Manufacturing Consent*

(Invited Paper)

Vivek S Borkar[1], Jayakrishnan Nair[2], and Nalli Sanketh[3]

[1]School of Technology and Computer Engineering, Tata Institute of Fundamental Research, India
[2]Department of Electrical Engineering, California Institute of Technology, USA
[3]Department of Information Technology, National Institute of Technology Karnataka, India

***Abstract*—A scheme for consensus formation is considered wherein the value of a certain variable associated with the nodes of a network is fixed a priori for a prescribed set of K nodes, and allowed to propagate throughout the network through an averaging process that mimics a gossip algorithm. The objective is to find the best choice of these K nodes that will achieve the fastest convergence to consensus. This objective is captured by the Perron-Frobenius eigenvalue of the resultant sub-stochastic matrix, which then is the quantity one seeks to minimize. We propose an algorithm for this optimization problem, as well as a greedy scheme with some performance guarantees for a variant of the problem that seeks to minimize a simpler objective. Some other related formulations are also considered.**

I. Introduction

'Gossip algorithms', which lead to a consensus value of an estimated quantity through local averaging over neighboring nodes of a graph, have been a topic of extensive research in recent years because of applications to sensor networks and social networks. See [1] for an excellent account of the state of the art. An important issue here is the rate of convergence, which is dictated by the second largest (in modulus) eigenvalue of the stochastic matrix with respect to which the averaging is performed. This has lead to some natural optimization issues concerning the minimization of this quantity, an account of which can be found in [2]. In this article we consider a related problem that has an element of control, leading to what one may call 'controlled gossip'. The idea is to partly or fully influence the opinion / estimate at a few, say K nodes, so that the consensus yields a targeted value. This corresponds to a spread of intended information across a network. The problem then is to find the optimal choice of these K nodes.

As we argue below, this amounts to the minimization of the top, or *Perron-Frobenius* eigenvalue of a sub-stochastic matrix obtainable from a stochastic matrix by deleting K rows and columns of the latter. The optimization is then over the possible choices of these rows and columns. This is a hard discrete optimization problem. We embed it into an optimization problem over continuous-valued variables and propose a projected gradient scheme for the same. This involves deriving an explicit expression for this gradient and combining the associated gradient descent with the 'successive projections' algorithm of Boyle, Dykstra and Han [3], [4]. As the objective function is not necessarily convex, we can expect convergence only to a local minimum. Hence we consider a greedy algorithm for a related objective function which, unlike the original, is supermodular. The supermodularity property allows us to provide (loose) performance guarantees for a greedy heuristic algorithm. We also discuss some variants of the basic theme: first the problem of choosing the above K nodes in presence of an adversary, then the problem of optimal placement of new edges, and finally a dynamic version of the above optimization problem, where the K nodes need to be prodded again and again over time to prevent 'forgetting'.

*with apologies to Noam Chomsky

II. The optimization problem

Consider an irreducible directed graph $\mathcal{G}$ with node set $\mathcal{S} = \{1, 2, \cdots, N\}$ and edge set $\mathcal{E}$, equipped with a compatible stochastic matrix $P = [[p(i,j)]]_{1 \le i,j \le N}$. The 'network' represented by $\mathcal{G}$ computes recursively a vector $\hat{x}_n = [\hat{x}_n(1), \cdots, \hat{x}_n(N)]^T$ according to the 'gossip' scheme

$$\hat{x}_{n+1} = P\hat{x}_n, \ n \ge 0.$$

Consider a scenario where K out of N nodes comprising a subset $A \subset \mathcal{S}$ fix their value of the corresponding components of $\hat{x}_n$ for good, at a fixed value c. Let P_A denote the principal submatrix of P corresponding to nodes in A^c and $\bar{P}_A$ the submatrix with row indices corresponding to A^c and column indices corresponding to A. Let I and $\mathbf{1}$ denote respectively the identity matrix and the vector of all 1's (with appropriate dimension depending on the context). Then the subvector x_n of $\hat{x}_n$ corresponding to nodes in A^c evolves according to

