

## Comaprison of Computational Requirements for Spectral and Kernel k-means Bisectioning of Power Systems

Ali Peiravi and Rahim Ildarabadi

Department of Electrical Engineering, School of Engineering Ferdowsi University of Mashhad

---

**Abstract:** The most probable optimum solution in the controlled islanding of power systems is when it is torn into two pieces since such an approach would lead to larger islands when compared with multisection tearing of the power system. Moreover, the restoration of such a bisectioned power system would be easier than the restoration of a power system torn into many pieces. Spectral and Multilevel Kernel k-means approaches to the islanding of power systems have previously been reported by the authors. In this paper, a review of the potential applications of graph partitioning is presented and the classical and the newer spectral and multilevel k-means approaches to solving these problems for power system islanding are described. The computational burden of each approach is presented for a large power system abstracted as a graph  $G(N,B)$ , with an example of the IEEE 9 bus bisectioned and the computational time requirements of actual bisectioning of the power system on the IEEE 118 bus, the IEEE 300 bus and the 2746 bus Polish power system are compared with previously reported computational requirements. The results presented may also be used in many other fields such as VLSI design, parallel processing, circuit and layout design, transient stability studies, optimal load flow, fault estimation, and dynamic security assessment.

**Key words:** Power System, Bisectioning, Fiedler vector, spectral partitioning, multilevel kernel k-means

---

### INTRODUCTION

Graph partitioning methods have been widely published in the literature and applied to many modern day very large scale system problems such as parallel processing, sparsity preserving orderings for sparse matrix factorizations, circuit placement, routing, system hierarchy, VLSI circuit testing, facility location, scattered network, hierarchical design of VLSI circuits, data mining, dynamic load balancing, parallel test pattern generation, power system islanding, power system fault section estimation in large scale power systems, annotation of protein sequence, etc.

Although many names are used for the algorithms used in each application, the most common grouping of such applications is spectral and kernel approaches. Application of spectral methods are reported in many areas such as VLSI, FPGA, multi-Chip modules, integrated circuits, macro cells and real-life pattern recognition gene network analysis, social network analysis, and image segmentation.

Recently several researchers investigated graph partitioning algorithms that reduce the size of the graph by collapsing nodes and branches, partition the smaller graph, and then uncoarsen it to construct a partition for the original graph. The effectiveness of many different choices for all three phases: coarsening, partition of the coarsest graph, and refinement were investigated by Karypis and Kumar (1998) who presented a new coarsening heuristic called heavy-edge heuristic. In this heuristic the size of the partition of the coarse graph is within a small factor of final partition. They also presented a much faster variation of the Kernighan-Lin algorithm for refining during uncoarsening. They tested the proposed scheme on graphs arising from finite element, linear programming, VLSI and transportation problems.

A graph partitioning method based on a Multilevel Kernel k-Means approach with a high speed performance in partitioning graphs with application on large-scale partitioning tasks such as image

---

**Corresponding Author:** Ali Peiravi, Associate Professor of Electrical Engineering Department of Electrical Engineering Faculty of Engineering Ferdowsi University of Mashhad Mashhad - IRAN  
Postal Code: 9177948974  
Phone: 0098-511-881-5100 Extension 653 Fax: 0098-511-8763302  
E-mail: Ali\_peiravi@yahoo.com

segmentation; social network analysis; and gene network analysis types of systems was reported by Dhillon *et al.* (2005) and Dhillon *et al.* (2007). It was shown that a general weighted kernel k-means objective function is mathematically equivalent to a weighted graph partitioning objective.

In a survey of kernel and spectral methods for clustering by Filippone *et al.* (2008) it was pointed out that although these methods are tested on several benchmarks, few applications to real world problems are found for them due to the high computational cost and that extensive validation of kernel and spectral clustering techniques on real world problems still remain as a big challenge

#### **Review of Reported Applications:**

Several researchers have reported application of spectral and kernel graph partitioning methods in various fields. In this section a brief review of reported applications of these techniques to real life problems is presented. Cherng *et al.* (1999) presented a two-level bisectioning algorithm combining a hybrid clustering technique with an iterative partitioning process for VLSI circuits. Later on, Cherng *et al.* (2003) presented a multi-level bisectioning algorithm by integrating a clustering technique and an iterative improvement based partitioning process for VLSI circuit design in order to minimize the number of interconnects between the subsets of the circuit in order to reduce interconnect delays in deep submicron technology. For application to partitioning internet-like topologies used in the field of large scale network simulation, a genetic algorithm called BC-GA was proposed by Lin *et al.* (2008) based on boundary crossing inspired by the analysis of characteristic of the Internet topology and its related solutions and tested on a large extent of graphs including snapshots of the real AS-level Internet, the topologies produced by the Internet model generator and many traditional benchmark graphs.

Another widely used area of application is power systems where graph partitioning could be used for many different purposes. The usual problem is to separate the busses of a system into two or more groups to satisfy a specified goal. For example, a group of generators is said to be coherent when they have identical dynamic response to events originating outside that group. Minimal cutset technique is used by Wang and Vittal (2004) to deal with islanding the actual system based on the grouping information. Wang *et al.* (2008) proposed an adaptive clustering algorithm based on power system network topology, initial power flow and given architecture to address power system transient stability studies. The sizes of the small cliques are derived using multi-constraint and multi-objective graph partitioning theory where the nodes represent units of computation, and the branches encode data dependencies.

#### **Graph Partitioning:**

Graph partitioning is a well known NP-complete problem in mathematics where a graph is divided into several pieces in such a way that the pieces are of about the same size with few connections between them. In a more general form, the weighted graph partitioning problem where both nodes and branches may be weighted, the problem may be stated as follows:

Given a graph  $G(N,B)$  with  $N$  nodes and  $B$  branches, and given an integer  $k > 1$ , partition the graph into  $k$  disjoint subsets of approximately equal weight such that the size of the branch cuts is minimized. The size of a cut is the sum of the weights of the branches contained in it, while the weight of a subset is the sum of the weights of the nodes in that subset. This partitioning problem may be solved by using graph-theoretic heuristics.

In practical applications, nodes and branches of the graphs to be partitioned represent different objects and this must be taken into consideration when developing the graph partitioning algorithm. In such cases, it would be wise to consider graphs with weights and costs assigned to the elements as suggested by Aleksadrov *et al.* (2006) who presented an algorithms for computing cutsets in planar graphs with costs and weights on the nodes, where weights are used to estimate the sizes of the partitions and costs are used to estimate the size of the cutset. They measured the quality of the partitioning by the total cost of the elements in the cutset and the imbalance between the total weights of the parts.

#### **Spectral Bisectioning:**

In this section, the background of the spectral partitioning method is presented and then its performance is presented by way of examples. The fundamental principles of Laplacian spectrum of graphs were initially introduced by Mohar (1991). In the spectral graph partitioning approach the eigenvalues of the Laplacian matrix of the graph are used for partitioning the graph as shown by Hagen *et al.* (1992).

In bisectioning applications, the graphs may be partitioned into two parts using the second smallest eigenvalues of the Laplacian matrix. The corresponding eigenvector is computed based on which the partitions are found.

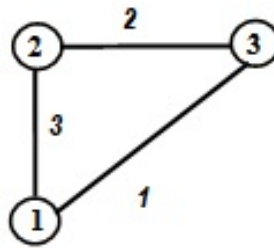
A graph's adjacency matrix and the degree matrix are needed to form the Laplacian matrix. The Laplacian matrix of an undirected, unweighted graph  $G(N,B)$  where the graph is without any self loops or multiple branches between any pair of nodes is an  $n$  by  $n$  symmetric matrix with one row and column for each node defined by

$$Q = D - A$$

where the degree matrix  $D = \text{diag}(d_1, d_2, \dots, d_n)$  well-known adjacency matrix. The

Laplacian matrix is symmetric and positive semidefinite, and it may be extended to weighted graphs where the weight of the branch is used in the adjacency matrix. If the eigenvalues of the Laplacian of a graph are sorted by increasing value, the eigenvector corresponding to the second smallest eigenvalue of the Laplacian matrix is called the Fiedler vector may be used in heuristics for various graph manipulations including spectral graph partitioning. The second smallest eigenvalue of the Laplacian matrix is greater than 0 if and only if  $G(N,B)$  is a connected graph.

Here, a simple example is used to illustrate the effort required in bisectioning a graph using the graph Laplacian. Consider the simple weighted graph shown in Figure 1.



