Network topology design to influence the effects of manipulative behaviors in a social choice procedure

Athanasis-Rafail Lagos and George P. Papavassilopoulos

Abstract—A social choice procedure is modeled as a repeated Nash game between the social agents, who are communicating with each other through a social communication network modeled by an undirected graph. The agents’ criteria for this game are describing a trade-off between self-consistent and manipulative behaviors. Their best response strategies are resulting in two dynamics rules, one for the agents’ opinions and one for their actions. The stability properties of these dynamics are studied. In the case of instability, the stabilization of these dynamics through the design of the network topology is formulated as a constrained integer programming problem. The constraints have the form of a Bilinear Matrix Inequality (BMI), which is known to result in a nonconvex feasible set in the general case. To deal with this problem a Genetic Algorithm is designed. Finally, simulations are presented for several different initial topologies and conclusions are derived concerning both the functionality of the algorithm and the advisability of the problem formulation.

Index Terms—Opinion dynamics, Nash game, Network’s topology design, Constrained integer programming, Bilinear Matrix Inequality, Genetic Algorithm

I. INTRODUCTION

In recent years great progress has been made in the mathematical modeling and study of social phenomena. A topic of current interest is the study of the evolution of social agents’ opinions about a certain issue. This is commonly called opinion dynamics, for which a first model has been proposed since 1974 by M.De-Groot [1]. Since then, a lot of work has been done in this field [2]-[11], many interesting cases have been modeled and analysed, some of which are summarized in [12], [20], and new ideas continue to be proposed and studied up to now [13]-[16]. The burst of interest in this area is due to the emerging new questions that a decentralised but interconnected system modeling and analysis poses in many fields.

However, from a sociological point of view, as in the case of our references, an underlying assumption is usually important for the justification of the study of these dynamics. This assumption states that if the opinions of the members of a society are known along with their dynamics, we can predict their social behavior, or we may even be able to determine this behavior by affecting the mechanism of the opinion dynamics. Thus, it is assumed that the agents are consistent and they act in accordance with their beliefs.

On the other hand, from a game-theoretic perspective the action of an agent is determined by her criteria as a best response to the other agents’ actions. This indicates that an agent’s action also depends on her neighbors’ actions and not only on her opinion, which shall be considered as a shaping characteristic of her criteria. Moreover, this perspective adds the useful insight that the agents usually act antagonistically to their neighbors and they are not just affected by them [18]. So, in this work we develop a model introduced in [16] describing a social choice procedure, where the population structure is modeled by an undirected graph and the agents’ actions depend both on their opinions, which evolve dynamically in our model, and on their neighbors’ actions.

Specifically, each agent has an internal belief or opinion, which evolves in time following some opinion shaping mechanism deriving from the minimization of a suitable criterion modeling the tendency of the agent to conform to the public opinion. Each agent has also an expressed action in the social choice procedure. Each opinion is mapped to a desired action. However, the action of each agent isn’t identical to her desired action but it derives from the minimization of a suitable criterion modeling the tendency of the agent to manipulate, i.e. to deviate from her desired action, in order to pull the social outcome to her favor. The resulting game between the agents is considered to take place in discrete time steps, where both the opinion shaping and the action shaping criteria of each agent retain the same form. So we formulate a repeated game where we seek for the stagewise Nash strategy profiles. These strategy profiles are consisted of two dynamic rules, one for the opinions and one for the actions of the agents.

We study the stability properties of these dynamics and we deduce a sufficient condition that guarantees the convergence of the system to a bounded state. This condition implicates the manipulative tendencies of the agents and the graph structure with the stability of the system, stating that the acceptable manipulative behavior of an agent is relative to her social cohesion and her position in the graph. Simulations are also presented in order to examine how the opinion and action dynamics behave over several well known graph structures, such as random, small world and special structured graphs and in order to study the resilience of the system when our sufficient assumption is violated.