$$x_{n+1} = P_A x_n + c\bar{P}_A \mathbf{1}, \ n \ge 0. \tag{1}$$

As P_A is substochastic with spectral radius < 1, this will converge to $x^* := c(I - P_A)^{-1}\bar{P}_A\mathbf{1}$. This has the well known stochastic representation

$$x^*(i) = E[h(x_\tau)|X_0 = i], i \in A^c, \tag{2}$$

for a Markov chain $\{X_n\}$ with transition matrix P, $\tau := \min\{n \ge 0 : X_n \in A\}$, and $h(\cdot) \equiv c$ on A. Thus $x^* = c\mathbf{1}$, i.e., there is asymptotic consensus on the value c as desired. The rate of convergence to the consensus will be dictated by the Perron-Frobenius eigenvalue $\lambda(P_A)$ of P_A, which then is

the objective function we seek to minimize. Formally, we are interested in the following optimization problem.

$$\min_{A\subset\mathcal{S},|A|=K} \lambda(P_A) \tag{3}$$

The minimizing set A will therefore correspond to the K most important nodes to influence from the point of view of rapid opinion dissemination. This is a distinct notion of rating nodes as compared to rating schemes such as Google's PageRank [5] or the 'hub and authority' model of Kleinberg [6], or the various centrality measures proposed in social network research [7].

Our discrete optimization problem (3) appears hard to solve exactly. In the following section, we consider a continuous relaxation of (3), which is more tractable.

III. Continuous relaxation

We now consider a relaxation of (3) to an optimization problem over reals as follows. Let $P'(\theta) := P\Theta$ where Θ is a diagonal matrix with entries $\theta_1, \cdots, \theta_N \in [0,1]$ on its diagonal. Let $\theta := [\theta_1, \cdots, \theta_N]^T$. We impose the constraint $\sum_i \theta_i = N - K$. Our relaxation of (3) is the following.

$$\min_{\theta\in C_1\cap C_2} \lambda(\theta), \tag{4}$$

where $\lambda(\theta) = \lambda(P'(\theta))$ denotes the the Perron-Frobenius eigenvalue of $P'(\theta)$, $C_1 := [0,1]^N$, and

$$C_2 := \{x = [x_1, \cdots, x_N]^T \in \mathcal{R}^N : \sum_i x_i = N - K\}.$$

Note that a $\{0,1\}$-valued θ will correspond to exactly K ones and $N-K$ zeros, thereby recovering our original formulation. Let $\pi, V \in \mathcal{R}^N$ denote respectively the Perron-Frobenius left- and right- eigenvectors of $P'(\theta)$, i.e., the eigenvectors corresponding to the eigenvalue $\lambda(\theta)$, where we suppress the θ-dependence of π, V for simplicity. That is, for $j \in \mathcal{S}$,

$$\lambda(\theta)V(j) = \sum_k p(j,k)\theta_k V(k), \tag{5}$$

$$\lambda(\theta)\pi(j) = \sum_k p(k,j)\theta_j \pi(k). \tag{6}$$

Differentiating (5) w.r.t. $\theta_i, 1 \le i \le N$, we have

$$\frac{\partial\lambda(\theta)}{\partial\theta_i}V(j)+\lambda(\theta)\frac{\partial V(j)}{\partial\theta_i} = p(j,i)V(i)+\sum_k p(j,k)\theta_k\frac{\partial V(k)}{\partial\theta_i}.$$