**Fig. 1:** The graph with 3 nodes and 3 branches

The Laplacian matrix  $Q$  for this graph is

$$Q = \begin{bmatrix} 4 & -3 & -1 \\ -3 & 5 & -2 \\ -1 & -2 & 3 \end{bmatrix}$$

Then the eigenvalues should be computed as shown below:

$$\det(Q - I\lambda) = \begin{vmatrix} 4 - \lambda & -3 & -1 \\ -3 & 5 - \lambda & -2 \\ -1 & -2 & 3 - \lambda \end{vmatrix} = 0$$

$$\Rightarrow (4 - \lambda)((5 - \lambda)(3 - \lambda) - 4) + (-1)(-3)((-3)(3 - \lambda) - (-1)(-2)) + (-1)((-3)(-2) - (-1)(5 - \lambda)) = 0$$

$$\Rightarrow -\lambda^3 + 12\lambda^2 - 33\lambda = 0$$

$$\Rightarrow \lambda = 6 \pm \sqrt{36 - 33} = 6 \pm 1.7321$$

$$\Rightarrow \lambda_1 = 0, \lambda_2 = 4.2679, \lambda_3 = 7.7321$$

Next the eigenvectors are computed as follows:

$$\Rightarrow AV_i = \lambda_i V_i \quad i=1,2,3 \quad , \quad V_i = \begin{bmatrix} V_{1i} \\ V_{2i} \\ V_{3i} \end{bmatrix}$$

The first eigenvector  $V_1$ :

$$\Rightarrow \begin{bmatrix} 4 & -3 & -1 \\ -3 & 5 & -2 \\ -1 & -2 & 3 \end{bmatrix} \begin{bmatrix} V_{11} \\ V_{21} \\ V_{31} \end{bmatrix} = 0 \quad \begin{bmatrix} V_{11} \\ V_{21} \\ V_{31} \end{bmatrix} \Rightarrow \begin{cases} 4V_{11} - 3V_{21} - V_{31} = 0 \\ -3V_{11} + 5V_{21} - 2V_{31} = 0 \\ -V_{11} - 2V_{21} + 3V_{31} = 0 \end{cases}$$

$$\Rightarrow V_{31} = 4V_{11} - 3V_{21} \Rightarrow -11V_{11} + 11V_{21} = 0 \Rightarrow V_{11} = V_{21} \Rightarrow V_{11} = V_{21} = V_{31}$$

suppose:  $V_{11} = 1$  then:

$$\Rightarrow V_{21} = V_{31} = 1 \Rightarrow V_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

↗

$$\hat{V}_1 = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.5774 \\ 0.5774 \\ 0.5774 \end{bmatrix}$$

Normalized

The second eigenvector  $V_2$ :

$$\begin{bmatrix} 4 & -3 & -1 \\ -3 & 5 & -2 \\ -1 & -2 & 3 \end{bmatrix} \begin{bmatrix} V_{12} \\ V_{22} \\ V_{32} \end{bmatrix} = 4.2679 \begin{bmatrix} V_{12} \\ V_{22} \\ V_{32} \end{bmatrix} \Rightarrow \begin{cases} 4V_{12} - 3V_{22} - V_{32} = 4.2679 V_{12} \\ -3V_{12} + 5V_{22} - 2V_{32} = 4.2679 V_{22} \\ -V_{12} - 2V_{22} + 3V_{32} = 4.2679 V_{32} \end{cases}$$

$$\Rightarrow \hat{V}_2 = \begin{bmatrix} -0.5774 \\ -0.2113 \\ 0.7887 \end{bmatrix}$$

↘

Normalized

The third eigenvector  $V_3$ :

$$\begin{bmatrix} 4 & -3 & -1 \\ -3 & 5 & -2 \\ -1 & -2 & 3 \end{bmatrix} \begin{bmatrix} V_{13} \\ V_{23} \\ V_{33} \end{bmatrix} = 7.7321 \begin{bmatrix} V_{13} \\ V_{23} \\ V_{33} \end{bmatrix} \Rightarrow \begin{cases} 4V_{13} - 3V_{23} - V_{33} = 7.7321V_{13} \\ -3V_{13} + 5V_{23} - 2V_{33} = 7.7321V_{23} \\ -V_{13} - 2V_{23} + 3V_{33} = 7.7321V_{33} \end{cases}$$

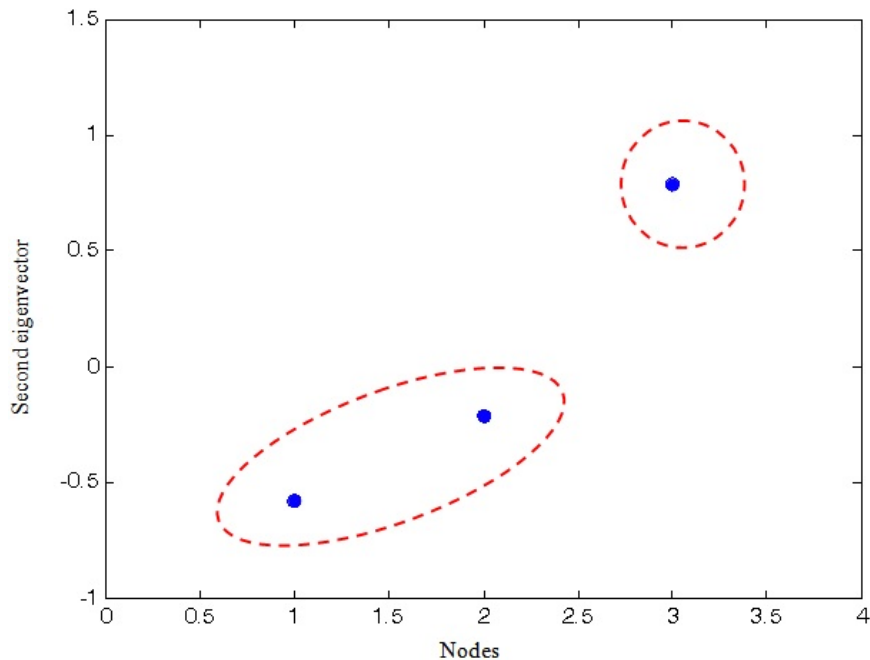
$$\Rightarrow \hat{V}_3 = \begin{bmatrix} -0.5774 \\ 0.7887 \\ -0.2113 \end{bmatrix}$$

Normalized

Therefore, the eigenvector matrix would be as follows:

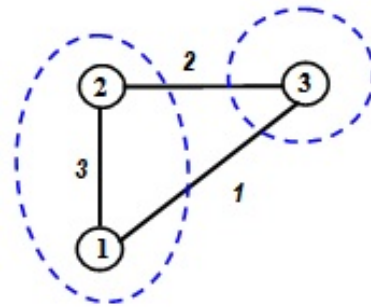
$$\Rightarrow V = \begin{bmatrix} 0.5774 & -0.5774 & -0.5774 \\ 0.5774 & -0.2113 & 0.7887 \\ 0.5774 & 0.7887 & -0.2113 \end{bmatrix}$$

The second column of this matrix, or the Fiedler vector, may be used to bisection the graph as shown in Figure 2.



**Fig. 2:** The sketch showing how the graph may be partitioned using the Fiedler vector

Therefore, the bisectioned graph is shown in Figure 3.



**Fig. 3:** The bisectioned graph

The mathematical effort required to compute the eigenvalues and the eigenvectors is a major problem in large power systems with hundreds or even thousands of busses. As can be seen from the above simple example, the bottleneck of the bisectioning algorithm presented above lies in the eigenvector calculation. The time requirement of this approach is  $O(n^3)$ . Notice that since only the sign of each component of the Fiedler vector is needed in order to partition the graph, an exact answer is not really required. This could be potentially useful in finding a faster solution.

**Kernighan-Lin Bisectioning:**

Kernighan and Lin (1970) proposed an iterative balanced minimum cutset bisectioning heuristic that starts with an initial bisection. Then, in each iteration, a subset of nodes from each partition is sought whose swapping would lead to bisections with a smaller branch cut set. This heuristic can yield local optimum partitions if it starts with a good choice of initial bisection. You may start with a random selection of bisection, too. The pseudocode for a single pass of the Kernighan-Lin algorithm is shown in Figure 4.

**Algorithm Kernighan-Lin-Single-Pass**

```

Repeat
  Start with partition int_partition:
  Best_partition:=init_partition
Repeat
  Select N1'⊂N1 and N2'⊂N2,
    Where |N1'|=|N2'|=k≥1, the gain of swapping N1' and N2' is maximum,
    And v∈(N1'∪N2') ⇒ v is unlocked
  Swap these subsets of nodes and lock them
  If(this partition is better than best_partition) then
    Best_partition:=this partition
  Endif
Until the solution does not improve
end algorithm
    
```

**Fig. 4:** A single pass of the Kernighan-Lin Algorithm

Multilevel versions of the Kernighan-Lin algorithm have been used for partitioning large graphs. In these algorithms, the graph is coarsened until it becomes so small that the processes for the problem at hand may be applied fast. Then the partitions are aggregated. Multilevel versions of the spectral method which are based on applying the spectral method at various levels have also been successfully used. In these methods, it is required to compute the Fiedler vector. Holzrichter and Oliveira (1999) proposed a purely spectral approach in which the calculation of the Fiedler vector is done using the Davidson algorithm. The problem at hand is to be setup in the form of a graphical pre-conditioner to the Davidson algorithm.

Spectral algorithms are usually based on the Fiedler vector of the Laplacian. Determining the Fiedler vector

of the Laplacian is the most computationally intensive part of graph partitioning. In many applications, an approximation of the Fiedler vector is used to speed up the solution.

