Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 50-1 (2017) 9211–9217

Network Design in the Presence of a Link Jammer: a Zero-Sum Game Formulation

Ioannis Kordonis [*,**] **George P. Papavassilopoulos** [**]

[*] *University of Southern California, Viterbi School of Engineering, 3740 McClintock Avenue, Los Angeles, CA 90089, (e-mail: jkordonis1920@yahoo.com)*

[**] *National Technical University of Athens, School of Electrical and Computer Engineering, 9 Iroon Polytechniou str., 157-80, Athens, (e-mail: yorgos@netmode.ntua.gr)*

Abstract: This paper considers the problem of designing a Network such that a set of dynamic rules converges as fast as possible to the Nash equilibrium in a class of repeated games, despite the attempt of a jammer to slow down the convergence by cutting a certain number of edges. Particularly we consider a class of quadratic games, motivated by the demand response problem in electricity markets. For a given network structure, a set of dynamic rules, based on approximate gradient decent is described. The convergence speed depends on the graph through a matrix which in turn depends on the graph Laplacian. The network design problem is formulated as a zero sum game between a network designer aiming to improve the convergence speed and a jammer who tries to deteriorate it. Simple heuristics for the designer and the jammer problems are proposed and a numerical example is presented.

Keywords: Network Design, Structural Optimization, Game Theory, Jamming, Zero Sum Game, Dynamic Rules

1. INTRODUCTION

Often in distributed systems and games, the sub-systems or players follow some simple dynamic rules in order to determine their actions, exchanging information through a communication graph. The performance of the overall system depends on the structure of that network. For example in consensus dynamics the convergence speed depends on the Laplacian of the communication graph. This work considers a simple quadratic aggregative game and focuses on the problem of designing the communication graph in order to optimize the system performance (i.e. the convergence speed to the Nash equilibrium) and its robustness with respect to link jamming. Specifically, we assume that the communication network is designed centrally and that there exists a jammer who wants to deteriorate the performance of the system, by cutting a certain rather small number of edges. This problem is formulated as a zero sum game between the network designer and the jammer. This work continues our previous work in Kordonis and Papavassilopoulos (2016).

The graph design for the fast convergence of the consensus dynamics and the related problems optimizing the network coherence, the Markov chain fastest mixing problem and the minimization of the effective resistance of a graph were studied in literature using several techniques and several criteria (Xiao and Boyd (2004), Dai and Mesbahi (2011), Charalampidis (2014), Xiao et al. (2007), Hassan-Moghaddam and Jovanović (2015), Boyd et al. (2004), Ghosh et al. (2008)). In El Chamie and Başar (2016), Liu and Başar (2014), Gharesifard and Başar (2012) and Khanafer et al. (2012) the communication graph or the dynamics are designed such that the overall system converges as close as possible to the average consensus, despite the attempt of an opponent to deteriorate the performance of the system.

In this work we focus on quadratic aggregative games. In this class of games, each one of the participants interacts with the aggregate actions of the rest of the players (Jensen (2010)). There are several applications of large aggregative games, such as the charging of electric vehicles (Ma et al. (2010), Parise et al. (2014)) and the demand response in the smart grid (Zhu and Başar (2011), Bagagiolo and Bauso (2014)). In this class of examples, the aggregative action of the players affects the energy price which in turn affects the individual costs.

The use of several of dynamic rules for the participants of large aggregative games was studied in Kizilkale and Caines (2013), Grammatico et al. (2016), Parise et al. (2015), Paccagnan et al. (2016a), Paccagnan et al. (2016b), Koshal et al. (2012). In many of these works, the dynamic rules use a network structure to transmit the necessary information. In the current work we consider a variant of the dynamic rule presented in Koshal et al. (2012). It turns out that the overall dynamics is linear and depend on the communication graph. Some convergence results are also proved.

Then the network design problem is formulated as a zero sum game between the network designer and a jammer. We consider a criterion describing the discounted cumulative quadratic distance from the Nash equilibrium. At first, the jammer's problem is considered and an approximate

10.1016/j.ifacol.2017.08.1278

algorithm is proposed. Particularly, using a set of relaxed variables, the effect of removing an edge is approximated by the partial derivative of the cost with respect to the corresponding adjacency matrix variable. We then study the problem of the network designer. In order to do so, we compute the approximate cost sensitivity with respect to edge removals and then add edges which reduce this sensitivity the most.

1.1 Notation

For a matrix $A \in \mathbb{R}^{m\times n}$, A_{ij} denotes the ij-th element. The standard basis vectors are denoted by e_i, i.e. e_i is a column vector having zeros in all its entries except entry i, which has the value 1. A column vector consisting of units is denoted by $\mathbf{1}$, i.e. $\mathbf{1} = [1\ 1\ \ldots 1]^T$.