Multiplying by the above equation by $\pi(j)$, summing over j, and using (6), we get

$$\frac{\partial\lambda(\theta)}{\partial\theta_i} = \frac{V(i)\pi^T p(\cdot,i)}{\pi^T V},$$

which gives an explicit expression for the gradient of $\lambda(\theta)$ w.r.t. θ. This suggests the following projected gradient scheme. Let $\eta > 0$ denote a prescribed step size. Then

$$\theta_{n+1} = \Gamma\left(\theta_n - \eta\frac{\text{diag}(V)P^T\pi)}{\pi^T V}\right), \tag{7}$$

where $\Gamma(\cdot)$ denotes the projection operator onto the set $C_1 \cap C_2$, i.e.,

$$\Gamma(x) = \arg\min_{y\in C_1\cap C_2} \|x - y\|.$$

The following 'successive projections' algorithm due to Boyle, Dykstra and Han [3], [4] can be used to compute $\Gamma(x)$.

- Set N-dimensional vectors y_1^0, y_2^0, x_1^0 equal to the zero vector and $x_2^0 = x$.
- At kth iteration $(k \ge 1)$, do:

$$\begin{aligned}
&z = x_2^{k-1} + y_1^{k-1}.\\
&\text{For } 1 \le i \le N,\\
&\qquad 0 \le z(i) \le 1 \Longrightarrow x_1^k(i) = z(i),\\
&\qquad z(i) > 1 \Longrightarrow x_1^k(i) = 1,\\
&\qquad z(i) < 0 \Longrightarrow x_1^k(i) = 0.\\
&y_1^k = z - x_1^k.\\
&w = x_1^k + y_2^{k-1}.\\
&x_2^k = w + \left(\frac{N-K}{N} - \frac{1}{N}\sum_j w(j)\right)\mathbf{1}\\
&y_2^k = w - x_2^k.
\end{aligned}$$

It can be proved that $\Gamma(x) = \lim_{k\to\infty} x_2^k$ [3], [4]. An alternative algorithm for computing $\Gamma(x)$ that exploits the structure of the sets C_1 and C_2 can be found in [8].

The objective function of (4) is not in general convex and hence we can only expect the gradient projection scheme (7) to converge to a local minimum. This suggests resorting to multi-start, simulated annealing, etc. to improve performance. From a candidate solution θ^* to (4), we may obtain a candidate solution to (3) by picking the nodes corresponding to the smallest K components of θ^*.

IV. A greedy algorithm

Let $\{X_n\}$ be a Markov chain with transition matrix P. The logarithm of the Perron-Frobenius eigenvalue λ of P_A has the interpretation of being the asymptotic rate of exit from A^c, i.e., the 'rate of exponential decay' of the tail probability of τ:

$$\log(\lambda(P_A)) = \lim_{t\uparrow\infty} \frac{\log P(\tau > t)}{t}$$

for $\tau := \min\{n \ge 0 : X_n \in A\}$. This suggests looking at a related, more amenable performance measure, the mean exit time $E[\tau]$. Assuming uniform initial distribution over A^c, we have the cost criterion

$$E[\tau] = \frac{1}{N-K}\mathbf{1}^T(I - P_A)^{-1}\mathbf{1} = \frac{1}{N-K}\mathbf{1}^T\left(\sum_{n=0}^{\infty} {P_A}^n\right)\mathbf{1}.$$

We now state our optimization objective formally. For $A \subseteq \mathcal{S}$, define $f(A) := \mathbf{1}^T\left(\sum_{n=0}^{\infty} {P_A}^n\right)\mathbf{1}$. Note that $f(\emptyset) = \infty$, and $f(A) < \infty$ for all non-empty $A \subseteq \mathcal{S}$. We seek to solve the following optimization problem:

$$\min_{A\subseteq\mathcal{S},|A|=K} f(A) \tag{8}$$

Another interpretation of the problem (8) is obtained as follows. In (1), fixing $x_0 = \mathbf{0}$ and $c > 0$, x_n evolves as $(c\mathbf{1} - x_n) = cP_A{}^n\mathbf{1}$, implying $\|c\mathbf{1} - x_n\|_1 = c\mathbf{1}^T P_A{}^n\mathbf{1}$. This means the optimization objective of (8) can be rewritten as $\frac{1}{c}\sum_{n=1}^{\infty}\|c\mathbf{1} - x_n\|_1$. Thus, (8) corresponds to minimizing an infinite horizon cost, the cost at each step n being the l_1 difference between x_n and the vector it will converge to.

