



HAL
open science

Opinion Shaping in Social Networks Using Reinforcement Learning

Vivek S Borkar, Alexandre Reiffers-Masson

► **To cite this version:**

Vivek S Borkar, Alexandre Reiffers-Masson. Opinion Shaping in Social Networks Using Reinforcement Learning. 2023. hal-04327548

HAL Id: hal-04327548

<https://imt-atlantique.hal.science/hal-04327548v1>

Preprint submitted on 11 Dec 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Opinion shaping in social networks using reinforcement learning

Vivek S. Borkar, Department of Electrical Engineering

Indian Institute of Technology

Powai, Mumbai 400076, India

Email: borkar.vs@gmail.com

Alexandre Reiffers-Masson, Robert Bosch Centre for Cyberphysical Systems,

Indian Institute of Science

Bengaluru 560012, India

Email: reiffers.alexandre@gmail.com

Abstract

Recent studies have demonstrated that the decisions of agents in society are shaped by their own intrinsic motivation, and also by the compliance with the social norm. In other words, the decision of acting in a particular manner will be affected by the opinion of society. This social comparison mechanism can lead to imitation behavior, where an agent will try to mimic the behavior of her neighbors. Using this observation, new policies have been designed, e.g., in the context of energy efficiency and transportation choice, to leverage social networks in order to improve altruism and prosocial behavior. One policy is to use targeting strategies. Indeed, by changing the behavior of influential actors in a social network, it is possible to reshape the global behavior of agents towards more prosocial behavior. However, discovering who are the influential agents requires a lot of information, such as the matrix of interactions between agents. In this paper, we study how to shape opinions in social networks when the matrix of interactions is unknown. We consider classical opinion dynamics with some stubborn agents and the possibility of continuously influencing the opinions of a few selected agents, albeit under resource constraints. We map the opinion dynamics to a value iteration scheme for policy evaluation for a specific stochastic shortest path problem. This leads to a representation of the opinion vector as an

Work supported in part by a J. C. Bose Fellowship from the Government of India and a grant ‘Monte Carlo and Learning Schemes for Network Analytics’ from Indo-French Centre for Promotion of Advanced Research (CEFIPRA).

approximate value function for a stochastic shortest path problem with some non-classical constraints. We suggest two possible ways of influencing agents. One leads to a convex optimization problem and the other to a non-convex one. Firstly, for both problems, we propose two different online two-time scale reinforcement learning schemes that converge to the optimal solution of each problem. Secondly, we suggest stochastic gradient descent schemes and compare these classes of algorithms with the two-time scale reinforcement learning schemes. Thirdly, we also derive another algorithm designed to tackle the curse of dimensionality one faces when all agents are observed. Numerical studies are provided to illustrate the convergence and efficiency of our algorithms.

Index Terms

Social Networks; Opinion Shaping; Reinforcement Learning; Stochastic Shortest Path

I. INTRODUCTION

In recent times there has been increasing interest in non-price based mechanisms to improve society's behavior in the context of, e.g., energy efficiency or traffic behavior. These policies are usually less expensive to implement and can be politically feasible as opposed to price based policies. One example is the use of lottery with the distribution of coupons for energy efficiency [24] or for promoting off-peak usage of cars [17]. Another example is of leveraging social network for enhancing pro-social behavior. Indeed, social interactions can impact day to day decisions of an agent. For instance, in transportation choice, several works ([10], [22]) have demonstrated that the preferences of people in the decision maker's peer group will impact her choice of mode of transport to work (public transport, bicycle, car). Another practical application concerns how to use social comparison to enhance energy efficiency [20]. One specific way of leveraging social network for such purposes is to exploit the word-of-mouth/imitation process in a network by using a targeted advertising campaign. Targeted advertising on a social network amounts to finding which agents should be convinced in a social network to be pro-social in such a way that by imitation, the maximum number of agents in the whole social network will also be pro-social. Designing such a targeting strategy, however, can be challenging because of: (1) computational issues, (2) unknown social network, (3) size or lack of convexity of the resulting optimization problem, and so on. The goal of this paper is to propose different algorithms based on reinforcement learning that address these issues.

Our initial model can be described as follows: The society is composed of a finite set of agents and each agent has an opinion concerning a given pro-social action that she has to take.

For instance, it could be the opinion concerning whether or not she should take the bus to work or how much she cares about energy efficiency of her apartment. Whether an agent performs or not the pro-social action, the rest of the society that is "close" to her in the social network will observe her and vice versa. Therefore each agent will have a tendency to imitate her neighbors and vice versa. A planner (government, owner of the social network) is interested in choosing which agents she will influence to shape the opinion in a given direction. The planner can influence an agent through two controls which will be described later on. The society is divided into three types of agents. The first set is composed of '*stubborn*' agents. In this set, we assume that they have a given opinion and they will not be impacted by the social network. The second set of agents is composed of '*uncontrolled*' agents. In this set, agents are influenced by the social network and their opinion will be impacted by the opinion of the others. However, the planner cannot directly influence them. The last set of agents is called the set of '*controlled*' agents. This set is composed of agents that care about the opinion of their neighbors and also can be influenced by the planner. The goal of the planner is to shape the opinion of the social network by targeting specific agents from the latter group. We call this problem the *opinion shaping problem*. One of the major drawbacks of this initial model is that for shaping the opinion of the society, the planner needs to know the influence matrix. Worse, even if the influence matrix is known, the number of agents is so big that it is not feasible to decide optimally which agent should be chosen. Finally, depending on how the planner can influence the users, the convexity of the problem can be lost.

Our initial model is inspired by [4], [8]. In these papers, the influence matrix is assumed to be known. Moreover the authors do not consider efficient and decentralized algorithms to solve the opinion shaping problem. In this paper, we extend these works further in order to address these issues. Throughout our paper, the main assumption is that the planner does not know the influence matrix but he observes when the agents interact. Using these observations, we are able to derive three reinforcement learning based algorithms that will address the aforementioned issues. The proposed algorithms are decentralized. We also provide supporting simulations for the different settings. From a mathematical point of view, we provide the equivalence of the opinion-shaping problem with a stochastic shortest path problem and use this correspondence to motivate our algorithms and their convergence. Additionally, we discuss several possible extensions and future directions.

A. Organization and Main Results

The remainder of this paper is organized into eight sections. In section II, we discuss related works. Section III introduces the opinion-shaping problem after first describing the model for the opinion adoption process. Section IV is the main section of this paper. We prove the equivalence of the opinion adoption process with a stochastic shortest path problem. Using this equivalence, we propose a decentralized algorithm accounting for the fact that the influence matrix is unknown. This is followed by two other variants. The first one is using a unbiased estimator for the gradient and the second one is designed to tackle the curse of dimensionality one faces when all agents are observed. In section V, we prove the convergence of our algorithms. In section VI, a new algorithm is proposed when the opinion-shaping problem is not convex. Numerical studies are discussed in VII. Moreover, we compare the efficiency of the second algorithm, where all the agents are not observed, with the one where all the agents are observed. Finally, we study the efficiency of the annealing scheme for the non-convex opinion-shaping problem. Section VIII concludes with pointers to some possible extensions of this work.

II. RELATED WORKS

The spread of opinions in social networks broadly falls into three families of mathematical models and the influence maximization problem has been studied in the context of each of these. The first category is cascade threshold models. In the last decade, initial models in this framework for control of user activity were dedicated to maximization of influence [14], including the seminal work of Kempe et al. [15]. One of the drawbacks of this line of work is that the state of each user is assumed to be finite, often even binary. Another key limitation is that it only focuses on the maximization of influence, which reduces its possible scope for applications. Indeed, one may also be interested in other objectives such as minimum activity in a social network or diverse activity and not just activity maximization. Based on these, a second category of models has been proposed, e.g., by Zha et al. in [11], who define a new mathematical problem dubbed the activity shaping problem. First, they use Hawkes processes to model the activity of users in a social network. Undeniably, in recent literature these point processes have proved to be a very effective method to capture users' activity [27]. Secondly, the authors consider an activity shaping problem wherein by controlling the exogenous rate vectors of the Hawkes processes, a central controller tries to minimize a convex function which depends on the expected overall

instantaneous intensity of the processes. Since then, other extensions have been suggested, e.g., [26], [23].