For any pair of matrices $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{q\times r}$, the Kronecker product $A \otimes B$ is defined as:

$$A \otimes B = \begin{bmatrix} A_{11}B & \ldots & A_{1n}B \\ \vdots & \ddots & \vdots \\ A_{m1}B & \ldots & A_{mn}B \end{bmatrix}. \qquad (1)$$

The vectorization of a matrix $A = [A_1|A_2|\ldots|A_m]$, where A_j is the j-th column of A, is denoted by vec(A) and is given by $[A_1^T A_2^T \ldots A_m^T]^T$. The identity (ex. Horn and Johnson):

$$\text{vec}(AXB) = (B^T \otimes A)\text{vec}(X), \qquad (2)$$

will be used.

We denote by $G = (V, E)$ an undirected graph, where $V = (v_1, \ldots, v_N)$ is the set of vertices and E the set of edges. The adjacency matrix is denoted by A, i.e. $A_{ij} = 1$ if there exists an edge between vertices i and j and 0 otherwise.

We denote by d_i the degree of the node i, i.e. the number of edges adjacent to vertex i. The Laplacian of the graph is given by $L = \Delta - A$ where $\Delta = diag(d_1, \ldots, d_N)$.

2. GAME DESCRIPTION AND DYNAMIC RULE

Let us describe a class of quadratic games, motivated by the demand response problem. The set of players is $i = 1, \ldots, N$ and the cost of each player is given by:

$$J_i = (x_i - \theta_i)^2 + \left(1 + \sum_{j=1}^{N} x_j/N\right) x_i, \qquad (3)$$

where N is the number of energy consumers, x_i is the amount of energy consumed by player i, θ_i is the desired consumption of player i and $1 + \sum_{j=1}^{N} x_j/N$ is the energy cost per unit (a more general price function of the form $c_1 + c_2 \sum_{j=1}^{N} x_j/N$ could be used, but for simplicity reasons we use $c_1 = c_2 = 1$). The first term of (3) represent the disutility of player i for having a level of energy consumption different from the desired and the second term is the total amount of money that she pays.

The game is played repeatedly over time and each player holds an estimation of the mean value of the actions of the players. Let us denote by $\hat{x}_i(k)$ the estimation of player i for the mean value of the action of the players.

The players exchange information through a network $G = (V, E)$ in which each player corresponds to a node. At every time step k, each player is informed for the estimated values $\hat{x}_i(k)$ of her neighbors. Throughout this section we assume that the graph G is connected.

We then describe a set of dynamic rules for the participants of the game. At each time step, the action of each player is updated in order to reduce her cost, using an approximate gradient decent rule:

$$x_i(k+1) = x_i(k) - \alpha(\partial J_i/\partial x_i)^{\text{aprox}}, \qquad (4)$$

where $(\partial J_i/\partial x_i)^{\text{aprox}}$ is an approximation of of $\partial J_i/\partial x_i$. The approximation of $\partial J_i/\partial x_i$ is derived substituting the last term of:

$$\frac{\partial J_i}{\partial x_i} = (2 + \frac{1}{N})x_i - 2\theta_i + 1 + \sum_{j=1}^{N} x_j/N, \qquad (5)$$

by $\hat{x}_i$. Thus, the dynamic rule for player i is given by:

$$x_i(k+1) = (1-\bar{\alpha})x_i(k) - \alpha\hat{x}_i(k) + \alpha(2\theta_i - 1) \qquad (6)$$
$$x_i(0) = \theta_i,$$

where $\bar{\alpha} = \alpha(2 + 1/N)$.

At every time step, the players update their estimates according to:

$$\hat{x}_i(k+1) = (1 - d_i\delta)\hat{x}_i(k) + \delta \sum_{j\in\mathcal{N}_i} \hat{x}_j(k) + \\ + (-\bar{\alpha}x_i(k) - \alpha\hat{x}_i(k) + \alpha(2\theta_i - 1)) \qquad (7)$$
$$\hat{x}_i(0) = \theta_i,$$

where $\mathcal{N}_i$ is the set of neighbors of player i and d_i is the degree (the number of neighbors) of the node of player i. The first two terms (7) correspond to the consensus dynamics and the last to the fact that x_i is actually changing.

Remark 1. The dynamic rule proposed, closely parallels the rule described in Koshal et al. (2012). The basic differences are that Koshal et al. (2012) considers a time varying stochastic pairwise information exchange and a Stochastic Approximation type decreasing step-size ($\alpha_k \to 0$) is used. Thus, the convergence rate may be slow. On the other hand the current work analyzes a simpler problem, in which there is a synchronous communication with all the neighbours. Hence, a constant step size may be used and the dynamic rule converges exponentially to the Nash equilibrium.

