2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

Performance Analysis of an Influence-Model-Based Graph Partitioning Algorithm

Rahul Dhal Sandip Roy

Abstract— **Recently, a stochastic automaton known as the influence model was advanced as a tool for flexible and distributed graph partitioning, which can find optimal solutions to numerous hard partitioning problems in an almost-sure sense. Here, we provide a performance analysis of the influence model-based partitioner, for the hard problem of *m*-way partitioning with reference vertices. Specifically, we show that the influence model algorithm finds the optimal partition quickly with high probability, whenever the optimal cut-set is sufficiently weak compared to other cuts in the graph.**

I. Introduction

A wide family of graph-partitioning algorithms have been developed, that permit fast partitioning according to a variety of metrics. These partitioning algorithms have found quite wide application in engineering fields ranging from VLSI circuit design to sensor networking. As graph partitioning is applied in an increasingly wide range of modern network applications, however, several new needs in graph partitioning are coming to the forefront. First, the metrics for partitioning are becoming increasingly complicated and varied, and tools for optimal or fast partitioning are needed for these new metrics. Additionally, there is also an increasing need for fast algorithms for partitioning in settings where decisions need to be made in real-time, e.g. in power system islanding. In these settings, good approximations of solutions are needed for partitioning problems that are NP-hard. Third, distributed- or self-partitioning algorithms, wherein network nodes must achieve partitioning through distributed communication and/or local knowledge of the network topology, are increasingly needed. These needs call for further research on graph partitioning.

Stochastic graph-partitioning algorithms are promising in that they yield fast solutions for some graph classes and partitioning metrics, and yet are potentially flexible enough to give globally optimal solutions for a range of problems including some NP-hard ones. Numerous stochastic algorithms have been developed, many originating from a classical work of Karger [4]. Specifically, [4] uses a stochastic "recursive contraction algorithm" to solve the min-cut problem, i.e. to find the minimum number of edges whose removal partitions an un-weighted multigraph of n nodes into two disjoint sub-graphs. The recursive contraction algorithm presented in [4] finds all the solutions of the min-cut problem in $O(n^2 \log^3 n)$ run-time with a very high probability (exceeding $1-{}^{1}\!/_{n}$); the computational cost and storage needs compare favorably with other deterministic algorithms for the min-cut problem.

Motivated by the need for an algorithm that finds the optimal partitions for broad classes of graphs and is flexible with regard to the metric for partitioning, Wan et al proposed an alternate stochastic algorithm for graph partitioning in [1]. The core construct underlying the proposed algorithm is a discrete-valued discrete-time stochastic network model known as the influence model. The influence model was initially advanced as a tractable representation for some stochastic interactions that occur among components in engineered networks [2, 3]. Recently, the influence model has also been used for various computational tasks, not only the partitioning task considered here but also distributed-agreement and social-network decision-making tasks [5, 6]. It was shown in [1] that the proposed algorithm is able to find optimal partitions for various metrics and graph classes, and is especially suited for graph partitioning problem in which certain nodes are constrained to be in specified partitions (referred to here as *partitioning with reference vertices*). Examples in [1] suggest that the algorithm can achieve fast solutions to these problems (many of which are NP-Hard) for some graph classes. However, a detailed convergence analysis of the algorithm was not developed in [1]. Here, we characterize the convergence rate of the algorithm for a particular class of graphs, namely ones whose optimal cut is sufficiently weak compared to other cuts in the graph. This convergence analysis lends credence to the algorithm's use in various engineering problems, where mixed strong/weak connections are common.

The remainder of the paper in organized as follows. Section 2 briefly reviews the influence model, while section 3 reviews influence model- based partitioning. The remainder of the paper is devoted to performance analysis of the algorithm for an *m*-way partitioning problem with reference vertices, for graphs with weak links (Sections 4 and 5). In our review of the influence model and the partitioning algorithm, we closely follow our group's previous works in those directions [1-3]. Due to space constraints, examples and proofs are excluded, see [11] for these.