Our model of opinion propagation falls in the last category, viz., consensus models. For several years, much effort has been devoted to the study of users' activities in a social network within this framework. The seminal work of Degroot [9] proposes a simple model to capture the diffusion of opinion. In his paper, the author assumes that each agent at each instant will compute the average opinion of her neighbors including possibly herself, and then replace her current opinion by this average. Several extensions of the Degroot model have been considered recently, especially the opinion shaping part [4], [8], [18]. In [4], the authors assume that a planner can directly contaminate a user in the social network by sending him some messages. The user reads the messages according to a certain probability and with the remaining probability, she will sample from the messages sent by her friends. In [8], the authors consider a more drastic control where they can freeze the opinion of a given user. Finally, in [18], the authors suggest a control based on the reduction of the interaction between different agents of the social network.

Concerning the opinion shaping problem without the knowledge of the network, the authors in [16] propose a data-driven model and a learning algorithm in case of a cascade with a linear threshold model. In [25], the authors suggest a Partially Observed Markov Decision Process (POMDP) framework in order to tackle the uncertainty over the topology of the network, again for the case of a cascade with linear threshold model.

To the best of our knowledge, our paper is the first one that tries to shape the opinion under resource constraints using a reinforcement learning approach, under the assumption that the opinion propagation is captured by a consensus model, but without the full knowledge of the topology.

III. PRELIMINARIES AND MODEL

We consider a social network given by a connected directed graph $\mathcal{G} = (\mathcal{S}, \mathcal{E})$ where \mathcal{S} is the set of its agents and \mathcal{E} the set of edges. To each edge $(i, j) \in \mathcal{E}$ we assign a probability weight $p_{ij} > 0$ with $\sum_{\{j:(i,j) \in \mathcal{E}\}} p_{ij} = 1$. We set $p_{ij} = 0$ if $(i, j) \notin \mathcal{E}$. The total number of agents is equal to I .

With each agent $i \in \mathcal{S}$, we also associate a process of valuations $x_i(k) \in [0, 1], n \geq 0$. Let $x(k) = [x_1(k), \dots, x_I(k)]$ be the associated vector for each k . We write \mathcal{S} as a disjoint union $\mathcal{S} = S \cup S_0 \cup S_1$ of three sets. Each time instant k , an agent i (more generally, a set of agents)

from \mathcal{S} is selected and updates her valuation. The update mechanism will change depending on which set an agent belongs to:

Set of ‘stubborn’ agents (S_0): the agents that belong to this set have their valuation frozen at some fixed value for good, i.e., $i \in S_0 \implies \forall k, x_i(k) \equiv h(i) \in [0, 1]$.

Set of ‘uncontrolled’ agents (S_1): This set stands for agents for which their valuations evolves according to a gossip mechanism, but they are not amenable to external influence. In this case the valuation update of an agent $i \in S_1$ is captured by the following mechanism: agent i polls a neighbor $\ell \in \mathcal{S}$ with probability $p_{i\ell}$ and updates $x_i(k)$ to $x_i(k+1) = x_\ell(k)$.

Set of ‘controlled’ agents (S): The remaining agents that constitute the set S also evolve according to a gossip mechanism, but are amenable to external influence or ‘control’. With probability $\alpha_i \in (0, 1)$, agent i will be influenced directly by the planner and with probability $1 - \alpha_i$, agent i will be influenced by her peer group. When influenced by the planner, agent i updates $x_i(k)$ to $x_i(k+1) = w_i(u_i)$, with $u_i \in \mathbb{R}_+$ without loss of generality and $w_i : \mathbb{R}_+ \mapsto [0, 1]$ are concave increasing and continuously differentiable maps. (Concavity captures the ‘diminishing returns’ effect.) If agent i is influenced by her peer group, she polls a neighbor ℓ with probability $p_{i\ell}$ and updates $x_i(k+1) = x_\ell(k)$.

To summarize, the overall dynamics is then described as follows. Suppose agent $i \in \mathcal{S}$ performs an update at time k . Then

$$\begin{aligned} x_i(k+1) &= \begin{cases} w_i(u_i) & \text{w.p. } \alpha_i \\ x_\ell(k) & \text{w.p. } 1 - \alpha_i \end{cases} \quad i \in S, \\ x_i(k+1) &= x_\ell(k), \quad i \in S_1, \\ x_i(k+1) &= h(i), \quad i \in S_0, \\ x_j(k+1) &= x_j(k) \quad \forall j \neq i. \end{aligned}$$

Analogous scheme holds when more than one agent updates.

For each $i \in \mathcal{S}$, when $k \rightarrow \infty$, $x_i^* = \lim_{k \rightarrow +\infty} E[x_i(k)]$ is the solution of the following fixed-point equation:

$$\begin{aligned} x_i^* &= \alpha_i w_i(u_i) + (1 - \alpha_i) \sum_{\ell \in \mathcal{S}} p_{i\ell} x_\ell^*, \quad i \in S, \\ x_i^* &= \sum_{\ell \in \mathcal{S}} p_{i\ell} x_\ell^*, \quad i \in S_1, \\ x_i^* &= h(i), \quad i \in S_0. \end{aligned}$$

For all $u \in \mathbb{R}_+^I$, let $W(u) \in [0, 1]^I$ be a vector-valued function where the i -th element, $W_i(u)$ is equal to $1_{i \in S} \alpha_i w_i(u_i) + 1_{i \in S_0} h(i)$. Let A be a $I \times I$ substochastic matrix where its ij -entry is equal to $a_{ij} = (1 - 1_{i \in S_0})(1 - 1_{i \in S} \alpha_i) p_{ij}$. The solution $x^* = [x_1^*, \dots, x_I^*]$ of the fixed-point equation, is given by:

$$x^* = (Id - A)^{-1} W(u), \quad (1)$$

with Id being the identity matrix with appropriate dimension depending on the context.

Optimization problem: The goal of the planner is to maximize the sum of the valuations when k goes to infinity, i.e., $\sum_{i \in S} x_i^*$, by controlling u_i , under the resource constraint $\sum_{i \in S} u_i \leq M$. Here $0 < M < |S|$ is a prescribed bound. Equivalently, the objective of the planner is to find $u^* = (u_i^*)_{i \in S}$, the solution of the following optimization problem:

$$u^* = \arg \max_{u_i \in \mathbb{R}_+, \forall i \in S} 1^T (Id - A)^{-1} W(u), \quad (2)$$

subject to:

$$\sum_{i \in S} u_i \leq M. \quad (3)$$

To compute u^* , we can use the gradient descent algorithm:

$$u_i(k+1) = \Gamma \left(u_i(k) + \frac{1}{k} 1^T (Id - A)^{-1} \frac{\partial}{\partial u_i} W(u(k)) \right), \quad (4)$$

with $\frac{\partial}{\partial u_i} W(u(k)) = [1_{j \in S} \alpha_j \frac{\partial w_j}{\partial u_i}(u_j)]_{j \in S}$. Our object is to do so in a data-driven manner using ideas from reinforcement learning and MCMC. In this paper we assume that the matrix P is *unknown*. The *known parameters* are the set of agents \mathcal{S} , the vectors $\alpha := [\alpha_i]_{i \in S}$, $h := [h_i]_{i \in S_0}$, the functional vector $w = [w_i]_{i \in S}$ and the budget M . The matrix P is *unknown*. At each time step $k \in \mathbb{N}_+$, the planner chooses a vector $u(k) = [u_i(k)]_{i \in S}$, then simultaneously, agent i is activated w.p. q_i , and polls agent j probabilistically as described earlier and observes her opinion. The planner *observes* this communication between i and j . The objective of the planner is to find an algorithm such that $\lim_{k \rightarrow +\infty} u(k) = u^*$.

