



Opinion Shaping in Social Networks Using Reinforcement Learning

Vivek S. Borkar , Fellow, IEEE, and Alexandre Reiffers-Masson 

Abstract—In this article, we consider a variant of the classical DeGroot model of opinion propagation with random interactions, in which a prescribed subset of agents is amenable to a control parameter. There are also some stubborn agents and some agents that are neither stubborn nor amenable to control. We map the problem to a shortest path problem, where the control parameter is coupled across controlled nodes because of a common resource constraint. Hence, the problem is not amenable to a pure dynamic programming approach, and the classical reinforcement learning schemes for the latter cannot be applied here for maximizing average influence in the long run. We view it instead as a parametric optimization problem and not a control problem and use a nonclassical policy gradient scheme. We analyze its performance theoretically and through numerical experiments. We also consider a situation when only certain interactions between agents are observed.

Index Terms—Opinion shaping, reinforcement learning (RL), social networks, stochastic shortest path.

I. INTRODUCTION

IN RECENT times, there has been increasing interest in nonprice-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 [35] or for promoting off-peak usage of cars [22]. These aim at leveraging the social network for enhancing prosocial behavior. Indeed, social interactions can impact day-to-day decisions of an agent. For instance, in transportation choice, several works (see [12] and [33]) have demonstrated that the preferences of people in the decision maker's peer group will

Manuscript received 31 January 2021; revised 3 February 2021 and 6 July 2021; accepted 17 August 2021. Date of publication 1 October 2021; date of current version 19 September 2022. The work of Vivek S. Borkar was supported in part by the J. C. Bose Fellowship from the Department of Science Technology (DST), Government of India, and in part by the project "Machine Learning for Network Analytics" from the joint DST-INRIA program administered by the Indo-French Centre for Promotion of Advanced Research. Recommended by Associate Editor G. Como. (Corresponding author: Alexandre Reiffers-Masson.)

Vivek S. Borkar is with the Department of Electrical Engineering, Indian Institute of Technology Bombay, Mumbai 400076, India (e-mail: borkar.vs@gmail.com).

Alexandre Reiffers-Masson was with the Robert Bosch Centre for Cyberphysical Systems, Indian Institute of Science, Bengaluru 560012, India. He is now with IMT Atlantique, 29238 Brest, France (e-mail: alexandre.reiffers-masson@imt-atlantique.fr).

Digital Object Identifier 10.1109/TCNS.2021.3117231

impact her choice of mode of transport to work (public transport, bicycle, and car). Another practical application concerns how to use social comparison to enhance energy efficiency [28]. 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 prosocial so that by imitation, a large number of agents in the whole social network will also become prosocial. Designing such a targeting strategy, however, can be challenging because of computational issues, unknown social network, size or lack of convexity of the resulting optimization problem, and so on. The goal of this article is to propose some algorithms that address these issues.

Our initial model can be described as follows: The society is composed of a finite set of agents. Each agent has an opinion concerning a given prosocial action. 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. The rest of the society that is "close" to the agent in the social network will observe whether she performs the prosocial action or not 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 should influence so as 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. These have a fixed opinion, and they will not be influenced by the social network or the planner. The second set of agents is composed of "uncontrolled" agents. These agents are influenced by the social network, i.e., their opinion will be influenced by the opinion of the others. However, the planner cannot directly influence them. The last set of agents is the set of "controlled" agents. These agents care about the opinion of their neighbors and can also 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 may be 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 Bimpikis *et al.* [3] and Borkar *et al.* [6]. In these papers, the influence matrix is assumed to be known. Moreover, the authors do not consider efficient decentralized algorithms to solve the opinion-shaping problem. In this article, we extend these works further in order to address these issues. Throughout our article, the main assumption is that the planner does not know the influence matrix, but she observes when the agents interact. Using these observations, we are able to derive two reinforcement learning (RL)-based algorithms that will address the aforementioned issues. We also provide supporting simulations for different settings. From a mathematical point of view, we establish the equivalence of the resource-constrained opinion-shaping problem with a stochastic shortest path problem, albeit one that amounts to parametric optimization rather than control. We use this correspondence to motivate our algorithms and their convergence. Additionally, we discuss possible extensions and future directions.

The remainder of this article is organized into eight sections. In Section II, we survey 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 article. We prove the equivalence of the opinion adoption process with a stochastic shortest path problem. Using this equivalence, we propose a decentralized algorithm that accounts for the fact that the influence matrix is unknown. This is followed by a variant that is designed to reduce the complexity of the task when all agents need to be observed. In Section V, we prove the convergence of our algorithms. In Section VI, a new algorithm is proposed, where the opinion-shaping problem is not convex. Numerical experiments are discussed in Section VII. Here, 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 nonconvex opinion-shaping problem. Finally, Section VIII concludes this article.

II. RELATED WORKS

The models for spread of opinions in social networks broadly fall into three categories, and the influence maximization problem has been studied in the context of each of them. The first category is cascade threshold models. In the last decade, initial models in this framework for control of user activity focused on maximization of influence alone [17], including the seminal work of Kempe *et al.* [19]. 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. Because of this, a second category of models was proposed, e.g., by Farajtabar *et al.* [14], who define a new mathematical problem dubbed the activity shaping problem. They use Hawkes processes to model the activity of users in a social network. Undeniably, these point processes have proved to be a very effective method

to capture users' activity [39] in recent literature. These works 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. Other extensions have been suggested since, e.g., [34] and [38].

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 [10] proposes a simple model to capture the diffusion of opinion. He 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 [3], [6], [26], [30]. In [3], Bimpikis *et al.* assume that a planner can directly contaminate a user in the social network by sending her 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 [37], Yildiz *et al.* study the impact of agents with fixed opinions in a social network when the evolution of opinions is captured by the classical voter model. They also investigate the optimal placements of such stubborn agents. A similar model has been studied in [6], again with the presence of stubborn agents. However, in this article, the authors assume that an agent can have opinion in $[0,1]$, and the opinions are propagated according to a DeGroot model. Again, the optimal placement of stubborn agents is investigated. In [32], an associated inference problem is also studied; see also [24], [25], and [31] for other works in this spirit. In [6], Borkar *et al.* consider a more drastic control where they can freeze the opinion of a given user. Finally, in [26], Reiffers-Masson *et al.* suggest a control based on the reduction of the interaction between different agents of the social network.

There are also works that do not exactly fit the above classification. For the opinion-shaping problem without knowledge of the network, Lin *et al.* [21] propose a data-driven model and a learning algorithm for a cascade with a linear threshold model. In [36], Yadav *et al.* suggest a partially observed Markov decision process framework in order to tackle the uncertainty over the topology of the network, again for the case of a cascade with a linear threshold model. To the best of our knowledge, our article is the first that studies shaping opinions under resource constraints using an RL 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 $\mathcal{S}, |\mathcal{S}| = s$ is the set of its agents and \mathcal{E} is 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 $(i, j) \notin \mathcal{E}$.

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 evolve 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_2): The remaining agents that constitute the set S_2 also evolve according to a gossip mechanism, but are amenable to external influence or “control.” With probability $\alpha_i \in (0, 1)$, agent $i \in S_2$ 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) denoting the control applied to node i . Specifying a control as a function of node or “state” in Markov decision processes would correspond to a “stationary Markov policy.” This will become relevant later when we draw a parallel with a stochastic shortest path problem. The maps $w_i : \mathbb{R}_+ \mapsto [0, 1]$ are strictly concave increasing and continuously differentiable. 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 according to $x_i(k+1) = x_\ell(k)$.

We assume that only one agent updates at a time. This could be the case, e.g., if the updates are done in continuous time with the conditional distribution of the next time instant when the update is done given the past being nonatomic. It is, however, not difficult to extend our results and analysis to the case where multiple updates are done concurrently. For ease of exposition, we stick to a single update at a time. We make the following assumption.

(A0) The matrix $P := [[p_{ij}]]$ restricted to $S_2 \cup S_0$, respectively, $S_2 \cup S_1$, is substochastic.