II. Influence Model: Review

In general terms, an influence model is a network of interacting discrete-time finite-state Markov chains. Specifically, we define the influence model to have

Both the authors are with the School of Electrical Engineering and Computer Science at Washington State University, Pullman, WA, 99164 USA. Correspondence should be sent to rdhal@eecs.wsu.edu.

n components or nodes or sites. Each node has a discrete-valued status (specifically, one of a finite set of possibilities) evolving in discrete time. Each site's status evolves according to an update rule that depends not only on the status of the site but also on those of its neighbors:

1. At each time k, each site i independently picks another site j (possibly itself) with probability d_{ij}. Site j is referred to as the *determining site* for site i.
2. Site i's next status is probabilistically selected based on the current status of site j. That is, the Probability Mass Function (PMF) of site i's next status is specified by the determining site j's current status.

We kindly ask the reader to see [1-3] for numerous illustrative examples of the influence model update.

We shall focus on the special case of a **copying influence model** wherein each site has the same number of statuses and, furthermore, each site simply copies the status of its determining site at every time step. To facilitate our analysis of the influence model based partitioning algorithm, let us introduce some notations and terminology regarding the copying influence model and briefly describe its analysis.

For the copying influence model, we assume that the status of each site may be one of the m possible values. We find it convenient to represent the status of a site i at time step k by an m-element column *status indicator* vector $s_i[k]$, that has a single unity entry indicating the status of the site and is otherwise zero. Also, we term the stochastic matrix $D=[d_{ij}]$ as the *network influence matrix*. We define a network graph for the influence model from D as follows. The graph is assumed to have n vertices, corresponding to n sites. The graph has a directed edge with weight d_{ij} from site j to site i when $d_{ij}>0$. The statuses of all sites at time step k is represented in a column vector $s[k]$ of length $n\times m$ as $s'[k]=\begin{bmatrix} s_1'[k] & \ldots & s_n'[k]\end{bmatrix}$.

Due to the special structure of the influence model update, the status probabilities of individual sites in the model can be computed easily via a low-order linear recursion. Specifically, the site status probabilities can be found as

$$E(s'[k+1]\mid s'[0]) = s'[0]H^{k+1} = E(s'[k]\mid s'[0])H$$

where the recursion matrix is given by $H=D'\otimes I_m$ (and I_m is an $m\times m$ identity matrix) and E is the expectation operator. The influence model admits numerous analyses beyond computing individual sites' status probabilities. We refer our readers to [2, 3] for details.

III. Influence Model based Partitioning Algorithm

This paper is focused on the problem of *partitioning with reference vertices*. We present a brief review of the problem before reviewing the influence model based partitioning algorithm. Specifically consider a directed and weighted graph $G(V,E:W)$, where V is a set of n vertices labeled as (1, …, n) and E is the set of directed edges in the graph. The weight of the edge from node i and to node j is denoted by w_{ij} and is assumed strictly positive. For *m-way* partitioning with reference vertices, we wish to divide the vertices in the graph into m different groups (partitions), subject to the constraint that m specified vertices (say $V_1, V_2, \ldots V_m$) are constrained to be in m different partitions. Our aim is to find partitions of the network of this form that minimize a cost defined from the edge weights between partitions. For the special case that the cost function is the sum of the edge weights between partitions (i.e., min-cut partitioning), this problem is known to be NP-Hard for $m\geq 3$ [7, 8], and good approximations are still needed.

Our group advanced an influence model based partitioning algorithm in [1] as a tool for solving several partitioning problems including *m-way* partitioning with reference vertices. We now review the algorithm briefly. See [1] for a more detailed description of the algorithm.

The algorithm consists of three phases: 1) Mapping, 2) Initialization and 3) Recursion and Stopping.

1) Mapping: The graph to be partitioned is mapped to a copying influence model whose sites can each take on m possible statuses. In particular, we associate an influence model site with each vertex of the graph to be partitioned. The copying probabilities d_{ij} in the influence model are defined from the edge weights as:

$$d_{ij} = \begin{cases} 0 & \text{if there is no edge from } j \text{ to } i, j\neq i \\ \Delta w_{ji} & \text{if there is an edge from } j \text{ to } i, j\neq i \\ 1-\sum\limits_{l\neq i} d_{il} & \text{for } i=j \end{cases}$$

where the scale factor Δ is chosen so that $\Delta \leq \dfrac{1}{\max\limits_i \sum\limits_j w_{ji}}$.