The dynamics can be written in compact form as:

$$\tilde{x}(k+1) = P\tilde{x}(k) + B\Theta \qquad (8)$$
$$\tilde{x}(0) = \begin{bmatrix}\Theta \\ \Theta\end{bmatrix}$$

where $x = [x_1\ \ldots\ x_N]^T$, $\hat{x} = [\hat{x}_1\ \ldots \hat{x}_N]^T$, $\tilde{x} = [x^T\ \hat{x}^T]^T$,

$$\Theta = [2\theta_1 - 1\ \ldots\ 2\theta_N - 1]^T, \qquad (9)$$

and

$$B = \begin{bmatrix}\alpha I \\ \alpha I\end{bmatrix}, \quad P = \begin{bmatrix}(1-\bar{\alpha})I & -\alpha I \\ -\bar{\alpha}I & (1-\alpha)I - \delta L\end{bmatrix}. \qquad (10)$$

The dynamics of the overall system depends on the matrix P, which in turn depends on the Laplacian matrix. Some properties of matrix P are shown in the following lemma.

Lemma 1. Assuming that G is connected, the dynamics (8) have the following properties:

(i) The matrix P has an eigenvalue 1 with left eigenvector $[\mathbf{1}^T \;\; -\mathbf{1}^T]$ and a unique right eigenvector

$$d = \begin{bmatrix} \mathbf{1} \\ -(2+1/N)\mathbf{1} \end{bmatrix}$$

(ii) The subspace:

$$C = \{\tilde{x} : [\mathbf{1}^T \;\; -\mathbf{1}^T]\tilde{x} = 0\}, \tag{11}$$

is invariant under (8)

(iii) If (8) converges to a fixed point:

$$\tilde{x}^N = \begin{bmatrix} x^N \\ \hat{x}^N \end{bmatrix},$$

then x^N is the unique Nash equilibrium of the game and $\hat{x}^N = (\mathbf{1}^T x^N/N)\mathbf{1}$.

Proof: For a similar result see Kordonis and Papavassilopoulos (2016). □

The Nash equilibrium of the game is given by:

$$x^N = \frac{N}{2N+1}\left[I - \frac{1}{3N+1}\mathbf{1}\mathbf{1}^T\right]\Theta = D_1\Theta \tag{12}$$

and the vector $\hat{x}^N$ by:

$$\hat{x}^N = \frac{1}{3N+1}\mathbf{1}\mathbf{1}^T\Theta = D_2\Theta. \tag{13}$$

Denote by D the matrix $[D_1^T D_2^T]^T$.

Lemma 1 shows that both $span\{d\}$ and C are P-invariant. Furthermore, $\mathbb{R}^{2N} = C + span\{d\}$. The dynamics (8) evolve in C. Thus, an equivalent description of (8) in C could be obtained using a matrix $\tilde{P}$ such that $\tilde{P}\tilde{x} = P\tilde{x}$ if $\tilde{x} \in C$ and $\tilde{P}d = 0$. The following lemma shows this possibility.

Lemma 2. Consider the matrix $\tilde{P} = P(I+M)$ where M is given by:

$$M = \frac{1}{3N+1}\begin{bmatrix} -\mathbf{1}\mathbf{1}^T & \mathbf{1}\mathbf{1}^T \\ (2+\frac{1}{N})\mathbf{1}\mathbf{1}^T & -(2+\frac{1}{N})\mathbf{1}\mathbf{1}^T \end{bmatrix}. \tag{14}$$

(i) It holds, $\tilde{P}\tilde{x} = P\tilde{x}$ if $\tilde{x} \in C$ and $\tilde{P}d = 0$.

(ii) Dynamics (8) has the same trajectories with:

$$\tilde{x}(k+1) = \tilde{P}\tilde{x}(k) + B\Theta, \tag{15}$$

under the initial conditions described.

Proof: Immediate □

We may observe that the matrix $\tilde{P}$ has the same eigenstructure with P except for the eigenpair $(1,d)$ which in $\tilde{P}$ becomes $(0,d)$.

The network will be designed such that the matrix $\tilde{P}$ is stable. Under this assumption, the dynamics (8) converges to the vector $\tilde{x}^N$ corresponding to the Nash equilibrium. The distance form equilibrium $y(k) = \tilde{x}(k) - \tilde{x}^N$ evolves according to:

$$y(k+1) = \tilde{P}y(k). \tag{16}$$

3. THE NETWORK DESIGN PROBLEM

The network transmitting the information is designed centrally and the aim of the planner is to design a network such that the dynamic rules of the players converge as quickly as possible to the Nash equilibrium, despite the fact that an opponent may destroy a number of links. The link cuts may be interpreted either as the actions of an actual opponent or as link failures. In the latter case the problem considered corresponds to the worst case analysis.