To summarize, the overall dynamics is then described as follows. Suppose agent $i \in \mathcal{S}$ performs an update at time k . Then, letting $\ell_i(k)$ denote the index of the node polled at time k by node i , we have

$$x_i(k+1) = \begin{cases} w_i(u_i) & \text{w.p. } \alpha_i \\ x_{\ell_i(k)}(k) & \text{w.p. } 1 - \alpha_i \end{cases} \quad i \in S_2$$

$$x_i(k+1) = x_{\ell_i(k)}(k), \quad i \in S_1$$

$$x_i(k+1) = h(i), \quad i \in S_0$$

$$x_j(k+1) = x_j(k) \quad \forall j \neq i.$$

Then, the vector $\bar{x}(k) := E[x(k)]$ satisfies the iteration

$$\bar{x}_i(k+1) = \alpha_i w_i(u_i) + (1 - \alpha_i) \sum_{\ell \in \mathcal{S}} p_{i\ell} \bar{x}_\ell(k), \quad i \in S_2 \quad (1)$$

$$\bar{x}_i(k+1) = \sum_{\ell \in \mathcal{S}} p_{i\ell} \bar{x}_\ell(k), \quad i \in S_1 \quad (2)$$

$$\bar{x}_i(k+1) = h(i), \quad i \in S_0. \quad (3)$$

Recall that for $d := |S_2 \cup S_1|$, the $d \times d$ matrix $[[p_{ij}]]$ restricted to $S_2 \cup S_1$ is substochastic, and therefore, the above is a stable affine iteration. Hence, for each $i \in \mathcal{S}$, $x_i^* = \lim_{k \rightarrow +\infty} \bar{x}_i(k)$

exists, and the vector $x^* := [x_1^*, \dots, x_s^*]$ is the solution of the equations:

$$x_i^* = \alpha_i w_i(u_i) + (1 - \alpha_i) \sum_{\ell \in \mathcal{S}} p_{i\ell} x_\ell^*, \quad i \in S_2 \quad (4)$$

$$x_i^* = \sum_{\ell \in \mathcal{S}} p_{i\ell} x_\ell^*, \quad i \in S_1 \quad (5)$$

$$x_i^* = h(i), \quad i \in S_0. \quad (6)$$

For all $u \in \mathbb{R}_+^s$, let $W(u) \in [0, 1]^d$ be a vector-valued function, where the i th element, $W_i(u)$, is equal to $1_{i \in S_2} \alpha_i w_i(u_i) + 1_{i \in S_0} h(i)$. Let A be the substochastic matrix whose ij entry is equal to $a_{ij} = (1 - 1_{i \in S_0})(1 - 1_{i \in S_2} \alpha_i) p_{ij}$. The solution $x^* = [x_1^*, \dots, x_s^*]$ of the fixed-point equation is given by

$$x^* = (Id - A)^{-1} W(u). \quad (7)$$

Here and later, Id is 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., $\sum_{i \in \mathcal{S}} x_i^*$, by controlling u_i , under the resource constraint $\sum_{i \in S_2} u_i \leq M$. Here, $0 < M < s$ is a prescribed bound. Equivalently, the objective of the planner is to find $u^* = (u_i^*)_{i \in S_2}$, the solution of the following optimization problem:

$$u^* = \arg \max_{u_i \in \mathbb{R}_+, \forall i \in S_2} 1^T (Id - A)^{-1} W(u) \quad (8)$$

subject to

$$\sum_{i \in S_2} u_i \leq M. \quad (9)$$

To compute u^* , we can use the projected gradient descent (GD) algorithm. Let

$$\Gamma(\cdot) : (\mathbb{R}_+)^s \mapsto Q := \left\{ u \in (\mathbb{R}_+)^s : \sum_{i \in S_2} u_i \leq M \right\}$$

denote the projection map $x \mapsto \arg \min_{y \in Q} \|x - y\|$, uniquely defined because of the convexity of Q . The algorithm is then

$$u_i(k+1) = \Gamma \left(u_i(k) + \frac{1}{k} 1^T (Id - A)^{-1} \frac{\partial}{\partial u_i} W(u(k)) \right) \quad (10)$$

with

$$\frac{\partial}{\partial u_i} W(u(k)) = \left[1_{j \in S_2} \alpha_j \frac{\partial w_j}{\partial u_i}(u_j) \right]_{j \in \mathcal{S}}.$$

Our objective is to do so in a data-driven manner using ideas from RL. We assume that the matrix P is *unknown*. The *known parameters* are the set of agents \mathcal{S} , the vectors $\alpha := [\alpha_i]_{i \in S_2}$, $h := [h_i]_{i \in S_0}$, the functional vector $w = [w_i]_{i \in S_2}$, and the budget M . At each step $k \in \mathbb{N}_+$, the planner chooses a vector $u(k) = [u_i(k)]_{i \in \mathcal{S}}$. Then, simultaneously, agent i is activated with a given probability η_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 $\lim_{k \rightarrow +\infty} u(k) = u^*$.

In the next section, we introduce a stochastic shortest path problem, which leads to an identical set of equations as (4)–(6)

and use this connection to propose a policy-gradient-based RL scheme, which can be mapped back to the original problem.

IV. RL SCHEME

A. Equivalent Controlled Markov Chain

As mentioned above, we identify (4)–(6) with a stochastic shortest path problem, albeit with a twist (viz., a resource constraint). For this purpose, consider an \mathcal{S} -valued Markov chain $\{Y_n\}$ with transition probabilities

$$\begin{aligned} q_{ij} &:= p_{ij}, & i \in S_1 \cup S_2 \\ &:= \delta_{ij}, & i \in S_0. \end{aligned} \quad (11)$$

Here, δ_{ij} is the Kronecker delta. Thus, the states in S_0 are absorbing states. We associate with a state-control pair $(i, u_i) \in S_2$ an instantaneous cost $-\alpha_i w_i(u)$ and a state-dependent discount factor $(1 - \alpha_i)$. The transition probabilities are independent of the control choice, which affects only the running cost. Let $\tau := \min\{k \geq 0 : Y_k \in S_0\}$ denote the first passage time to S_0 . Set $\alpha_i = 0$ and $w_i(u) = 0$ for all $i \in S_0 \cup S_1$. We consider the problem of maximizing the total discounted reward till the first hitting time of S_0 , given by $\sum_i x_i^*$, where

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

Here, $E_i[\cdot]$ denotes the expectation when $Y_0 = i$ and $i \mapsto u_i$ is a fixed stationary policy that specifies the control as a function of the current state alone. This makes it a stochastic shortest path problem as mentioned above, albeit nonclassical, because we also impose the constraint (9).

The reuse of notation x_i^* here is not accidental. We aim to identify our opinion-shaping problem with this nonclassical stochastic shortest path problem, so we are already using the notation that will facilitate establishing this correspondence. ‘‘One step analysis’’ applied to this problem leads to the standard linear system

$$x_i^* = \alpha_i w_i(u_i(k)) + (1 - \alpha_i) \sum_{\ell \in \mathcal{S}} p_{i\ell} x_\ell^*, \quad i \in S_2 \quad (12)$$

$$x_i^* = \sum_{\ell \in \mathcal{S}} p_{i\ell} x_\ell^*, \quad i \in S_1 \quad (13)$$

$$x_i^* = h(i), \quad i \in S_0. \quad (14)$$

The objective is then to maximize $\sum_i x_i^*$ subject to the above and (9), which is exactly the same problem as before.

The constraint (9) 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 (e.g., without regard to how often each state is visited). This puts it beyond the reach of traditional dynamic-programming-based computations such as value or policy iteration. Therefore, we

treat this as a parametric optimization problem over the parameters u_i s instead of as a control problem. This in particular means that we cannot hope to use standard RL schemes such as Q -learning, actor-critic, TD(λ), etc. But we can and do use a policy gradient scheme, which can also treat parametric optimization problems such as ours.