2) Initialization and Recursion: Let the reference vertices be denoted as $(V_1, V_2, \ldots, V_m)$. The site in the mapped copying influence model corresponding to each reference vertex is initialized to a different one of the m possible statuses (say V_1 to status 1, V_2 to status 2 and so forth without loss of generality). Also, the copying probabilities for the influence model site associated with the i^{th} reference vertex V_i are then modified as:

$$d_{ij} = \begin{cases} 0 \text{ if } i\neq j \\ 1 \text{ if } i=j \end{cases}$$

for each *i=1,…,m*. Thus, the influence model sites corresponding to the reference vertices always choose themselves as their determining site, and the statuses of these nodes remains the same. The remaining nodes are initialized at random to any of the m possible statuses. The influence model is allowed to run as per the above mentioned update rules. At each time-step, the influence model identifies a valid *m-way* partition of the network, i.e. sites in the same status specify a partition of the original graph, with each reference vertex guaranteed to be in a different partition. Typically, the algorithm will favor partitions with weak cuts between them, since strongly-connected sites are likely to adopt the same status while weakly connected sites will remain different.

3) *Stopping:* Centralized and decentralized stopping strategies for an influence model based partitioning algorithm have been developed, see [1] for details. Our concern here is on how fast the algorithm achieves the optimal partition as it scans through potential partitions (for *m-way* partitioning with reference vertices assuming a min-cut cost), so we do not concern ourselves with the details of the stopping strategy.

It was shown in [1] that the algorithm would eventually find the optimal partition for *m-way* partitioning with reference vertices with probability 1. In the following section, we show that the influence model algorithm is capable of finding the partitions quickly when the graphs have a weak-cut.

IV. PERFORMANCE ANALYSIS

We take a two-step approach to show that the influence model based partitioning algorithm is able to quickly find the optimal solution to the partitioning with reference vertices problem, when the cuts between the optimal partitions are sufficiently weak. First, we show that, asymptotically, the sites in each optimal partition of the mapped copying influence model have statuses equal to that of the reference vertex in that partition with high probability, given that there are weak links between the partitions. Second, we provide an expression for the number of time-steps after which the status probabilities are close to their asymptotic (steady state) value. Using simple probability constructs, we then show that the above characterization of the statuses of the sites implies that the partitions are indentified correctly with high probability in a short time. The results in this section assume that the cuts between the partitions are arbitrarily weak (which is a limiting case). In Section 5, we present bounds on how weak these links need to be for fast partitioning, in terms of sub-graph eigenvalues and the number of nodes in the graph.

For simplicity of presentation, we develop and prove the results for the 2-way partitioning problem with reference vertices. We then present the results for m-way partitioning with reference vertices after proving the results for 2-way partitioning. The proofs for the results for m-way partitioning case generally follow the same logic as for the 2-way partitioning case. We stress here, in beginning the analysis, that the m-way partitioning results are of particular importance because the m-way problem is NP-hard; thus, for this case, our results identify a sub-class of the problem for which fast algorithms can be found.

For 2-way partitioning, the network is mapped to a binary copying influence model, with two reference vertices. For notational convenience, the reference vertices are indexed as *1* and n where n is the number of the vertices in the network. To characterize the probabilities that each site takes on a certain status asymptotically, we show that these probabilities are specified by the eigenvector(s) of the network influence matrix associated with its eigenvalue(s) at unity. The following theorem characterizes these eigenvectors and hence allows analysis of individual sites status probabilities.

THEOREM 1:

Consider 2-way partitioning with reference vertices for a connected graph, comprising two connected sub-graphs which each contain one of the reference vertices. Assume that the edge weights between the sub-graphs (or optimal partitions) can be scaled down by an arbitrary factor. Now consider applying the influence model-based partitioning algorithm. Then the asymptotic probability that the status of a site in a given partition is equal to the status of the reference vertex of that partition approaches unity, as the edges between the partitions are made arbitrarily weak (i.e., the edge weights are scaled down sufficiently).

The above theorem shows that the sites in partition 1 have a high probability of being in status 1 and sites in partition 2 have a high probability of being in status 2 in steady state when each optimal partition is internally strongly connected, but the two partitions are weakly linked. Now let us present the result for the m-way case.