We assume that the planner does not know the types Θ of the players. Furthermore, the same topology is designed for many repetitions of the game. Thus, a stochastic model for the types of the players is used. Based on the stochastic description a discounted quadratic criterion is introduced and the network design problem is described as a zero sum game.

3.1 A Discounted Quadratic Criterion

We assume that the type of each player is given by:

$$\theta_i = \mu_i + w_i, \tag{17}$$

where μ_i is the mean of player i's type and w_i a zero mean random vector. Denote by θ the vector of types. We further assume that the vector of means μ and the covariance matrix $\Sigma = E[ww^T]$ of θ are known to the planner.

The network design problem has two objectives. At first the matrix $\tilde{P}$ should should have spectral radius less than or equal to 1, such that the players actions remain bounded. The second objective is captured by a discounted quadratic criterion, quantifying the speed of convergence:

$$J^d = E\left[\sum_{k=0}^{\infty}\rho^k \sum_{i=1}^{N}(x^i(k) - x^{i,N})^2\right]. \tag{18}$$

Remark 2. The jammer may cut a certain number of edges resulting to a disconnected network. In this case we may not expect convergence to the Nash equilibrium and we will have a steady state error. An undiscounted infinite horizon cost would lead to an infinite cost. The reason we choose a discounted criterion is to take into account both the transient and the steady state error. An alternative representation is that the dynamic rule will stop after a random number of steps (see for example Boukas et al. (1990)).

The criterion (18) can be written as:

$$J^d = E\left[y^T(0)\left(\sum_{k=0}^{\infty}\rho^k(\tilde{P}^k)^T Q\tilde{P}^k\right)y(0)\right], \tag{19}$$

where:

$$Q = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}. \tag{20}$$

If $\sqrt{\rho}\tilde{P}$ is stable then the matrix $X = \sum_{k=0}^{\infty}\rho^k(\tilde{P}^T)^k Q\tilde{P}^k$ is the unique solution of the Lyapunov equation:

$$\rho\tilde{P}^T X\tilde{P} - X + Q = 0, \tag{21}$$

which can be expressed in terms of the Kronecker product as:

$$f_Q(\tilde{P}, \mathrm{vec}(X)) = \left(I - \rho\tilde{P}^T \otimes \tilde{P}^T\right)\mathrm{vec}(X) - \mathrm{vec}(Q) = 0 \tag{22}$$

The cost can be written as:

$$J^d = \sum_{i=1}^{2N}\sum_{j=1}^{2N} X_{ij}E\left[y^i(0)y^j(0)\right] = \mathrm{vec}(S)^T\mathrm{vec}(X), \tag{23}$$

where $S = E[y(0)y(0)^T]$. In order to compute S recall that:

$$y(0) = \begin{bmatrix} I \\ I \end{bmatrix} \theta - D(2\theta - \mathbf{1}) = \left(\begin{bmatrix} I \\ I \end{bmatrix} - 2D \right) \theta + D\mathbf{1}. \quad (24)$$

Denoting by $\bar{D}$ the matrix $[I\ I]^T - 2D$ we get:

$$S = \bar{D}(\mu\mu^T + \Sigma)\bar{D}^T + \bar{D}\mu\mathbf{1}^T D^T + D\mathbf{1}\mu^T\bar{D}^T + D\mathbf{1}\mathbf{1}^T D^T \quad (25)$$

3.2 The Network Design Problem

Let us then state the network design problem. Assume that there is a graph $G_0 = (V, E_0)$ describing the existing links among the N players. Then, the designer should choose the place where a set of additional links $E_{new} = \{l_1, \ldots, l_{max}\}$ will be introduced to the graph in order to minimize J^d, knowing that possibly a set of at most m edges will be removed. The jammer, knowing the set of edges $E = E_0 \cup E_{new}$, chooses a subset of $E_j \subset E \cap E_{vul}$ to cut, with $|E_j| = m$ in order to maximize J_d. The set E_{vul} represents all the links which the jammer has the ability to cut. The designer's problem is:

$$\begin{aligned} \underset{E_{new}}{\text{minimize}} \quad & \max_{E_j \subset E \cup E_{vul}, |E_j| \le m} \{\text{vec}(S)^T \text{vec}(X)\} \\ \text{subject to} \quad & \tilde{P} = \tilde{P}(A) \\ & \left(I - \rho\tilde{P}^T \otimes \tilde{P}^T\right) \text{vec}(X) = \text{vec}(Q) \\ & \tilde{P}\text{: has spectral radius less than or equal to 1} \end{aligned} \quad (26)$$

where $\tilde{P} = \tilde{P}(A)$ is given by $\tilde{P} = P(I + M)$, (10) and (14).

Remark 3. The solution of (26) represents the security level (ex. Basar and Olsder (1999)) of the network designer in the zero sum game where the network designer has cost J^d and the jammer $-J^d$.