IV. REINFORCEMENT LEARNING SCHEME

A. Equivalent controlled Markov chain

Consider an \mathcal{S} -valued controlled Markov chain $\{Y_n\}$ with controlled transition probabilities

$$\begin{aligned} q_{ij}(u) &:= p_{ij}, & i \in S_1 \cup S, \\ &:= \delta_{ij}, & i \in S_0. \end{aligned}$$

Here δ_{ij} is the Kronecker delta. Thus the states in S_0 are absorbing states. Also note that the transition probabilities are independent of the control choice, which affects only the running reward. Let $\tau := \min\{n \geq 0 : Y_n \in S_0\}$ denote the first passage time to S_0 . We associate with a state-control pair (i, u_i) an instantaneous cost $w_i(u)$ and a state-dependent discount factor $(1 - \alpha_i)$. Note that $\alpha_i = 0$ and $w_i(u) = 0$ for all $i \in S_0 \cup S_1$. Suppose $u_i(k) = v(i)$ for some $v : S \mapsto [0, 1]$, i.e., a ‘stationary Markov policy’ in Markov decision theoretic parlance [6]. Let $\bar{x}_i(k) := E[x_i(k)] \forall i, k$. Then $\{\bar{x}_i(k)\}$ satisfy the dynamics

$$\begin{aligned}\bar{x}_i(k+1) &= \alpha_i w_i(u_i(k)) + (1 - \alpha_i) \sum_{\ell} p_{i\ell} \bar{x}_\ell(k), \quad i \in S, \\ \bar{x}_i(k+1) &= \sum_{\ell} p_{i\ell} \bar{x}_\ell(k), \quad i \in S_1.\end{aligned}$$

The matrix $P := [[p_{ij}]]_{i,j \in S \cup S_0}$ is substochastic, hence the above linear system of equations is stable. Then as $k \uparrow \infty$,

$$\bar{x}_i(k) \rightarrow \bar{x}_i(\infty),$$

where $\bar{x}(\infty)$ satisfies the equation

$$\begin{aligned}\bar{x}_i(\infty) &= \alpha_i w_i(u_i(k)) + (1 - \alpha_i) \sum_{\ell \in S} p_{i\ell} \bar{x}_\ell(\infty), \quad i \in S, \\ \bar{x}_i(\infty) &= \sum_{\ell \in S} p_{i\ell} \bar{x}_\ell(\infty), \quad i \in S_1.\end{aligned}$$

This equation can be rewritten as:

$$\begin{aligned}\bar{x}_i(\infty) &= \alpha_i w_i(u_i(k)) + (1 - \alpha_i) \sum_{\ell \in S \cup S_1} p_{i\ell} \bar{x}_\ell(\infty) \\ &\quad + \sum_{\ell' \in S_0} p_{i\ell'} h(\ell'), \quad i \in S, \\ \bar{x}_i(\infty) &= \sum_{\ell \in S \cup S_1} p_{i\ell} \bar{x}_\ell(\infty) + \sum_{\ell' \in S_0} p_{i\ell'} h(\ell'), \quad i \in S_1.\end{aligned}$$

By standard ‘one step analysis’, one sees that $\bar{x}_i(\infty)$ has the representation

$$\begin{aligned}\bar{x}_i(\infty) &:= E_i \left[\sum_{m=0}^{\tau-1} \left(\prod_{k=0}^{m-1} (1 - \alpha_{Y_k}) \right) \alpha_{Y_m} w_{Y_m}(u(m)) \right. \\ &\quad \left. + \left(\prod_{k=0}^{\tau} (1 - \alpha_{Y_k}) \right) h(Y_\tau) \right].\end{aligned}$$

This suggests that we can view the opinion dynamics as the value iteration for evaluating a fixed stationary Markov policy $v(\cdot)$ for the controlled Markov chain $\{Y_n\}$, the objective being

$\sum_{i \in S \cup S_1} \bar{x}_i(\infty)$. This will be recognized as a discounted reward for the *stochastic shortest path problem* (‘Longest path problem’, to be precise, since we are maximizing a reward rather than minimizing a cost. The equivalent stochastic shortest path problem corresponds to viewing $-w_i(\cdot)$ as a running cost function and $-h(\cdot)$ as the terminal cost.)

We do, however, have an additional constraint (3). This is hard to incorporate in a Markov decision process as a constraint on controls, because it couples actions across different states in a manner unrelated to the dynamics (i.e., without regard to, e.g., how often they are visited). This puts it beyond the reach of traditional dynamic programming based computations such as value or policy iteration, or linear programming version of the dynamic program. Therefore we treat this as a parametric optimization problem over the parameters u_i ’s instead of as a control problem - this will become apparent from the algorithms we propose. The ‘uncontrolled’ but parameter-dependent ‘dynamic programming’¹ equation is given by standard arguments, as the linear system

$$V(i) = \alpha_i w_i(u_i) + (1 - \alpha_i) \sum_j p_{ij} V(j), \quad i \in S, \quad (5)$$

$$V(i) = \sum_j p_{ij} V(j), \quad i \in S_1, \quad V(i) = h(j), \quad i \in S_0. \quad (6)$$

B. First algorithm

Let $k \in \mathbb{N}_+$ be k^{th} time an agent polls another. A gradient based learning scheme for this problem is as follows. Let

$$I\{Y_n = i\} = \begin{cases} = 1 & \text{if } Y_n = i, \\ = 0, & \text{if } Y_n \neq i, \end{cases} \quad \nu(i, n) := \sum_{m=0}^n I\{Y_m = i\}.$$

for $n \geq 0$. Then $\nu(i, n), n \geq 0$, can be interpreted as a ‘local clock’ at agent i , counting its own number of updates till ‘time’ (i.e., the overall iterate count) n . Pick stepsize sequences $\{a(k)\}, \{b(k)\} \subset (0, \infty)$ such that

$$\begin{aligned} \sum_k a(k) &= \sum_k b(k) = \infty, \quad \sum_k (a(k)^2 + b(k)^2) < \infty, \\ \frac{b(k)}{a(k)} &\rightarrow 0. \end{aligned} \quad (7)$$

We shall also make the following additional assumptions on $\{a(k)\}$:

¹‘one step analysis’, to be precise

- 1) $a(n+1) \leq a(n)$ from some n onwards;
- 2) there exists $r \in (0, 1)$ such that $\sum_n a(n)^{1+q} < \infty$, $q \geq r$;
- 3) for $x \in (0, 1)$, $\sup_n \left(\frac{a(\lfloor xn \rfloor)}{a(n)} \right) < \infty$, where $\lfloor \cdot \rfloor$ stands for the integer part of ‘ \cdot ’;
- 4) for $x \in (0, 1)$ and $A(n) := \sum_{m=0}^n a(m)$, $\lim_{n \rightarrow \infty} \left(\frac{A(\lfloor yn \rfloor)}{A(n)} \right) = 1$ uniformly in $y \in [x, 1]$.

These conditions are satisfied, e.g., by the popular stepsize $a(n) = \frac{1}{n+1}$, $n \geq 0$.

The algorithm then is as follows. For $k \geq 0, i, j \in S$, do: for a prescribed state i_0 ,

$$\begin{aligned} \Psi_{ij}(k+1) &= \Psi_{ij}(k) + a(\nu(i, k)) I\{Y_k = i\} \times \\ &\quad \left[\alpha_i w'_i(u_i(k)) \delta_{ij} + (1 - \alpha_i) \Psi_{Y_{k+1}j}(k) \right. \\ &\quad \left. - \Psi_{ij}(k) \right], \end{aligned} \tag{8}$$

$$u_i(k+1) = \Gamma(u_i(k) + b(k) \sum_j \Psi_{ji}(k)), \tag{9}$$

$$\Psi_{ij}(k) = 0, \quad i \in S_0. \tag{10}$$

Here $\Gamma(\cdot)$ is the projection onto the simplex

$$\{x = [x_1, \dots, x_{|S|}]^T : x_i \geq 0 \forall i, \sum_i x_i \leq M\}.$$

This is a gradient-based reinforcement learning scheme which is better suited for our purposes than, e.g., the classical Q-learning scheme of [1]. This is because it allows us to treat the optimization over control parameters as parametric optimization which can handle the constraint

(3). The explanation of this scheme is as follows:

- The iteration (8) estimates the partial derivatives $\frac{\partial V(i)}{\partial u_j}$ by $\Psi_{ij}(k), k \geq 0$. This is arrived at by considering the constant policy dynamic programming equation

$$\tilde{V}(i) = \alpha_i w_i(u_i) + (1 - \alpha_i) \sum_j p_{ij} \tilde{V}(j), \tag{11}$$

for $i \in S$. Differentiating both sides w.r.t. u_j , we see that $\Phi_{ij} := \frac{\partial V(i)}{\partial u_j}$ satisfy

$$\Phi_{ij} = \alpha_i w'_i(u_i) \delta_{ij} + (1 - \alpha_i) \sum_\ell p_{i\ell} \Phi_{\ell j}, \tag{12}$$

for $i \in S$, with $\Phi_{ij} = 0$ for $i \in S_0$. The iteration (8) then is the stochastic approximation scheme to solve this equation.

