Network Design for Fast Convergence to the Nash Equilibrium in a Class of Repeated Games

I. Kordonis[1], G.P. Papavassilopoulos[1],

Abstract— **This work studies the problem of designing a Network such that a set of dynamic rules, in a class of repeated games, converges quickly to the Nash equilibrium. Particularly a very simple class of repeated games with mean field interactions is considered and we assume that the actions of the participants are determined using some simple myopic gradient based dynamic rules. The information about the actions of the other players is transmitted through a Network, using a consensus type dynamics. The speed of the convergence to equilibrium is characterized, using the Lyapunov equation involving a Laplacian like matrix. A topology optimization problem for the communication graph is then stated and an algorithm, based on the effects of new edges to the speed of convergence, is proposed. Numerical results are also given.**

I. INTRODUCTION

This work considers a Network Design problem for the fast convergence of dynamic rules in a class of repeated games. Network design problems were already studied in Multiagent Systems and Cooperative Control literature (ex. [1]). In this class of problems, several subsystems such as sensors, mobile robots or local controllers cooperate in order to achieve a certain common goal, such as to agree on a certain quantity to estimate a value or to form a certain shape. The graph design for the fast convergence of the consensus dynamics was studied in literature using several techniques and several criteria [2], [3], [4]. Another related problem is the graph topology design to optimize the network coherence, [5] - [6]. Other related network design problems include the Markov chain fastest mixing problem [7], the minimization of the effective resistance of a graph [8], the design of optimal synchronization [9] and the design of optimal sparse feedback gains [10].

In the last several years there is a large and growing literature for large Aggregative Games. In this class of games, each one of the individuals interacts with the aggregate actions of the rest of the players [11]. When the number of players is large the Mean Field Games approach was proposed [12], [13]. There are several applications of large aggregative games, such as the charging of electric vehicles [14], [15] and the demand response in the smart grid [16], [17].

The use of several of dynamic rules for the participants of large aggregative games was studied in [18], [19], [20], [21]. In many of these works, the dynamic rules use a network structure to transmit the necessary information. In the current work we focus on a variant of the dynamic rule presented in [21] and study the problem of designing the information network such that there is a fast convergence to the Nash equilibrium.

The network design problem is studied in a very simple class of repeated quadratic aggregative games. A slightly modified version of the set of dynamic rules used in [21] is proposed (see Rem. 1). The dynamics of the overall system depends on a variant of the Laplacian matrix. Assuming a random structure for the parameters of the game, the speed of convergence is characterized using the Lyapunov equation. Using this characterization, the problem of choosing the network structure such that the dynamic rules converge to the Nash equilibrium as fast as possible is stated as an optimization problem. An algorithm based on the approximate effect of new links is proposed. The algorithm aims, at each iteration, to find the edge that reduces the cost more and add that link to the existing graph. The optimization technique used closely parallels [4]. Some numerical examples are then given. It turns out that the speed of convergence of the dynamic rules, at least in some cases, may be much faster when the communication graph is designed using the algorithm, compared to the case where a random graph is used.

A. 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. $\mathbf{1} = [1\ 1\ \dots 1]^T$.

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:

$$A \otimes B = \begin{bmatrix} A_{11}B & \dots & A_{1n}B \\ \vdots & \ddots & \vdots \\ A_{m1}B & \dots & A_{mn}B \end{bmatrix}. \tag{1}$$

The vectorization of a matrix $A = [A_1|A_2|\dots|A_m]$, where A_j is the j-th column of A, is denoted by $\mathrm{vec}(A)$ and is given by $[A_1^T A_2^T \dots A_m^T]^T$. The identity (ex. [22]):

$$\mathrm{vec}(AXB) = (B^T \otimes A)\mathrm{vec}(X) \tag{2}$$

*This research has been cofinanced by the European Union (European Social Fund ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: THALES. Investing in knowledge society through the European Social Fund and the program ARISTEIA, project name HEPHAISTOS.