**Application of the Kernighan-lin Bisectioning:**

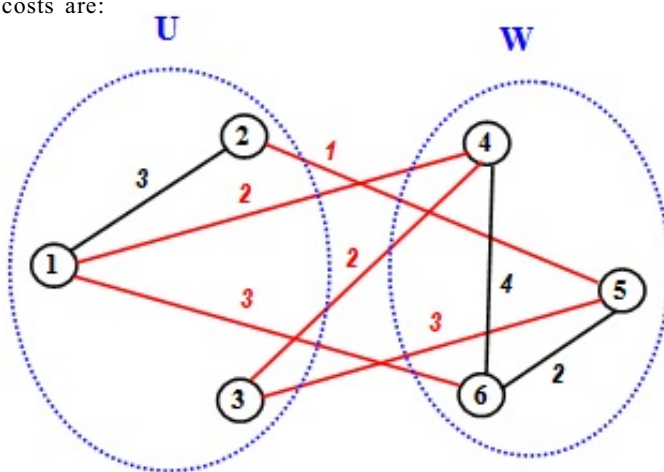
Consider the weighted graph G with 6 nodes and 8 branches initially bisectioned into two subgraphs U and W of the same size as shown in Figure 6. The cost matrix is as follows:

$$c = \begin{bmatrix} 0 & 3 & 0 & 2 & 0 & 3 \\ 3 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 & 3 & 0 \\ 2 & 0 & 2 & 0 & 0 & 4 \\ 0 & 1 & 3 & 0 & 0 & 2 \\ 3 & 0 & 0 & 4 & 2 & 0 \end{bmatrix}$$

The cost of each node is defined as the sum of the costs of the branches that are incident with it and incident with the nodes within the other partition, minus the sum of the branches that are incident with it and incident with nodes within the same partition, namely:

$$\text{for } i \in U \quad D_i = \sum_{j \in W} c_{ij} - \sum_{j \in U} c_{ij}$$

For example, costs for nodes in the example shown in Figure 5 are computed as follows. For nodes in partition U the node costs are:



**Fig. 5:** A graph with 6 nodes and 8 branches showing initial bisections

$$D_1 = \sum_{j \in W} c_{1j} - \sum_{j \in U} c_{1j} = (2 + 3) - (3) = 2$$

$$D_2 = \sum_{j \in W} c_{2j} - \sum_{j \in U} c_{2j} = (1) - (3) = -2$$

$$D_3 = \sum_{j \in W} c_{3j} - \sum_{j \in U} c_{3j} = (2 + 3) - (0) = 5$$

For nodes in partition W the node costs are:

$$D_4 = \sum_{i \in U} c_{4,i} - \sum_{j \in W} c_{4,j} = (2 + 2) - (4) = 0$$

$$D_5 = \sum_{i \in U} c_{5,i} - \sum_{j \in W} c_{5,j} = (1 + 3) - (2) = 2$$

$$D_6 = \sum_{i \in U} c_{6,i} - \sum_{j \in W} c_{6,j} = (3) - (4 + 2) = -3$$

It is observed that whenever the cost of two nodes the have been placed in different partitions are larger, it is better to swap them. Therefore, one may define an index such as cost reduction or gain "g", that is the sum of the costs of two nodes that should be swapped, namely:

$$g_{i,j} = g_{j,i} = D_i + D_j - 2c_{ij}; i \in U, j \in W$$

For the example above, the cost reduction or gain for all node are listed as follows:

$$g_{1,4} = g_{4,1} = D_1 + D_4 - 2c_{1,4} = 2 + 0 - 2 \times 2 = -2$$

$$g_{1,5} = g_{5,1} = D_1 + D_5 - 2c_{1,5} = 2 + 2 - 2 \times 0 = 4$$

$$g_{1,6} = g_{6,1} = D_1 + D_6 - 2c_{1,6} = 2 + (-3) - 2 \times 3 = -7$$

$$g_{2,4} = g_{4,2} = D_2 + D_4 - 2c_{2,4} = (-2) + 0 - 2 \times 0 = -2$$

$$g_{2,5} = g_{5,2} = D_2 + D_5 - 2c_{2,5} = (-2) + 2 - 2 \times 1 = -2$$

$$g_{2,6} = g_{6,2} = D_2 + D_6 - 2c_{2,6} = (-2) + (-3) - 2 \times 0 = -5$$

$$g_{3,4} = g_{4,3} = D_3 + D_4 - 2c_{3,4} = 5 + 0 - 2 \times 2 = 1$$

$$g_{3,5} = g_{5,3} = D_3 + D_5 - 2c_{3,5} = 5 + 2 - 2 \times 3 = 1$$

$$g_{3,6} = g_{6,3} = D_3 + D_6 - 2c_{3,6} = 5 + (-3) - 2 \times 0 = 2$$

One may conclude from these cost reductions that nodes 1 and 5 should be swapped since the cost reduction between these nodes is maximal that it called  $\hat{g}_1 = 4$ , so  $\hat{g}_1 = \max\{g_{i,j}\}_{i \in U, j \in W}$ . Next we repeat the procedure without considering nodes 1 and 5 as follows:



$$D_2 = (c_{2,1} + c_{2,4} + c_{2,6}) - (c_{2,3} + c_{2,5}) = (3) - (1) = 2$$

$$D_3 = (c_{3,1} + c_{3,4} + c_{3,6}) - (c_{3,2} + c_{3,5}) = 2 - 3 = -1$$

$$D_4 = (c_{4,2} + c_{4,5} + c_{4,3}) - (c_{4,1} + c_{4,4}) = 2 - (2 + 4) = -4$$

$$D_6 = (c_{6,2} + c_{6,5} + c_{6,3}) - (c_{6,1} + c_{6,4}) = 2 - (3 + 4) = -5$$

Therefore, one can write:

$$g_{2,4} = g_{4,2} = D_2 + D_4 - 2c_{2,4} = 2 + (-4) - 2 \times 0 = -2$$

$$g_{2,6} = g_{6,2} = D_2 + D_6 - 2c_{2,6} = 2 + (-5) + 2 \times 0 = -3$$

$$g_{3,4} = g_{4,3} = D_3 + D_4 - 2c_{3,4} = -1 + (-4) - 2 \times 2 = -9$$

$$g_{3,6} = g_{6,3} = D_3 + D_6 - 2c_{3,6} = -1 + (-5) - 2 \times 0 = -6$$

It is observed that cost reduction  $g_{2,4}$  is maximized, thus  $\hat{g}_2 = -2$  and these two nodes can be candidates for swapping. Now after swapping these nodes with each other, repeat this procedure without considering nodes 1,5 and nodes 2,4 nodes as follows:

$$D_6 = (c_{6,4} + c_{6,5} + c_{6,3}) - (c_{6,1} + c_{6,2}) = 6 - 3 = 3$$

$$D_3 = (c_{3,1} + c_{3,2} + c_{3,4}) - (c_{3,5} + c_{3,6}) = 0 - 5 = -5$$

Therefore, one can write:

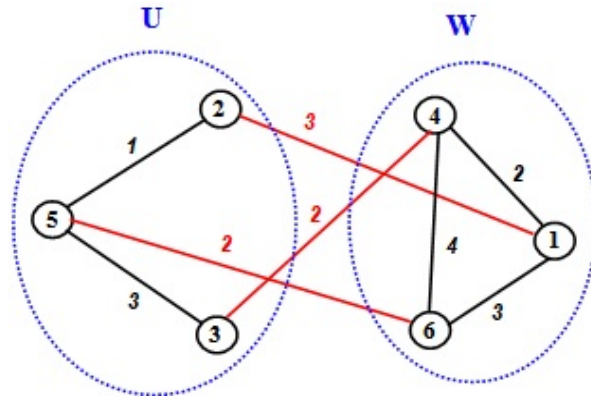
$$g_{3,6} = g_{6,3} = D_3 + D_6 - 2c_{3,6} = 3 + (-5) - 2 \times 0 = -2$$

Thus  $\hat{g}_3 = -2$ ,

Now the largest  $\sum_{i=1}^k \hat{g}_i$  can create the swapping with the largest cost reduction. For instance in this example we can write:

$$\left. \begin{aligned} k=1 &\Rightarrow \sum_{i=1}^k \hat{g}_i = \hat{g}_1 = 4 \\ k=2 &\Rightarrow \sum_{i=1}^k \hat{g}_i = \hat{g}_1 + \hat{g}_2 = 4 - 2 = 2 \\ k=3 &\Rightarrow \sum_{i=1}^k \hat{g}_i = \hat{g}_1 + \hat{g}_2 + \hat{g}_3 = 4 - 2 - 2 = 3 \end{aligned} \right\} \Rightarrow \max \sum_{i=1}^k \hat{g}_i \quad \text{occurs for } K=1$$

This means that nodes 1 and 5 must be chosen to be swapped and the changed partitions are shown in Figure 6.