B. First Algorithm

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

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

for $k \geq 0$. Then, $\nu(i, k)$, $k \geq 0$, can be interpreted as a ‘‘local clock’’ at agent i , counting its own number of updates till ‘‘time’’ (i.e., the overall iterate count) k . We assume that

$$\lim_{k \uparrow \infty} \frac{\nu(i, k)}{k} \geq \delta \quad \forall i, \text{ a.s.} \quad (15)$$

for some $\delta > 0$. This means that all i are sampled *comparably often* with probability 1. Pick step-size sequences $\{a(k)\}, \{b(k)\} \subset (0, \infty)$ such that

$$\begin{aligned} \sum_k a(k) &= \sum_k b(k) = \infty, \quad \sum_k (a(k)^2 + b(k)^2) < \infty \\ \frac{b(k)}{a(k)} &\rightarrow 0. \end{aligned} \quad (16)$$

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

- 1) $a(k+1) \leq a(k)$ from some k onwards.
- 2) $\exists r \in (0, 1)$ such that $\sum_k a(k)^{1+q} < \infty$, $q \geq r$.
- 3) For $x \in (0, 1)$, $\sup_k \left(\frac{a(\lfloor xk \rfloor)}{a(k)} \right) < \infty$, where $\lfloor \cdot \rfloor$ stands for the integer part of ‘‘ \cdot ’’.
- 4) For any $x \in (0, 1)$ and $A(k) := \sum_{m=0}^k a(i)$, we have $\lim_{n \uparrow \infty} \left(\frac{A(\lfloor yk \rfloor)}{A(k)} \right) = 1$ uniformly in $y \in [x, 1]$.
- 5) For $N(k, x) := \min\{m \geq k' : \sum_{k'=m}^k a(k') \geq x\}$, the limit $\lim_{k \uparrow \infty} \frac{\sum_{m=\nu(i, k)}^{\nu(i, N(k, x))} a(m)}{\sum_{m=\nu(j, k)}^{\nu(j, N(k, x))} a(m)}$ exists a.s. $\forall i \neq j, x > 0$.

These conditions are satisfied, e.g., by the popular step size $a(k) = \frac{1}{k+1}$, $k \geq 0$. They allow us to apply to our algorithm the results of [5] for asynchronous stochastic approximation.

The algorithm is then as follows. Set $\alpha_i \equiv 0$ for $i \in S_1$. For $k \geq 0, i \in \mathcal{S}, j \in S_2$, we have

$$\begin{aligned} \Psi_{ij}(k+1) &= \Psi_{ij}(k) + a(\nu(i, k)) I\{Y_k = i\} \\ &\quad \times [\alpha_i w'_i(u_i(k)) \delta_{ij} + (1 - \alpha_i) \Psi_{Y_{k+1}j}(k) \\ &\quad - \Psi_{ij}(k)], \quad i \notin S_0 \end{aligned} \quad (17)$$

$$u_i(k+1) = \Gamma \left(u_i(k) + b(k) \sum_j \Psi_{ji}(k) \right), \quad i \in S_2 \quad (18)$$

$$\Psi_{ij}(k) = 0, \quad i \in S_0. \quad (19)$$

Lemma 1: Iterations (17)–(19) constitute a projected stochastic gradient ascent to solve the above stochastic shortest path problem with constraints.

Proof: The argument is in two steps.

- 1) The iteration (17) estimates the partial derivatives $\frac{\partial V(i)}{\partial u_j}$ by $\Psi_{ij}(k)$, $k \geq 0$. Consider the constant policy dynamic programming equation

$$\tilde{V}(i) = \alpha_i w_i(u_i) + (1 - \alpha_i) \sum_j p_{ij} \tilde{V}(j), \quad i \in S_2 \cup S_1 \quad (20)$$

$$\tilde{V}(i) = h(i), \quad i \in S_0. \quad (21)$$

This is a well-posed linear system of equations in \tilde{V} for given u_i s and, by Cramer's rule, is a rational function of the continuously differentiable functions $u_i \mapsto w_i(u_i)$, $i \in S_2$, with a nonvanishing denominator. Hence, it is continuously differentiable in the u_i s. These equations are seen to be identical to (1)–(3). Thus, we can identify \tilde{V} with x^* . Differentiating both sides of (20) w.r.t. u_j , we see that $\Phi_{ij} := \frac{\partial \tilde{V}(i)}{\partial u_j}$ satisfy

$$\Phi_{ij} = \alpha_i w'_i(u_i) \delta_{ij} + (1 - \alpha_i) \sum_\ell p_{i\ell} \Phi_{\ell j} \quad (22)$$

for $i \in S_2$, with $\Phi_{ij} = 0$ for $i \in S_0$. The iteration (17) is then the standard stochastic approximation scheme to solve this equation. That is, it replaces the conditional expectation w.r.t. the $p_{i\ell}$ s by an actual evaluation at a random variable with conditional law p_i . and then uses the incremental nature of stochastic approximation to do the averaging over successive iterations.

- 2) Having identified \tilde{V} with x^* , iteration (18), operating on a slower time scale [in view of (16)], constitutes a stochastic gradient ascent to maximize the reward $\sum_i x_i^*$. That is, (18) is a stochastic gradient ascent over the control variables, which takes the outputs $\{\Psi_{ij}(k)\}$ of (17) as estimates of the relevant partial derivatives and, summing them up over the first index, generates an estimate of the corresponding partial derivative of the reward itself. The application of the projection $\Gamma(\cdot)$ makes it a projected stochastic gradient scheme that imposes the constraint (9). ■

Let Z_k be the index of the agent that updated its valuation at time k , i.e., it is the Z_k th component $x_{Z_k}(k)$ that got updated at time k ; the rest were left unperturbed. Also, suppose that this was done by the Z_k th agent by polling a neighbor \tilde{Z}_k according to the transition probabilities q_{z_k} defined in (11). The algorithm for our original problem is as follows.

Algorithm 1:

$$\begin{aligned} \Psi_{ij}(k+1) = & \Psi_{ij}(k) + a(\nu(i, k)) I\{Z_k = i\} \\ & \times \left[(\alpha_i w'_i(u_i(k)) \delta_{ij} + (1 - \alpha_i) \Psi_{\tilde{Z}_k j}(k) \right. \\ & \left. - \Psi_{ij}(k) \right], \quad i \in S_2 \cup S_1 \end{aligned} \quad (23)$$

$$u_i(k+1) = \Gamma \left(u_i(k) + b(k) \sum_j \Psi_{ji}(k) \right), \quad i \in S_2 \quad (24)$$

$$\Psi_{ij}(k) = 0, \quad i \in S_0. \quad (25)$$

Theorem 1: The algorithm (23)–(25) is a valid algorithm for the proposed problem for $\{Z_k\}$ independent identically distributed (i.i.d.) η .

Proof: This is precisely the scheme (17)–(19) applied to our original problem after identifying it with the shortest path problem described above, as follows. We identify Y_k with Z_k and Y_{k+1} with \tilde{Z}_k . Note that the algorithm needs only the pairs (Y_k, Y_{k+1}) with the correct conditional probability of the latter given the former. As we observe later in course of the convergence proof, the distribution of Y_k in this pair can be ignored as long as the basic requirement (15) is satisfied. This is indeed satisfied by $\{Y_k\}$ because of its irreducibility. On the other hand, it is also satisfied by $\{Z_k\}$, which are i.i.d. η . So, we can replace (Y_k, Y_{k+1}) by (Z_k, \tilde{Z}_k) without affecting its convergence analysis. Thus, iterations (17)–(19) above reduce to Algorithm 1 for our problem. ■