THEOREM 1.A:

Consider *m-way* partitioning with reference vertices for a connected graph comprising m strongly connected sub-graphs (each of which contain one of the reference vertices), and assume that the weights of the edges between the sub-graphs are scaled down by an arbitrary factor. Consider partitioning the graph using the influence-model algorithm. Then the asymptotic probability that the status of a site in a given optimal partition is equal to the status of the reference vertex of that partition approaches unity, as the edges between the partitions are made arbitrarily weak. ■

It is worth stressing that, *a priori*, a user of the influence model-based partitioning algorithm would not know that the edge weights between partitions were sufficiently weak – after all, finding weak cuts is the goal of the partitioning algorithm! However, as we will formalize in Section 5, the strength of cuts needed for fast partitioning can be bounded, i.e. the cuts do not need to be arbitrarily weak; we will then argue that many graph classes encountered in practice have such weak cuts, and so the algorithm will work quickly in many cases. We also reiterate that the algorithm will eventually find the weakest cut (or the optimal according to an appropriate cost measure), regardless of how strong or weak the cut is.

Now that the asymptotic dynamics of the mapped influence model have been characterized, we focus on the transients of the model with the aim of characterizing how quickly the model can identify optimal partitions. Specifically, for such graphs we show the following: after running the influence model algorithm for a specified number of time-steps, the sites in each partition have a high probability of being in the same status as that of the corresponding reference site.

Let us first give a conceptual overview of the transient analysis and then present the formal result. As is made clear from the proof of Theorem 1 (see [11]), the probability of sites having status "**1**" during the transient is given by:

$$P_1'([k]) = P_1'([0]) \times (D')^k$$

where the vector $P_1[k]$ contains the probabilities that each site has status "**1**". From the above relation, we can see that eigen-analysis of the network influence matrix is needed to determine the settling behavior of the model. Since the occupancy probability vector at the kth time step depends on D'^k, a Jordan decomposition of D^k shows the modal contribution from each eigenvalue. Applying the Frobenius theorem for stochastic matrices, we immediately see that D has some eigenvalues at unity while the rest of its eigenvalues are strictly less than 1 in magnitude. Therefore, for sufficiently large values of k, the modal contributions from eigenvalues with magnitude strictly less than unity should die out, and hence the state occupancy probabilities should approach the steady state probabilities (as formalized in Theorem 1). This forms the basis of our argument, which is formalized in the next theorem.

THEOREM 2

Consider a connected graph with two connected subgraphs containing the two reference vertices, and assume that the weights of the edges between the optimal partitions can be scaled down arbitrarily. Sites in the mapped influence model in each partition have statuses equal to that of the corresponding reference vertex with a high probability ($>1-0.1^q-\nu$, where ν becomes arbitrarily small as the edges between the partitions are made weaker and q can be chosen as an arbitrary positive quantity) after k-time steps, if

$$k \geq \frac{-q-\log_{10}(n-2)}{\log_{10}(|\lambda_{\max}|)}$$

Here, $\lambda_{\max}$ is the largest eigenvalue of the network influence matrix of the mapped influence model with magnitude strictly less than unity.

THEOREM 2.A:

Consider a connected graph with m strongly connected subgraphs, and assume that the edges between the optimal partitions can be scaled down arbitrarily. Sites in the mapped influence model in each partition have statuses equal to that of the corresponding reference vertex with a high probability ($>1-0.1^q-\nu$, where ν becomes arbitrarily small as the edges between the partitions are made weaker and q can be chosen as an arbitrary positive quantity) after k-time steps, if

$$k \geq \frac{-q-\log_{10}(n-m)}{\log_{10}(|\lambda_{\max}|)}$$

Here, $\lambda_{\max}$ is the largest eigenvalue of the network influence matrix of the mapped influence model with magnitude strictly less than unity. ■

We note that our analysis depends on having "sufficiently weak" link between the partitions. We will revisit the perturbation of the eigenvector and present explicit bounds on ν (which represents the eigenvector perturbation due to graph perturbation) in terms of graph structure in Section 5.