It should be pointed out that if (26) is to be considered as a zero sum game and a zero sum solution is sought, then several issues may arise. First, it can actually be seen equivalently as a matrix a game for which we seek a pure strategy solution, which may not exist. Then we could seek for a mixed solution via the known Linear Programming formulation. The dimensions are quite formidable and thus we can resort to other algorithms such as Simulated Annealing and its variants. Clearly such random search methods can be employed also for the security level formulation we chose to consider here since even for this formulation the dimensions are quite large. We will pursue this line in future work.

The optimization problem (26) is a nonlinear mixed integer programming problem (Wolsey and Nemhauser (2014)). Thus, in general it could be difficult to solve. In the following simple heuristics for the designer's and jammer's problems are suggested.

4. HEURISTICS FOR THE OPTIMIZATION PROBLEM (26)

4.1 The Jammer's Problem

Let us start with the jammer's problem. A simple algorithm based on the approximate influence of removing edges is proposed. In order to do so, the binary variables A_{ij} are temporarily assumed to have continuous values in $[0, 1]$ (relaxation) and the derivative of the cost with respect to A_{ij} is considered. We assume that A_{ij} with $j < i$, corresponding to existing edges are the free variables and that $A_{ji} = A_{ij}$. For a given matrix A the effect of removing link (i, j) to the cost is approximated by the derivative $\partial J^d(\tilde{P}(A), X(\tilde{P}(A)))/\partial A_{i,j}$.

This partial derivative can be computed using the chain rule and implicit function theorem:

$$\frac{\partial J^d}{\partial A_{i,j}} = -\text{vec}(S)^T \left(\frac{\partial f_Q}{\partial \text{vec}(X)} \right)^{-1} \frac{\partial f_Q}{\partial \text{vec}(\tilde{P})} \frac{\partial \text{vec}(\tilde{P})}{\partial A_{ij}} \quad (27)$$

The first two terms satisfy:

$$(\text{vec}(S))^T \left(\frac{\partial f_Q}{\partial \text{vec}(X)} \right)^{-1} = \\ = \left(\left(I - \rho\tilde{P} \otimes \tilde{P} \right)^{-1} \text{vec}(S) \right)^T = \text{vec}(Y)^T \quad (28)$$

where Y satisfies the Lyapunov equation:

$$\rho(\tilde{P}^T)^T Y (\tilde{P}^T) - Y = S, \quad (29)$$

or $f_S(\tilde{P}^T, \text{vec}(Y)) = 0$.

For the third term it holds,

$$\frac{\partial f_Q}{\partial \tilde{P}_{i'j'}} = -\rho \left(e_{j'} e_{i'}^T \otimes \tilde{P}^T + \tilde{P}^T \otimes e_{j'} e_{i'}^T \right) \text{vec}(X). \quad (30)$$

Denote by $\bar{f}_{i'j'}(\tilde{P}, X)$ the value of $\frac{\partial f_Q}{\partial \tilde{P}_{i'j'}}$. Furthermore, it holds:

$$\frac{\partial \tilde{P}}{\partial A_{ij}} = \frac{\partial P}{\partial A_{ij}} = \begin{bmatrix} 0 & 0 \\ 0 & \delta\left(e_i e_j^T + e_j e_i^T - e_j e_j^T - e_i e_i^T\right) \end{bmatrix}. \quad (31)$$

The equality, $\frac{\partial \tilde{P}}{\partial A_{ij}} = \frac{\partial P}{\partial A_{ij}}$ holds true due to the fact that $\frac{\partial P}{\partial A_{ij}} M = 0$.

Thus, the last two terms of (27) satisfy:

$$\frac{\partial f_Q}{\partial \text{vec}(\tilde{P})} \frac{\partial \text{vec}(\tilde{P})}{\partial A_{ij}} = \delta \left(\bar{f}_{i+N,j+N}(\tilde{P}, X) + \right. \\ \left. + \bar{f}_{j+N,i+N}(\tilde{P}, X) - \bar{f}_{i+N,i+N}(\tilde{P}, X) + \bar{f}_{j+N,j+N}(\tilde{P}, X) \right) \quad (32)$$

A simple technique for the jammer is described in Algorithm 1

Algorithm 1

1: Get the designed graph $G = (V, E)$.
2: Set $l_{cnt} \leftarrow 0$ and $E_j \leftarrow \emptyset$.
3: Compute the matrices $\tilde{P}$ and X.
4: For every edge $(i, j) \in E$ compute $\partial J^d/\partial A_{ij}$ and choose $(i^\star, j^\star)$ which minimizes $\partial J^d/\partial A_{ij}$.
5: Set $E \leftarrow E \setminus (i^\star, j^\star)$ and $l_{cnt} \leftarrow l_{cnt} + 1$
5: If $l_{cnt} < m$ go to Step 3. Else halt.