(8) appears to be a hard problem to solve exactly. However, it can be shown that f is supermodular. This motivates a greedy heuristic algorithm [9].

Proposition 1. *f is supermodular, i.e., for $A, B \subseteq \mathcal{S}$,*

$$f(A) + f(B) \leq f(A \cup B) + f(A \cap B).$$

Proof: For $A \subseteq \mathcal{S}$, define $f^{(n)}(A) = \mathbf{1}^T P_A^n \mathbf{1}$. We will prove that $f^{(n)}(\cdot)$ is supermodular. It is easy to see that this implies supermodularity of f. Define $g : [0,1]^N \to \mathbb{R}$ as $g(\theta) = \mathbf{1}^T \left(\text{diag}(\theta)\ P\ \text{diag}(\theta)\right)^n \mathbf{1}$. Define

$$\hat{\theta}(A) := (\mathrm{I}_{\{i \notin A\}}, i = 1, 2, \cdots, N),$$

where $\mathrm{I}_{\{z\}}$ equals 1 if z is true and 0 otherwise. Note that $f^{(n)}(A) = g(\hat{\theta}(A))$.

It is easy to see that $\frac{\partial^2 g(\theta)}{\partial\theta_i \partial\theta_j} \geq 0$ for all $i \neq j$. This implies that g is supermodular [10, Theorem 10.4], i.e., g satisfies $g(\theta) + g(\tilde{\theta}) \leq g(\theta \vee \tilde{\theta}) + g(\theta \wedge \tilde{\theta})$ for all $\theta, \tilde{\theta} \in [0,1]^N$. For $A, B \subseteq \mathcal{S}$,

$$\begin{aligned} f^{(n)}(A) + f^{(n)}(B) &= g(\hat{\theta}(A)) + g(\hat{\theta}(B)) \\ &\leq g(\hat{\theta}(A) \vee \hat{\theta}(B)) + g(\hat{\theta}(A) \wedge \hat{\theta}(B)) \\ &= g(\hat{\theta}(A \cap B)) + g(\hat{\theta}(A \cup B)) \\ &= f^{(n)}(A \cap B) + f^{(n)}(A \cup B). \end{aligned}$$

This proves that $f^{(n)}(\cdot)$ is supermodular. ∎

Greedy Algorithm:
The supermodularity of f (equivalently, the submodularity of $-f$) motivates the following simple greedy heuristic to compute an approximate solution A^G to (8) (see [9]).

1) Set $A_0 = \emptyset$.
2) For $i = 1, 2, \cdots, K$ do

$$\begin{aligned} j_i^* &= \arg\min_{j \in A_{i-1}^C} f(A_{i-1} \cup \{j\}), \\ A_i &= A_{i-1} \cup \{j_i^*\}. \end{aligned}$$

3) Set $A^G = A_K$.

Note that the objective function f is monotone non-increasing, i.e., $f(A) \geq f(B)$ whenever $A \subseteq B$. The algorithm constructs the set A^G in K stages. In each stage, the node that produces the greatest marginal decrease in the objective function is added to the set.

The greedy algorithm described above involves several evaluations of the function f. For non-empty $A \subset \mathcal{S}$, the computation of $f(A)$ involves the inversion of the typically 'large' matrix $(I - P_A)$. However, these inversions can be greatly simplified using the Sherman-Morrison-Woodbury (SMW) formula (see [11, Section 2.1.3]). For example, in Stage $i \geq 2$ of the algorithm, the SMW formula can be used to compute $(I - P_{A_{i-1} \cup \{j\}})^{-1}$ efficiently from $(I - P_{A_{i-1}})^{-1}$.