- Iteration (9) operating on a slower time scale (in view of (7)), constitutes a stochastic gradient ascent. That is, (9) is a stochastic gradient ascent over the control variables which takes the outputs $\{\Psi_{ij}(k)\}$ of (8) as estimates of the relevant partial derivatives and summing

them over the first index, generates an estimate of the corresponding partial derivative of the reward itself. In turn, the application of the projection $\Gamma(\cdot)$ makes it a projected stochastic gradient scheme which also imposes the constraint (3).

The chain $\{Y_n\}$, however, is an imaginary object. To map this scheme back to our original framework, let Z_n be the index of the agent that updated its valuation at time n . In other words, it is the Z_n -th component $x_{Z_n}(n)$ that got updated at time n , the rest were left unperturbed. Also suppose that this was done by the Z_n -th agent by polling a neighbor \tilde{Z}_n according to the transition probabilities q_{z_n} defined above.

The iteration (8) of the above scheme can then be written for our original framework as

$$\begin{aligned} \Psi_{ij}(k+1) = & \Psi_{ij}(k) + a(\nu(i, k))I\{Z_k = i\} \times \\ & \left[(\alpha_i w'_i(u_i(k))\delta_{ij} + (1 - \alpha_i)\Psi_{\tilde{Z}_k j}(k) - \Psi_{ij}(k) \right]. \end{aligned} \quad (13)$$

The third iteration, i.e., (9), remains unaltered, as do the boundary conditions for the Ψ_i 's.

Now that we no longer have to think of Z_k, \tilde{Z}_k as realizations of a single trajectory of a Markov chain, we can generalize this further and let Z_k be a subset of S . For each $i \in Z_k$, we generate a random variable \tilde{Z}_k^i according to the probability distribution q_i above. The iteration (14) then gets replaced by

$$\begin{aligned} \Psi_{ij}(k+1) = & \Psi_{ij}(k) + a(\nu(i, k))I\{i \in Z_k\} \times \\ & \left[(\alpha_i w'_i(u_i(k))\delta_{ij} + (1 - \alpha_i)\Psi_{\tilde{Z}_k^i j}(k) - \Psi_{ij}(k) \right]. \end{aligned} \quad (14)$$

In particular when $Z_n = S$, it is a completely synchronous iteration. Note that $\nu(i, n)$ has to be defined now as $\nu(i, n) := \sum_{m=0}^n I\{i \in Z_m\}$.

C. Stochastic gradient scheme

Along the line of the previous algorithm, we define a new algorithm, where instead of having a biased but consistent estimator of the gradient, we can derive a sampling scheme that will provide, at each iteration, an unbiased estimator of the gradient. Recall the Z_n, \tilde{Z}_n defined in the preceding section. There we had considerable freedom in choosing how Z_n is generated, the key requirement was that \tilde{Z}_n should have the prescribed conditional law given Z_n . This is because the algorithm at each step calls for a single transition executed according to the given transition matrix. That is, one has to generate a pair of random variables with the conditional law of the latter given the former completely specified and the (marginal) law of the former having full

support at each step. In the algorithm we propose below (with its natural extension), however, we require at each step a path of random duration in \mathcal{S} . Generating pairs (Z_n, \tilde{Z}_n) as before does not provide that. Hence unlike the previous scheme with full observations, we now need a probing mechanism. Thus we define Z_n as before and do (for each fixed n): Let $\delta_{..}$ denote the Kronecker delta.

- 1) For each $i \in S$, set $m = 0$ and set $Y_{j_0} = Z_n = j$ (say). Here $j \in \mathcal{S} \setminus S_0$ can be picked uniformly at random. Initialize $\xi_{ji}(n) = 0$.
- 2) With probability $\alpha_{Y_{j_0}} = \alpha_j$ ($\alpha_j = 0$ if $j \in S_1$), stop and set $\xi_{ji}(n) \rightarrow \xi_{ji}(n) + \delta_{ji}$. If not,
- 3) with probability $(1 - \alpha_{Y_{j_0}})p_{Y_{j_0}k}$, continue by setting $Y_{j_1} = k$, $\xi_{ji}(n) \rightarrow \xi_{ji}(n)$.
- 4) At step m , stop if $Y_{j_m} \in S_0$. If not, stop with probability $\alpha_{Y_{j_m}}$ and set $\xi_{ji}(n) \rightarrow \xi_{ji}(n) + \delta_{Y_{j_m}i}$, or else continue with probability $(1 - \alpha_{Y_{j_m}})$ by setting $Y_{j(m+1)} = \ell$ with probability $(1 - \alpha_{Y_{j_m}})p_{Y_{j_m}\ell}$.
- 5) Repeat 4) above for $m \geq 1$ till stopping.
- 6) Perform the following gradient descent step:

$$u_i(n+1) = \Gamma \left(u_i(n) + a(n)w'_i(u_i(n)) \sum_i \xi_{ij}(n) \right). \quad (15)$$

An alternative scheme is:

- 1) For each $i \in \mathcal{S}$, and for each $j \in S$, set $m = 0$ and $Y_{i_0} = i$, kept fixed for this run. Initialize $\zeta_i = 1$. Set

$$\xi_{ij} \rightarrow \xi_{ij} + \zeta_i \delta_{Y_{i_0}j} \alpha_{Y_{i_0}}.$$

Continue by setting $Y_{i_1} = k$ with probability $p_{Y_{i_0}k}$.

- 2) At step m , stop if $Y_{i_m} \in S_0$. If not, set

$$\zeta_i \rightarrow \zeta_i(1 - \alpha_{Y_{i(m-1)}}), \quad \xi_{ij} \rightarrow \xi_{ij} + \zeta_i \delta_{Y_{i_m}j} \alpha_{Y_{i_m}},$$

and continue by setting $Y_{i(m+1)} = \ell$ with probability $p_{Y_{i_m}\ell}$.

- 3) Repeat 2) above for $m \geq 1$ till stopping. Freeze ξ_{ij} on stopping and label it $\xi_{ij}(n)$.
- 4) Perform the following gradient descent step:

$$u_i(n+1) = \Gamma \left(u_i(n) + a(n)w'_i(u_i(n)) \sum_j \xi_{ji}(n) \right). \quad (16)$$

By construction, for the two sampling schemes, $w'_i(u_i(n))E[\sum_\ell \xi_{i\ell}(n)]$, is the solution of the linear system (12). Therefore, the previous scheme will converge to the optimal u^* as long as

the variance of $\xi_{ij}(n)$ is bounded for all k [7]. For the stochastic gradient iterate (16), a good step-size in this context is $a(n) = A/(\lceil \frac{n}{M} \rceil)$ for some $A > 0$ and $M \geq 1$.