[1]The authors are with the National Technical University of Athens, School of Electrical and Computer Engineering, 9 Iroon Polytechniou str., Athens, Postal Code 157 80, Greece. I. Kordonis e-mail: jkordonis1920@yahoo.com, G.P. Papavassilopoulos e-mail: yorgos@netmode.ntua.gr.

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 $\Delta = diag(d_1, \ldots, d_N)$.

B. Organization

The rest of the current work is organized as follows: In Section II the game among the players is described and a set of dynamic rules is proposed for the convergence to the Nash equilibrium. In Section III the speed of convergence is characterized by a Lyapunov equation and the network design problem is stated. In Section IV a simple Algorithm is developed based on the derivative approximation of the influence of a new edge to to the designers cost. Section V presents some numerical results and Section VI summarizes the main contributions of the current work and proposes some future directions.

II. Game Description and Dynamics

Let us first describe the game among the players. There are N players, each one having a type θ_i. The cost for player p_i is given by:

$$J_i = (x_i - \theta_i)^2 + w_i \left(x_i - \frac{1}{N-1} \sum_{j \neq i} x_j \right)^2, \quad (3)$$

where $x_i \in \mathbb{R}$ is the action of player p_i. For simplicity, we assume that $w_i = 1$, for all the players. We also assume that each player is not informed about the types of the other players and that the types of the players belong to a bounded interval $[\underline{\theta}, \bar{\theta}]$.

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 p_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 of the graph G. At every time step k, each player is informed for the estimated values $\hat{x}_i(k)$ of her neighbors. We, further, assume that the graph G is connected.

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

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

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:

$$\frac{\partial J_i}{\partial x_i} = \left(4 + \frac{2}{N-1}\right) x_i - 2\theta_i - \frac{2}{N-1} \sum_{j=1}^{N} x_j, \quad (5)$$

by $\frac{2N}{N-1}\hat{x}_i$. Thus, the dynamic rule for player p_i is given by:

$$\begin{aligned} x_i(k+1) &= (1 - \bar{\alpha}) x_i(k) + \bar{\beta} \hat{x}_i + 2\alpha\theta_i \quad (6) \\ x_i(0) &= \theta_i, \end{aligned}$$

where $\bar{\alpha} = \alpha(4 + 2/(N-1))$ and $\bar{\beta} = 2\alpha N/(N-1)$.

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

$$\begin{aligned} \hat{x}_i(k+1) = (1 - d_i\delta)\hat{x}_i(k) + \delta \sum_{j \in \mathcal{N}_i} \hat{x}_j(k) + \\ + (-\bar{\alpha} x_i(k) + \bar{\beta}\hat{x}_i + 2\alpha\theta_i) \quad (7) \\ \hat{x}_i(0) = \theta_i. \end{aligned}$$

The first two terms 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 [21]. The basic differences are that [21] considers a time varying stochastic pairwise information exchange and a Stochastic Approximation type decreasing step-size ($\alpha_k \rightarrow 0$) 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:

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

where $x = [x_1 \ \ldots \ x_N]^T$, $\hat{x} = [\hat{x}_1 \ \ldots \hat{x}_N]^T$, $\tilde{x} = [x^T \ \hat{x}^T]^T$, $\Theta = [\theta_1 \ \ldots \ \theta_N]^T$ and

$$B = \begin{bmatrix} 2\alpha I \\ 2\alpha I \end{bmatrix}, \quad P = \begin{bmatrix} (1-\bar{\alpha})I & \bar{\beta} I \\ -\bar{\alpha} I & (1+\bar{\beta})I - \delta L \end{bmatrix}. \quad (9)$$

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

Lemma 1: 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

$$d = \begin{bmatrix} \mathbf{1} \\ (\bar{\alpha}/\bar{\beta})\mathbf{1} \end{bmatrix}$$

(ii) The subspace:

$$C = \{\tilde{x} : [\mathbf{1}^T \ -\mathbf{1}^T]\tilde{x} = 0\}, \quad (10)$$

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: (i) It holds:

$$[\mathbf{1}^T \ -\mathbf{1}^T]P = [\mathbf{1}^T \quad -\mathbf{1}^T + \delta\mathbf{1}^T L].$$

Due to the fact that $L = \Delta - A$, we have $\mathbf{1}^T L = 0$. Thus, the matrix has left eigenvector $[\mathbf{1}^T \ -\mathbf{1}^T]$ with eigenvalue 1.