C. Alternative Learning Scheme

The problem with the above scheme is that it involves all agents in \mathcal{S} . Worse, it requires all communications between agents to be observed. A more realistic assumption is that only a few agents can be monitored. These should include in particular those in $S_0 \cup S_2$. Without loss of generality, we assume that only the updates of agents in S_2 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 $S^* := S_2 \cup S_0$ as our effective state space for the algorithm. But the situation is much more difficult here. Recall Z_k, \tilde{Z}_k defined in the preceding section. There, we had considerable freedom in choosing how Z_k is generated; the key requirement was that \tilde{Z}_k should have the prescribed conditional law given Z_k . 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; the (stationary marginal) law of the former only needs to have full support at each step. Now, we require a path from one state in S^* to another, passing through a possibly nonempty set of unobserved states in $\mathcal{S} \setminus S^*$. Generating pairs (Z_k, \tilde{Z}_k) as before does not provide that. We now need a probing mechanism. We use one suggested by respondent-driven sampling [18]. That is, we define Z_k as before, but when node $Z_k = i \in S^*$ polls a neighbor $i_1 \in \mathcal{S}$, 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 $\tilde{Z}_k = j$. By analogy to the above stochastic shortest path formulation, we consider an \mathcal{S} -valued Markov chain $\{Y_k\}$ with transition probabilities $\{q_{ij}\}$, but observed only at the successive return times $T_k, k \geq 0$, of $\{Y_k\}$ to S^* . These are

defined recursively by

$$T_0 := \min\{m \geq 0 : Y_m \in S^*\}$$

$$T_{k+1} := \min\{m > T_k : Y_m \in S^*\}, \quad k \geq 0.$$

The sampled chain $\{Y_k^* := Y_{T_k}\}$ eventually gets absorbed into S_0 as before. Strictly speaking, if we keep track of the T_k 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 [15], 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 imaginary object, the actual process is the simple averaging or ‘‘gossip’’ dynamics. Thus, the only thing that matters is that the conditional law of Y_{k+1}^* given Y_k^* is the same as the conditional law of \tilde{Z}_k given Z_k above. The actual values of T_k s are irrelevant for us. Let $\varphi(j|\ell) := P(Y_\zeta = j | Y_0 = \ell)$, for $\zeta := \min\{k > 0 : Y_k \in S^*\}$. In particular, $\varphi(\cdot|\cdot)$ is independent of the control choice u . This is because once the chain $\{Y_k\}$ leaves state i , it does not hit any other controlled state before hitting another state (j above) in S^* . For the same reason, the running cost associated with this transition is $-\alpha_i w_i(u)$ as before. We now consider the restricted reward $\sum_{i \in S} x_i^*$, 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

$$V(i) = \alpha_i w_i(u_i) + (1 - \alpha_i) \left(p_{ij} + \sum_{j \in S^*, \ell \notin S^*} p_{i\ell} \varphi(j|\ell) \right) V(j)$$
(26)

$$V(i) = h(i), \quad i \in S_0.$$
(27)

By exactly the same reasoning as before, this leads to our second algorithm.

Algorithm 2: For $i \in S, j \in S_2$,

$$\Psi_{ij}(k+1) = \Psi_{ij}(k) + a(\nu(i, k)) I\{i \in Z_k\}$$

$$\times \left[(\alpha_i w'_i(u_i(k)) \delta_{ij} + (1 - \alpha_i) \Psi_{\tilde{Z}_k^i j}(k) - \Psi_i(k) \right], \quad i \notin S_0$$
(28)

$$u_i(k+1) = \Gamma \left(u_i(k) + b(k) \sum_j \Psi_{ji}(k) \right), \quad i \in S_2$$
(29)

$$\Psi_{ij}(k) = 0, \quad i \in S_0.$$
(30)

V. CONVERGENCE ANALYSIS

The convergence analyses of both schemes are similar and use known facts from the theory of two time scales and distributed asynchronous stochastic approximation. With this in mind, we sketch it in outline only for the first scheme and write a more detailed proof in Appendix A. To begin with, note that condition (16) implies that the iterates (24) move on a slower, in fact asymptotically negligible, time scale compared to (23). Hence, they can be viewed as quasi-static, i.e., $u_i(k) \approx u_i \forall i$, for purposes of analyzing (23) (see [4, Sec. 6.1]). Then, (23) constitutes a stochastic approximation scheme to estimate the partial derivatives of V^* w.r.t. the u_i s by solving the linear system (22), 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 imposed on $\{a(n)\}$ play a crucial role.

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

$$\Psi_{ij}(k) - \left. \frac{\partial x_i^*}{\partial u_j} \right|_{u_i = u_i(k)} \rightarrow 0$$

a.s. $\forall i, j$, i.e., Ψ_{ij} s track the corresponding partial derivatives of x_i^* with an asymptotically negligible error, as desired. Then, (24) is a legitimate stochastic gradient ascent scheme. We need the following lemma.

Lemma 2: The solution $V(\cdot)$ of the constant policy dynamic programming equation (26), (27) 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., [1]. \blacksquare

Our main result is then the following.

Theorem 2: Algorithms 1 and 2 are a.s. asymptotically optimal for their respective optimality criteria.

A detailed proof is given for Algorithm 1 in Appendix A. That for Algorithm 2 is similar. Since both are two time-scale stochastic approximations, standard results concerning convergence rates of such algorithms apply; see, e.g., [7], [11], and [23] for results in this vein.

VI. MORE GENERAL MODEL

We can also consider the situation where the α_i s depend on the control choice u_i at $i \in S_2$. We shall illustrate the changes for the second scheme above, the situation for the first scheme being completely analogous. Thus, the ‘‘dynamic programming equations’’ become

$$V(i) = \alpha_i(u_i) w_i(u_i)$$

$$+ (1 - \alpha_i(u_i)) \left(p_{ij} + \sum_{\ell \in S^*} p_{i\ell} \varphi(j|\ell) \right) V(j), \quad i \notin S_0$$

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

and the corresponding RL scheme is as follows.

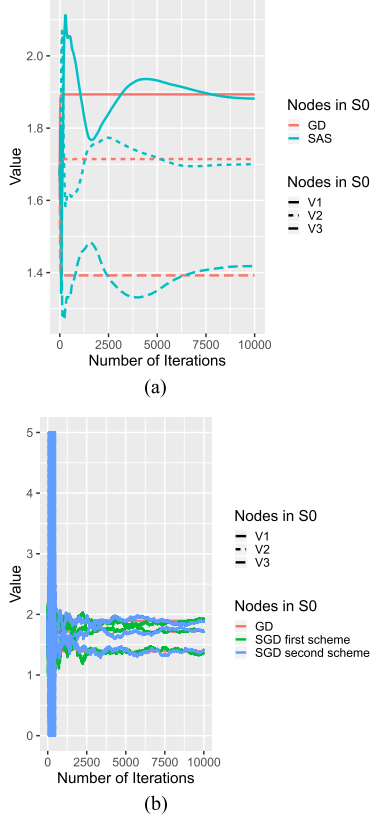


Fig. 1. Evolution of $u(k)$ for each algorithm. (a) Comparison of the evolution of the Algorithm 1 with the classical gradient scheme. The simulation is performed over the Karate network. (b) Comparison of the evolution of the stochastic gradient schemes (see the Appendixes) with the classical stochastic gradient scheme. The simulation is performed over the Karate network.

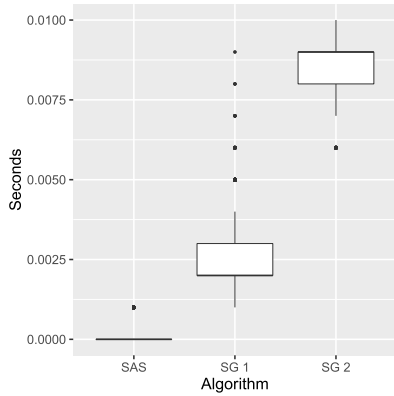


Fig. 2. Boxplot for the time (in second) to perform one iteration of each algorithm. [SAS = Algorithm 1, SG1 = (42), SG2 = (43).] The simulations are performed on the Karate network.

Algorithm 3: For $i \in S^*$, $j \in S_2$,

$$\begin{aligned}
 V_i(k+1) &= V_i(k) + a(\nu(i, k))I\{i \in Z_k\} \\
 &\times \left[\alpha_i(u_i(k))w_i(u_i(k)) + \right. \\
 &\left. (1 - \alpha_i(u_i(k)))V_{\tilde{Z}_k^i}(k) - V_i(k) \right], \\
 &i \notin S_0
 \end{aligned} \tag{31}$$

TABLE I
DESCRIPTION OF THE NETWORKS

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

$$\begin{aligned}
 \Psi_{ij}(k+1) &= \Psi_{ij}(k) + a(\nu(i, k))I\{i \in Z_k\} \\
 &\times \left[(\alpha_i(u_i(k))w_i'(u_i(k)) \right. \\
 &+ \alpha_i'(u_i(k))w_i(u_i(k)))\delta_{ij} \\
 &- \alpha_i'(u_i(k))\delta_{ij}V_{\tilde{Z}_k^i}(k) \\
 &\left. + (1 - \alpha(u_i(k)))\Psi_{\tilde{Z}_k^i j}(k) - \Psi_{ij}(k) \right], \\
 &i \notin S_0
 \end{aligned} \tag{32}$$

$$\begin{aligned}
 u_i(k+1) &= \Gamma \left(u_i(k) + b(k) \sum_j \Psi_{ji}(k) \right), \\
 &i \in S_2 \\
 V_k(i) &= h(i), \Psi_{ij}(k) = 0, \quad i \in S_0.
 \end{aligned} \tag{34}$$

The difference with the previous scheme is that (20) gets replaced by

$$\tilde{V}(i) = \alpha_i(u_i)w_i(u_i) + (1 - \alpha_i(u_i)) \sum_j p_{i\ell} \tilde{V}(\ell). \tag{35}$$

Differentiating through with respect to u_i in (22) 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 terms inside the square brackets on the right-hand side of (33) correspond to these additional terms. This involves $\tilde{V}(\cdot)$ as well, unlike the previous scheme, which did not. Therefore, one needs the additional iteration (31) to estimate it, this being the stochastic approximation scheme to solve (20).