A note about Theorem 2 is worthwhile. The result we have given is phrased in terms of the largest eigenvalue of D that is strictly less than 1 in magnitude. The value of this eigenvalue depends on how strongly connected the sub-graphs are. The stronger the connections within the sub-graphs, the smaller the largest magnitude among the non-unity eigenvalues and hence the faster the occupancy probabilities settle to steady-state values. A family of results is available in the algebraic graph theory literature, that formalize the relationship between connectivity and sub-dominant eigenvalue locations. We omit the details here.

Since each site has been shown to adopt the correct partition status with high probability quickly, we intuitively believe that our proposed algorithm can identify partitions correctly after the specified number of time steps with a high probability. To formalize this intuition, we should prove that the statuses of the sites jointly identify the partitions. Computations of joint probabilities are developed and discussed in [2]. However, explicitly computing joint probabilities is computationally taxing. Instead of directly characterizing the joint status probabilities, we here will bound the joint status probabilities from the individual site status probabilities. Specifically, we have shown that in steady state sites in different partitions have high probability (nearly equal to 1) of being in the same status as the reference vertex of the partition. Thus, we can expect that the joint probability of the sites in the same partition having the same status, and of sites in different partitions having different statuses, is high. This notion is formalized in Theorem 3.

THEOREM 3:

Consider using the influence model for m-way partition with reference vertices. The optimal (min-cut) partitions are denoted as Partition 1, Partition 2 and so on for notational convenience. Say that at some time k in the influence model algorithm, the individual site status occupancy probabilities are defined as $\eta_i = p(s_i[k]=r_i)$, where r_i is the proper partition of site i in the optimal partitioning. Then the joint probability $P_J = p(\bigcap s_i[k]=r_i)$ is bounded by:

$$\max(0,(1-\sum_{i=1}^{n}(1-\eta_i))) \leq P_J \leq \min_i(\eta_i)$$

■

Let us first interpret this result in the context of the previous two theorems. From Theorems 1.A and 2.A, if the number of time-steps k is greater than the specified value, then the sites individually have a high probability of being in the same status as that of the reference vertices of the partition. Hence, from Theorem 3, the joint probability of all sites having the status as that of the reference vertex is high after the specified number of time-steps. Thus, the algorithm finds the correct partitions after the specified number of time steps with a high probability for graphs that have a weak cut.

V. CHARACTERIZING WEAK CUTS

The results in the Section 4 show that for graphs with sufficiently weak cuts, fast solution to the *m*-way partitioning problem with reference vertices is possible. However, thus far the presented results have only guaranteed partition for sufficiently weak cuts without explicit characterization of how weak the edge weight needs to be to achieve fast partitioning. Here, we aim to give graphical bounds on the cut sizes for fast partitioning. The reason that the above results are not explicit in the graph structure is because they are based on a limiting perturbation argument. Specifically, the network influence matrix of the mapped influence model of the graph with weak links can be viewed as a perturbation of the network influence matrix for the case with the weak links removed. We have used the following two ideas regarding the perturbation of the network influence matrix to show fast convergence for sufficiently weak cuts:

1) The eigenvalues of the network influence matrix do not change much due to perturbation.

2) The eigenvectors of the network influence matrix associated with the two eigenvalues at unity remains almost the same after perturbation.

If we are able to give explicit bounds on the extent of eigenvalue and eigenvector perturbation due to graph perturbations, we may be able to give explicit graph theoretic bounds on the partitioning algorithm performance. In the following discussion, we characterize the size of the weak cuts (and hence of the perturbation) that ensure that the eigenvector and eigenvalue of the network influence matrix are not significantly perturbed. It should be noted that the results of the perturbation theory for general matrices are typically conservative. This is mainly because there are no convenient tight characterizations of eigenvalue perturbations for Jordan block of sizes greater than one. Nevertheless, we believe that the development of explicit graph theoretic bounds on eigenvalue and eigenvector perturbation (for both general and symmetric cases) is significant in characterizing the influence model partitioner, in that they make explicit class of graphs that permit fast partitioning.

A. Perturbation of Eigenvalues

The following result gives a bound on size of the weak cut (equivalently, the norm of the perturbation on the network influence matrix effected by the weak cut) such that the eigenvalues of the network influence matrix of the mapped influence model with weak cuts are close to those of the mapped influence model with weak-cut removed (where closeness is in a proportional sense with respect to the spectral gap).