A vector $[x^T \ \hat{x}^T]^T$ is a right eigenvector of 1 if and only if: $\bar{\alpha}x = \bar{\beta}\hat{x}$ and $L\bar{x} = 0$. Due to the fact that the graph is connected, the Laplacian has a unique eigenvector $\mathbf{1}$ corresponding to the 0 eigenvalue [1]. Thus, there is a unique (up to scalar multiplication) right eigenvalue of P corresponding to 1 which has the form:

$$d = \begin{bmatrix} \mathbf{1} \\ (\bar{\alpha}/\bar{\beta})\mathbf{1} \end{bmatrix}$$

(ii) If $\tilde{x}(k) \in C$, then:

$$\begin{aligned} [\mathbf{1}^T \ -\mathbf{1}^T]\tilde{x}(k+1) &= [\mathbf{1}^T \ -\mathbf{1}^T]P\tilde{x}(k) + [\mathbf{1}^T \ -\mathbf{1}^T]B\Theta \\ &[\mathbf{1}^T \ -\mathbf{1}^T]\tilde{x}(k) + [\mathbf{1}^T \ -\mathbf{1}^T]B\Theta = 0. \end{aligned}$$

Thus, C is invariant under (8).

(iii) Simple manipulations show that the game has a unique Nash equilibrium given by:

$$x_i = \frac{N-1}{2N-1}\theta_i + \frac{1}{2N-1}\sum_{j=1}^{N}\theta_j, \qquad (11)$$

or in matrix form:

$$x^N = \frac{1}{2N-1}\left((N-1)I + \mathbf{1}\mathbf{1}^T\right)\Theta \qquad (12)$$

If $\tilde{x} = [x^T \ \hat{x}^T]^T$ is a fixed point of (8) and $\tilde{x}(k) \to \tilde{x}$ with the given initial conditions, then $\tilde{x} \in C$, i.e. $\mathbf{1}^T x = \mathbf{1}^T \hat{x}$ and:

$$\begin{aligned} x &= (1-\bar{\alpha})x + \bar{\beta}\hat{x} + 2\alpha\Theta \\ \hat{x} &= -\bar{\alpha}x + (1+\bar{\beta})\hat{x} - \delta L\hat{x} + 2\alpha\Theta \end{aligned}$$

Subtracting the first from the second, we obtain $L\hat{x} = 0$. Due to the fact that the graph is connected, the Laplacian has a unique right eigenvector $\mathbf{1}$ corresponding to the eigenvalue 0. Furthermore, $\tilde{x} \in C$ implies $\hat{x} = (\mathbf{1}^T x/N)\mathbf{1}$. Thus, the i-th component of x satisfies:

$$x_i = \frac{\bar{\beta}}{\bar{\alpha}}\hat{x} + 2\alpha\theta_i/\bar{\alpha}. \qquad (13)$$

Equation (13) is for each i is equivalent to (11). □

Using (12), the vector $\tilde{x}^N$ can be computed by:

$$\tilde{x}^N = \begin{bmatrix} \frac{1}{2N-1}\left((N-1)I + \mathbf{1}\mathbf{1}^T\right) \\ \mathbf{1}\mathbf{1}^T/N \end{bmatrix}\Theta = D\Theta \qquad (14)$$

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:

$$M = \frac{1}{N(\bar{\alpha}-\bar{\beta})}\begin{bmatrix} \bar{\beta}\mathbf{1}\mathbf{1}^T & -\bar{\beta}\mathbf{1}\mathbf{1}^T \\ \bar{\alpha}\mathbf{1}\mathbf{1}^T & -\bar{\alpha}\mathbf{1}\mathbf{1}^T \end{bmatrix}. \qquad (15)$$

(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:

$$\tilde{x}(k+1) = \tilde{P}\tilde{x}(k) + B\Theta, \qquad (16)$$

under the initial conditions described.