The convergence analysis applies as before except for the fact that we can no longer claim concavity. Hence, only convergence to a local maximum 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

$$\begin{aligned}
 u_i(k+1) &= \Gamma \left(u_i(k) + \frac{B}{k} \sum_j \Psi_{ji}(k) \right. \\
 &\left. + \frac{C}{\sqrt{k \log \log k}} W_{k+1} \right)
 \end{aligned} \tag{36}$$

where $\{W_k\}$ are i.i.d. $N(0, 1)$ and $B, C > 0$ are suitably chosen constants as in [16].

VII. NUMERICAL EXPERIMENTS

We select three real-world networks for our evaluation. The three networks are Karate, Macaque, and Rfid (see Table I) and have 34–75 nodes and 78–2278 links. Each network was

Scheme	Convergence Criterion			
	0.1	0.01	0.001	0.0001
SAS	0.202s	1.043s	5.237s	16.71s
SG 1	30.542s	30.67s	NA	NA
SG 2	0.174s	1.918s	30.868s	NA

Fig. 3. Table in this figure summarizes the system time (in second) taken by each algorithm before meeting the following convergence criterion $\frac{1}{T100} \sum_{k'=k-99}^k \sum_{i=1}^I |u_i(k') - u_i(k' - 1)| \leq CC$, with $CC \in \{0.1, 0.01, 0.001, 0.0001\}$. When NA is appearing in one cell of the table, it means that after 30 000 iterations, the associated algorithm was not able to meet the convergence criterion.

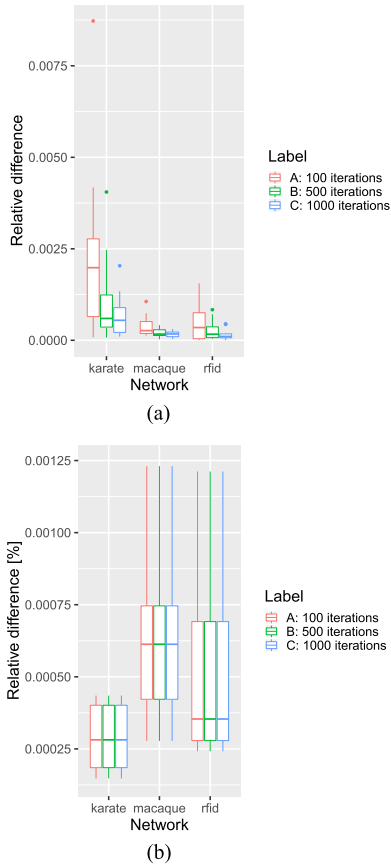


Fig. 4. Boxplot of the relative difference between the payoff obtained at k and the optimum over ten simulations for Algorithm 1 and the SGD (42). The simulations are performed over three networks. (a) Algorithm 1. (b) SGD first scheme (42).

retrieved from the R package `igraphdata` [27]. The numerical experiments reported here are for the synchronous case, i.e., all components are updated at each iteration. The results are compared with, first, the exact solution computed offline using a classical gradient ascent (10) and, second, the stochastic gradient ascent schemes described in Appendix B.

Inputs: The matrix P , the number of agents in each set (S_2, 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 C of our step-size functions

$$a(k) = \frac{A}{\lceil (1 + k \log(1 + k)) / C \rceil}$$

and

$$b(k) = \frac{B}{\lceil k / C \rceil}$$

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

Initial setting: First, we specify the number of agents in each set (S_2, S_1 , and S_0) and then randomly allocate an agent to a given set. We assume that $\alpha_i = \alpha$ for each $i \in S_0 \cup S_2$ 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, $\alpha = 0.6$, $M = 5$, $A = 0.6$, $B = 0.6$, $C = 100$, and $w(x) = \frac{x}{x+0.1}$.

Study of Algorithm 1 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 other networks. In Fig. 1(a), the x -axis denotes the number of iterations and the y -axis captures the evolution of $u(k)$ for the stochastic approximation (see Algorithm 1) (17)–(19) (SAS for short). In Fig. 1(b), the x -axis denotes the number of iterations and the y -axis captures the evolution of $u(k)$ for the stochastic gradient descent (SGD) with the two sampling schemes (42) and (43) described in the Appendixes. The red curve captures the evolution of $u(k)$ using the gradient ascent (GD) [see (10)]. The number of controlled agents is equal to 3. Twenty-eight agents belong to S_1 and three agents are in S_0 . In Fig. 1(a), before 7500 iterations, we can observe that the GD algorithm already converges and the RL scheme did not. In fact, the SGD seems to converge faster (see Fig. 1(b) after 2500 iterations). However, we observe that the variance over the iterates of SGD is higher than the SAS. The tradeoff is, therefore, between speed and fluctuations. Moreover, we can observe in Fig. 2 that one iteration of the SAS is much faster than 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. This finding is confirmed by Fig. 3, where we observe that the stochastic approximation scheme is converging much faster in terms of system time compared to the two stochastic gradient schemes. *Study of Algorithm 1 for 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 conclusions are

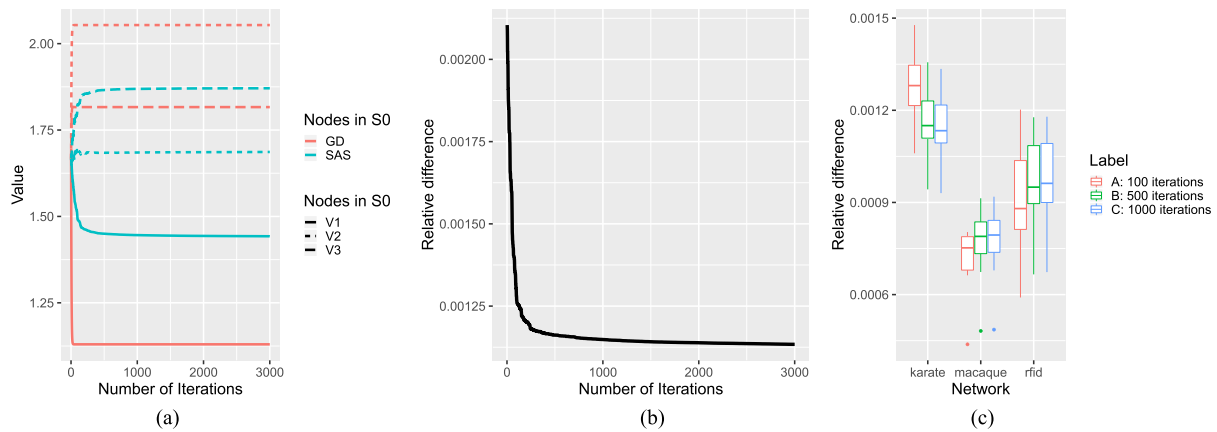


Fig. 5. Simulations for Algorithm 2. (a) Evolution of $u(k)$. In this figure, we compare the evolution of the classical GD scheme with Algorithm 2. (b) Evolution of the relative difference between the current payoff generated by Algorithm 2 and the optimal payoff. (c) Boxplots of the relative difference between the current payoff generated by Algorithm 2 and the optimal payoff. The relative difference is studied after Algorithm 2 has performed 100, 500, and 1000 iterations. The simulations are performed over three different networks.