**Fig. 6:** The bisectioned graph after swapping nodes 1 and 5.

Now repeat all of this procedure again until the  $\hat{g}_i$  become zero. For this example, we have:

$$D_1 = 3 - 5 = -2$$

$$D_2 = 3 - 1 = 2$$

$$D_3 = 2 - 3 = -1$$

$$D_4 = 2 - 6 = -4$$

$$D_5 = 2 - 4 = -2$$

$$D_6 = 2 - 7 = -5$$

Therefore, the cost reduction for all nodes are as follows:

$$g_{1,2} = g_{2,1} = D_1 + D_2 - 2c_{1,2} = -2 + 2 - 2 \times 3 = -6$$

$$g_{1,3} = g_{3,1} = D_1 + D_3 - 2c_{1,3} = -2 + (-1) - 2 \times 0 = -3$$

$$g_{1,5} = g_{5,1} = D_1 + D_5 - 2c_{1,5} = -2 + (-2) - 2 \times 0 = -4$$

$$g_{4,5} = g_{5,4} = D_4 + D_5 - 2c_{4,5} = -4 + (-2) - 2 \times 0 = -6$$

$$g_{4,3} = g_{3,4} = D_4 + D_3 - 2c_{3,4} = -4 + (-1) - 2 \times 2 = -9$$

$$g_{4,2} = g_{2,4} = D_4 + D_2 - 2c_{2,4} = -4 + (2) - 2 \times 0 = -2$$

$$g_{6,2} = g_{2,6} = D_6 + D_2 - 2c_{2,6} = -5 + (2) - 2 \times 0 = -3$$

$$g_{6,3} = g_{3,6} = D_6 + D_3 - 2c_{3,6} = -5 + (-1) - 2 \times 0 = -6$$

$$g_{6,5} = g_{5,6} = D_6 + D_5 - 2c_{5,6} = -5 + (-2) - 2 \times 2 = -11$$

Thus, the maximum cost reduction would be  $g_{4,2} = \hat{g}_1 = -2$  and nodes 4 and 2 can be candidates for swapping. Therefore, we repeat finding additional candidates for swapping without considering nodes 2 and 4 as follows:

$$D_1 = 2 - 6 = -4$$

$$D_3 = 0 - 5 = -5$$

$$D_5 = 3 - 3 = 0$$

$$D_6 = 6 - 3 = 3$$

The cost reductions in this iteration are as follows:

$$g_{6,5} = g_{5,6} = D_6 + D_5 - 2c_{5,6} = 3 + 0 - 2 \times 2 = -1$$

$$g_{6,3} = g_{3,6} = D_6 + D_3 - 2c_{3,6} = 3 + (-5) - 2 \times 0 = -2$$

$$g_{1,5} = g_{5,1} = D_1 + D_5 - 2c_{5,1} = -4 + 0 - 2 \times 0 = -4$$

$$g_{1,3} = g_{3,1} = D_1 + D_3 - 2c_{3,1} = -4 + (-5) - 2 \times 0 = -9$$

Therefore,  $g_{6,5} = \hat{g}_2 = -1$  indicating a swapping of nodes 6 and 5. And repeating the above procedure as follows we have:

$$D_1 = 5 - 3 = 2$$

$$D_3 = 3 - 2 = 1$$

Therefore, we can write

$$g_{1,3} = g_{3,1} = D_1 + D_3 - 2c_{1,3} = 2 + 1 - 0 = 3$$

Thus  $\hat{g}_{1,3} = \hat{g}_3 = 0 \Rightarrow \hat{g}_1 + \hat{g}_2 + \hat{g}_3 = -2 + -1 + 3 = 0$

Now the largest  $\sum_{i=1}^k \hat{g}_i$  can create the swapping with the largest cost reduction. For instance, in this example we can write

$$\left. \begin{aligned} k=1 &\Rightarrow \sum_{i=1}^k \hat{g}_i = \hat{g}_1 = -2 \\ k=2 &\Rightarrow \sum_{i=1}^k \hat{g}_i = \hat{g}_1 + \hat{g}_2 = -2 - 1 = -3 \\ k=3 &\Rightarrow \sum_{i=1}^k \hat{g}_i = \hat{g}_1 + \hat{g}_2 + \hat{g}_3 = -2 - 1 + 3 = 0 \end{aligned} \right\} \Rightarrow \max \sum_{i=1}^k \hat{g}_i \quad \text{occurs for } k=3$$

This means that nodes 2,3,5 must be exchanged with nodes 4,6,1 respectively. Since this would result in the previous bisection, it means that this is the best bisection with the least cut cost as shown in Figure 7. Figure 7 - The final bisection with the least cut cost

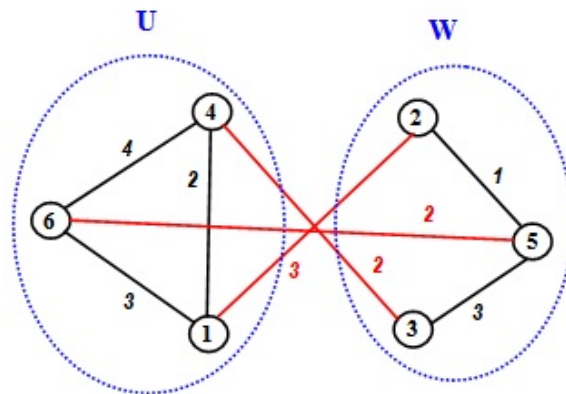


Fig. 7: The final bisection with the least cut cost

**Bisectioning of Power Systems as a Means of Controlled Islanding:**

Modern power systems experience a variety of stresses which may cause them to lose stability or lead to a catastrophic failure. In the absence of intelligent supervisory action, a power system may be driven into an emergency state which could either cause system collapse by natural islanding or total system blackout. One possible action in an emergency state is controlled power system islanding. Many methods have been proposed for this purpose. However these methods are slow, whereas islanding must be done rapidly for it to be an effective countermeasure.

As modern power systems are highly interconnected and are operating under stress due to deregulation and restructuring, a fault in a part of an interconnected power system may lead it towards instability or complete blackout as shown by Vittal *et al.* (2005). This would happen more often than in the past as reported by Venkatasubramanian *et al.* (2005), and Anderson *et al.* (2005) with the reliability situation even worse in the developing countries as indicated by Al-Odienat, (2006), or Sanaye-Pasand and Dadashzadeh, (2004) since the power systems there suffer from a larger gap between demand and generation; inadequate transmission capacity; and large distances between load centers and generation units.

Controlled power system islanding can improve restoration of the power system since it creates an equilibrium between load and generation. Several types of approaches have been presented in power systems in India without which catastrophic events could have caused serious power system blackouts as reported by Rajamani *et al.* (1999). The controlled islanding approach based on active and reactive power balancing control installed in the metropolitan area in Tokyo prevented a blackout in 1999 after an airplane ran into with a 275-kV transmission tie line as reported by Agematsu *et al.* (2001).

The spectral partitioning was applied to power system islanding by Li *et al.* (2005). Power system controlled islanding based on the multilevel kernel k-means was proposed for the intentional islanding of large scale power systems by Peiravi and Ildarabadi, (2009) who also made a comparison with the spectral partitioning. Many issues such as generation/load imbalance in each island to be formed, dynamic response of the system, load shedding, generation tripping which may occur if there is a severe generation inadequacy after the island is formed, transmission system capacity constraints, consideration of priority ranking of various loads, and computational efficiency of the approach should be considered. It is felt that fast bisectioning of the power system could result in the best solution to this problem.

**Spectral Bisectioning for Power System Islanding:**

Although minimal generation/load imbalance is a very important factor in islanding since it affects the amount of under-frequency load shedding required in the island after its formation, and makes it easier to restore the island, other issues should also be considered. Spectral bisectioning of the power system may be performed as follows as proposed by Rehtanz, (2003):

- 1- Compute the power flow of the power system network
- 2- Convert the power system network into a graph G(N,B) with n nodes and b branches.
  - 2-1 Each bus of the power network is a node of the graph G.
  - 2-2 Each transmission line of the power system is a branch of G.
  - S2-3 Assign the weight of each branch of G as the absolute value of real power flow of the corresponding transmission line.
- 3- Compute the Laplacian matrix Q of Graph G(N,B) as follows:
  - 3-1 Compute the adjacency matrix A and diagonal degree matrix D of graph G(N,B).
  - 3-2 The Laplacian matrix Q=D-A
- 4- Compute the second smallest eigenvalue  $\lambda_2$  of the Laplacian matrix Q.
- 5- Compute the real eigenvector associated with  $\lambda_2$  call it  $x_2$ ,
- 6- Map  $x_2$  into a heuristic partition vector of the graph G(N,B).
  - 6-1 Sort entries of  $x_2$  to obtain the sorted vector v of node indices.
  - 6-2 Place all nodes in partition U.
  - 6-3 For i=1 to n-1
    - Move  $v_i$  from partition U to partition W
    - Calculate the cut set size of the (U, W) partition.
- 7 Find the optimal  $(U^*, W^*)$  partition that has the minimum cut set size among the above n-1 different partitions.

The flowchart of the spectral power system bisectioning algorithm is shown in Figure 8.

An example of spectral power systems bisectioning is presented below to illustrate the effort required in bisectioning a power system using the graph Laplacian.

**Spectral Bisectioning the IEEE 9 Bus Test Power System:**

Consider the simple weighted graph shown in Figure 9 which shows the graph of the IEEE 9 bus power system with active power flow through the branches, and with the node numbers shown on the graph.