D. An alternative learning scheme

The problem with the above scheme is that it involves all agents in \mathcal{S} , which may lead to a curse of dimensionality. Worse, it requires that all communications between agents be observed. It makes sense to assume that only a few agents can be monitored. These should include in particular those in S . Without any loss of generality, we assume that only the updates of agents in S are observed. The algorithm we propose next and its analysis extend easily to the case when a few uncontrolled agents are also observed (by using, e.g., the trivial device of setting $\alpha_i \equiv 0$ for such agents). Then it also makes sense that we should treat $S^* := S \cup S_0$ as our effective state space for the algorithm. By analogy to the above stochastic shortest path formulation, consider an \mathcal{S} -valued Markov chain $\{Y_n\}$ with transition probabilities $\{p_{ij}\}$. If we restrict $\{Y_n\}$ to S^* , it means that we observe only $\{Y_{T_n}\}$ where $T_n, n \geq 0$, are the successive return times of $\{Y_n\}$ to S^* define recursively by

$$\begin{aligned} T_0 &:= \min\{m \geq 0 : Y_m \in S^*\}, \\ T_{n+1} &:= \min\{m > T_n : Y_m \in S^*\}, \quad n \geq 0. \end{aligned}$$

The chain eventually gets absorbed into S_0 as before. Strictly speaking, if we keep track of the T_n 's as well, it is a *semi-Markov* process. Exercising control only when the chain is in S^* leads to a supervisory control problem as in [12], albeit with a different reward structure compared to theirs. Nevertheless, we do not need to view it in this manner. This is because our controlled Markov chain is an artifact, the actual process is the simple averaging or ‘gossip’ dynamics. Thus the actual values of T_n 's are irrelevant for us and we can work with the chain $Y_n^* := Y_{T_n}, n \geq 0$. Let $q^*(j|i), i, j \in S^*$, denote the probability that $Y_{n+1}^* = j$ given $Y_n^* = i$. It is of the form

$$q^*(j|i) = p_{ij} + \sum_{j \neq \ell \in S'} p_{i\ell} \varphi(j|\ell)$$

where $\varphi(j|\ell) := P(Y_\zeta^* = j | Y_0^* = \ell)$, for $\zeta := \min\{n \geq 0 : Y_n^* \in S\}$. In particular, $\varphi(\cdot|\cdot)$ is independent of the control choice u . After the chain leaves state i , it does not hit any other controlled state before hitting another state (j above) in S^* , so the associated running cost is $\alpha_i w_i(u)$ as before. We now consider the restricted reward $\sum_{i \in S} x_i(\infty)$ which is *not* the same as the original, so this is an approximation. The advantage of this reward is that it is expected to

be positively correlated with the full reward, i.e., increase in the former should lead to increase in the latter. More importantly, it depends only on observed quantities. This passage is purely heuristic and avoids in particular having to contend with the full complications of the ‘partial observations’ framework. The associated (constant policy) dynamic programming equation is then given by

$$V(i) = \alpha_i w_i(u_i) + (1 - \alpha_i)(p_{ij} + \sum_{j, \ell \in S^*: j \neq \ell} p_{i\ell} \varphi(j|\ell))V(j), \quad (17)$$

$$V(i) = h(i), \quad i \in S_0. \quad (18)$$

One could write down a reinforcement learning scheme for approximate solution of (17)-(18) along the lines of the preceding subsections, but the situation is much more difficult here. The problem is similar to the one faced in the stochastic gradient scheme above. We require a path from one state in S^* to another, passing through a possibly nonempty set of unobserved states in $S \setminus S^*$. Again, generating pairs (Z_n, \tilde{Z}_n) as before does not provide that. We now need a probing mechanism. Thus we define Z_n as before, but when node $Z_n = i \in S^*$ polls a neighbor $i_1 \in S$, it passes to i_1 a time-stamped token tagged with i . The node i_1 , if not in S^* , does likewise, but retaining the original tag and time stamp. This continues till the token reaches some $j \in S^*$. Then set $\tilde{Z}_n = j$. The corresponding reinforcement learning scheme now becomes

$$\begin{aligned} \Psi_i(k+1) &= \Psi_i(k) + a(\nu(i, k))I\{i \in Z_k\} \times \\ &\quad \left[(\alpha_i w'_i(u_i(k)) + (1 - \alpha_i)\Psi_{\tilde{Z}_k^i}(k) \right. \\ &\quad \left. - \Psi_{ij}(k) \right], \end{aligned} \quad (19)$$

$$u_i(k+1) = \Gamma\left(u_i(k) + b(k)\Psi_i(k)\right), \quad (20)$$

$$\Psi_i(k) = 0, \quad i \in S_0. \quad (21)$$

V. CONVERGENCE ANALYSIS

The convergence analysis of the first scheme goes along standard lines, essentially piecing together known facts from the theory of two time scale and distributed asynchronous stochastic approximation. With this in mind, we only sketch it in outline. To begin with, note that condition (7) implies that the iterates (20) move on a slower, in fact asymptotically negligible, time scale compared to (14). Hence they can be viewed as quasi-static, i.e., $u_i(k) \approx u_i \forall i$, for purposes

of analyzing (14) ([7], Section 6.1). Then (14) constitutes a stochastic approximation scheme to estimate the partial derivatives of V^* w.r.t. the u_i 's by solving (12), which has a unique solution. Its convergence to this solution follows from the theory of asynchronous stochastic approximation developed in [5], wherein the conditions we have imposed on $\{a(n)\}$ play a crucial role.

But this is under the assumption that $u_i(k) \approx u_i \forall i$, whereas in reality the $u_i(k)$'s are changing on a slower time scale. Thus what the foregoing entails in reality is that

$$\Psi_{ij}(k) - \frac{\partial V^*(i)}{\partial u_j} \Big|_{u.=u.(k)} \rightarrow 0$$

a.s. $\forall i, j$, i.e., Ψ_{ij} 's track the corresponding partial derivatives of V^* with an asymptotically negligible error, as desired. Thus (20) is a legitimate stochastic gradient ascent scheme. We need the following lemma.

Lemma 1 The solution $V(\cdot)$ of the constant policy dynamic programming equation (17)-(18) is componentwise concave and continuous in the variables $\{u_i\}$.

Proof (Sketch) This follows by considering the associated constant policy value iteration and using induction, along with the fact that pointwise limits of concave functions are concave and uniform limits of continuous functions are continuous. The details are routine, see, e.g., [2]. \square

Our main result then is the following:

Theorem 1 The above learning policy is asymptotically optimal, a.s.

Proof This is immediate from the fact that the projected stochastic gradient ascent for a concave function on a compact interval converges to the set of its global maxima a.s. (see, e.g., [7], Chapter 10). \square

The stochastic gradient scheme above, by virtue of (12), is already of the form

$$u_i(n+1) = u_i(n) + a(n)I\{Z_n = i\} \left[\frac{\partial V(i)}{\partial u_i}(u(n)) + M_i(n+1) \right],$$

where $M(n) := [M_1(n), M_2(n), \dots]^T$ is a martingale difference sequence. That is, it is a classical asynchronous stochastic gradient scheme with a.s. convergence to a local maximum, which is also a global maximum by concavity of $V(i)$, under reasonable conditions on $\{M(n)\}$ – see Chapter 10 of [5].

Finally, the ‘alternative scheme’ above based on observing few nodes is of the same form as the reinforcement learning scheme above and is analyzed exactly the same way.