Subsequently, we consider the problem of changing the social network’s topology in order to influence the effects of manipulative behaviors. The network topology has been chosen as our designing parameter for two basic reasons. At first, the network topology design is an emerging problem in many scientific fields nowadays, such as security [19], UAVs navigation [21], cyberphysical systems [22], convergence of mean field games [20], etc. So, a general formulation and study of a network topology design for the stabilisation of...
a system of unstable dynamics of interconnected agents can be applied to many practical problems of current interest. Secondly, the network topology is a parameter that the social network’s administrator can affect and thus influence the agents’ behaviors in an indirect way, that does not raise social reactions. So, undesirable social phenomena can be avoided. It must be specified that in our work the term social network corresponds to its digital realisation and not to its abstract concept of a representation of human relationships, so an administrator exists and the topology can be affected. We would like to note here that in contrast to its practicality the existence of one or more administrators in such networks raises the more intriguing question of who will control the administrators, who have the power to affect the other agents’ manipulability and the final outcome.

Analytically, we study the case of an initial topology resulting in unstable dynamics and we want to find a new topology that results in stable dynamics and that is close to the initial topology with respect to the number and the exact position of their edges. This problem is formulated as an integer programming problem with a Lyapunov inequality for discrete time systems (known also as Schur’s inequality) as constraint. Each decision variable of this optimisation problem represent either the existence of an edge between two agents or one of the components of the Lyapunov matrix. The constraint is nonlinear with respect to our decision variables and is shown to be also nonconvex, by the time it can be written as a Bilinear Matrix Inequality, which is known to be a nonconvex problem in its general case [23]. To the best of our knowledge, a similar approach involving nonconvex integer optimisation for the graph topology design problem does not exist and since it may arise in many similar design problems when no simplifying assumptions are considered there is a merit in studying it. Since no efficient algorithm is known for this type of problems, we developed a genetic algorithm to deal with it. This algorithm searches only for the values of the integer decision variables representing the edges of the graph, while a Linear Matrix Inequality solver is used to check the feasibility of each new topology by solving the Lyapunov inequality with the topology variables fixed, which results to be linear with respect to the Lyapunov matrix. This procedure is repeated for many generations, where new topologies are produced by the application of the genetic operators. Finally, simulations of the results of the proposed algorithm are presented. The behavior of the algorithm is studied over several different initial topologies, where the agents’ parameters have been chosen properly to raise instabilities in the dynamics. Through the examination of these test cases we derive conclusions on the functionality of the proposed algorithm and the relevancy of our results with the expected ones from our empirical perception of social networks and social choice procedures.

II. PROBLEM FORMULATION

A. Notation

We consider an undirected graph $G = (V,E)$. By $n$ we denote the size of the graph or equivalently the number of its vertices. We denote by $N_i$ the neighborhood of the agent $i$, $N_i = \{ j : (i,j) \in E \}$ and by $d_i$ the degree of node $i$, that is the size of its neighborhood. Let $A$ be the adjacency matrix of the graph, it is a $n \times n$ symmetric matrix and its $(i,j)$ entry is 1 if nodes $i$ and $j$ are adjacent to each other and 0 otherwise. Let $D = diag\{d_i\}$ be the diagonal degree matrix, $C = diag\{c_i\}$ be the diagonal self-confidence matrix and $G = diag\{g_i\}$ be the diagonal manipulation matrix. The symbol $1$ stands for the $n \times 1$ vector with all his coordinates equal to 1, the symbol $\mathcal{I}$ stands for the identity $n \times n$ matrix and the symbols $e_i, i = 1...n$ stand for the standard basis of $\mathbb{R}^n$. For a set $A$ we denote $\mathcal{X}_A$ its indicator function, $\mathcal{X}_A(x) = 1$ if $x \in A$ and $\mathcal{X}_A(x) = 0$ elsewhere. The symbolism $\lceil \cdot \rceil$ denotes rounding to the next natural number and the symbolism $\lceil \cdot \rceil_{even}$ denotes rounding to the next even natural number. The symbol $A^T$ stands for the transpose of the matrix $A$.