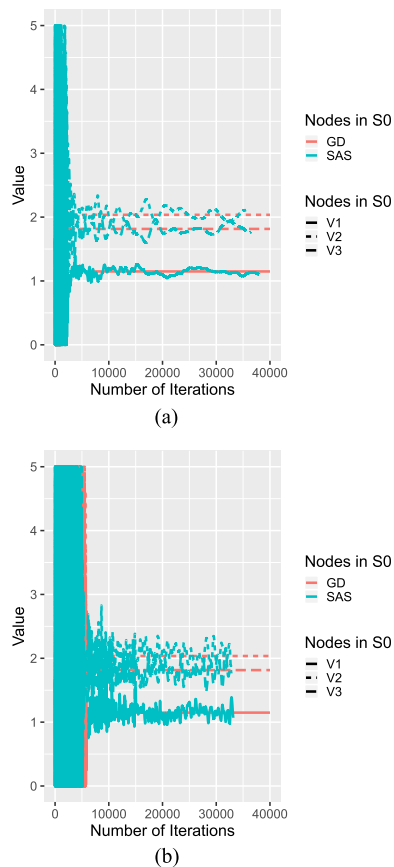


Fig. 6. Convergence of Algorithm 3 with/without the gradient knowledge. The simulation are performed over the Karate network. (a) $C = 1000$. (b) $C = 5000$.

similar. In Fig. 4(a) and (b), we perform ten simulations of the stochastic approximation scheme (see Algorithm 1) and stochastic gradient for each network. The performance measure on the 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 the relative difference is much lower than 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. *Study of Algorithm 2:* In the third numerical study, we are interested in understanding how the second learning scheme (see Algorithm 2) 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 second one, you can only observe a part thereof. In order to be able to compare with the previous simulations, we assume the following: the set of controlled agents S is the same. Only 50% of the agents in S_0 and S_1 are observed. The results are depicted in Fig. 5(a)–(c). In Fig. 5(a), we note that Algorithm 2 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 Fig. 5(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 Fig. 5(c) for other networks. These preliminary simulations encourage the use of algorithm 2.

Study of Algorithm 3: The final numerical study is dedicated to the last scheme based on the annealing method for nonconvex optimization. We restrict this study to the Karate network. The noisy term of (36) is parameterized by $c(k) := \lceil k/C \rceil$ and $C = 10$. We study two schemes. The first one is described by (31), (33), and (36) (see Algorithm 3). The second one is Algorithm 3 with the direct computation of the gradient [see (36)]. We are interested in understanding how the first scheme tracks the behavior of the second scheme. In Fig. 6(a) [respectively, Fig. 6(b)], we observe that Algorithm 3 begins to track the trajectory of (36) after 2000 iterations (15 000 iterations).

Moreover, we observe that in both cases, the two schemes are converging to the same values.

Take-away from the numerical section: The different numerical experiments lead to the following conclusions.

- 1) We observed that Algorithm 1 converges much faster (in terms of system time) in comparison with the two stochastic gradient schemes. This is a consequence of the fact that computing a biased estimate of the gradient (as in Algorithm 1) is faster than computing an unbiased estimate of it. Therefore, Algorithm 1 outperforms classical stochastic gradients.
- 2) Algorithm 1 still requires a lot of iterations to converge to the optimal solution. Algorithm 2, which uses partial information of the network, will converge faster than Algorithm 1. Also, even when it does not converge to the optimal strategy, we have demonstrated by examples that it is still performing well. Therefore, for large networks, we recommend using Algorithm 2.
- 3) Finally, when the problem is not convex, we have shown that Algorithm 1 can be adapted suitably as Algorithm 3 and will still be able to converge to the optimal solution.

VIII. CONCLUSION AND FUTURE DIRECTIONS

In this article, we have proposed a dynamic model for opinion shaping in a social network by a social planner, where only some agents are amenable to influence by the central planner, and in addition, there is a resource constraint that couples the decisions across the nodes. We separately introduced a stochastic shortest path problem, which leads to an identical optimization problem mathematically. Because of the coupling resource constraint, this problem cannot be solved by dynamic programming and, therefore, is not amenable to classical RL schemes for approximate dynamic programming. It is, however, amenable to a policy gradient scheme with the difference that its gradient ascent component gets modified to a projected gradient ascent in order to accommodate this constraint. There is also a problem specific gradient estimation component. In addition to the vanilla variant, we also introduce another variant based on a reward that only approximates the original one, but requires fewer interactions to be observed. Empirically, this is seen to be a faster scheme with very small loss of optimality. Both schemes perform better than some natural Markov-chain-Monte-Carlo-based gradient schemes for larger problems. They cannot, however, be compared with the standard RL schemes such as Q -learning for reasons already stated. We also introduce a harder problem, where, in addition to influencing the reward, the planner can directly influence the probabilities with which the controlled agents poll their neighbors. This leads to loss of convexity, and we can claim only local optimality.

Some potential future directions are as follows.

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., [29] for some initial efforts toward the dynamic programming aspects.

Selection of the initial set of agents: One of the results of this article is the fact that by observing a smaller number of agents, we can increase drastically the speed of convergence of the algorithm. Even if the solution obtained is suboptimal, the relative difference observed between the optimal payoff and the suboptimal one in simulations was low (about 0.01%). Therefore, one interesting question is to find an algorithm to choose the initial set of agents. This question can be related to the problem of selecting k sensors among n potential sensors. In future work, we plan to adapt this well-known problem to our setting. A closely related formulation and its resolution by a message passing algorithm appears in [30]. See also [6] for a greedy scheme for agent selection with performance guarantees.

APPENDIX A PROOF OF THEOREM 2

Theorem 2: The proposed learning policy (23)–(25) is asymptotically optimal, a.s.

Proof: Let $\mathcal{F}_k := \sigma(\Phi(k'), u(k'), M(k'), k' \leq k)$. We can rewrite (17)–(19) as

$$\begin{aligned} \Psi_{ij}(k+1) = & \Psi_{ij}(k) + a(k) \left[\alpha_i w'_i(u_i(k)) \delta_{ij} \right. \\ & + (1 - \alpha_i) \sum_l p_{il} \Psi_{lj}(k) - \Psi_{ij}(k) \\ & \left. + M_{ij}(k+1) \right], \quad i \notin S_0 \end{aligned} \quad (37)$$

$$u(k+1) = \Gamma(u(k) + b(k)1^T \Psi(k)), \quad i \in S_2 \quad (38)$$

$$\Psi_{ij}(k) = 0, \quad i \in S_0 \quad (39)$$

where $M_{ij}(k+1) = (1 - \alpha_i)[\Psi_{\bar{z}_k}(k) - \sum_l p_{il} \Psi_{lj}(k)]$ for all i . We denote by $M(k) := [M_{ij}(k)]_{1 \leq i, j \leq I}$ the associated matrix. Define the Gateaux derivative

$$\gamma(x; y) := \lim_{\delta \rightarrow 0} \frac{\Gamma(x + \delta y) - x}{\delta}.$$

Using a first-order Taylor expansion, we can rewrite (38) as

$$u(k+1) = u(k) + b(k)[\gamma(u(k); 1^T \Psi(k)) + \epsilon_1(k+1)] \quad (40)$$

where $\epsilon_1(k+1)$ is the error term from the Taylor expansion, which is $o(b(k))$.