VI. A MORE GENERAL MODEL

We can also consider the situation where α_i 's depend on the control choice u_i at $i \in S$. We shall illustrate the changes for the second, i.e., the improved learning scheme above, the situation for the first scheme being completely analogous. Thus the 'dynamic programming equations' become

$$\begin{aligned} V(i) &= \alpha_i(u_i)w_i(u_i) + \\ &\quad (1 - \alpha_i(u_i))(p_{ij} + \sum_{\ell \in S'} p_{i\ell}\varphi(j|\ell))V(j), \\ V(i) &= h(i), \quad i \in S_0, \end{aligned}$$

and the corresponding reinforcement learning scheme is

$$\begin{aligned} V_i(k+1) &= V_i(k) + a(\nu(i, k))I\{i \in Z_k\} \times \\ &\quad \left[\alpha_i(u_i(k))w_i(u_i(k)) + \right. \\ &\quad \left. (1 - \alpha_i(u_i(k)))V_{\tilde{Z}_k^i}(k) - V_i(k) \right], \end{aligned} \quad (22)$$

$$\begin{aligned} \Psi_{ij}(k+1) &= \Psi_{ij}(k) + a(\nu(i, k))I\{i \in Z_k\} \times \\ &\quad \left[(\alpha_i(u_i(k))w'_i(u_i(k)) + \alpha'_i(u_i(k))w_i(u_i(k)))\delta_{ij} \right. \\ &\quad \left. - \alpha'_i(u_i(k))\delta_{ij}V_{\tilde{Z}_k^i}(k) \right. \\ &\quad \left. + (1 - \alpha(u_i(k)))\Psi_{\tilde{Z}_k^i, j}(k) - \Psi_{ij}(k) \right], \end{aligned} \quad (23)$$

$$+ (1 - \alpha(u_i(k)))\Psi_{\tilde{Z}_k^i, j}(k) - \Psi_{ij}(k) \Big], \quad (24)$$

$$u_i(k+1) = \Gamma\left(u_i(k) + b(k) \sum_j \Psi_{ji}(k)\right), \quad (25)$$

$$V_k(i) = h(i), \quad i \in S_0, \quad \Psi_{ij}(k) = 0, \quad i \in S_0.$$

The convergence analysis applies as before except for the fact that we can no longer claim concavity and convergence only to a local minimum can be guaranteed. This could be improved, e.g., by resorting to simulated annealing for the slow time scale iterates, i.e., replacing them by

$$\begin{aligned} u_i(k+1) &= \Gamma\left(u_i(k) + b(k) \sum_j \Psi_{ji}(k) \right. \\ &\quad \left. + \frac{C}{\sqrt{1/b(k) \log \log(c(k))}} W_{k+1}\right), \end{aligned} \quad (26)$$

where $\{W_k\}$ are IID $N(0,1)$, $C > 0$ is a suitably chosen constant as in [13]. The difference with the previous scheme is that (11) gets replaced by

$$\tilde{V}(i) = \alpha_i(u_i)w_i(u_i) + (1 - \alpha_i(u_i)) \sum_j p_{i\ell} \tilde{V}(\ell). \quad (27)$$

Differentiating through with respect to u_i in (12) and after replacing α_i by $\alpha(u_i)$, we have the additional term $\alpha'_i(u_i)(w_i(u_i) - \sum_{\ell} p_{i\ell} \tilde{V}(\ell))$ on the right hand side. The second and third term on the right had side of (24) correspond to these additional terms. As this involves $\tilde{V}(\cdot)$ as well unlike the previous scheme which did not, one needs the additional iteration (22) to estimate it, this being the stochastic approximation scheme to solve (11).

For comparison purposes later on in the numerical section, we also state the iterations in the case where we know the matrix P explicitly. Then the only difference would be in the update of $V_i(k)$ and $\Psi_{ij}(k)$ which will follow the following scheme:

$$\begin{aligned}
V_i(k+1) &= V_i(k) + a(k) \times \\
&\quad \left[\alpha_i(u_i(k))w_i(u_i(k)) + \right. \\
&\quad \left. (1 - \alpha_i(u_i(k))) \sum_{l \in \mathcal{S}} p_{il} V_l(k) - V_i(k) \right], \\
\Psi_{ij}(k+1) &= \Psi_{ij}(k) + a(k) \times \\
&\quad \left[(\alpha_i(u_i(k))w'_i(u_i(k))\delta_{ij} + \right. \\
&\quad \left. [(\alpha'_i(u_i(k))w_i(u_i(k)) - \alpha'_i(u_i(k)))\delta_{ij} \times \right. \\
&\quad \left. \sum_{l \in \mathcal{S}} p_{il} V_l(k), \right. \\
V_k(i) &= h(i), \quad i \in S_0, \quad \Psi_{ij}(k) = 0, \quad i \in S_0.
\end{aligned} \tag{28}$$

VII. NUMERICAL EXPERIMENTS

We select three real-world networks for our evaluation. The three networks are Karate, Macaque, Rfid (see Table I) and have from 34 to 75 nodes and from 78 to 2278 links. Each network was retrieved from the R package `igraphdata` [19]. The numerical experiments reported here are for the synchronous case, i.e., all components are updated each time. The results are compared with the exact solution computed off-line using gradient descent described in (4).

Inputs: the matrix P , the number of agents in each set (S, S_1, S_0), the upper bound in the resource constraint M , the number of iterations, the function $w(\cdot)$ and finally the parameters A, B and $denom$ of our step-size functions $a(k) = \frac{A}{\lceil (1+k \log(1+k))/denom \rceil}$ and $b(k) = \frac{B}{\lceil k/denom \rceil}$.

Construction of P : Given an adjacency matrix A , which can be weighted or not, we transform this matrix in a stochastic matrix by dividing each row by the sum of its elements. This matrix is our communication matrix P .

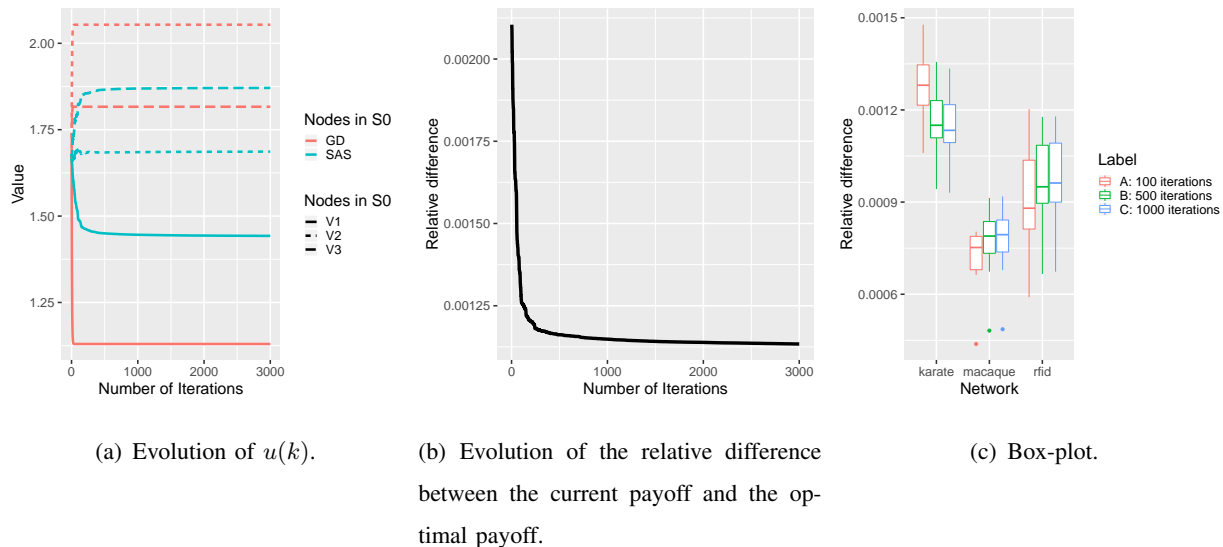


Fig. 1. Simulations for the improved learning scheme.

Initial setting: First we specify the number of agents in each set (S , S_1 , and S_0) and then randomly allocate an agent to a given set. We assume that $\alpha_i = \alpha$ for each $i \in S_0 \cup S$ and $\alpha_i = 0$ for all $i \in S_1$. For each $i \in S_0$, $h(i)$ is sampled from a uniform distribution. In our simulations, $\alpha = 0.6$, $M = 5$, $A = 0.6$, $B = 0.6$, $denom = 100$ and $w(x) = \frac{x}{x+0.1}$.