B. Derivation of the Opinion Dynamics

At first, the mechanism that determines the evolution of the agents’ opinions is studied. It is considered that in this procedure the main factor that shapes the opinions in time is conformity. That is, the agents’ opinions tend to be affected and finally conform with their neighbors’ opinions. This model of opinions’ evolution is well known and studied for many years [1]–[3]. [13]. In fact, in [3, [13] the model has been enriched with the inclusion of stubborn agents, i.e. people who insist on their initial beliefs, but since their presence affects primarily the equilibrium of the opinion dynamics and not their stability properties, we shall not include such agents in our model. Following the notation of the aforementioned works, the opinion of the agent $i$ is denoted by $\theta_i(k)$ at each time step $k$, and its value is a real number. Every agent has an initial opinion $\theta_i(0)$ and she changes her opinion at each time step according to her stepwise opinion shaping criterion:

$$J\theta_i(k+1) = c_i(\theta_i(k+1) - \theta_i(k))^2 + \frac{d_i(\theta_i(k+1) - \sum_{j \in N_i} \theta_j(k))}{d_i}$$

(1)

where $c_i$ is a factor analogue to the self-confidence of the agent for her opinion and $d_i$, which is the number of her friends, is assumed to affect her opinion according to the rule that the more social the agent is the more her opinion is affected by his peers. In fact, these two measures have no meaning as absolute values but the fractions $c_i/d_i$ and $d_i/c_i$ indicate the obstinateness or the volatility of the agent. It shall be pointed here that in several cases of practical interest where the opinion shaping procedure is considered very slow, these parameters should be chosen: $c_i/d_i >> d_i/c_i$, which will slow down drastically the rate of convergence of the opinions to the social average.

By the minimization of the above criteria arise the following best-response dynamic rule according to which the opinions evolve in time:

$$\theta_i(k+1) = \frac{c_i}{d_i + c_i} \theta_i(k) + \frac{1}{d_i + c_i} \sum_{j \in N_i} \theta_j(k)$$

(2)
C. Derivation of the Action Dynamics

In contrast with the opinion shaping criteria, which are basically modelling a progressive conformity to the average beliefs, the criteria determining the action of each agent in every time step depict the tendency of the agents to manipulate the social outcome to their favor. That is, each agent may deviate her action from the one dictated by her beliefs in order to pull the social outcome towards her desired direction. In other words, as pointed in [15], it is a common phenomenon in politics that the people who disagree with what they perceive as the expected social outcome tend to overstate their opinions, leading their neighbors to misperceptions of the public opinion and conform to these false estimations, thus pulling the social outcome to their favor.

A basic aspect of this model is that of the agent’s perception of the social outcome. For that, it is assumed that the agents have local information of the other agents’ actions, that is they know only the actions of their neighbors. Moreover, it is assumed that this information pattern is Markovian, i.e. at each time step they know only the last actions of their neighbors forgetting the past. According to these two assumptions the estimated social outcome for each agent is:

\[
\tilde{u}_i(k) = \frac{\sum_{j \in N_i} u_j(k-1) + u_i(k)}{d_i + 1}
\]

where each agent’s action is denoted by \(u_i(k)\) at each time step \(k\) and its value is a real number.

**Remark II.1.** If the second assumption is relaxed by adding memory to the agents, so as to be able to predict the social outcome based on all the previous actions of their neighbors, the repeated Nash game examined here will be converted to a dynamic one, whose analysis is far more difficult yet interesting, and may be part of a future work.

Based on the aforementioned concepts the criteria that determine the actions of each agent are dependent on her current opinion and on her locally estimated social outcome, so they are defined in each stage as follows:

\[
J_{a_i}(\theta_i(k), u_1(k-1) \ldots u_i(k) \ldots u_n(k-1)) = (u_i(k) - \phi(\theta_i(k)))^2 + g_i(\tilde{u}_i(k) - \phi(\theta_i(k)))^2
\]

where \(\phi(.)\) is a common for all agents continuous function matching each opinion to a desired outcome. The constant parameters \(g_i\) are indicating how manipulative each agent is. That is because the first term of the cost function \((u_i(k) - \phi(\theta_i(k)))^2\) indicates the self-consistency of the agent, i.e. how close her action is to her opinion, while the second term \((\tilde{u}_i - \phi(\theta_i(k)))^2\) indicates the manipulative/opportunistic ends of the agent, i.e. how much she cares to affect the social outcome through her action so as to bring it close to her desirable outcome. So the parameters \(g_i\) determine the ratio between self-consistent and manipulative behaviour for each agent.