Suboptimality bound: Since $f(\emptyset) = \infty$, one cannot bound the suboptimality of $f(A^G)$ relative to the optimum value f^* of (8) directly, as in Theorem 4.2 of [9]. We can, however make the following weaker statement. Suppose that we restrict the optimization in (8) to sets A containing a special node m; let f_m^* denote the optimum value of this relaxed problem. Consider the following modification to the greedy algorithm above: start with $A_1 = \{m\}$, and run the iterations for $i = 2, \cdots, K$ to obtain the set $A^G(m)$. Then

$$f(m) - f(A^G(m)) \geq \left(1 - \frac{1}{e}\right)(f(m) - f_m^*),$$

where e denotes the base of the natural logarithm.

V. Experiments

In this section, we present simulation results for the algorithms presented in the preceding sections on two small real-world network datasets. Each dataset provides us with an undirected network graph, represented by its adjacency matrix R. From R, we generate the stochastic matrix P (that determines the dynamics of opinion propagation) as follows.

$$P_{ij} = \begin{cases} \alpha & \text{for } i = j \\ \frac{1-\alpha}{\deg(i)} R_{ij} & \text{for } i \neq j \end{cases}.$$

Here, $\deg(i)$ denotes the degree of node i. In our experiments, we set $\alpha = 0.7$. We compute the set A of K 'key' nodes in the network and $\lambda(P_A)$ (which determines the rate of convergence to consensus) using the following algorithms.

1) *grad_desc*: This refers to the projected gradient descent algorithm described in Section III. The step size η was set by trial and error. We ran the algorithm five times, with random initializations of θ; the result corresponding to the best of these runs is reported.
2) *greedy_alg*: This refers to the greedy algorithm presented in Section IV.
3) *pagerank*: This refers to the well known PageRank algorithm [5]. We compute the stationary distribution of the Markov chain with transition matrix P. The set A is composed of the K nodes (states) having the largest stationary probabilities.
4) *HITS*: The set A consists of the K nodes that have the highest rating as hubs/authorities according to the well known HITS algorithm [6]. Since the adjacency matrix R in our examples is symmetric, the hub and authority ratings for a node are equal; the vector of hub/authority ratings of the nodes is the Perron-Frobenius eigenvector of R^2.

We use the following two datasets.

1) *Zachary Karate club*: This is a well known network representing friendships between 34 members of a Karate club over a period of two years [12]. We take $K = 5$.

TABLE I
RESULTS: EACH TABLE ENTRY GIVES THE PERRON-FROBENIUS EIGENVALUE OF P_A

	grad_desc	*greedy_alg*	*pagerank*	*HITS*
Zachary Karate Club	0.930	0.908	0.930	0.930
Net. Sci. coauthorships	0.995	0.989	0.996	0.999

2) *Coauthorships in Network Science*: A dataset describing a collaboration network of scientists working in network theory and experiments has been prepared by Newman [13]. In this network graph, nodes are scientists and two scientists are connected by an (undirected) edge if they have co-authored a paper. For our experiment, we use the largest connected component of this graph, which contains 379 nodes. We take $K = 20$.

Our results are presented in Table I. The greedy algorithm produces the lowest value of $\lambda(P_A)$, followed by the projected gradient descent algorithm. These results suggest that our notion of ranking (subsets of) nodes from the point of view of rapid opinion dissemination is indeed distinct from other popular centrality notions in the literature.

VI. VARIATIONS

In this section we describe some related problems.

