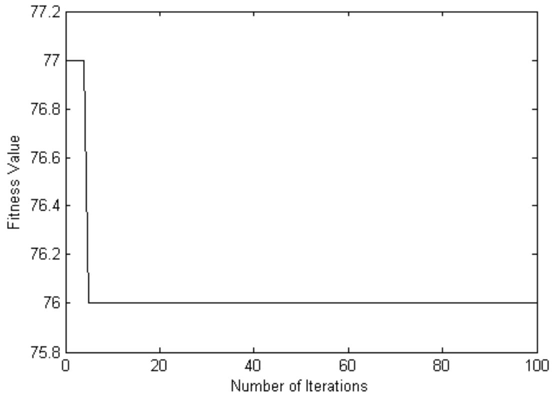


Fig. 4. Convergence characteristics of DPSO Algorithm

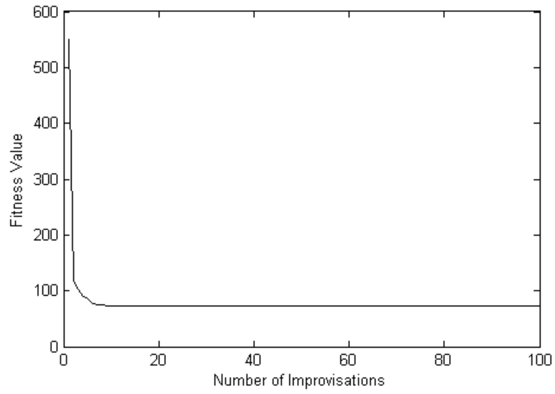


Fig. 5. Convergenve characteristics of HS Algorithm

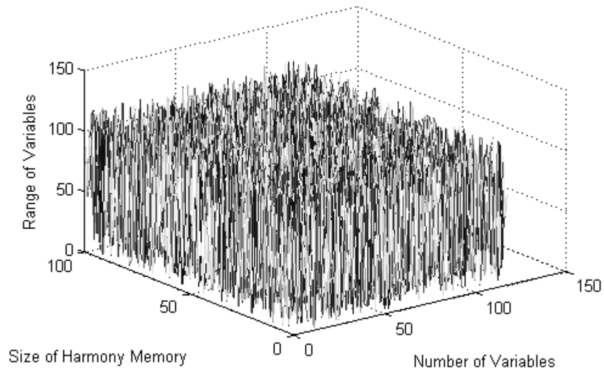


Fig. 6. Initial Harmony Memory of 118 bus system

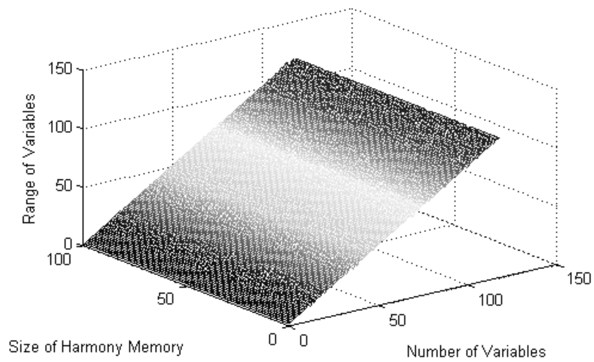
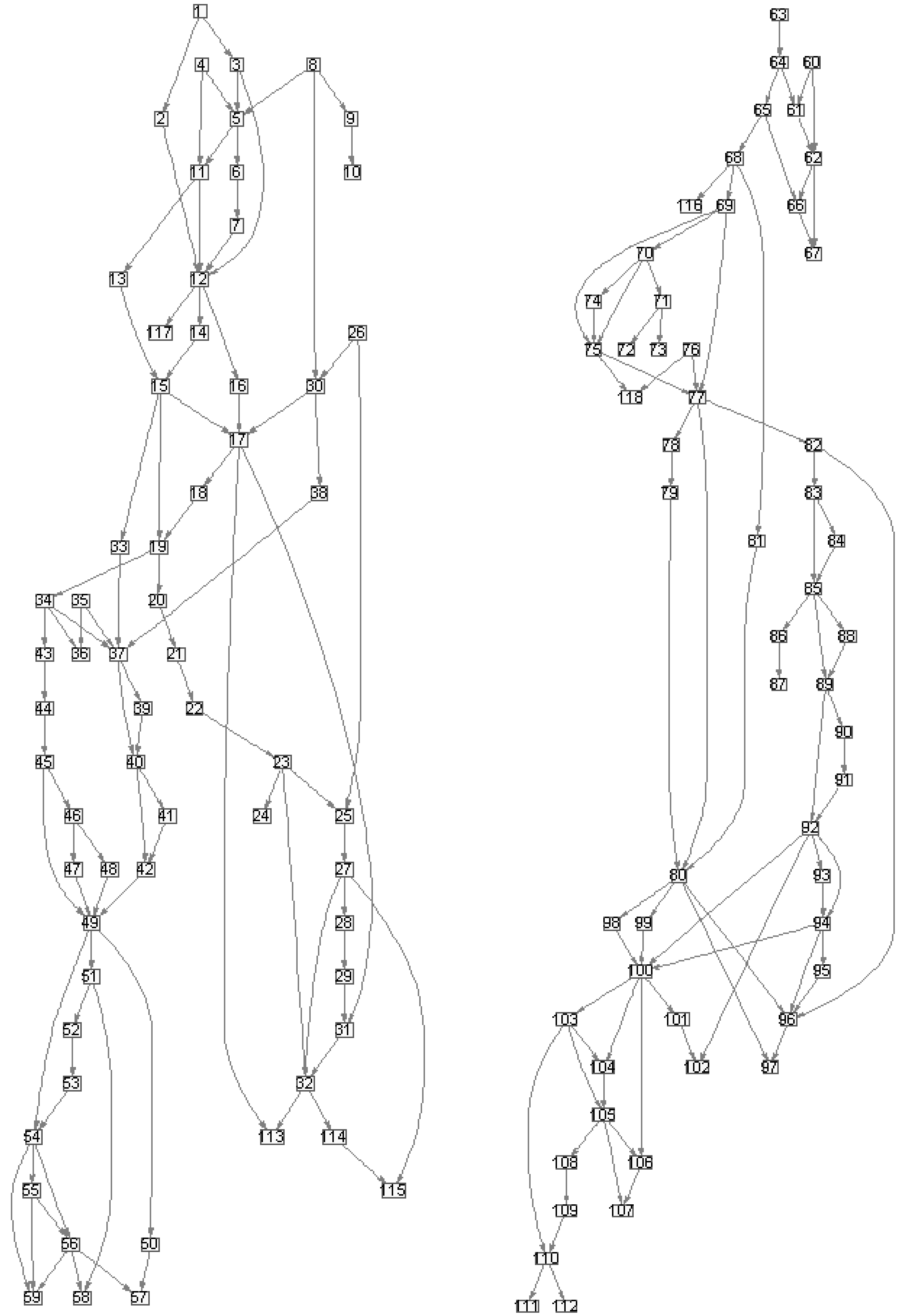