Assuming that the agents choose their actions rationally based on their criteria we seek for the Nash equilibrium solution in each stage of the game. In fact, we deal with a repeated game rather than a dynamic game since the structure of the criteria remain the same in each stage of the game and we don’t examine any aggregative or discounted criteria. So, each stage of the game consists itself a static game, where the players are assumed to choose their actions so as to achieve a Nash equilibrium. These actions derive from the solution of the following system of equations:

\[
\frac{\partial J_{a_i}}{\partial u_i} = 0
\]

The action dynamics of each agent in the context of the examined stagewise Nash strategy profile are derived form the system of equations [5]

\[
\frac{\partial J_{a_i}}{\partial u_i} = 0 \Rightarrow 2(u_i(k) - \phi(\theta_i(k))) + 2g_i \left( \sum_{j \in N_i} u_j(k-1) + u_i(k) \right) - \phi(\theta_i(k)) \left( \frac{1}{d_i + 1} \right).\]

Solving these equations with respect to \(u_i(k)\) and shifting the time index \(k\) to \(k+1\) for presentation coherence:

\[
u_i(k + 1) = \left(1 + \frac{d_i g_i}{g_i + (d_i + 1)^2}\right) \phi(\theta_i(k + 1)) - \frac{g_i}{g_i + (d_i + 1)^2} \sum_{j \in N_i} u_j(k)
\]

Defining now the diagonal matrices

\[
G_\theta = diag\{1 + \frac{d_i g_i}{g_i + (d_i + 1)^2}\}
\]

and

\[
G_u = diag\{\frac{g_i}{g_i + (d_i + 1)^2}\}
\]

we rewrite the equation in matrix form:

\[
u(k + 1) = G_\theta \Phi(\theta(k + 1)) - G_u Au(k)
\]

where \(u(k) = [u_1(k) \ldots u_n(k)]^T\) and \(\Phi(\theta(k + 1)) = [\phi(\theta_1(k + 1)) \ldots \phi(\theta_n(k + 1))]^T\).

III. Stability Analysis

A. Known results on opinion dynamics

For the evolution of the opinions of the agents we consider the following dynamics:

\[
\theta_i(k + 1) = \frac{c_i}{d_i + c_i} \theta_i(k) + \frac{1}{d_i + c_i} \sum_{j \in N_i} \theta_j(k)
\]

which can be summarized using matrix notation to the following expression:

\[
\theta(k + 1) = (D + C)^{-1}(A + C)\theta(k)
\]

where \(\theta(k) = [\theta_1(k) \ldots \theta_n(k)]^T\) and \((D + C)^{-1}(A + C)\) is a matrix with its rows summing to the unit (like a stochastic matrix) and thus a stable one, so \(\theta(k)\) converges to a limit distribution \(\theta^*\) which is actually a consensus on each connected subgraph. For some results on these the reader could study [11] and for a more general description one could study...
the criteria summarised in [17], which are satisfied in our case. So the following statements hold:

\[
\|\theta(k+1) - \theta(k)\| \to 0 \quad \text{(12)}
\]

\[
\|\theta(k) - \theta^*\| \to 0 \quad \text{(13)}
\]

**Remark III.1.** It is easy to observe that the parameters \(\frac{c_i}{\delta_1 + \cdots + \delta_i}\), describing the resistance of each agent to adopt new ideas, affect the rate of convergence. Specifically, as \(\frac{c_i}{\delta_1 + \cdots + \delta_i}\) increases the matrix \(A\) tends to approximate the identity matrix so its eigenvalues tend to be on the unit circle and the rate of convergence decreases.

**B. Stability analysis of the coupled opinion and action dynamics**

We continue our analysis by considering the augmented vector \(z(k) = [\theta_1(k), \ldots, \theta_n(k), u_1(k), \ldots, u_n(k)]^T\) and the resulting augmented system describing its dynamics. For simplicity reasons of the presentation we will use the notation \(\Phi(\theta(k+1))\) to denote the nonlinear function \(\Phi(\theta(k+1))\). So we obtain the following dynamics:

\[
z(k+1) = \left[\begin{array}{cc}
(D + C)^{-1}(A + C) & 0 \\
G_\theta \Phi \circ (D + C)^{-1}(A + C) & -G_\theta A
\end{array}\right] z(k)
\]  \quad \text{(14)}

1) A decoupling lemma:

**Lemma III.2.** If the matrix \(A_u = G_u A\) is asymptotically stable, ie \(|\lambda_1(A_u)| < 1, \forall i\) and the function \(\Phi\) is continuous in \(\mathbb{R}^n\) and locally Lipschitz in a neighborhood of \(\theta^*\) with a Lipschitz constant \(L_\Phi\), then the coupled dynamics (14) will be stable.