Network	Karate	macaque	rfid
Number of nodes	34	45	75
Number of edges	78	463	2278

TABLE I

DESCRIPTION OF THE NETWORKS

Convergence for the Karate network. In the first numerical study, we are interested in understanding the convergence of the stochastic approximation scheme and the stochastic gradient to the optimal strategy. We restrict this study to the Karate network. Later on, we shall extend it to the remaining networks. In Figure 2(a), the x-axis denotes the number of iterations and the y-axis captures the evolution of $u(k)$ for the stochastic approximation/reinforcement based scheme (8)-(10). We will abbreviate the name of this scheme by SAS (for stochastic approximation scheme). In Figure 2(b), the x-axis denotes the number of iterations and the y-axis captures the evolution of $u(k)$ for the stochastic gradient (SGD) with the two sampling schemes (15) and (16). In the two figures, the red curve captures the evolution of $u(k)$ using the gradient descent

(GD). The number of controlled agents is equal to 3. Twenty-eight agents belong to S_1 and three agents are in S_0 . In Figure 2(a), before 7500 iterations, we can observe that the gradient descent algorithm already converges and the reinforcement learning scheme did not. In fact the SGD seems to converge faster (see Figure 2(b) after 2500 iterations). However, we observe that the variance over the iterates of SGD is higher than the SAS. The tradeoff therefore is between speed and fluctuations. Moreover we can observe in Figure 2 that one iteration of the SAS is much faster than the ones of the two SGD algorithms. Therefore there is a clear tradeoff between the complexity of a single iteration and the number of iterations, so the latter cannot be the sole basis for comparison.

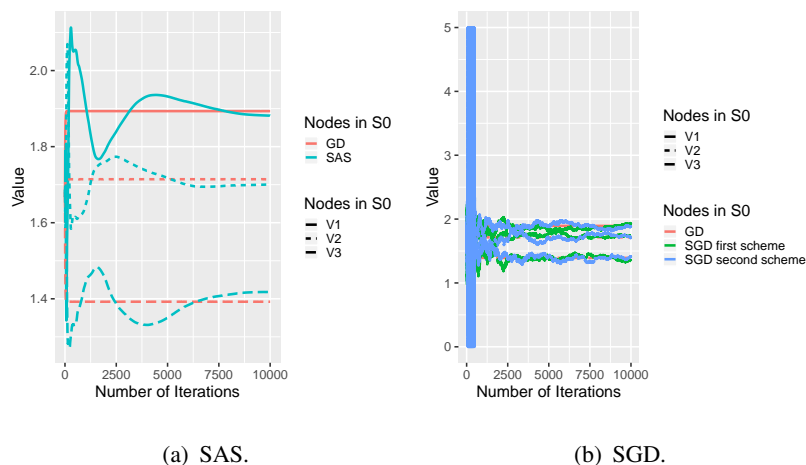


Fig. 2. Evolution of $u(k)$ for each algorithm.

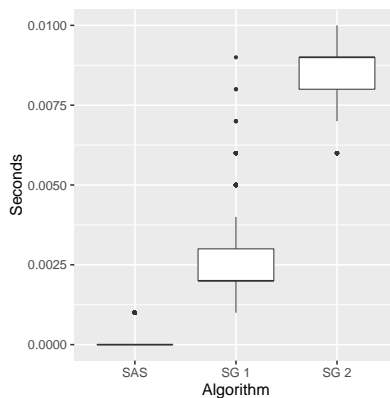


Fig. 3. Boxplot for the time (in seconds) to perform one iteration of each algorithm.

Extension to other networks: The second numerical study applies the same schemes to the other datasets and observes whether or not the same conclusions apply. We do not present the

SGD with the second sampling scheme because the conclusion are similar. In Figure 4(a) and in Figure 4(b), we perform 10 simulations of the stochastic approximation scheme and stochastic gradient for each network. The performance measure on y -axis is the relative difference between the optimal payoff and the current payoff generated by $u(k)$ at iteration k . For the SAS, we observe that for each network, even if we stop the stochastic approximation after 100 iterations, the third quantile will have a relative difference lower than 1%. For each network, when we use the stochastic gradient, we note that relative difference is much lower that for the SAS. The last observation highlights the fact that when the number of iterations is low (under 1000 in this case), the SAS uses a biased estimator of the gradient compared to the stochastic gradient and therefore has lower performance.

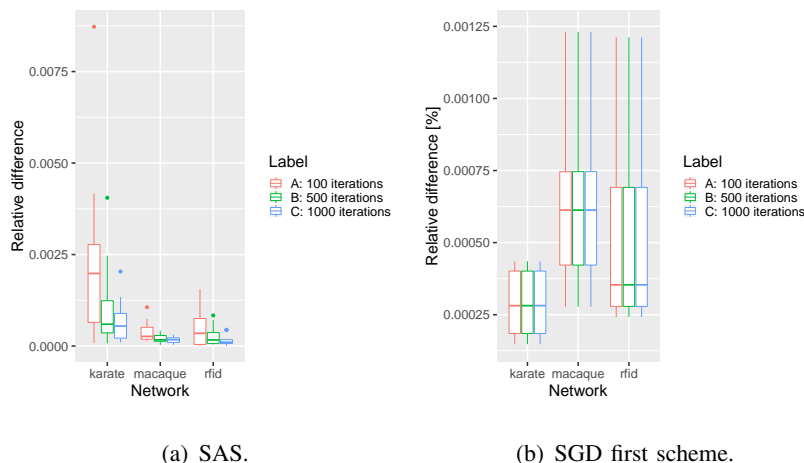


Fig. 4. Box-plot of the relative difference between the payoff obtained at k and the optimum over 10 simulations for the stochastic approximation scheme and the stochastic gradient descent.

Study of the improved learning scheme: In the third numerical study, we are interested in understanding how the second learning scheme compares with the first. The main difference between the two algorithms is that in the first one you have to observe the communication between all the agents and in the improved one, you can only observe a part of it. In order to be able to compare with the previous simulations, we assume the following: The set of controlled agents is the same (S is the same). Only 50% of the agents in S_0 and S_1 are observed. The results are depicted in Figures 1(a), 1(b) and 1(c). In Figure 1(a), we note that the improved stochastic approximation scheme already converges after a number of iterations less than 3000. The convergence is not to the optimal one but in this case, we can observe that in Figure 1(b), the relative difference of the current payoff and the optimal is below 0.1%, therefore nearly optimal.

We can conclude that even if the improved stochastic approximation does not converge to the optimal u^* , the strategy reached is already quite good. We can observe a similar conclusion in Figure 1(c) for the remaining networks. These preliminary simulations encourage the use of the improved stochastic approximation scheme.

Study of the more general problem: The final numerical study is dedicated to the last reinforcement scheme based on annealing method for non-convex optimization. We restrict this study to the Karate network. We assume that $w_i(u_i) = h(i)$ for all $i \in \mathcal{S}$ and $u_i \in [0, M]$. Also in this simulation study for $i \in \mathcal{S}$, $\alpha_i(u_i) = \frac{u_i}{u_i+0.1}$ with $\#\mathcal{S} = 3$. The noisy term of (26) is parametrized by $c(k) := \lceil k/denom \rceil$ and $C = 10$. We study two schemes. The first one is reinforcement learning ((22), (24) and (26)). The second one is without the approximation of the reinforcement scheme for the computation of the gradient ((28) and (26)). We are interested in understanding how the first scheme tracks the behavior of the second scheme. In Figure 5(a) (resp. Figure 5(b)), we observe the first algorithm starts to track the trajectory of the second trajectory after 2000 iterations (15000 iterations). Moreover, we observe that in both cases, the two schemes are converging to the same values.

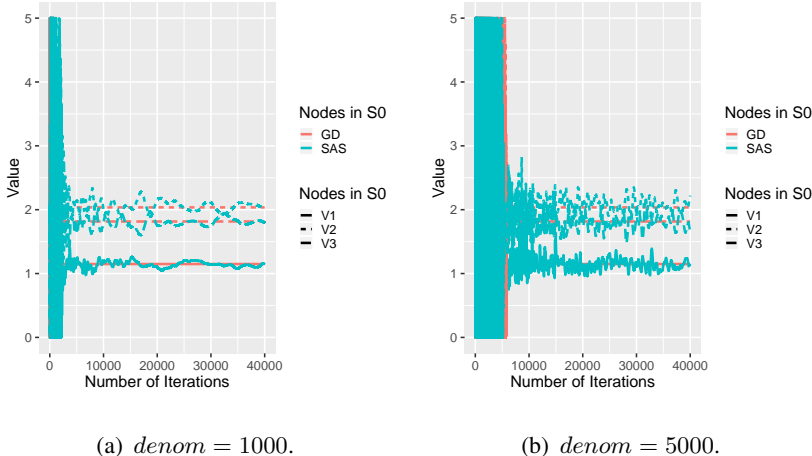


Fig. 5. Convergence of the annealing scheme with/without the reinforcement learning scheme.