Proof: (i) It is not difficult to see that $Md = -d$. Thus, $\tilde{P}d = 0$. Then consider an $\tilde{x} = [x^T \ \hat{x}^T]^T \in C$. Then, $\mathbf{1}^T x = \mathbf{1}^T \hat{x}$. Then,

$$M\tilde{x} = \begin{bmatrix} \bar{\beta}\mathbf{1}(\mathbf{1}^T x - \mathbf{1}^T\hat{x}) \\ \bar{\alpha}\mathbf{1}(\mathbf{1}^T x - \mathbf{1}^T\hat{x}) \end{bmatrix} = 0.$$

(ii) C is invariant under (8) and $\tilde{P}\tilde{x} = P\tilde{x}$ in C. □

We may observe that the matrix $\tilde{P}$ has the same eigenstructure with P except 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:

$$y(k+1) = \tilde{P}y(k), \qquad (17)$$

and the initial conditions are given by:

$$y^i(0) = y^{i+N}(0) = \frac{(N-1)\theta_i - \sum_{j=1}^{N}\theta_j}{2N-1}. \qquad (18)$$

III. The Network Design Problem

Let us turn to 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 quickly to the Nash equilibrium, without using too many links. However, 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 introduced.

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

$$\theta_i = \mu_i + w_i, \qquad (19)$$

where μ_i is the mean of player p_i's type and w_i a zero mean random vector. In compact form:

$$\Theta = \mu + w. \qquad (20)$$

We, further, assume that the vector of means μ and the covariance matrix $\Sigma = E[ww^T]$ are known to the planner.

The network design problem has two objectives. At first the matrix $\tilde{P}$ should be stable and thus the actions of the players converge to the Nash equilibrium, for any set of types. The second objective is the expected distance from the Nash equilibrium to decrease as fast as possible. A quadratic criterion, called the "designer's cost", quantifying the speed of convergence, is thus introduced:

$$J^d = E\left[\sum_{k=0}^{\infty}\sum_{i=1}^{N}(x^i(k) - x^{i,N})^2\right]. \qquad (21)$$

The criterion can be written as:

$$J^d = E\left[y^T(0)\left(\sum_{k=0}^{\infty}(\tilde{P}^k)^T Q\tilde{P}^k\right)y(0)\right], \qquad (22)$$

where:

$$Q = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}. \quad (23)$$

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

$$\tilde{P}^T X \tilde{P} - X + Q = 0, \quad (24)$$

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

$$f(\tilde{P}, \text{vec}(X)) = \left(I - \tilde{P}^T \otimes \tilde{P}^T\right) \text{vec}(X) - \text{vec}(Q) = 0 \quad (25)$$

Remark 2: Let us note that the solution of the Lyapunov equation depending on the graph Laplacian, appears also in the network coherence literature ex. [5].

The designer's cost can, thus, be written as:

$$J^d = \sum_{i=1}^{2N} \sum_{j=1}^{2N} X_{ij} E\left[y^i(0) y^j(0)\right] = \text{vec}(S)^T \text{vec}(X), \quad (26)$$

where $S = E[y(0)y(0)^T]$.

In order to compute S, let us observe that $\tilde{x}^N = D\Theta$ and $\tilde{x}(0) = \begin{bmatrix} I \\ I \end{bmatrix} \Theta$. Hence,

$$\begin{aligned} S &= E[y(0)y(0)^T] = \bar{D} E[\Theta\Theta^T] \bar{D}^T \\ &= \bar{D}(\mu\mu^T + \Sigma)\bar{D}^T, \end{aligned} \quad (27)$$

where $\bar{D} = \begin{bmatrix} I \\ I \end{bmatrix} - D$.