**Proof.** For the opinion dynamics, \(\theta(k+1) = A_\theta \theta(k)\), it is known to be stable as we have already discussed in a previous section. So, \(\exists K : \forall k > K \theta(k)\) belongs to a neighborhood of \(\theta^*\) where the mapping \(\Phi\) is Lipschitz. Thus \(\forall k > K\) the following holds for the holds:

\[
\|u(k+1) - u(k)\| =
\|G_\theta \Phi(\theta(k+1)) - A_u u(k) - G_\theta \Phi(\theta(k)) + A_u u(k-1)\|
\leq \|G_\theta \Phi(\theta(k+1)) - G_\theta \Phi(\theta(k))\| + \|A_u u(k) - A_u u(k-1)\|
\leq L_\Phi \|G_\theta\| \|\theta(k+1) - \theta(k)\| + \|A_u\| \|u(k) - u(k-1)\|
\]  \quad \text{(15)}

let now \(a = \|A_u\| < 1\) by our hypothesis that \(A_u\) is asymptotically stable, \(\delta_k = L_\Phi \|G_\theta\| \|\theta(k+1) - \theta(k)\| \to 0\) due to (12) and \(x_k = [u(k) - u(k-1)]\) thus we rewrite the previous inequality:

\[
x_{k+1} \leq ax_k + \delta_k
\]  \quad \text{(16)}

with \(a < 1\) and \(\frac{\delta_k}{1-a} \to 0\). So this inequality satisfy the conditions of lemma 3, p.45 of [30] and consequently it converges to zero, thus the sequence \(\{u(k+1) - u(k)\}\) is convergent to zero, so the sequence \(u(k)\) is Cauchy and thus convergent to an equilibrium point. So finally, the coupled dynamics are stable.

The usefulness of this lemma arises form the fact that the opinion dynamics are stable for every graph structure as the matrix \(A_u = (D + C)^{-1}(A + C)\) has the desired properties for every adjacency matrix \(A\) and its degree matrix \(D\). So this lemma enables us to focus on the stabilization of the action dynamics, through the graph design and the consequent tuning of the matrix \(A_u\), guaranteeing that the coupled dynamics will remain stable for every such design.

2) A sufficient condition: From the previous lemma and the substitution \(G_u A = G_u (D + I)(D + I)^{-1} A\) we can derive the following simple stability condition.

The matrix \((D + I)^{-1} A\) is a substochastic, ie its rows have sum equal or less than one, thus as shown in [13] its spectral radius \(\rho((D + I)^{-1} A) = \max\{\|\lambda_i((D + I)^{-1} A)\|, i = 1...N\}\) is less than one, so this matrix is stable.

So, a simple but restrictive sufficient condition for the stability of the whole system is the spectral radius of \(G_u (D + I)^{-1} A\), \(\rho(G_u (D + I)^{-1} A) = \max\{\|\lambda_i(G_u (D + I)^{-1} A)\|, i = 1...N\}\) to be less than one as well or equivalently

\[
\frac{(d_i + 1) g_i}{g_i + (d_i + 1)^2} \leq 1 \Rightarrow g_i \leq d_i + 2, \forall i
\]  \quad \text{(17)}

**Remark III.3.** We state this simple observation here because we can exploit its simplicity to use it as a heuristic for a stable topology design. That is, since this condition guarantees that the coupled dynamics converge on a graph with \(\min\{d_i\} \geq \max\{g_i\} - 2\) we know that a ring lattice of degree \(d_1 = \lceil\max\{g_i\} - 2\rceil\) is a topology that stabilizes these dynamics.

**C. Simulations on the model’s stability properties**

Several illustrative examples are presented here. On the one hand, the stability properties of the coupled dynamics proven in the previous section will be verified through the presentation of suitable examples on different graph structures. On the other hand, some observations will be pointed, that motivated us to formulate a general topology design problem.