The Laplacian matrix for the above graph is as follows:

$$Q_{i,j} = \begin{cases} \sum_{h=1}^{14} e_{ih}; & i = j \\ -e_{ij} & ; i \neq j \end{cases}$$

where  $e_{ij}$  is weight of the branch that connects the two nodes  $v_i$  and  $v_j$  (note that  $e_{ii} = 0$ ). Therefore,

the Laplacian matrix is given as follows:

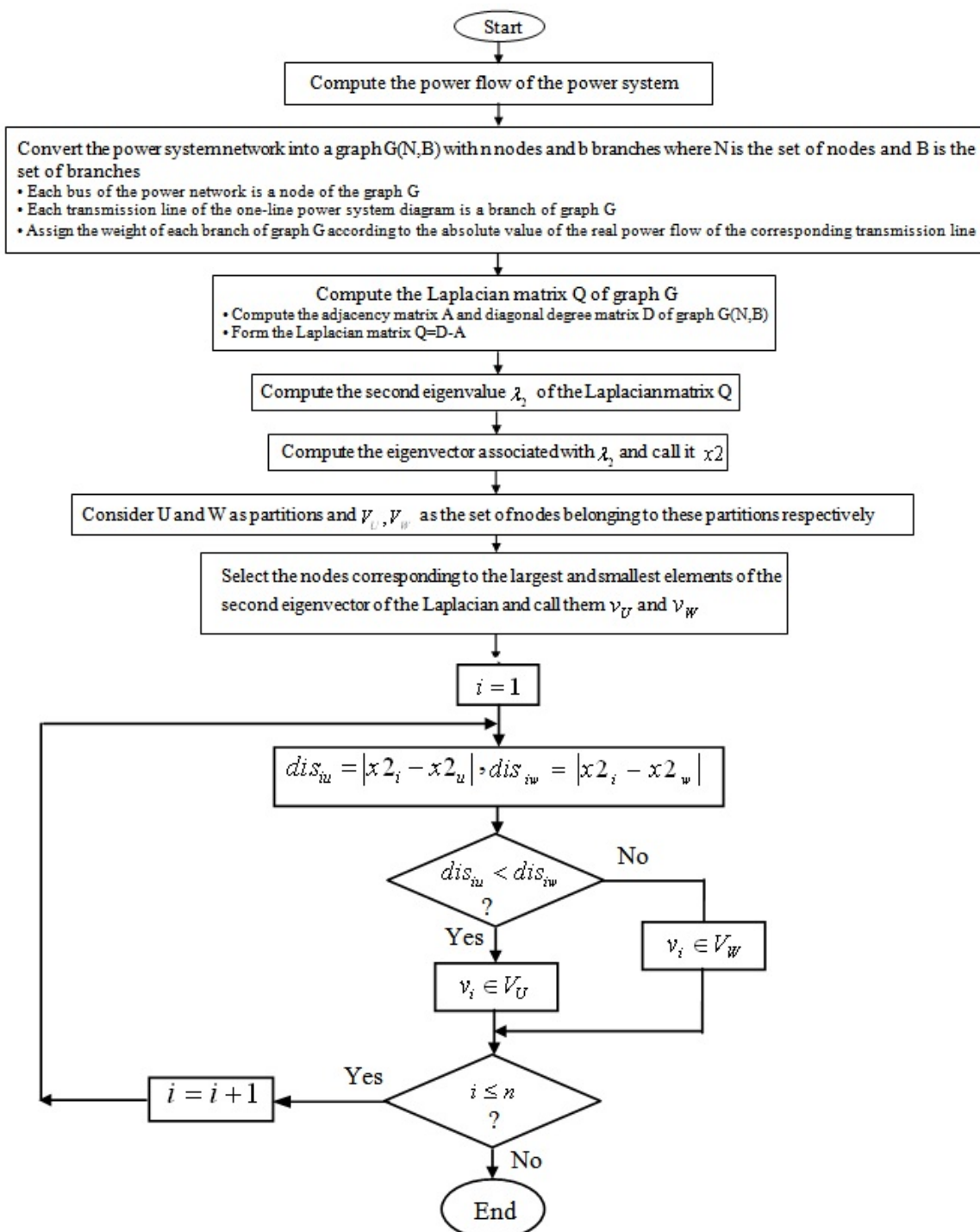


Fig. 8: The flowchart of spectral power system bisectioning algorithm

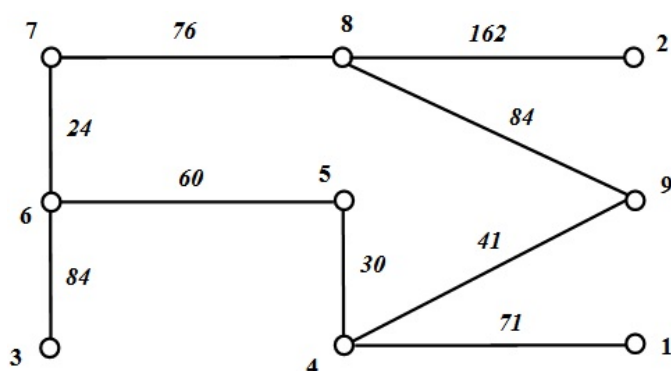


Fig. 9: The graph of the IEEE 9 bus power system with 9 nodes and 9 branches

$$Q = \begin{bmatrix} -71 & 0 & 0 & 71 & 0 & 0 & 0 & 0 & 0 \\ 0 & -162 & 0 & 0 & 0 & 0 & 0 & 162 & 0 \\ 0 & 0 & -84 & 0 & 0 & 84 & 0 & 0 & 0 \\ 71 & 0 & 0 & -142 & 30 & 0 & 0 & 0 & 41 \\ 0 & 0 & 0 & 30 & -90 & 60 & 0 & 0 & 0 \\ 0 & 0 & 84 & 0 & 60 & -168 & 24 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 24 & -100 & 76 & 0 \\ 0 & 162 & 0 & 0 & 0 & 0 & 76 & -322 & 84 \\ 0 & 0 & 0 & 41 & 0 & 0 & 0 & 84 & -125 \end{bmatrix}$$

The eigenvalues of Q are as follow:

$\lambda_1$	$\lambda_2$	$\lambda_3$	$\lambda_4$	$\lambda_5$	$\lambda_6$	$\lambda_7$	$\lambda_8$	$\lambda_9$
0	20	27	87.6	106	130	206	245	454

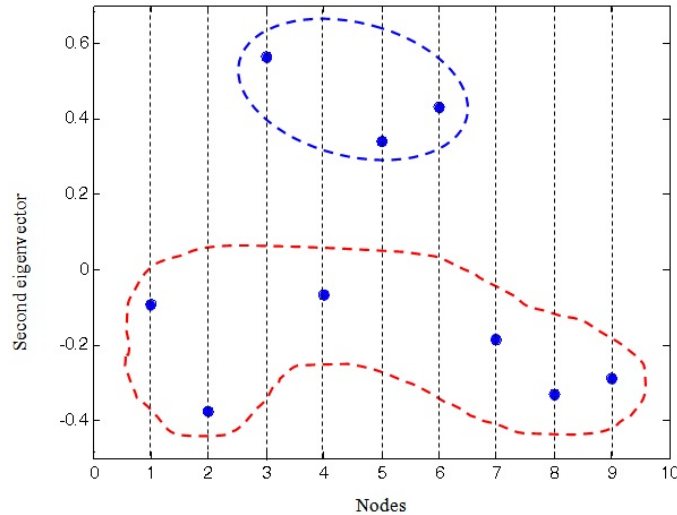
The eigenvector of the Laplacian matrix Q are computed and shown below.

$$V = \begin{bmatrix} 0.3333 & -0.0924 & 0.7039 & 0.3074 & 0.2259 & -0.244 & 0.4179 & 0.0713 & 0.0058 \\ 0.3333 & -0.3753 & -0.2926 & 0.0241 & -0.19 & -0.621 & -0.1298 & -0.0599 & 0.4688 \\ 0.3333 & 0.5657 & -0.2121 & 0.5342 & -0.1976 & 0.009 & -0.1371 & 0.425 & 0.0041 \\ 0.3333 & -0.0663 & 0.4432 & -0.0668 & -0.1057 & 0.1955 & -0.777 & -0.1718 & -0.0309 \\ 0.3333 & 0.34 & 0.0722 & -0.7817 & 0.0218 & -0.149 & 0.104 & 0.3514 & 0.0056 \\ 0.3333 & 0.4303 & -0.1456 & -0.0162 & 0.048 & -0.0047 & 0.1947 & -0.8011 & -0.0179 \\ 0.3333 & -0.1845 & -0.3009 & 0.0359 & 0.7727 & 0.338 & -0.0692 & 0.1169 & 0.1823 \\ 0.3333 & -0.3284 & -0.2448 & 0.0112 & -0.0669 & -0.1272 & 0.034 & 0.0302 & -0.8356 \\ 0.3333 & -0.2891 & -0.0233 & -0.0481 & -0.5082 & 0.6032 & 0.3624 & 0.0379 & 0.2178 \end{bmatrix}$$

The second eigenvector  $V_2$  or the Fiedler vector is

$$V_2^T = [-0.0924 \quad -0.3753 \quad 0.5657 \quad -0.0663 \quad 0.34 \quad 0.4303 \quad -0.1845 \quad -0.3284 \quad -0.2891]$$

The Fiedler vector may be used to bipartition the graph as shown in Figure 10.