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 where to introduce additional links to the graph, such that the dynamic rule converges as fast as possible and the total number of edges does not exceed a maximum number E_{max}. The minimization problem for the designer is, thus, given by:

$$\begin{aligned} \underset{A}{\text{minimize}} \quad & \text{vec}(S)^T \text{vec}(X) \\ \text{subject to} \quad & \tilde{P} = \tilde{P}(A) \\ & \left(I - \tilde{P}^T \otimes \tilde{P}^T\right) \text{vec}(X) = \text{vec}(Q) \\ & A_{ij} \in \{0, 1\} \\ & \sum_{i=1}^{N} \sum_{j=1}^{i-1} A_{i,j} \le E_{Max} \\ & A_{ij} = 1 \text{ if } (i,j) \in E_0 \\ & \tilde{P}\text{: Stable} \\ & A\text{: Symmetric with zero diagonal} \end{aligned} \quad (28)$$

where $\tilde{P} = \tilde{P}(A)$ is given by $\tilde{P} = P(I + M)$, (9) and (15).

Remark 3: The parameters α and δ (describing the gradient decent step size (6) the consensus dynamics constant (7)) could be also considered as design parameters. Furthermore, different players could have different δ and α. However, in order to keep the analysis simple we consider only the network design problem with given δ and α.

The optimization problem (28) is a nonlinear mixed integer programming problem [23]. Thus, in general it could be difficult to solve. In the following section a simple algorithm which leads to suboptimal solutions is proposed.

IV. Algorithm for the Network Design Problem

A simple algorithm based on the approximate influence of new edges is proposed. In order to do so, the binary variables A_{ij} are temporarily assumed to have continuous values in $[0, 1]$ and the derivative of the cost with respect to A_{ij} is considered. We assume that A_{ij} with $j < i$ are the free variables and that $A_{ji} = A_{ij}$. For a given matrix A for which matrix $\tilde{P}$ is stable, the effect of adding a link (i, j) to the cost is approximated by the derivative $\partial J^d(\tilde{P}(A), X(\tilde{P}(A)))/\partial A_{i,j}$. The detailed computations for this derivative are first given.

If the matrix $\tilde{P}(A)$ is stable, implicit function theorem implies that there is a function $\text{vec}(X) = g(\tilde{P})$ such that $f(\tilde{P}, g(\tilde{P})) = 0$, locally and the partial derivative of J^d with respect to the elements of the adjacency matrix is given by:

$$\begin{aligned} \frac{\partial J^d(\tilde{P}(A), g(\tilde{P}(A)))}{\partial A_{i,j}} &= -\left(\text{vec}(S)\right)^T \left(\frac{\partial f}{\partial \text{vec}(X)}\right)^{-1} \cdot \\ & \cdot \sum_{i',j'=1}^{2N} \frac{\partial f}{\partial \tilde{P}_{i'j'}} \frac{\partial \tilde{P}_{i'j'}}{\partial A_{ij}}, \end{aligned} \quad (29)$$

for $1 \le j < i \le N$.

The first two terms satisfy:

$$\begin{aligned} & \left(\text{vec}(S)\right)^T \left(\frac{\partial f}{\partial \text{vec}(X)}\right)^{-1} = \\ & = \left(\left(I - \tilde{P} \otimes \tilde{P}\right)^{-1} \text{vec}(S)\right)^T = \text{vec}(Y)^T \end{aligned} \quad (30)$$

where Y satisfies the Lyapunov equation:

$$(\tilde{P}^T)^T Y (\tilde{P}^T) - Y = S. \quad (31)$$

For the other terms it holds,

$$\frac{\partial f}{\partial \tilde{P}_{i'j'}} = -\left(e_{j'} e_{i'}^T \otimes \tilde{P}^T + \tilde{P}^T \otimes e_{j'} e_{i'}^T\right) \text{vec}(X) \quad (32)$$

and:

$$\frac{\partial \tilde{P}}{\partial A_{ij}} = \frac{\partial P}{\partial A_{ij}} = \begin{bmatrix} 0 & 0 \\ 0 & \delta\left(e_i e_j^T + e_j e_i^T - e_j e_j^T - e_i e_i^T\right) \end{bmatrix}. \quad (33)$$

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

Equation (33) implies that:

$$\frac{\partial \tilde{P_{i'j'}}}{\partial A_{ij}} = \begin{cases} \delta & \text{if } i' = i + N, \ j' = j + N \\ \delta & \text{if } i' = j + N, \ j' = i + N \\ -\delta & \text{if } i' = i + N, \ j' = i + N \\ -\delta & \text{if } i' = j + N, \ j' = j + N \\ 0 & \text{otherwise} \end{cases} \quad (34)$$

Thus, only four terms contribute to the summation in (29).