1) *Adversarial action:*
Consider the following variant of our basic model. Let $A = A_1 \cup A_2$, where A_1, A_2 are disjoint with K_1, K_2 nodes respectively. The values at the K_1 nodes in A_1 have been frozen at a given number c_1 by an adversary. We freeze the values at the K_2 nodes in A_2 at a fixed number $c_2 > c_1$. Then the vector x_n corresponding to the nodes in A^c will converge to x^*, given by the representation (2) with $h(i) = c_1$ on A_1 and c_2 on A_2. For $j = 1, 2$, let τ_j denote the first hitting time of A_j for the Markov chain $\{X_n\}$ with transition matrix P. (2) implies that for $i \in A^c$,

$$\begin{aligned} x^*(i) &= c_1 P(\tau_1 < \tau_2 | X_0 = i) + c_2 P(\tau_2 < \tau_1 | X_0 = i) \\ &= c_1 + (c_2 - c_1) P(\tau_2 < \tau_1 | X_0 = i). \end{aligned}$$

Note that x^* in general depends on our choice of the set A_2. We consider the problem of choosing the set $A_2 \subset A_1^c$, such that $|A_2| = K_2$, so as to maximize $\sum_{i \in A^c} x^*(i)$. Define, for $i \in \mathcal{S}$,

$$y(i) := P(\tau_2 < \tau_1 | X_0 = i).$$

The optimization described above is equivalent to the following.

$$\max_{A_2 \subset A_1^c,\ |A_2| = K_2} h(A_2), \tag{9}$$

where $h : 2^{A_1^c} \to \mathbb{R}$ is defined by $h(A_2) := \sum_{i \in A_1^c} y(i)$. We have

$$\begin{aligned} y(i) &= \sum_j p(i,j) y(j), \quad i \in A^c, \\ y(i) &= 0, \quad i \in A_1, \\ y(i) &= 1, \quad i \in A_2. \end{aligned}$$

Let $y = (y(i), i \in A^c)$. The above equations can be written in the form

$$y = P_A y + \check{P}\mathbf{1},$$

where P_A is defined as before and $\check{P}$ is the submatrix of P with row indices corresponding to A^c and column indices corresponding to A_2 (the dependence of $\check{P}$ on A_1 and A_2 is supressed for simplicity). Therefore, $y = (I - P_A)^{-1}\check{P}\mathbf{1}$, implying

$$h(A_2) = |A_2| + \mathbf{1}^T (I - P_A)^{-1} \check{P}\mathbf{1}.$$

As we prove next, h is non-decreasing and submodular, implying that a simple greedy heuristic algorithm for solving (9) with bounded suboptimality can be devised [9].

Proposition 2. *h is non-decreasing and submodular.*

Proof: Assuming X_0 is distributed uniformly over A_1^c, it is easy to see that

$$h(A_2) = (N - K_1) P(\tau_2 < \tau_1).$$

From the above representation, it is clear that h is a non-decreasing set function.
Let $\tilde{P}$ denote the submatrix of P with row indices corresponding to A^c and column indices corresponding to A_1, and $\bar{P}_A$ denote the submatrix of P with row indices corresponding to A^c and column indices corresponding to A. Then $\check{P}\mathbf{1} = \bar{P}_A\mathbf{1} - \tilde{P}\mathbf{1}$. Therefore, we may write

$$\begin{aligned} h(A_2) &= |A_2| + \mathbf{1}^T (I - P_A)^{-1} \check{P}\mathbf{1} \\ &= |A_2| + \mathbf{1}^T (I - P_A)^{-1} \bar{P}_A \mathbf{1} - \mathbf{1}^T (I - P_A)^{-1} \tilde{P}\mathbf{1} \\ &= |A_2| + (N - K_1 - |A_2|) - \mathbf{1}^T \left(\sum_{n=0}^{\infty} {P_A}^n \right) \tilde{P}\mathbf{1} \\ &= N - K_1 - \mathbf{1}^T \left(\sum_{n=0}^{\infty} {P_A}^n \right) \tilde{P}\mathbf{1} \end{aligned}$$