**Fig. 10:** The sketch showing how the graph may be partitioned using the Fiedler vector

The number of elements of the second eigenvector corresponds to the number of nodes in the graph. Consider U and W as graph partitions and  $V_U, V_W$  as the set of nodes belonging to these partitions, respectively. Select the nodes that correspond to the largest and smallest values in the Fiedler vector and call them  $v_U$  and  $v_W$ . In this example,  $v_U = 2$ ,  $v_W = 3$ , and  $V_2(v_U) = -0.3753$ ,

$V_2(v_W) = 0.5657$ . Then node 2 is the first element of the U partition and node 3 is the first element

of the W partition. The rest of the nodes will be checked as follows to see which partition they belong to. Thus, every other node is checked and the absolute value of the difference between the value of the corresponding element in the eigenvector and that of  $V_2(v_U) = -0.3753$  and  $V_2(v_W) = 0.5657$  is checked. If the value that is relative to  $V_2(v_U) = -0.3753$  is less than that relative to  $V_2(v_W) = 0.5657$ ,

then that node belongs to the U partition. Otherwise it belongs to the W partition. Therefore, for this example,

$$\left. \begin{aligned} D_{1,2} &= |V_2(1) - V_2(2)| = |-0.0924 - (-0.3753)| = 0.2829 \\ D_{1,3} &= |V_2(1) - V_2(3)| = |-0.0924 - (0.5657)| = 0.6581 \end{aligned} \right\} \Rightarrow \text{node}(1) \in U$$

and

$$\left. \begin{aligned} D_{4,2} &= |V_2(4) - V_2(2)| = |-0.0663 - (-0.3753)| = 0.3090 \\ D_{4,3} &= |V_2(4) - V_2(3)| = |-0.0663 - (0.5657)| = 0.632 \end{aligned} \right\} \Rightarrow \text{node}(4) \in U$$



and

$$\left. \begin{aligned} D_{5,2} &= |V_2(5) - V_2(2)| = |0.34 - (-0.3753)| = 0.7153 \\ D_{5,3} &= |V_2(5) - V_2(3)| = |0.34 - (0.5657)| = 0.2257 \end{aligned} \right\} \Rightarrow \text{node}(5) \in W$$

and

$$\left. \begin{aligned} D_{4,2} &= |V_2(6) - V_2(2)| = |0.4303 - (-0.3753)| = 0.8056 \\ D_{4,3} &= |V_2(6) - V_2(3)| = |0.4303 - (0.5657)| = 0.1354 \end{aligned} \right\} \Rightarrow \text{node}(6) \in W$$

and

$$\left. \begin{aligned} D_{7,2} &= |V_2(7) - V_2(2)| = |-0.1845 - (-0.3753)| = 0.1908 \\ D_{7,3} &= |V_2(7) - V_2(3)| = |-0.1845 - (0.5657)| = 0.7502 \end{aligned} \right\} \Rightarrow \text{node}(7) \in U$$

And

$$\left. \begin{aligned} D_{8,2} &= |V_2(8) - V_2(2)| = |-0.3284 - (-0.3753)| = 0.0469 \\ D_{8,3} &= |V_2(8) - V_2(3)| = |-0.3284 - (0.5657)| = 0.8941 \end{aligned} \right\} \Rightarrow \text{node}(8) \in U$$

And

$$\left. \begin{aligned} D_{9,2} &= |V_2(9) - V_2(2)| = |-0.2891 - (-0.3753)| = 0.0862 \\ D_{9,3} &= |V_2(9) - V_2(3)| = |-0.2891 - (0.5657)| = 0.8548 \end{aligned} \right\} \Rightarrow \text{node}(9) \in U$$

Thus, the original graph is partitioned into two parts as follows:

Partition 1: {3,5,6}

Partition 2: {1,2,4,7,8,9}

Figure 11 shows the bisectioned form of the IEEE9 bus power system based on the above calculations with the minimum cut set cost.

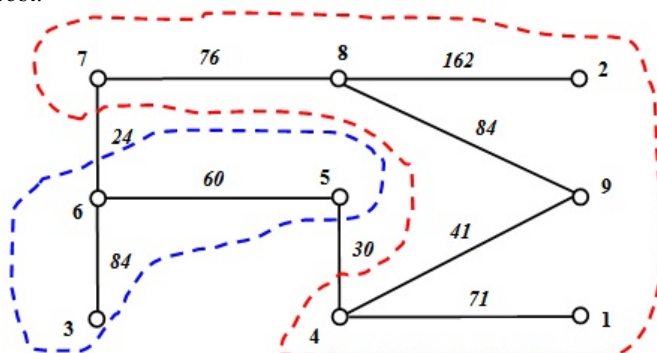


Fig. 13: The bisectioned IEEE 9 bus power system with minimum branch cutest cost

**Power System Partitioning Based on Multilevel Kernel k-Means:**

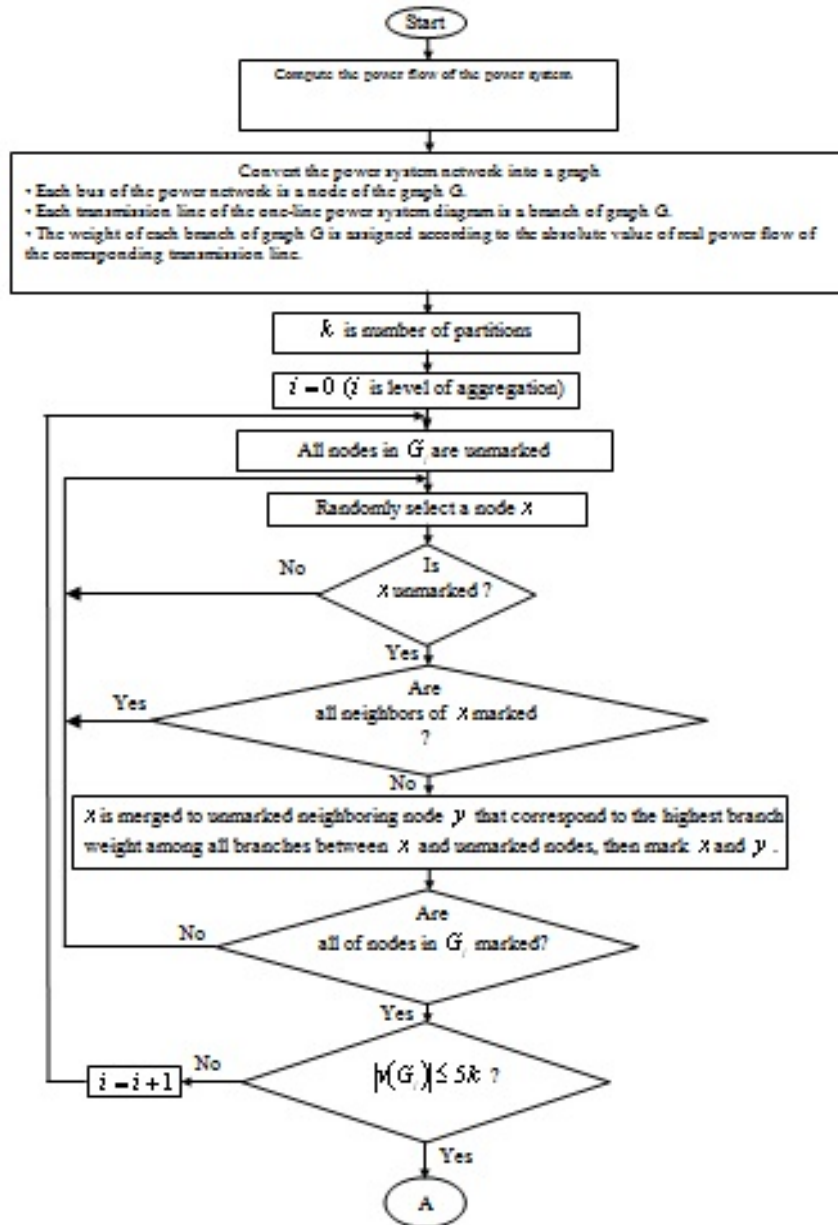
Spectral partitioning and kernel k-means are two seemingly different methods for partitioning graphs. However, they are very similar mathematically and this helps us design a fast kernel-based multilevel graph partitioning algorithm that is better in terms of speed, memory storage and quality as shown by Dhillon *et al.* (2007)]. Peiravi and Ildarabadi (2009) proposed to apply the Multilevel Kernel k-Means method to the intentional power system islanding problem. Since in this approach there is no need to calculate the eigenvalues and eigenvectors, it is much faster than the spectral method.