Step I (Convergence of the fast-time scale):

We can rewrite (37) and (40) as: for $j \in S_2$,

$$\begin{aligned} \Psi_{ij}(k+1) = & \Psi_{ij}(k) + a(k) \left[\alpha_i w'_i(u_i(k)) \delta_{ij} \right. \\ & + (1 - \alpha_i) \sum_l p_{il} \Psi_{lj}(k) - \Psi_{ij}(k) \\ & \left. + M_{ij}(k+1) \right], \quad i \notin S_0 \end{aligned}$$

$$u(k+1) = u(k) + a(k)[\epsilon_2(k) + \epsilon_3(k+1)], \quad i \in S_2$$

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

with

$$\epsilon_2(k) = a(k)^{-1}b(k)\gamma(u(k); 1^T\Psi(k))$$

$$\epsilon_3(k+1) = a(k)^{-1}b(k)\epsilon_1(k+1).$$

Both $\{\epsilon_2(k)\}$ and $\{\epsilon_3(k+1)\}$ are bounded random sequences that are $o(1)$ because $a(k)^{-1}b(k) \xrightarrow{k \rightarrow +\infty} 0$. The martingale noise $M_{ij}(k+1)$ satisfies

$$\mathbb{E}[(M_{ij}(k+1))^2 | \mathcal{F}_k] = K_1(1 + \|\Psi(k)\|^2) \quad \forall i, j.$$

Therefore, in the fast-time scale regime corresponding to step sizes $\{b(k)\}$, it follows from [4, Th. 7, p. 74] that the limiting ordinary differential equation (o.d.e.) for the first iteration above is

$$\dot{\Psi}_{ij}(t) = \eta(i) \left[\alpha_i w'_i(u_i) \delta_{ij} + (1 - \alpha_i) \sum_l p_{il} \Psi_{il}(t) - \Psi_i(t) \right],$$

$$\forall i \notin S_0, j \in S_2$$

$$\Psi_{ij}(t) = 0, \quad i \in S_0 \quad \forall t \geq 0.$$

Since A is a substochastic matrix, this o.d.e. has a globally asymptotically stable equilibrium $\Phi_{ij}(u)$, which is the unique solution of the fixed-point equation: for $i \notin S_0, j \in S_2$,

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

with $\Phi_{ij} = 0$ for $i \in S_0$. Define the matrix $\Phi(u) := [\Phi_{ij}(u)]_{1 \leq i, j \leq I}$. By standard arguments of two time-scale stochastic approximation (see [4, Sec. 6.1]), it follows that

$$\Psi_{ij}(k) - \Phi_{ij}(u(k)) \rightarrow 0 \text{ a.s. } \forall i, j.$$

Step II (Convergence of the slow-time scale):

In the slow-time scale regime, we analyze

$$u(k+1) = u(k) + b(k) [\gamma(u(k); 1^T\Phi(u(k))) + \epsilon_1(k+1) + \epsilon_4(k+1)]$$

where $\epsilon_4(k+1) = \gamma(u(k); 1^T\Psi(k)) - \gamma(u(k); 1^T\Phi(u))$. The sequence $\{\epsilon_4(k)\}$ is bounded and, by the results of Step I above, is $o(1)$. Recall that $\Gamma(\cdot)$ is the projection to the simplex Q . Define the normal cone

$$\mathcal{N}_Q(x) := \{z \in \mathbb{R}^s : \langle z, x - y \rangle \geq 0 \forall y \in Q\}.$$

Then, the above iterates track the so-called ‘‘projected dynamical system’’ [20], which is equivalent to the differential inclusion (see [13, Lemma 4.6])

$$\dot{u}(t) \in 1^T\Phi(u(t)) - \mathcal{N}_Q(u(t)). \quad (41)$$

Since Q is convex, (41) has a unique solution (see [8, Corollary 2] and [9, Ths. 3.1 and 3.2]). Thus, (41) is simply a projected gradient ascent for the strictly concave function $u \mapsto 1^T(Id - A)^{-1}W(u)$ and, therefore, must converge to its global

maximizer u^* . By the theory of stochastic approximation with differential inclusion limits (see [2, Prop. 3.27]), it follows that the iterates (4) converge to u^* a.s.

APPENDIX B STOCHASTIC GRADIENT SCHEMES

For comparison purposes, we propose two algorithms, where, instead of having a biased but consistent estimator of the gradient, we have a sampling scheme that will provide, at each iteration, an unbiased estimator of the gradient. Let $\{Z_n\}$ be as before. Let $\delta_{..}$ denote the Kronecker delta.

- 1) For each $i \in S$, set $m = 0$ and set $Y_{j0} = Z_k = j$ (say). Here, $j \in S \setminus S_0$ can be picked uniformly at random. Initialize $\xi_{ji}(k) = 0$.
- 2) With probability $\alpha_{Y_{j0}} = \alpha_j$ ($\alpha_j = 0$ if $j \in S_1$), stop and set $\xi_{ji}(k) \rightarrow \xi_{ji}(k) + \delta_{ji}$.
- 3) If not, with probability $(1 - \alpha_{Y_{j0}})p_{Y_{j0}j'}$, continue by setting $Y_{j1} = j'$, $\xi_{ji}(k) \rightarrow \xi_{ji}(k)$.
- 4) At step m , stop if $Y_{jm} \in S_0$. If not, stop with probability $\alpha_{Y_{jm}}$ and set $\xi_{ji}(k) \rightarrow \xi_{ji}(k) + \delta_{Y_{jm}i}$, or else continue with probability $(1 - \alpha_{Y_{jm}})$ by setting $Y_{j(m+1)} = \ell$ with probability $(1 - \alpha_{Y_{jm}})p_{Y_{jm}\ell}$.
- 5) Repeat step 4 above for $m \geq 1$ till stopping.
- 6) Perform the following GD step:

$$u_i(k+1) = \Gamma \left(u_i(k) + a(k)w'_i(u_i(k)) \sum_i \xi_{ij}(k) \right). \quad (42)$$

An alternative scheme is the following.

- 1) For each $i \in S$, and for each $j \in S$, set $m = 0$ and $Y_{i0} = i$, kept fixed for this run. Initialize $\zeta_i = 1$. Set

$$\xi_{ij} \rightarrow \xi_{ij} + \zeta_i \delta_{Y_{i0}j} \alpha_{Y_{i0}}.$$

Continue by setting $Y_{i1} = k$ with probability $p_{Y_{i0}k}$.

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

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

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

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

$$u_i(k+1) = \Gamma \left(u_i(k) + a(k)w'_i(u_i(k)) \sum_j \xi_{ji}(k) \right). \quad (43)$$

By construction, for both sampling schemes, $w'_i(u_i(k))E[\sum_\ell \xi_{i\ell}(k)]$ is the solution of the linear system (22). Therefore, the schemes will converge to the optimal u^* as long as the variance of $\xi_{ij}(k)$ is bounded for all k [4]. For the stochastic gradient iterate (43), a good step size is $a(k) = A/(\lceil \frac{k}{M} \rceil)$ for some $A > 0$ and $M \geq 1$.