4.2 The designer's problem

We then propose a simple heuristic for the designer's problem. The technique is based on the idea to reduce the sensitivity of the designed graph with respect to link removals. Particularly, the designer adds $l_{max} - m$ edges to reduce the cost J^d and the rest m of the edges to reduce the sensitivity of the designed graph in edge removal.

The sensitivity of a given graph may be approximated by $max_{(i,j)\in E}\{\partial J^d/\partial A_{ij}\}$. Fix the edge $(i^\star, j^\star)$ which attains the maximum. The dependence of the sensitivity on the addition of a new edge (i_n, j_n) may be approximated by $\frac{\partial[\partial J^d/\partial A_{i^\star j^\star}]}{\partial A_{i_n j_n}}$.

It holds:

$$\frac{\partial\left[\partial J^d/\partial A_{i^\star j^\star}\right]}{\partial A_{i_n,j_n}} = -\frac{\partial \mathrm{vec}(Y)^T}{\partial A_{i_n j_n}} \frac{\partial f_Q}{\partial \mathrm{vec}(\tilde{P})} \frac{\partial \mathrm{vec}(\tilde{P})}{\partial A_{i^\star j^\star}} + $$
$$+ \mathrm{vec}(Y)^T \frac{\partial}{\partial A_{i_n j_n}} \left[\frac{\partial f_Q}{\partial \mathrm{vec}(\tilde{P})} \frac{\partial \mathrm{vec}(\tilde{P})}{\partial A_{i^\star j^\star}}\right] \quad (33)$$

Let us first compute the first term of (33). The vector Y satisfies the Lyapunov equation (29). If $\tilde{P}$ is stable, implicit function theorem implies that there exists locally a function $g_S(\tilde{P}^T)$, such that $f_S(\tilde{P}^T, g_S(\tilde{P}^T)) = 0$ and that:

$$\frac{\partial \mathrm{vec}(Y)}{\partial A_{i_n j_n}} = \left(\frac{\partial \mathrm{f}_S}{\partial \mathrm{vec}(Y)}\right)^{-1} \frac{\partial f_S}{\partial \mathrm{vec}(\tilde{P}^T)} \frac{\partial \mathrm{vec}(\tilde{P}^T)}{\partial A_{i_n j_n}}. \quad (34)$$

Furthermore,

$$\frac{\partial f_S}{\partial \tilde{P^T}_{j'i'}} = -\rho\left(e_{j'}e_{i'}^T \otimes \tilde{P} + \tilde{P} \otimes e_{j'}e_{i'}^T\right) \mathrm{vec}(Y), \quad (35)$$

and $\frac{\partial \mathrm{vec}(\tilde{P}^T)}{\partial A_{i_n j_n}}$ was already computed in (31).

Let us then compute the second term of (33). Using (32), it remains to compute $\frac{\partial \bar{f}_{i'j'}(\tilde{P},X)}{\partial A_{i_n j_n}}$. It holds:

$$\frac{\partial \bar{f}_{i'j'}(\tilde{P},X)}{\partial A_{i_n j_n}} = \sum_{i,j} \frac{\partial \bar{f}_{i'j'}}{\partial \tilde{P}_{ij}} \frac{\partial \tilde{P}_{ij}}{\partial A_{i_n j_n}} \quad (36)$$

where

$$\frac{\partial \bar{f}_{i'j'}}{\partial \tilde{P}_{ij}} = -\rho\left(e_{j'}e_{i'}^T \otimes e_j e_i^T + e_j e_i^T \otimes e_{j'}e_{i'}^T\right) \mathrm{vec}(X) +$$
$$+\rho^2 \left(e_{j'}e_{i'}^T \otimes \tilde{P}^T + \tilde{P}^T \otimes e_{j'}e_{i'}^T\right)(I - \rho\tilde{P}^T \otimes \tilde{P}^T)^{-1} \cdot$$
$$\cdot \left(e_j e_i^T \otimes \tilde{P}^T + \tilde{P}^T \otimes e_j e_i^T\right) \mathrm{vec}(X) \quad (37)$$

Combining (32)-(37) $\frac{\partial[\partial J^d/\partial A_{i^\star j^\star}]}{\partial A_{i_n j_n}}$ is computed.

A simple technique for the network designer is described in Algorithm 2.