In the following simulations we consider a network of \(n = 20\) agents participating in a repeated social choice procedure for \(T = 100\) times. The parameters \(c_i\) indicating the obstinateness of the agents are randomly chosen from the interval \([10, 100]\). The parameters \(g_i\) indicating the manipulative tendencies of the agents are randomly chosen from the interval \([0, 15]\). Their initial opinions are randomly chosen from the interval \([0, 10]\) interval. Their initial actions are the desired ones according to their initial opinions \(u_i(0) = \phi(\theta_i(0))\), where the function \(\Phi\) is considered to be \(\Phi(\theta) = 10 \tanh(\theta/10)\), which is both continuous and locally Lipschitz.

Firstly, we present the convergent opinion and action dynamics Fig.2 on a realization of a random graph [24] with edge probability \(p = 0.4\) Fig.8 which in the presented case has \(|E| = 81\) edges and the spectral radius of the resulting matrix \(A_u\) equals \(\lambda_{max}\{A_u\} = 0.7774\), so it has the necessary stability properties.

Subsequently, a case of nonconvergent dynamics will be presented. The dynamics Fig.11 result from a realization of a random graph with edge probability \(p = 0.3\) Fig.8 which has \(|E| = 54\) edges and \(\lambda_{max}\{A_u\} = 1.0418\).
We consider now the problem of choosing a proper graph structure, which would result in stable dynamics and be as close as possible to the aforementioned unstable one with respect to the edge number $|E|$ in this case. We make several experiments beginning from an $L^*$-lattice which satisfies our sufficient condition ($L^* > g_{\text{max}} - 2$), $L^* = 14$ in this example, and relaxing it by considering lattices of smaller node degree until the dynamics become unstable, as shown in table I.

In Fig. 5 is depicted an example of a lattice graph of degree 8.

The most interesting observation we made from our experiments was that while the 6 degree lattice results in unstable dynamics if we rewire some of its edges and thus create a small world graph, as introduced by J. Watts and S. Strogatz (1998), the dynamics become stable. This indicates that a well structured topology -whose properties can be studied analytically- is not necessarily a best choice for our problem, on the opposite the introduction of some random rewirings results in better structures. This was a motivation for the following general formulation of the topology design problem and the avoidance of restrictions on several special classes of topologies.

The small world graph is presented in Fig. 6 and the stable dynamics resulting from this structure are presented in Fig. 7.

### IV. NETWORK TOPOLOGY DESIGN FOR THE STABILIZATION OF THE ACTION DYNAMICS

#### A. Notation and Problem statement

For the graph topology design we will consider the vector $x \in \{0, 1\}^{n(n-1)/2}$, which denotes the occurrence of a change in the existing graph structure and constitutes our decision variables.
Let \( \{P^k, k = 1...n(n+1)/2\} \) be a basis of the symmetric \( n \times n \) matrices. Specifically, consider the matrices \( P^k \) with \( P^k_{ij} = P^k_{ji} = 1 \) if \( i = \max m \geq 0 \{ \sum_{l=1}^{m-1} (n - (l - 1)) \leq k \} \) and \( j = i + k - \sum_{l=1}^{i-1} (n - (l - 1)) \) and \( P^k = 0 \) elsewhere. The diagonal matrices of this basis, ie \( \{P^k : k \in K_d = \{ \sum_{l=1}^{i-1} (n - (l - 1)) + 1, i = 1...n \} \} \), we will denote them \( P_d^i \) since each \( k \in K_d \) corresponds to an \( i \in \{1...n\} \).

Now with this notation we can write \( A_0 = \sum_{k \notin K_d} x_0(k)P^k \), where the vector \( x_0 \) stands for the coordinates of \( A_0 \) with respect to the aforementioned basis \( \{P^k, k = 1...n(n+1)/2\} \) except its diagonal elements whose coordinates are all zero. It is profound that \( x_0(k) \in \{0,1\} \).

The topology design procedure consists of the addition of some new edges and the removal of some existing edges. So we define the following sign function \( S_{x_0}(k) = 1 \) if \( x_0(k) = 0 \) and \( S_{x_0}(k) = -1 \) if \( x_0(k) = 1 \), which multiplied with the vector of changes \( x \) indicates which changes correspond to an addition of an edge and which to a removal.