The step before last above uses the fact that $(I - P_A)^{-1}\bar{P}_A\mathbf{1} = \mathbf{1}$; this follows easily from Section II. Define $\tilde{h}^{(n)}(A_2) := \mathbf{1}^T {P_A}^n \tilde{P}\mathbf{1}$. We will prove that $\tilde{h}^{(n)}$ is supermodular. This in turn implies supermodularity of $\mathbf{1}^T \left(\sum_{n=0}^{\infty} {P_A}^n\right) \tilde{P}\mathbf{1}$, which in turn implies submodularity of h.
To prove that $\tilde{h}^{(n)}$ is supermodular, we use the same technique as in the proof of Proposition 1. Define the N dimensional column vector d to be the sum of the K_1 columns of P corresponding to A_1. Define $g : [0,1]^N \to \mathbb{R}$ as

$$g(\theta) = \mathbf{1}^T \left(\mathrm{diag}(\theta)\ P\ \mathrm{diag}(\theta) \right)^n (\theta \circ d).$$

Here, $a \circ b$ denotes the elementwise product of the vectors a and b. Define

$$\hat{\theta}(A_2) := (\mathrm{I}_{\{i \notin A\}}, i = 1, 2, \cdots, N).$$

Note that $\tilde{h}^{(n)}(A_2) = g(\hat{\theta}(A_2))$.

As before, it is easy to see that $\frac{\partial^2 g(\theta)}{\partial\theta_i \partial\theta_j} \geq 0$ for all $i \neq j$. This implies that g is supermodular [10, Theorem 10.4], which in turn implies submodularity of $\tilde{h}^{(n)}$, following the steps in the proof of Proposition 1. ∎

2) *Adding of new edges:*
Let P be a substochastic matrix, e.g., P_A above. Consider the problem of adding K new directed edges between pairs that did not have an edge in P. For simplicity, consider adding an edge from node i to node i', where $p(i,i') = 0$. Suppose we assign a probability of $\theta \in [0,\beta], 0 < \beta < 1$, to this new edge and reduce the existing transition probabilities out of i by a multiplicative factor of $(1-\theta)$. Let $\check{P}(\theta) = [[p_\theta(j,j')]]$ denote the resultant matrix and consider the problem of minimizing its Perron-Frobenius eigenvalue $\Lambda(\theta)$ over $\theta \in [0,\beta]$. This turns out to be a special case of the 'risk-sensitive' control problem with the associated dynamic programming equation

$$V(j) = \frac{\min_{\theta\in[0,\beta]} \sum_k p_\theta(j,k)V(k)}{\Lambda},\ j \in \mathcal{S}, \quad (10)$$

which specifies $\Lambda > 0$ as the optimal Perron-Frobenius eigenvalue and specifies the 'value function' V, which is a positive vector unique modulo a multiplicative factor [14]. This extends to K edges in a straightforward manner. Value and policy iteration algorithms to solve (10) are available [14]. Note that the expression being minimized on the right hand side is linear in θ and therefore the minimum will be at either 0 or 1 or everywhere in $[0,1]$, indicating that the edge must be added with full weight, or should not be added at all, or either. There is never the case that a weight in $(0,\beta)$ is the only option.
If one imposes a 'resource constraint' $\sum_i \theta_i = N - K$ as in the preceding sections, dynamic programming does not work and one has to resort to a projected gradient scheme along the lines of Section III.

3) *A dynamic optimization problem:*
Fix $A \subset \mathcal{S}$ of cardinality K. We now introduce a 'forgetting factor' $\alpha \in (0,1)$ and consider the dynamics

$$x_{n+1} = \alpha P_A x_n + c\bar{P}_A \nu_n,\ n \geq 0.$$

The interpretation is as follows. In absence of repeated prodding, the nodes 'forget' at an exponential rate. The input process $\{\nu_n\}$ consists of $\{0,1\}$-valued vectors which repeatedly 'prod' the nodes in A, with the objective being to keep average 'interest level' x_n as high as possible. Let r_k denote the cost of prodding node k in A once, and $r := [r_1, \cdots, r_K]^T$. The objective then is to minimize the average cost