Algorithm 1 uses the derivative approximation to add new edges to the graph. Particularly, at each time step, the approximate reduction that the cost would have by adding each one of the possible edges is computed. The edge which reduces more the designer's cost is added and the process is repeated.

Remark 4: The Use of the solution of the Lyapunov equation (31) simplifies the computations, due to the fact that we avoid the matrix inversion which is much more computationally consuming.

Algorithm 1

1: Get an initial graph $G = (V, E_0)$.
2: Compute the matrices $\tilde{P}$ and X.
3: For every pair of vertices i, j such that $(i, j) \notin E$ use (29) - (34) to compute $\partial J / \partial A_{ij}$.
4: Find a pair $(i^\star, j^\star)$ such that:

$$(i^\star, j^\star) \in \arg \min_{(i,j) \notin E} \partial J / \partial A_{ij}$$

5: Set $E \leftarrow E \cup (i^\star, j^\star)$
6: If $\sum_{i=1}^{N} \sum_{j=1}^{i-1} A_{i,j} < E_{Max}$ then go to Step 2. Else halt.

Remark 5: At each iteration of the algorithm the Lyapunov equation (31) and the partial derivatives $\frac{\partial f}{\partial \tilde{P}_{i'j'}}$ are computed only one time. Only the matrix given by (34) is different among the candidate edges.

Remark 6: At each iteration of the algorithm one new edge is introduced and the maximum number of edges is finite. Thus, Algorithm 1 always terminates. Furthermore, numerical results show that in all the executions of Algorithm 1 the cost reduces in any iteration.

V. Numerical Results

In this section three examples of the application of the algorithm are presented in order to illustrate the effectiveness of the proposed scheme. In these examples we assume that there is an initial communication graph, such that each player is connected with the previous and the next one, that is $A_{ij} = 1$ if $|i - j| = 1$ and there are $N = 40$ players.

Example 1: In the this example $\mu = 0$ and $W = I$. The algorithm will add 15 more edges. The designers cost after the network design is $J^d = 68.65$. Figure 1 illustrates the designed network. Figure 2 illustrates the cost of the designed graph compared with 100 graphs in which 15 edges were drawn uniformly at random.

Example 2: In this example the types of the players remain uncorrelated but the mean values depend on the position. We further assume that the variance of the types is small. Particularly, $W = I/20$ and $\mu_i = 2(i - N/2)/N$. The designed graph is illustrated in Figure 3. The designers cost in the designed graph, compared to 100 graphs in which the 15 edges were drawn uniformly at random is illustrated in Figure 4.

Example 3: In this example, the types of the players are correlated and the mean values are zero. Particularly, we assume that the $\Sigma = LL'$ where L is a row vector with

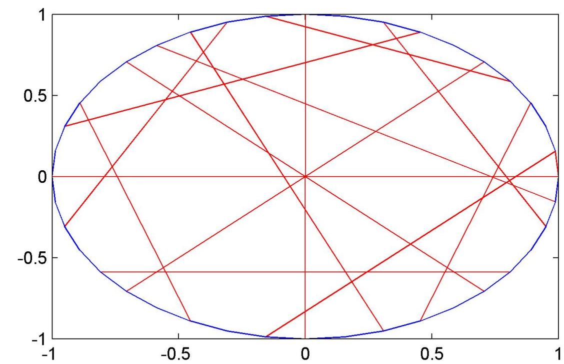

Fig. 1. The graph designed in Example 1

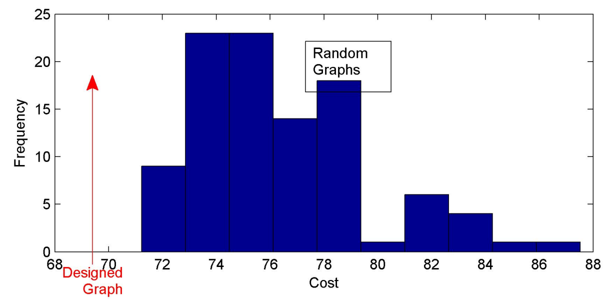

Fig. 2. The designer's cost for the graph designed in Example 1 compared with 100 graphs drown at random