VIII. FURTHER DIRECTIONS

Incorporating subjective risk measures: Since we are modeling social networks, it is desirable that we incorporate behavioral aspects into our model explicitly, such as the risk-measures suggested by behavioral economics. This makes the problem a lot harder, see, e.g., [21] for some initial efforts towards the dynamic programming aspects.

How to select the initial set of agents: One of the results of this paper is the fact that by observing a small number of agents, we can increase drastically the speed of convergence of the algorithm. Even if the obtained solution is suboptimal, the relative difference observed between the optimal payoff and the suboptimal one, in the simulation, was low (about 0.01%). Therefore, one interesting question would be to find a possible algorithm to choose the initial set of agents? This question can be related to the problem of selecting sensors k , among n potential sensors. In future work, we will try to adapt this well-known problem to our setting. See also a greedy scheme for agent selection with performance guarantees proposed in [8].

Other learning schemes: In our current scheme, we observe communications between a set of particular agents. In [3], the authors prove that agents in a social network can easily guess who is central in a diffusion process. Therefore a potential scheme would be to ask a small number of agents who they think is central in the network and factor this information into the opinion-shaping optimization problem.

Pricing scheme for accessing communication data: Accessing the data in the age of information is getting more and more important. Agents start to realize the value of their data. The question that we should ask in our setting is the following: how much should a planner pay an agent to access her information in order to be able to perform opinion shaping, i.e., design an incentive-compatible pricing mechanism for data acquisition.

ACKNOWLEDGMENT

The work of VSB was supported in part by a J. C. Bose Fellowship from the Department of Science and Technology, Government of India, and the project ‘*Machine Learning for Network Analytics*’ from the joint DST-INRIA program administered by the Indo-French Centre for Promotion of Advanced Research.

REFERENCES

- [1] Jinane Abounadi, Dimitri P Bertsekas, and Vivek Borkar. Stochastic approximation for nonexpansive maps: Application to q-learning algorithms. *SIAM Journal on Control and Optimization*, 41(1):1–22, 2002.
- [2] Mukul Agarwal, Vivek S Borkar, and Abhay Karandikar. Structural properties of optimal transmission policies over a randomly varying channel. *IEEE Transactions on Automatic Control*, 53(6):1476–1491, 2008.
- [3] Abhijit V Banerjee, Arun G Chandrasekhar, Esther Dufo, and Matthew O Jackson. Using gossips to spread information: Theory and evidence from two randomized controlled trials. *MIT Department of Economics Working Paper No. 14-15.*, 2017.

- [4] Kostas Bimpikis, Asuman Ozdaglar, and Ercan Yildiz. Competitive targeted advertising over networks. *Operations Research*, 64(3):705–720, 2016.
- [5] Vivek S Borkar. Asynchronous stochastic approximations. *SIAM Journal on Control and Optimization*, 36(3):840–851, 1998.
- [6] Vivek S Borkar. Convex analytic methods in markov decision processes. In *Handbook of Markov Decision Processes (A. Shwartz, E. Feinberg, eds.)*, pages 347–375. Springer, 2002.
- [7] Vivek S Borkar. *Stochastic Approximation: A Dynamical Systems Viewpoint*. Hindustan Publ. Agebcy, New Delhi, and Coambridge Uni. Press, Cambridge, UK, 2008.
- [8] Vivek S Borkar, Aditya Karnik, Jayakrishnan Nair, and Sanketh Nalli. Manufacturing consent. *IEEE Transactions on Automatic Control*, 60(1):104–117, 2015.
- [9] Morris H DeGroot. Reaching a consensus. *Journal of the American Statistical Association*, 69(345):118–121, 1974.
- [10] Elenna Dugundji and Joan Walker. Discrete choice with social and spatial network interdependencies: an empirical example using mixed generalized extreme value models with field and panel effects. *Transportation Research Record: Journal of the Transportation Research Board*, 1921:70–78, 2005.
- [11] Mehrdad Farajtabar, Nan Du, Manuel Gomez-Rodriguez, Isabel Valera, Hongyuan Zha, and Le Song. Shaping social activity by incentivizing users. In *Advances in Neural Information Processing Systems*, pages 2474–2482, 2014.
- [12] J-P Forestier and Pravin Varaiya. Multilayer control of large markov chains. *IEEE Transactions on Automatic Control*, 23(2):298–305, 1978.
- [13] Saul B Gelfand and Sanjoy K Mitter. Recursive stochastic algorithms for global optimization in \mathcal{R}^d . *SIAM Journal of Control and Optimization*, 29(5):999–1018, 1991.
- [14] Adrien Guille, Hakim Hacid, Cécile Favre, and Djamel A Zighed. Information diffusion in online social networks: A survey. *ACM SIGMOD Record*, 42(2):17–28, 2013.
- [15] David Kempe, Jon Kleinberg, and Éva Tardos. Influential nodes in a diffusion model for social networks. In *Automata, Languages and Programming*, pages 1127–1138. Springer, 2005.
- [16] Su-Chen Lin, Shou-De Lin, and Ming-Syan Chen. A learning-based framework to handle multi-round multi-party influence maximization on social networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 695–704. ACM, 2015.
- [17] Deepak Merugu, Balaji S Prabhakar, and N Rama. An incentive mechanism for decongesting the roads: A pilot program in bangalore. In *Proc. of ACM NetEcon Workshop*. Citeseer, 2009.
- [18] Alexandre Reiffers-Masson, Yezekael Hayel, and Eitan Altman. Posting behaviour dynamics and active filtering for content diversity in social networks. *IEEE transactions on Signal and Information Processing over Networks*, 3(2):376–387, 2017.
- [19] Gabor Scardi. igraphdata: A collection of network data sets for the igraph package. *R package version 1.0*, 1, 2015.
- [20] P Wesley Schultz, Jessica M Nolan, Robert B Cialdini, Noah J Goldstein, and Vladas Griskevicius. The constructive, destructive, and reconstructive power of social norms: Reprise. *Perspectives on Psychological Science*, 13(2):249–254, 2018.
- [21] Yun Shen, Wilhelm Stannat, and Klaus Obermayer. Risk-sensitive markov control processes. *SIAM Journal on Control and Optimization*, 51(5):3652–3672, 2013.
- [22] Joan L Walker, Emily Ehlers, Ipsita Banerjee, and Elenna R Dugundji. Correcting for endogeneity in behavioral choice models with social influence variables. *Transportation Research Part A: Policy and Practice*, 45(4):362–374, 2011.
- [23] Yichen Wang, Evangelos Theodorou, Apurv Verma, and Le Song. A stochastic differential equation framework for guiding online user activities in closed loop. *arXiv preprint arXiv:1603.09021*, 2016.

- [24] Bainan Xia, Hao Ming, Ki-Yeob Lee, Yuanyuan Li, Yuqi Zhou, Shantanu Bansal, Srinivas Shakkottai, and Le Xie. Energycoupon: A case study on incentive-based demand response in smart grid. In *Proceedings of the Eighth International Conference on Future Energy Systems*, pages 80–90. ACM, 2017.
- [25] Amulya Yadav, Hau Chan, Albert Xin Jiang, Haifeng Xu, Eric Rice, and Milind Tambe. Using social networks to aid homeless shelters: Dynamic influence maximization under uncertainty. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pages 740–748, 2016.
- [26] De A. Rabiee H. & Gomez Rodriguez M. Zarezade, A. Cheshire: Algorithm for activity maximization in social networks. In *55th Annual Allerton Conference on Communications, Control, and Computing, Oct. 3-6, 2017, Monticello, IL, 2017*.
- [27] Ke Zhou, Hongyuan Zha, and Le Song. Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, pages 641–649, 2013.