5. NUMERICAL EXAMPLES

Example 1. In this example we study how the designer's cost varies when an edge is removed. We consider a very simple linear graph with 20 nodes (Figure 1). The mean of the type of each player μ_i is 0.5. The matrix covariance of the types is given by:

$$\Sigma = \begin{bmatrix} 0.5 & p & 0 & \dots & 0 \\ p & 0.5 & p & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \end{bmatrix}. \quad (38)$$

Algorithm 2

1: Get the designed graph $G = (V, E_0)$.
2: Set $l_{cnt} \leftarrow 0$ and $E_{new} \leftarrow \emptyset$.
3: Compute the matrices $\tilde{P}$ and X.
4: For every edge $(i,j) \notin E_0 \cup E_{new}$ compute $\partial J^d/\partial A_{ij}$ and choose $(i^\star, j^\star)$ which minimizes $\partial J^d/\partial A_{ij}$.
5 If $l_{cnt} > l_{max} - m$ go to Step 7.
6: Set $E \leftarrow E \cup (i^\star, j^\star)$ and $l_{cnt} \leftarrow l_{cnt} + 1$. Go to Step 9.
7: For every edge $(i_n, j_n) \notin E_0 \cup E_{new}$ compute $\frac{\partial[\partial J^d/\partial A_{i^\star j^\star}]}{\partial A_{i_n j_n}}$ and choose (i^{new}, j^{new}) which minimizes the partial derivative.
8: Set $E \leftarrow E \cup (i^{new}, j^{new})$ and $l_{cnt} \leftarrow l_{cnt} + 1$.
9: If $l_{cnt} < m$ go to Step 3. Else halt.

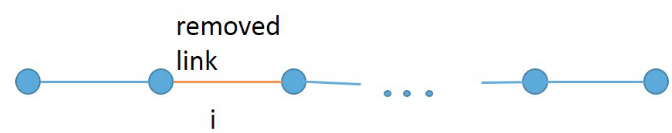

Fig. 1. The graph of Example 1

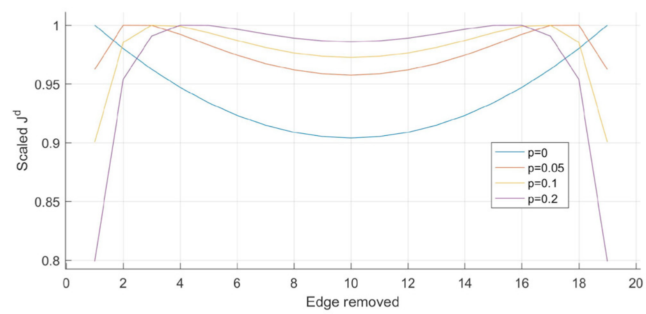

Fig. 2. The cost J^d for the graph in Example 1. The horizontal axis is the index of the link removed by the jammer and the vertical axis is the scaled cost.

That is, each player has a type correlated only with his immediate neighbors. Figure 2 illustrates the value of the criterion J^d when the edge among nodes $(i, i+1)$ has been removed, for various values of i and p.

We may observe that the effects of an edge removal depend essentially on the stochastic characteristics of the players' types.

Example 2. In this example, the initial graph is shown in Figure 3. The means and covariance matrix of the players' types is as in Example 1 with $p = 0.1$. We assume that both the network designer can design 1 additional link and the jammer could cut 1 link. Applying Algorithm 1, the jammer's choice would be link $(3, 7)$. Using Algorithm 2, the network designer will add the edge $(2, 7)$.

The heuristics examined do not necessarily give the optimal solution and in the future we plan to use or develop alternative techniques for the problem (26).

6. CONCLUSION

We considered the problem of designing an information exchange network such that a set gradient based rules to converge to the Nash equilibrium as fast as possible, despite the presence of a link jammer. The underlying

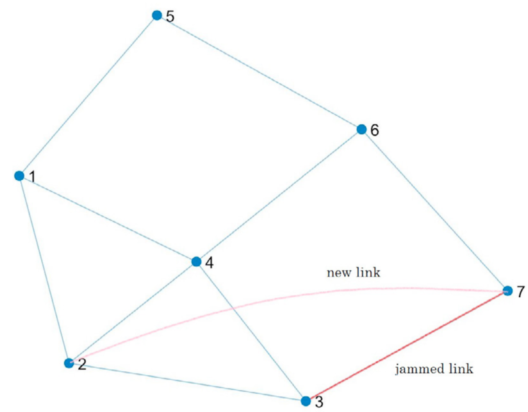

Fig. 3. The graph of Example 2.

game is motivated by the demand response problem. We first characterized the convergence to the Nash equilibrium using the eigenstructure of a matrix depending on the graph Laplacian and then stated a discounted infinite horizon criterion capturing both the transient and steady state errors. The network design problem was formulated as an integer valued zero sum game among the designer and the jammer, for which a security level solution, i.e. minimax was sought. Some simple heuristic rules for both players were proposed.

There are several possible directions for future research. A very interesting one is to study alternative methods to solve the optimization problems. Due to the high dimensionality, good candidates would be simulated annealing and other organized random searches. Another direction is to generalize the underlying game structure to games involving players with different influence, multidimensional or constrained actions and non-quadratic cost functions.