**Bisectioning of the Power System using Multilevel Kernel k-Means:**

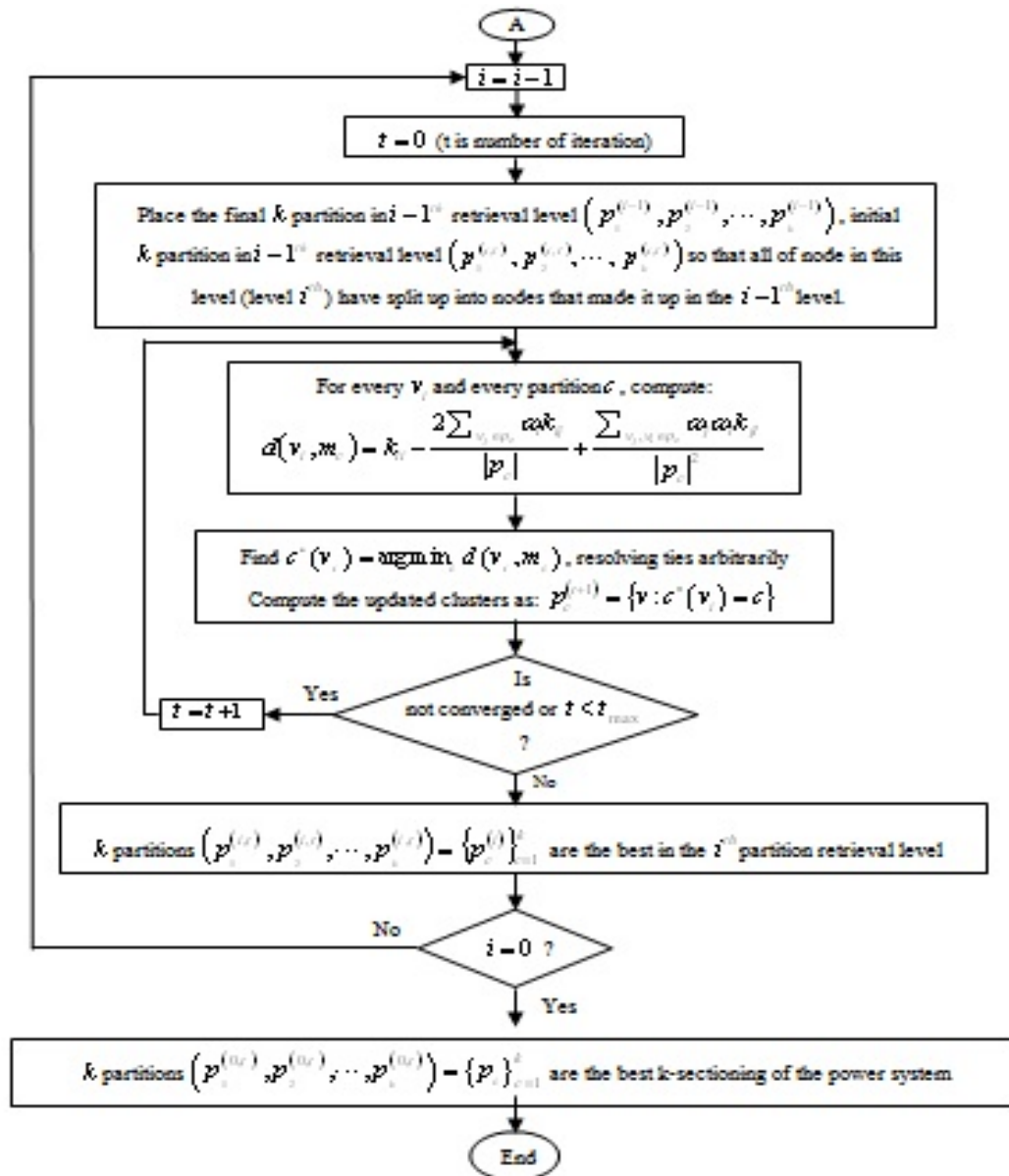
Since a good reduction in computational time can be achieved using the Multilevel Kernel k-Means method, it is applied to the bisectioning of power systems. To do so, the power system must first be abstracted into a graph as follows:

- Compute the power flow of the power system network.
- Convert the power system network into a graph G.
- Each bus of the power network is a node of the graph G.
- Each transmission line of the one-line power system diagram is a branch of graph G.
- The weight of each branch of graph G is assigned according to the absolute value of real power flow of the corresponding transmission line.

The created graph of the power system is considered as an input for Multilevel Kernel k-Means approach. Figures 12a and b show the proposed algorithm for intentional islanding of power systems using multilevel kernel k-means adopted from an earlier method called Metis as reported by Karypis and Kumar (1999).



**Fig. 12a:** The first part of the flowchart of finding the optimum form of intentional islanding of the power system by multilevel Kernel K-means



**Fig. 12b:** The second part (continuation) of the flowchart of finding the optimum form of intentional islanding of the power system by multilevel Kernel K-means

**Post Islanding Measures:**

The usual thing to worry about after island formation is the frequency changes that may pursue due to load/generation imbalance within the island. If there is excessive load in an island, there will be a decline in the frequency which should be controlled by load shedding. Otherwise, the frequency decline may cause the underfrequency protective relays of the generating units to trip leading to a worsening of the situation. In islands in which there is excessive generation, there will be an increase in frequency which must be dealt with by generation shedding. This expected form of behavior should be included in the objective function of the controlled islanding algorithm so that in addition to a reduction in load/generation imbalance in the islands to be formed, the partitioned solution results in minimum rate of change of frequency after island formation in all islands formed. The inclusion of this idea in the objective function may be done by dividing the sum of the power flows in the cut sets by the sum of the positive node weights in the island which is proportional to the total inertia of the machines in that island.

**Computational Requirements:**

The computational requirements of applying various graph partitioning schemes to power system islanding are presented here based on the findings of previous researchers. The computational burden of the approach determines whether or not the proposed algorithm has a potential for real time application. Since controlled islanding of power systems in the face of severe interruptions is a major power system emergency operation issue, its real time applicability is highly desirable.

The computational bottleneck of the application of the spectral approach to the intentional islanding of large scale power systems is the amount of mathematical effort required to compute the eigenvalues and the eigenvectors of the Laplacian matrix. In large scale power system islanding where a real time solution is needed and thousands of busses are involved, the computational requirements become burdensome. However, this could be eased a little since only the sign of each component of the Feidler vector is needed in order to partition the power system's equivalent graph and an exact answer is not really required. In applications where the Laplacian matrix is dense, there exist routines such as eig in Matlab that require  $O(n^3)$ . However, in power system applications in which the graph is sparse with relatively few connections compared to a complete graph, this would not be computationally wise. In such cases, it is more suitable to resort to the Lanczos algorithm which is an iterative algorithm.

The computational requirements of computing distances in the kernel k-means approach to bisectioning power systems is  $O(n^2)$  operations for each iteration. In modern day large scale power systems where sparsity holds, each iteration will only be of the  $O(nz)$   $z$  denotes the number of nonzero entries in the matrix that is proportional to the number of transmission lines in the power system. Since in the multilevel kernel k-means approach to power system bisectioning, we need not compute the eigenvectors at all and this drastically reduces the computational burden of the problem making it  $O(nz)$ . This is much less than that required for the best possible implementation of the spectral approach.

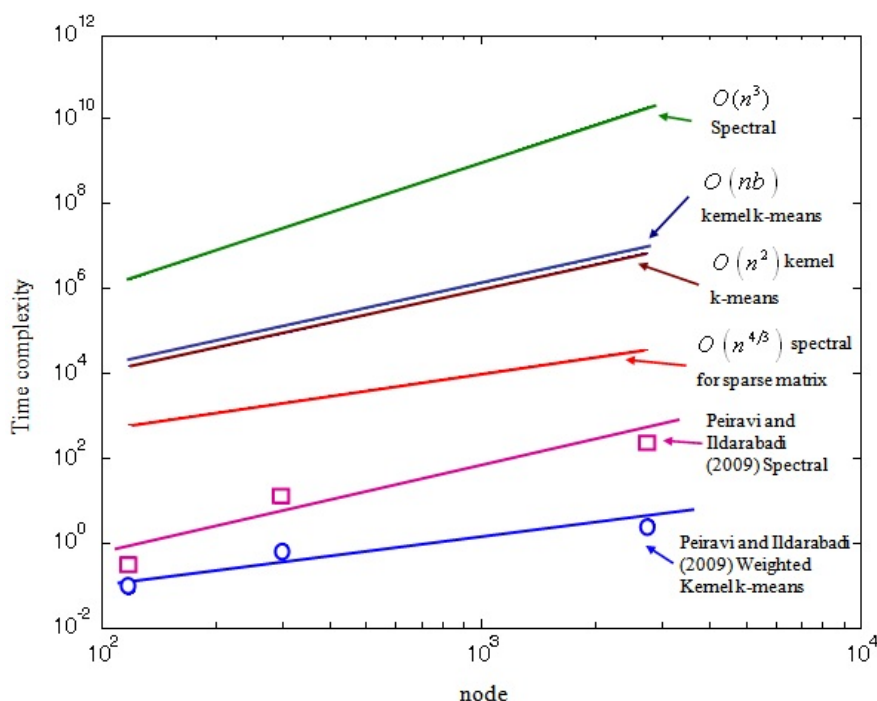
Peiravi and Ildarabadi (2009) presented a case study performed on the application of spectral vs. multilevel kernel k-means to the intentional islanding of three sample power systems namely the IEEE 118 bus, the IEEE 300 bus and the 2746 bus Polish power system showing an increasing trend in computational time, but a much less computational time for the multilevel kernel k-means approach as compared with the spectral approach and a much lower rate of increase in computational time versus system size which shows the computational advantage obtained in solving the intentional islanding of large scale power systems. The multilevel kernel k-means approach resulted in very impressive computational time reduction when compared with the spectral approach in the 2746 bus Polish power system. More work needs to be carried out to make a better comparison since the code used for the Multilevel Kernel k-Means approach indirectly uses C++ and is somewhat faster than the code that was written for the spectral approach in Matlab. The simulations reported were carried out on an IBM PC computer with a 2.8GHz Celeron processor using Matlab for the spectral approach and existing Graclus Software, version 1.2 for the Multilevel Kernel k-Means approach. These computational times are compared with the reported time complexities in Table 1 and shown in Figure 13.