$$\limsup_{n\uparrow\infty} \frac{1}{n} E\left[\sum_{m=0}^{n-1} (r^T\nu_m - \gamma^T x_m)\right] \quad (11)$$

for a prescribed positive vector γ. Under stationarity, this becomes

$$r^T\bar{\nu} - \gamma^T\bar{x}, \quad (12)$$

where $\bar{\nu}, \bar{x}$ denote resp. the stationary averages of ν_n, x_n. In turn, the dynamics implies that these satisfy

$$\bar{x} = \alpha P_A\bar{x} + c\bar{P}_A\bar{\nu}. \quad (13)$$

Thus the problem reduces to the linear program of minimizing (12) subject to the constraints (13) and the non-negativity constraints.
To justify restricting a priori to stationary solutions, argue as follows. First note that if the initial condition is taken to be from a bounded set, then the x_n remain in a bounded set. Arguments leading to Corollary 11.2 of [15] then allow us to restrict ourselves to stationary solutions. What's more, (12) depends solely on the averages $\bar{\nu}, \bar{x}$, of which the latter is uniquely specified by the former. Thus an open loop, i.e., deterministic choice of $\{\nu_n\}$ with time average $\bar{\nu}$ will also be optimal.

ACKOWLEDGEMENT

This work was supported in part by the project 'Dynamics of Decision Making on Networks: Estimation and Control' from General Motors India Lab. The first author was also supported by a J. C. Bose Fellowship. This work was done when the second and the third author were visiting the School of Technology and Computer Science at the Tata Institute of Fundamental Research. The visit of the third author was supported by a joint Summer Fellowship Program of the Academies of Sciences in India.

REFERENCES

[1] D. Shah, *Gossip Algorithms*. Now Publishers, 2009.
[2] S. Boyd, P. Diaconis, and L. Xiao, "Fastest mixing Markov chain on a graph," *SIAM Review*, vol. 46, no. 4, pp. 667–689, 2004.
[3] H. Boyle and R. Dykstra, "A method for finding projections onto the intersection of convex sets in Hilbert space," in *Advances in Order Restricted Statistical Inference*, ser. Lecture Notes in Statistics. Springer Verlag, 1985, pp. 28–47.
[4] S.-P. Han, "A successive projection method," *Mathematical Programming*, vol. 40, pp. 1–14, 1988.
[5] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Comput. Netw. ISDN Syst.*, vol. 30, no. 1-7, pp. 107–117, 1998.
[6] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *J. ACM*, vol. 46, no. 5, pp. 604–632, 1999.
[7] M. Jackson, *Social and Economic Networks*. Princeton University Press, 2008.
[8] N. Maculan, C. Santiago, E. Macambira, and M. Jardim, "An O(n) algorithm for projecting a vector on the intersection of a hyperplane and a box in R^n," *Journal of Optimization Theory and Applications*, vol. 117, pp. 553–574, 2003.
[9] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions - I," *Mathematical Programming*, vol. 14, pp. 265–294, 1978.
[10] R. Sundaram, *A First Course in Optimization Theory*. Cambridge University Press, 1996.
[11] G. H. Golub and C. F. Van Loan, *Matrix computations (3rd ed.)*. Baltimore, MD, USA: Johns Hopkins University Press, 1996.
[12] W. Zachary, "An information flow model for conflict and fission in small groups," *Journal of Anthropological Research*, vol. 33, no. 4, pp. 452–473, 1977.
[13] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Phys. Rev. E*, vol. 74, no. 3, 2006.
[14] V. Borkar and S. Meyn, "Risk-sensitive optimal control for Markov decision processes with monotone cost," *Mathematics of Operations Research*, vol. 27, no. 1, pp. 192–209, 2002.

[15] V. Borkar, "Convex analytic methods in Markov decision processes," in *Markov Decision Processes: Models, Methods, Directions and Open Problems*, E. Feinberg and A. Shwartz, Eds. Kluwer Academic, 2001, pp. 347–375.