THEOREM 4:

Let us denote the network influence matrix of the mapped influence model as $\bar{D}$ and that of the influence model with the weak cuts between the optimal partitions removed as D. The perturbation matrix is defined as $E = \bar{D} - D$. Let $\lambda_{\max}$ be the largest eigenvalue of D with magnitude strictly less than unity. We seek to bound the change in $\lambda_{\max}$ due to the perturbation by a fraction of the spectral gap, or more generally by an amount $G(1-\lambda_{\max})$ where $G(x)$ is any function such that $G(x) \to 0$ as $|x| \to 0$. Then a perturbation matrix satisfying the following inequality guarantees that the bounds on eigenvalue change is met.

$$\log(\|E\|_2) \le n \times \left[\log\left(G(1-\lambda_{\max})\right) - \log(2n-1) - \log(4n) \times \frac{n-1}{2n} \right]$$

This bound on the perturbation matrix is weak. The bound can be improved for the case where the eigenvalues are simple i.e. the size of the largest Jordan block is one, through the application of Henrici's Theorem [9, pp-172]. Unfortunately, it is difficult to provide general graph conditions such that the eigenvalues are simple for directed graphs, and so it is difficult to obtain a stronger bound on the perturbation of the eigenvalues in this way. For the special case of undirected graphs with weak links (which is a subset of the case that the network influence matrix has real eigenvalues), the Courant-Fischer theorem can provide strong bounds [11].

B. Eigenvector Sensitivity

Another perturbation result that we have relied on is that the eigenvectors associated with the eigenvalues at unity (which determine the steady state status probabilities) do not change much after the perturbation. While there are many well-known results that deal with sensitivity of eigenvectors to perturbation of general matrices, we find it worth our while to present a result that is applicable to our development. In particular, we aim to develop an explicit bound on the eigenvector change, in terms of the strength of the weak links between the partitions of the graph. We exploit the block lower-triangular structure of the network influence matrix and the fact that the perturbation is small compared to the matrix itself to characterize the eigenvector. Let us give a conceptual discussion of the bound methodology before presenting the formal result.

To do so, we define two $(n-1)\times(n-1)$ matrices, called the *reduced network influence matrices*, by deleting n^{th} row and column of the network influence matrices for the connected as well as the disconnected graphs. We denote them as $\tilde{D}$ for the connected graph (i.e. the graph after perturbation) and

$\hat{D}$ for the disconnected graph. It is easy to see that $\hat{D}$ and $\tilde{D}$ can be written in the following lower triangular block structure:

$$\tilde{D} = \begin{bmatrix} 1 & 0,...,0 \\ \vec{\upsilon}_1 & A \end{bmatrix} \text{ and } \hat{D} = \begin{bmatrix} 1 & 0,...,0 \\ \hat{\vec{\upsilon}}_1 & \hat{A} \end{bmatrix}$$

It is easy to see that, $\hat{A} = A + E$, $O(E) = \varepsilon$ and $\hat{\vec{\upsilon}}_1 = \vec{\upsilon}_1 + O(\varepsilon)$, where ε is an arbitrarily small quantity and $O(\varepsilon)$ is the usual Big-o notation. We will show that the relevant eigenvectors of these two matrices (which give the steady state probabilities) can be found by solving a certain related systems of linear equations, and bound the change in the eigenvector from this perspective. Here is the formal result:

THEOREM 5:

Let the reduced network influence matrix of the mapped binary copying influence model be $\hat{D}$ and that of the influence model with the weak cuts between the optimal partitions removed be $\tilde{D}$. Note that $\hat{D}$ and $\tilde{D}$ have the following structure:

$$\tilde{D} = \begin{bmatrix} 1 & 0,...,0 \\ \vec{\upsilon}_1 & A \end{bmatrix} \quad \hat{D} = \begin{bmatrix} 1 & 0,...,0 \\ \hat{\vec{\upsilon}}_1 & \hat{A} \end{bmatrix},$$

Where, $\hat{A} = A + E$, and $\hat{\vec{\upsilon}}_1 = \vec{\upsilon}_1 + \vec{e}$.