**Conclusions:**

In this paper, the spectral and multilevel kernel k-means approaches to bisectioning power systems for islanding for avoiding power system blackouts were compared. When a fault is sensed that requires islanding, relay and breaker operation for islanding needs to be done within a few cycles. This implies that all islanding computations should be performed in much less than 500ms. Therefore, islanding must be done fast for it to be affective. The various intentional islanding methods presented previously are slow and are not suitable for real time application to large scale interconnected power systems. The time requirements of the various schemes were compared with real implementation results indicating that the reported time requirements for both the spectral and the weighted kernel k-means approach follow the same time requirement as expected as can be seen from the results having a similar slope. Even more important is the fact that the weighted kernel k-means implementation shows the best performance.

**Table 1:** A comparison of computational requirements of spectral vs. Kernel k-means bisectioning of three sample power systems studied

	Computational Complexity	IEEE 118 bus	IEEE 300 bus	2746 bus Polish
Number of Nodes	n	118	300	2746
Number of Branches	b	186	417	3514
Spectral Computational Requirements	$O(n^3)$	1643032	27000000	20706256936
Spectral Computational Requirements for sparse applications	$O(n^{4/3})$	578.77	2008.29	38453.33
Kernel K-means Computational Requirements for dense graphs per iteration	$O(n^2)$	13924	90000	7540516
Kernel K-means Computational Requirements for sparse graphs per iteration	$O(nz) \approx O(nb)$	21984	125100	9649444
Peiravi and Ildarabadi (2009) Spectral		0.312	12.639	234.563
Peiravi and Ildarabadi (2009) Weighted Kernel k-means		0.083	0.654	2.587



**Fig. 13:** A comparison of expected computational time requirements and implemented time requirements for spectral and multilevel kernel k-means bisectioning of power systems

### REFERENCES

Agematsu, S., S. Imai, R. Tsukui, H. Watanabe, T. Nakamura, T. Matsushima, 2001. "Islanding Protection System with Active and Reactive Power Balancing Control for Tokyo Metropolitan Power System and Actual Operational Experiences," in Proceedings of the 7th IEE Int. Conf. Developments in Power System Protection, pp: 351-354.

Aleksadrov, L., H. Djidjev, H. Guo and A. Maheshwari, 2006. Partitioning Planar Graphs with Costs and Weights, *ACM Journal of Experimental Algorithmics (JEA)*, 11(1.5): 1-28.

Al-Odienat, A.I., 2006. Power System Blackouts: Analysis and Simulation of August 9, 2004 Blackout in Jordan Power System, *Information Technology Journal*, 5(6): 1078-1082.

Anderson, G., P. Donalek, R. Farmer, N. Hatzigryriou, I. Kamwa, P. Kundur, N. Martins, J. Paserba, P. Pourbeik, J. Sanchez-Gasca, R. Schulz, A. Stankovic, C. Taylor and V. Vittal, 2005. "Causes of the 2003 Major Grid Blackouts in North America and Europe, and Recommend Means to Improve System Dynamic Performance", *IEEE Trans. On Power Systems*, 20(4): 1922-1928.

Cherng, J., S. Chen, C. Tsai, J. Ho, 1999. An efficient two-level partitioning algorithm for VLSI circuits, *Proceedings of the 1999 Design Automation Conference, ASP-DAC '99, Asia and Pacific*, 1: 69-72, Wanchai, Hong Kong, 18-21 Jan. 1999.

Cherng, J., S. Chen, 2003. An efficient multi-level partitioning algorithm for VLSI circuits, *Proceedings of the 16<sup>th</sup> International Conference on VLSI Design (VLSI'03)*, 4-8 January 2003, New Delhi, India, pp: 70-75.

Dhillon, Inderjit S., Guan, Yuqiang, Kulis, Brian, 2005. "A Fast Kernel Based Multilevel Algorithm for Graph Partitioning," In the *Proceedings of the 11<sup>th</sup> ACM SIGKDD International Conference on Knowledge Discovery Data Mining (KDD)*, pp: 629-634.

Dhillon, Inderjit S., Guan, Yuqiang, Kulis, Brian, 2007. "Weighted Graph Cuts Without Eigenvectors: a Multilevel Approach," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29(11): 1944-1957.

Filippone, M., F. Camastrab, F. Masullia, S. Rovettaa, 2008. A Survey of Kernel and Spectral Methods for Clustering, *Pattern Recognition*, 41(1): 176-190.

Hagen, L., A.B. Kahng, 1992. "New Spectral Methods for Ratio Cut Partitioning and Clustering," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 11(9): 1074-1085.

Holzrichter, M., S. Oliveira, 1999. A graph based method for generating the fiedler vector of irregular problems, *Proceedings of the 11 IPSP/SPDP'99 Workshops Held in Conjunction with the 13th International Parallel Processing Symposium and 10th Symposium on Parallel and Distributed Processing*, San Juan, P.R., pp: 978-985.

Karypis, G., V. Kumar, 1999. "A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs", *SIAM J. Scientific Computing*, 20(1): 359-392.

<http://glaros.dtc.umn.edu/gkhome/fetch/papers/mlSIAMSC99.pdf>

Kernighan, B.W., S. Lin, 1970. "An efficient heuristic procedure for partitioning graphs," *The Bell System Technical Journal*, 49(2).

Li, Hao, Rosenwald, W. Gary, Jung, Juhwan, Liu Chen-Ching, 2005. "Strategic Power Infrastructure Defense," In *Proceedings of the IEEE*, 93(5): 918-933.

Lin, S., X. Cheng, 2008. BC-GA: A graph partitioning algorithm for parallel simulation of internet applications, *Proceedings of the 16th Euromicro Conference on Parallel, Distributed and Network-Based Processing, PDP 2008*, art. no. 4457144, pp: 358-365.

Mohar, B., 1991. "The Laplacian Spectrum of Graphics," in *Graph Theory, Combinatorics, and Applications*, Vol. 2, Ed. Y. Alavi, G. Chartrand, O. R. Oellermann, A. J. Schwenk, Wiley, New York, pp: 871-898.

Peiravi, A., R. Ildarabadi, 2009. A fast algorithm for intentional islanding of power systems using the multilevel kernel k-means approach, *Journal of Applied Sciences*, 9(12): 2247-2255.

Rajamani, K., U.K. Hambarde, 1999. "Islanding and Load Shedding Schemes for Captive Power Plants," *IEEE Trans. on Power Del.*, 14(3): 805-809.

Rehtanz, C., 2003. *Autonomous systems and intelligent agents in power system control and operation*, Springer, pp: 256.

Sanaye-Pasand, M., M.R. Dadashzadeh, 2004. Iran national grid blackout, power system protection point of view, *Eighth IEE International Conference on Developments in Power System Protection (CP500)*, pp: 20-23, Amsterdam, Netherlands.

Saab, Y., 2002. A Contraction-based Ratio-cut Partitioning Algorithm, *VLSI design*, 15(2): 485-489.

Venkatasubramanian, V., J. Quintero, 2005. "Detection, Prevention and Mitigation of Cascading Events-Part II", *Power System Engineering Research Center, A National Science Foundation Industry/University Cooperative Research Center Since 1996, PSERC Publication 05-60*.

Vittal, V., Xiaoming Wang, 2005. "Detection, Prevention and Mitigation of Cascading Events-Part III", *Power System Engineering Research Center, A National Science Foundation Industry/University Cooperative Research Center Since 1996, PSERC Publication 05-61, October 2005*.

Wang, X.Z., Z. Yan, W. Xue, 2008. An adaptive clustering algorithm with high performance computing application to power system transient stability simulation, 3rd International Conference on Deregulation and Restructuring and Power Technologies, *DRPT 2008*, art. no. 4523578, pp: 1137-1140.

Wang, X., Vittal, V., 2004. "System Islanding using Minimal Cutsets with Minimum Net Flow", In Proceedings of the IEEE Power Eng. Soc. Power Syst. Conf. Expo., New York, 1: 379-384.

Yang, B., V. Vittal, Gerald T. Heydt, Arunabha Sen, 2007. A Novel Slow Coherency Based Graph Theoretic Islanding Strategy, IEEE Power Engineering Society General Meeting, pp: 1-7.