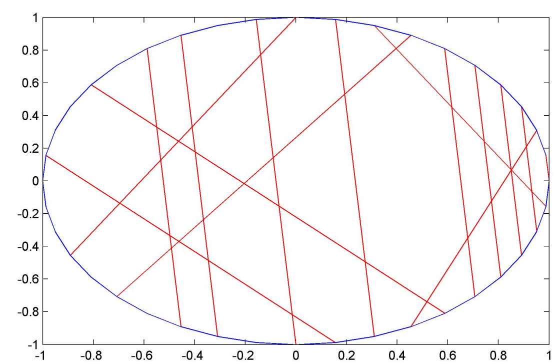

Fig. 3. The graph designed in Example 2

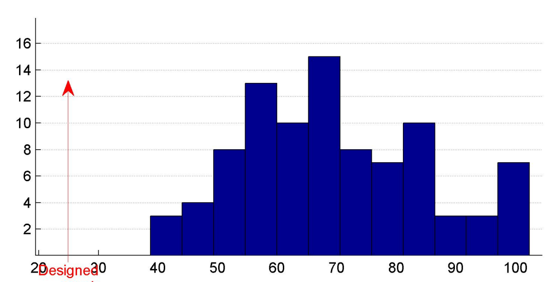

Fig. 4. The designer's cost for the graph designed in Example 2 compared with 100 graphs drown at random

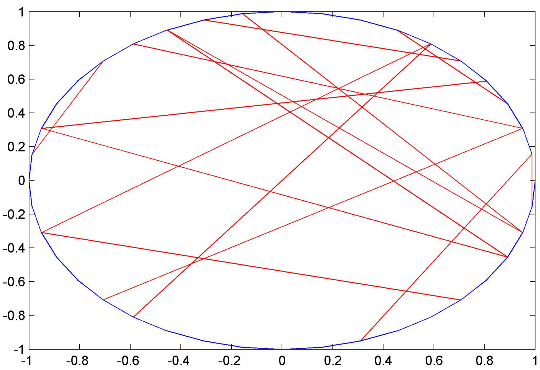

Fig. 5. The graph designed in Example 3

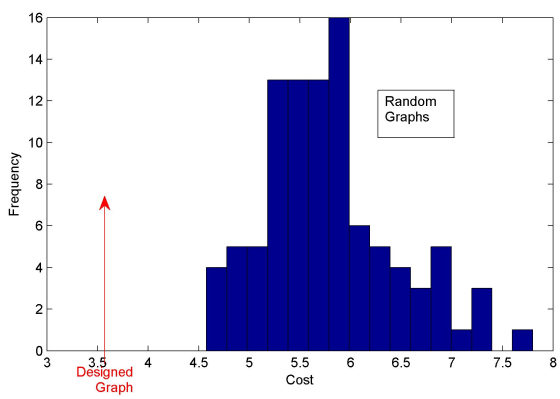

Fig. 6. The designer's cost for the graph designed in Example 3 compared with 100 graphs drown at random

values chosen uniformly at random from $[0, 1]$. The designed graph is illustrated in Figure 5 and the designers cost in the designed graph, compared to 100 graphs in which the 15 edges were drawn uniformly at random is illustrated in Figure 6.

Remark 7: The designed networks are performing at least in some cases much better than the networks chosen at random, despite the fact that random graphs and small world networks are known in the literature to have fast convergence in the consensus dynamics [24].

Remark 8: It seems that there is a relationship between the statistical description of the types and the optimal network design. The graph of Figure 1 is quite regular. In the graph of Figure 2 "opposite" players are connected and in the graph of Figure 3, many edges are concentrated in around some players.