REFERENCES

- [1] M. Agarwal, V. S. Borkar, and A. Karandikar, "Structural properties of optimal transmission policies over a randomly varying channel," *IEEE Trans. Autom. Control*, vol. 53, no. 6, pp. 1476–1491, Jul. 2008.
- [2] M. Benaïm, J. Hofbauer, and S. Sorin, "Stochastic approximations and differential inclusions," *SIAM J. Control Optim.*, vol. 44, no. 1, pp. 328–348, 2005.
- [3] K. Bimpikis, A. Ozdaglar, and E. Yildiz, "Competitive targeted advertising over networks," *Oper. Res.*, vol. 64, no. 3, pp. 705–720, 2016.
- [4] V. S. Borkar, *Stochastic Approximation: A Dynamical Systems Viewpoint*. New Delhi, India: Hindustan Book Agency, 2008.
- [5] V. S. Borkar, "Asynchronous stochastic approximations," *SIAM J. Control Optim.*, vol. 38, no. 3, pp. 662–663, 1998.
- [6] V. S. Borkar, A. Karnik, J. Nair, and S. Nalli, "Manufacturing consent," *IEEE Trans. Autom. Control*, vol. 60, no. 1, pp. 104–117, Jan. 2015.
- [7] V. S. Borkar and S. Pattathil, "Concentration bounds for two time scale stochastic approximation," in *Proc. 56th Annu. Allerton Conf. Commun., Control, Comput.*, Monticello, IL, USA, 2018, pp. 504–511.
- [8] B. Brogliato, A. Daniilidis, C. Lemarechal, and V. Acary, "On the equivalence between complementarity systems, projected systems and differential inclusions," *Syst. Control Lett.*, vol. 55, no. 1, pp. 45–51, 2006.
- [9] M.-G. Cojocaru and L. Jonker, "Existence of solutions to projected differential equations in Hilbert spaces," *Proc. Amer. Math. Soc.*, vol. 132, no. 1, pp. 183–193, 2004.
- [10] H. M. DeGroot, "Reaching a consensus," *J. Amer. Statist. Assoc.*, vol. 69, no. 345, pp. 118–121, 1974.
- [11] T. T. Doan, "Finite-time convergence rates of nonlinear two-time-scale stochastic approximation under Markovian noise," 2021, *arXiv:2104.01627*.
- [12] E. Dugundji and J. Walker, "Discrete choice with social and spatial network interdependencies: An empirical example using mixed generalized extreme value models with field and panel effects," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 1921, pp. 70–78, 2005.
- [13] P. Dupuis, "Large deviations analysis of reflected diffusions and constrained stochastic approximation algorithms in convex sets," *Stochastics: Int. J. Probab. Stochastic Processes*, vol. 21, no. 1, pp. 63–96, 1987.
- [14] M. Farajtabar, N. Du, M. Gomez-Rodriguez, I. Valera, H. Zha, and L. Song, "Shaping social activity by incentivizing users," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2474–2482.
- [15] J.-P. Forestier and P. Varaiya, "Multilayer control of large Markov chains," *IEEE Trans. Autom. Control*, vol. 23, no. AC-2, pp. 298–305, Apr. 1978.
- [16] B. S. Gelfand and S. K. Mitter, "Recursive stochastic algorithms for global optimization in \mathcal{R}^d ," *SIAM J. Control Optim.*, vol. 29, no. 5, pp. 999–1018, 1991.
- [17] A. Guille, H. Hacid, C. Favre, and D. A. Zighed, "Information diffusion in online social networks: A survey," *ACM SIGMOD Rec.*, vol. 42, no. 2, pp. 17–28, 2013.
- [18] D. D. Heckathorn, "Respondent-driven sampling: A new approach to the study of hidden populations," *Social Problems*, vol. 44, pp. 174–199, 1997.
- [19] D. Kempe, J. Kleinberg, and É. Tardos, "Influential nodes in a diffusion model for social networks," in *Automata, Lang. Program.* New York, NY, USA: Springer, 2005, pp. 1127–1138.
- [20] H. J. Kushner and D. S. Clark, *Stochastic Approximation Algorithms for Constrained and Unconstrained Systems*. New York, NY, USA: Springer, 1978.
- [21] S.-C. Lin, S.-D. Lin, and M.-S. Chen, "A learning-based framework to handle multi-round multi-party influence maximization on social networks," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2015, pp. 695–704.
- [22] D. Merugu, S. B. Prabhakar, and N. Rama, "An incentive mechanism for decongesting the roads: A pilot program in Bangalore," in *Proc. ACM NetEcon Workshop*, 2009, pp. 1–6.
- [23] A. Makkadem and M. Pelletier, "Convergence rate and averaging of nonlinear two-time-scale stochastic approximation algorithms," *Ann. Appl. Probab.*, vol. 16, no. 3, pp. 1671–1702, 2006.
- [24] C. Ravazzi, S. Hojjatinia, M. C. Lagoa, and F. Dabbene, "Ergodic opinion dynamics over networks: Learning influences from partial observations," *IEEE Trans. Autom. Control*, vol. 66, no. 6, pp. 2709–2723, Jun. 2021.
- [25] C. Ravazzi, R. Tempo, and F. Dabbene, "Learning influence structure in sparse social networks," *IEEE Control Netw. Syst.*, vol. 5, no. 4, pp. 1976–1986, Dec. 2018.
- [26] A. Reiffers-Masson, Y. Hayel, and E. Altman, "Posting behaviour dynamics and active filtering for content diversity in social networks," *IEEE Trans. Signal Inf. Process. Over Netw.*, vol. 3, no. 2, pp. 376–387, Jun. 2017.
- [27] G. Scardi, "Igraphdata: A collection of network data sets for the 'igraph' package," *R Package Version 1.0.1*, 2015.
- [28] P. W. Schultz, J. M. Nolan, R. B. Cialdini, N. J. Goldstein, and V. Griskevicius, "The constructive, destructive, and reconstructive power of social norms: Reprise," *Perspectives Psychol. Sci.*, vol. 13, no. 2, pp. 249–254, 2018.
- [29] Y. Shen, W. Stannat, and K. Obermayer, "Risk-sensitive Markov control processes," *SIAM J. Control Optim.*, vol. 51, no. 5, pp. 3652–3672, 2013.
- [30] L. Vassio, F. Fagnani, P. Frasca, and A. Ozdaglar, "Message passing optimization of harmonic influence centrality," *IEEE Control Netw. Syst.*, vol. 1, no. 1, pp. 109–120, Mar. 2014.
- [31] H.-T. Wai, A. Scaglione, B. Barzel, and A. Leshem, "Joint network topology and dynamics recovery from perturbed stationary points," *IEEE Trans. Signal Process.*, vol. 67, no. 17, pp. 4582–4596, Sep. 2019.
- [32] H.-T. Wai, A. Scaglione, and A. Leshem, "Active sensing of social networks," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 3, pp. 406–419, Sep. 2016.
- [33] J. L. Walker, E. Ehlers, I. Banerjee, and E. R. Dugundji, "Correcting for endogeneity in behavioral choice models with social influence variables," *Transp. Res. Part A: Policy Pract.*, vol. 45, no. 4, pp. 362–374, 2011.
- [34] Y. Wang, E. Theodorou, A. Verma, and L. Song, "A stochastic differential equation framework for guiding online user activities in closed loop," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2018, pp. 1077–1086.
- [35] B. Xia et al., "EnergyCoupon: A case study on incentive-based demand response in smart grid," in *Proc. 8th Int. Conf. Future Energy Syst.*, 2017, pp. 80–90.
- [36] A. Yadav, H. Chan, A. X. Jiang, H. Xu, E. Rice, and M. Tambe, "Using social networks to aid homeless shelters: Dynamic influence maximization under uncertainty," in *Proc. Int. Conf. Auton. Agents Multiagent Syst.*, 2016, pp. 740–748.
- [37] A. Yildiz, A. Ozdaglar, D. Acemoglu, A. Saberi, and A. Scaglione, "Binary opinion dynamics with stubborn agents," *ACM Trans. Econ. Comput.*, vol. 1, no. 4, pp. 1–30, 2013.
- [38] A. Zarezade, A. De, H. Rabiee, and M. G. Rodriguez, "Cheshire: Algorithm for activity maximization in social networks," in *Proc. 55th Annu. Allerton Conf. Commun., Control, Comput.*, Monticello, IL, USA, 2017, pp. 1–16.
- [39] K. Zhou, H. Zha, and L. Song, "Learning social infectivity in sparse low-rank networks using multi-dimensional Hawkes processes," in *Proc. 16th Int. Conf. Artif. Intell. Statist.*, 2013, pp. 641–649.



Vivek S. Borkar (Fellow, IEEE) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology Bombay (IIT Bombay), Mumbai, India, in 1976, the M.S. degree in systems and control from Case Western Reserve University, Cleveland, OH, USA, in 1977, and the Ph.D. degree in electrical engineering and computer sciences from the University of California, Berkeley, CA, USA, in 1980.

He was with the TIFR Center for Applicable Mathematics and the Indian Institute of Science, Bengaluru, India, and the Tata Institute of Fundamental Research, Mumbai, before joining the IIT Bombay. His research interests include stochastic optimization and control.

Dr. Borkar is a Fellow of the American Mathematical Society, the World Academy of Sciences, and the Indian National Academy of Engineering.



Alexandre Reiffers-Masson received the B.Sc. degree in mathematics from Aix-Marseille University, Marseille, France, in 2010, the M.Sc. degree in applied mathematics from Pierre and Marie Curie University, Paris, France, in 2012, and the Ph.D. degree in computer science from the National Institute for Research in Computer Science and Automation, Rocquencourt, France, and Avignon University, Avignon, France, in 2016, under the supervision of Eitan Altman and Yezekael Hayel.

He is currently an Assistant Professor with IMT Atlantique, Brest, France. He was a Researcher with SafranTech, Magny-les-Hameaux, France, and a Postdoctoral Fellow with the Robert Bosch Centre for Cyberphysical Systems, Indian Institute of Science, Bengaluru, India. His research interests include game theory, optimization, stochastic processes, and machine learning to real-world problems in networks, economics, and manufacturing.