Fig. 7. Final Harmony Meomory of 118 bus system

The partitioned subnetworks of IEEE 118 bus system is shown Fig. 8 with compete connectivity.



(a) Cluster1 (b) Cluster2

Fig. 8. Decomposed network of IEEE 118 bus test system showing two clusters

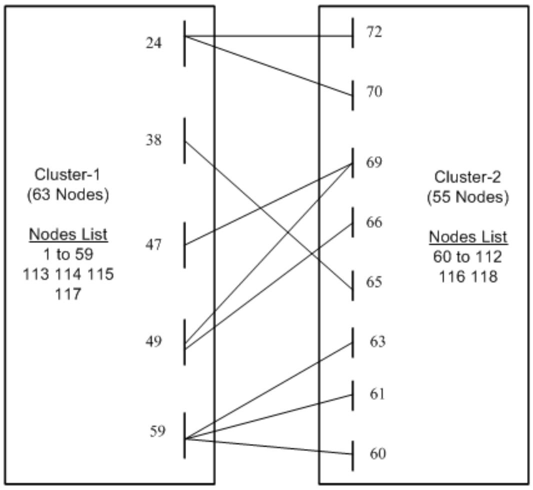


Fig. 9. IEEE 118 system with the partitioned clusters showing their inter partition connectivity by tie lines.

Typically the 118 bus system has 9 tie lines between the border buses as shown in Fig. 9, which on removal gives two independent clusters.

VII. Conclusion

This paper presents two new evolutionary methods for power systems network decomposition namely Discrete Particle Swarm Optimization (DPSO) and Harmony Search Algorithm (HSA). DPSO works on binary variables wheras the HSA uses real variables as the particles and harmony respectively. The algorithm was implemented in a high performance computing environment which supports distributed computing and parallel processing. The network partitioning is formulated as an optimization problem where the objective is to minimize the number of tie lines between the clusters and to attain a balance of nodes between the clusters. The connectivity of the clusters obtained is check by means od the AI search algorithms namely BFS and DFS. The proposed method has been tested on IEEE standard test cases and the case study on 118 Bus System is given in this paper. The simulation results clearly shows the implementation of the proposed methods to solve the network decomposition problem in a simple and efficient way. This partitioning can be adopted in a deregulated environment for distributed computing applications of a large interconnected power system network. The sub network aids parallel processing thereby improving speed of intensive power system computations and reduces the computational burden.