VI. Conclusion

We considered a simple class of repeated quadratic games with mean field interactions. A set of simple gradient based dynamic rules was proposed. The overall dynamics depends on a variant of the Laplacian matrix. Using a stochastic model for the type of the players the quadratic criterion for the speed of convergence was stated its value was characterized using the Lyapunov equation. The minimization problem for this criterion was stated and a local search algorithm was proposed. Numerical results show that the designed graphs achieve a significantly faster convergence to the Nash equilibrium than a graph chosen at random.

There are several possible extensions of the current work. At first, the class of games under consideration could be generalized. For example it is possible that not all the players influence the overall system the same or that the different players do not care the same about the actions of the others. Furthermore, the decisions of the players may be multidimensional or constrained and the costs may be non-quadratic. Another interesting extension is to study games in which the players are influenced by the actions of their neighbours instead of the aggregate action all the players. Furthermore, the total number of players could be also a design variable. Finally, the optimization with respect to the parameters δ and α, as well as the use of alternative search techniques for the optimization problem (28) are of certain interest.

References

[1] M. Mesbahi and M. Egerstedt, *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010.

[2] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

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

[4] A. C. Charalampidis, "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*. IEEE, 2014, pp. 1452–1457.

[5] L. Xiao, S. Boyd, and S.-J. Kim, "Distributed average consensus with least-mean-square deviation," *Journal of Parallel and Distributed Computing*, vol. 67, no. 1, pp. 33–46, 2007.

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

[7] S. Boyd, P. Diaconis, and L. Xiao, "Fastest mixing markov chain on a graph," *SIAM review*, vol. 46, no. 4, pp. 667–689, 2004.

[8] A. Ghosh, S. Boyd, and A. Saberi, "Minimizing effective resistance of a graph," *SIAM review*, vol. 50, no. 1, pp. 37–66, 2008.

[9] M. Fardad, F. Lin, and M. R. Jovanovic, "Design of optimal sparse interconnection graphs for synchronization of oscillator networks," *Automatic Control, IEEE Transactions on*, vol. 59, no. 9, pp. 2457–2462, 2014.

[10] F. Lin, M. Fardad, and M. R. Jovanovic, "Design of optimal sparse feedback gains via the alternating direction method of multipliers," *Automatic Control, IEEE Transactions on*, vol. 58, no. 9, pp. 2426–2431, 2013.

[11] M. K. Jensen, "Aggregative games and best-reply potentials," *Economic theory*, vol. 43, no. 1, pp. 45–66, 2010.

[12] J.-M. Lasry and P.-L. Lions, "Mean field games," *Japanese Journal of Mathematics*, vol. 2, no. 1, pp. 229–260, 2007.

[13] M. Huang, R. P. Malhamé, and P. E. Caines, "Large population stochastic dynamic games: closed-loop mckean-vlasov systems and the nash certainty equivalence principle," *Communications in Information & Systems*, vol. 6, no. 3, pp. 221–252, 2006.

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

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

[16] Q. Zhu and T. Başar, "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*. IEEE, 2011, pp. 1–8.

[17] F. Bagagiolo and D. Bauso, "Mean-field games and dynamic demand management in power grids," *Dynamic Games and Applications*, vol. 4, no. 2, pp. 155–176, 2014.

[18] A. C. Kizilkale and P. E. Caines, "Mean field stochastic adaptive control," *Automatic Control, IEEE Transactions on*, vol. 58, no. 4, pp. 905–920, 2013.

[19] S. Grammatico, F. Parise, M. Colombino, and J. Lygeros, "Decentralized convergence to nash equilibria in constrained deterministic mean field control," *IEEE Transactions on Automatic Control*, vol. PP, no. 99, pp. 1–1, 2016.

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

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

[22] R. A. Horn and C. R. Johnson, "Topics in matrix analysis, 1991," *Cambridge University Presss, Cambridge*, vol. 37, p. 39.

[23] L. A. Wolsey and G. L. Nemhauser, *Integer and combinatorial optimization*. John Wiley & Sons, 2014.

[24] R. Olfati-Saber, "Ultrafast consensus in small-world networks," in *American Control Conference, 2005. Proceedings of the 2005*. IEEE, 2005, pp. 2371–2378.