REFERENCES

Bagagiolo, F. and Bauso, D. (2014). Mean-field games and dynamic demand management in power grids. *Dynamic Games and Applications*, 4(2), 155–176.

Basar, T. and Olsder, G.J. (1999). *Dynamic noncooperative game theory*, volume 23. Siam.

Boukas, E.K., Haurie, A., and Michel, P. (1990). An optimal control problem with a random stopping time. *Journal of optimization theory and applications*, 64(3), 471–480.

Boyd, S., Diaconis, P., and Xiao, L. (2004). Fastest mixing markov chain on a graph. *SIAM review*, 46(4), 667–689.

Charalampidis, A.C. (2014). Graph topology optimization for fast consensus based on the influence of new links to a quadratic criterion. In *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, 1452–1457. IEEE.

Dai, R. and Mesbahi, M. (2011). Optimal topology design for dynamic networks. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, 1280–1285. IEEE.

El Chamie, M. and Başar, T. (2016). A zero-sum game between the network designer and an adversary in consensus protocols. In *Advances in Dynamic and Evolutionary Games*, 117–137. Springer.

Gharesifard, B. and Başar, T. (2012). Resilience in consensus dynamics via competitive interconnections. *IFAC Proceedings Volumes*, 45(26), 234–239.

Ghosh, A., Boyd, S., and Saberi, A. (2008). Minimizing effective resistance of a graph. *SIAM review*, 50(1), 37–66.

Grammatico, S., Parise, F., Colombino, M., and Lygeros, J. (2016). Decentralized convergence to nash equilibria in constrained deterministic mean field control. *IEEE Transactions on Automatic Control*, PP(99), 1–1. doi: 10.1109/TAC.2015.2513368.

Hassan-Moghaddam, S. and Jovanović, M.R. (2015). An interior-point method for topology identification and optimal design of noisy networks. *arXiv preprint arXiv:1506.03437*.

Horn, R.A. and Johnson, C.R. (????). Topics in matrix analysis, 1991. *Cambridge University Presss, Cambridge*, 37, 39.

Jensen, M.K. (2010). Aggregative games and best-reply potentials. *Economic theory*, 43(1), 45–66.

Khanafer, A., Touri, B., and Başar, T. (2012). Consensus in the presence of an adversary. *IFAC Proceedings Volumes*, 45(26), 276–281.

Kizilkale, A.C. and Caines, P.E. (2013). Mean field stochastic adaptive control. *Automatic Control, IEEE Transactions on*, 58(4), 905–920.

Kordonis, I. and Papavassilopoulos, G. (2016). Network design for fast convergence to the nash equilibrium in a class of repeated games. In *2016 24th Mediterranean Conference on Control and Automation (MED)*, 1026–1032. IEEE.

Koshal, J., Nedic, A., and Shanbhag, U.V. (2012). A gossip algorithm for aggregative games on graphs. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, 4840–4845. IEEE.

Liu, J. and Başar, T. (2014). Toward optimal network topology design for fast and secure distributed computation. In *International Conference on Decision and Game Theory for Security*, 234–245. Springer.

Ma, Z., Callaway, D., and Hiskens, I. (2010). Decentralized charging control for large populations of plug-in electric vehicles. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, 206–212. IEEE.

Paccagnan, D., Gentile, B., Parise, F., Kamgarpour, M., and Lygeros, J. (2016a). Distributed computation of generalized Nash equilibria in quadratic aggregative games with affine coupling constraints. In *IEEE Conference on Decision and Control.*

Paccagnan, D., Kamgarpour, M., and Lygeros, J. (2016b). On aggregative and mean field games with applications to electricity markets. In *European Control Conference*, 1–6.

Parise, F., Colombino, M., Grammatico, S., and Lygeros, J. (2014). Mean field constrained charging policy for large populations of plug-in electric vehicles. In *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, 5101–5106. IEEE.

Parise, F., Grammatico, S., Gentile, B., and Lygeros, J. (2015). Network aggregative games and distributed mean field control via consensus theory. *arXiv preprint arXiv:1506.07719.*

Wolsey, L.A. and Nemhauser, G.L. (2014). *Integer and combinatorial optimization*. John Wiley & Sons.

Xiao, L. and Boyd, S. (2004). Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53(1), 65–78.

Xiao, L., Boyd, S., and Kim, S.J. (2007). Distributed average consensus with least-mean-square deviation. *Journal of Parallel and Distributed Computing*, 67(1), 33–46.

Zhu, Q. and Başar, T. (2011). A multi-resolution large population game framework for smart grid demand response management. In *Network Games, Control and Optimization (NetGCooP), 2011 5th International Conference on*, 1–8. IEEE.