If

$$E_{i,j} < \frac{0.1^q}{0.1^q(n-2)+n} \times \frac{1}{\left\|(I-A)^{-1}\right\|_\infty} \text{ and}$$

$$e_i < \frac{0.1^q}{0.1^q(n-2)+n} \times \frac{1}{\left\|(I-A)^{-1}\right\|_\infty}$$

then, the change in the entries of the eigenvector for the unity eigenvalue due to perturbation is at most 0.1^q where q is an arbitrarily chosen positive quantity.

THEOREM 5.A:

Let the reduced network influence matrix of the mapped copying *m*-state influence model be $\hat{D}$ and that of the influence model with the weak cuts between the optimal partitions removed be $\tilde{D}$. Note that $\hat{D}$ and $\tilde{D}$ have the following structure:

$$\tilde{D} = \begin{bmatrix} 1 & 0,...,0 \\ \vec{\upsilon}_1 & A \end{bmatrix} \quad \hat{D} = \begin{bmatrix} 1 & 0,...,0 \\ \hat{\vec{\upsilon}}_1 & \hat{A} \end{bmatrix},$$

Where, $\hat{A} = A + E$, and $\hat{\vec{\upsilon}}_1 = \vec{\upsilon}_1 + \vec{e}$. If,

$$E_{i,j} < \frac{0.1^q}{0.1^q(n-m)+n} \times \frac{1}{\left\|(I-A)^{-1}\right\|_\infty}$$

and

$$e_i < \frac{0.1^q}{0.1^q(n-m)+n} \times \frac{1}{\left\|(I-A)^{-1}\right\|_\infty}$$

then, the change in the entries of the eigenvector for the unity eigenvalue due to perturbation is at most, 0.1^q where q is an arbitrarily chosen positive quantity.

Remark: - The term $\frac{1}{\left\|(I-A)^{-1}\right\|_\infty}$ is undesirable to some extent. Numerous results are available in the algebraic graph theory and stochastic matrix literature that permit bounding of the quantity in terms of the network graph; we omit the details.

ACKNOWLEDGMENTS

This project was partially supported by NSF grants ECS-0725589, ECS-0901137, CNS-1035369, and CNS-1058124.

REFERENCES

[1] Y. Wan, S. Roy, A. Saberi, B. Lesieutre, "A stochastic automaton-based algorithm for flexible and distributed network partitioning," *International Journal of Distributed Sensor Networks,* vol. 4, 2008, pp – 223-246

[2] C. Asavathiratham, "*The Influence Model: A Tractable Representation for the Dynamics of Networked Markov Chains," Ph.D. dissertation*, EECS Dept., MIT, Oct. 2000.

[3] C. Asavathiratham, S. Roy, B. Lesieutre and G. Verghese. "The influence model," *IEEE Control Systems Magazine,* Dec 2001.

[4] D. R. Karger, C. Stein, "A new approach to the minimum cut problem," *Journal of the ACM*, Vol. 43, No. 4, July 1996, pp. 601–640.

[5] David Kempe , Jon Kleinberg , Éva Tardos, "Maximizing the spread of influence through a social network," *Proc. 9th ACM SIGKDD international conference on Knowledge discovery and data mining*, August 24-27, 2003, Washington, D.C.

[6] S. Roy, K. Herlugson, A. Saberi, "A control-theoretic approach to distributed discrete-valued decision-making in networks of sensing agents," *IEEE Trans. on Mobile Computing,* vol. 5, 2006, pp 945-957.

[7] A compendium of NP optimization problems," http://www.nada.kth.se/~viggo/wwwcompendium/

[8] E. Dahlhaus , D. S. Johnson , C. H. Papadimitriou , P. D. Seymour, M. Yannakakis, "The complexity of multiterminal cuts", *SIAM Journal on Computing*, vol.23, no.4, pp.864-894, Aug. 1994

[9] G. W. Stewart, J. Sun, "*Matrix Perturbation Theory*," Academic Press Inc., USA, 1990, Ch 3- 4.

[10] B. Chamberlain, "Graph partitioning algorithms for distributed workloads of parallel computations," Technical Report UW-CSE-98-10-03, University of Washington, Oct. 1998

[11] R. Dhal, S. Roy, "Performance analysis of an influence-model-based graph partitioning algorithm" (Extended Version), http://www.eecs.wsu.edu/~rdhal