References

[1] Chang, CS and Lu, LR and Wen, FS. Power system network partitioning using tabu search. Electric power systems research, 49(1):55--61, 1999.
[2] Ding, H. and El-Keib, AA and Smith, R. Optimal clustering of power networks using genetic algorithms. Electric Power Systems Research, 30(3):209--214, 1994.
[3] M.R. Irving, BEng, PhD, CEng, MIEE, Prof. M.J.H. Sterling,, "Optimal network tearing using simulated annealing", 1EE Proceedings of Generation, Transmission and Distribution, Vol. 137, No. I, Jan 1990, 69 – 72.
[4] Mori, H. and Komatsu, Y. Power Network Decomposition with New Ant Colony Optimization. International Conference on Probabilistic Methods Applied to Power Systems, 2006. PMAPS 2006, pages 1--6, 2006.
[5] Zambroni, D.E.S. and others. New technique of network partitioning for voltage collapse margin calculations. IEE proceedings. Generation, transmission and distribution, 141(6):630--636, 1994.
[6] Mahdavi, M. and Chehreghani, M.H. and Abolhassani, H. and Forsati, R. Novel meta-heuristic algorithms for clustering web documents. Applied Mathematics and Computation, 201(1-2):441--451, 2008
[7] Zhongxu Li, Yutian Liu, Rushui Liu and Xinsheng Niu. Network Partition for Distributed Reactive Power Optimization in Power Systems. IEEE International Conference on Networking, Sensing and Control, pages 6--8, 2008.
[8] Jong - Bae Park, Member, IEEE, Ki-Song Lee, Joong-Rin Shin, and Kwang Y. Lee, Fellow, IEEE, "A Particle Swarm Optimization for Economic Dispatch With Nonsmooth Cost Functions", IEEE Transactions on Power Systems, Vol. 20, No. 1, February 2005
[9] Qian-Li Zhang, Xing Li, Quang-Ahn Tran, "A Modified Particle Swarm Optimization Algorithm", Proceedings Of The Fourth International Conference On Machine Learning And Cybernetics, Guangzhou, 18-21 August 2005
[10] Li-Yeh Chuanga, Hsueh-Wei Changb, Chung-Jui Tu C, Cheng-Hong Yang C, "Improved binary PSO for feature selection using gene expression data", Computational Biology and Chemistry, Elsevier, Vol 32, 2008, pp. 29-38.
[11] X.H. Shi', X.L. Xing, Q.X. Wang, L.H. Zhang, X.W. Yang, C.G. Zhou, Y.C. Liang, "A Discrete PSO Method For Generalized TSP Problem", Proceedings of the Third International Conference on Machine Learning and Cybernetics, Shanghai, 26-29 August 2004
[12] H. Shayeghi, M. Mahdavi, A. Kazemi, "Discrete Particle Swarm Optimization Algorithm Used for TNEP Considering Network Adequacy Restriction", International Journal of Electrical, Computer, and Systems Engineering, 3:1, 2009
[13] P. Kanakasabapathy and K. Shanti Swarup, "Optimal Bidding Strategy for Multi-unit Pumped Storage Plant in Pool-Based Electricity Market Using Evolutionary Tristate PSO", IEEE International Conference on Sustainable Energy Technologies, ICSET 2008, 24-27 Nov. 2008, 95 – 100.
[14] Lee, K.S. and Geem, Z.W. A new structural optimization method based on the harmony search algorithm. Computers & Structures, 82(9-10):781--798, 2004.
[15] Coelho, L.S. and Mariani, V.C. An improved harmony search algorithm for power economic load dispatch. Energy Conversion and Management, 50(10):2522--2526, 2009.
[16] Verma, A. and Panigrahi, BK and Bijwe, PR. Harmony search algorithm for transmission network expansion planning. IET Generation Transmission and Distribution, 4(6):663-673, 2010.
[17] Zou, Dexuan and Gao, Liqun and Li, Steven and Wu, Jianhua and Wang, Xin. A novel global harmony search algorithm for task assignment problem. J. Syst. Softw., 83(10):1678--1688, 2010.
[18] Wang, C.M. and Huang, Y.F. Self-adaptive harmony search algorithm for optimization. Expert Systems with Applications, 37(4):2826--2837, 2010.
[19] Geem, Z.W. Novel derivative of harmony search algorithm for discrete design variables. Applied mathematics and computation, 199(1):223--230, 2008.
[20] Fesanghary, M. and Mahdavi, M. and Minary-Jolandan, M. and Alizadeh, Y. Hybridizing harmony search algorithm with sequential quadratic programming for engineering optimization problems. Computer methods in applied mechanics and engineering, 197(33-40):3080--3091, 2008.

G. Angeline Ezhilarasi received her bachelor's degree in electrical engineering from Madras University and master's degree in power systems engineering from Anna University, Chennai, INDIA. She has worked as a Project Associate at IIT Madras, India for about 6 years and was involved in the content management and administration team of National Project on Technology Enhanced Learning. Currently she is a Research Scholar at IIT Madras with Distributed Computing and Parallel Processing as her major interest of research.

K. Shanti Swarup received the Ph.D. degree in electrical engineering from the Indian Institute of Science, Bangalore, India. He is currently a Professor in the Department of Electrical Engineering, Indian Institute of Technology Madras, Chennai, India. He was an Associate Professor with Kitami Institute of Technology, Hokkaido, Japan. He was a Research Scientist with the R&D Department, Mitsubishi Electric Corporation, Osaka, Japan. His current research interests include restructuring and deregulation of power systems, power system automation, energy management systems and (SCADA), object modeling, and design of electric power systems.