So the adjacency matrix of the graph depends linearly on the changes’ vector \( x \):

\[
A(x) = A_0 + \sum_{k=1}^{n(n-1)/2} x(k)P^kS_{x_0}(k)
\]

and the degree matrix changes accordingly:

\[
D(x) = \sum_{i=1}^{n} e_i(A(x)1)^TP_d^i
\]

which is also a linear function of \( x \).

Subsequently, we define the matrix functions:

\[
G_u(x) = G(G + (D(x) + I)^2)^{-1}
\]

and

\[
A_u(x) = G_u(x)A(x)
\]

which are nonlinear with respect to the decision variables \( x \).

Applying the Lyapunov stability criterion on the matrix \( A_u(x) = G_u(x)A(x) \) we obtain the following matrix inequality for \( P > 0 \) and \( x \):

\[
A(x)G_u(x)PG_u(x)A(x) - P \leq Q
\]

The matrix \( Q \) is a negative definite matrix, for example \( Q = -qI \), where \( q \) is a design parameter affecting the stability properties of the system as well as the size of the feasible region of the optimisation problem. In the simulations presented in the next section this parameter is considered very small (\( q = 10^{-2} \)).

Let the \( F_1 = \{ x : \exists P > 0 : A(x)G_u(x)PG_u(x)A(x) - P \leq -qI \} \). This set contains all the feasible designs, ie the vectors \( x \) for which the induced graph described by the adjacency matrix \( A(x) \) has the desired stability properties.

In order to choose an element of the aforementioned feasible set as a best design, we consider the criterion of the minimum change from the initial graph structure, which is a natural criterion as especially on graphs representing social interactions it may be very difficult to persuade someone to abandon a friend or make a new one. So we consider the minimization of \( \|x\|_1 \), which is equivalent to the minimization of the linear objective \( 1^T x \). The resulting problem is:

\[
\begin{align*}
\text{minimize} & \{1^T x \} \\
x,P & \in \{0,1\}^{n(n-1)/2} \\
\exists P > 0 : A(x)G_u(x)PG_u(x)A(x) - P \leq -qI
\end{align*}
\]

In this problem formulation several linear constraints may be added so as to describe restrictions on the design parameters due to special structural characteristics of the network, which may be important to be preserved or due to special characteristics of several nodes, whose neighborhood cannot be affected. However, these extra constraints do not increase the difficulty of the problem as it lies on the constraint (25).

This constraint \( \exists P > 0 : A(x)G_u(x)PG_u(x)A(x) - P \leq -qI \) is nonlinear in the decision variables \( x \). Furthermore, considering the change of variables \( Z = G_u(x)PG_u(x) \), we obtain the equivalent constraint:

\[
\exists Z > 0 : A(x)ZA(x) - G_u^{-1}(x)ZG_u^{-1}(x) \leq -qI
\]

This last constraint (26) is polynomial in the decision variables \( x \) and since \( x(k) \in \{0,1\} \) we have a proper change of variables \( (y(m) = x(k)x(l) \forall m,l \text{ and } w(n) = y(m)z_j \forall m,i,j) \) which transforms (26) to a Bilinear Matrix Inequality (BMI). The feasibility of a BMI is known to be a nonconvex problem in its general case (25), so the same holds for our initial problem (23,25).

### B. A genetic algorithm for the topology design problem

We now present the genetic algorithm developed to obtain a feasible solution for the nonconvex integer programming problem (23,25). In order to avoid the explosion of the dimensionality which results to a very slow convergence for the genetic algorithm, we use the genetic algorithm to search only in the space of the decision variables \( x \) rather than in the whole space (\( x,P \)). However, this search may lead to several topologies which will not satisfy the constraint (25).

To deal with this we observe that the constraint (25) is linear with respect to the matrix variable \( P \), so its feasibility can be efficiently checked with the use of a projective method based algorithm for Linear Matrix Inequalities (LMIs). So, for each new topology produced by the genetic operations we check its feasibility with an LMI solver and we drop it out of the next
generation if it is infeasible. The basic characteristics of this algorithm are enlisted below:

Chromosomes: Each chromosome of the genetic algorithm is a 0-1 sequence of length \( \frac{n(n-1)}{2} \) representing the vector \( x_0 + x \cdot S_{x_0} \) for some changes’ vector \( x \). The vectors \( x_0 \), \( x \) and \( S_{x_0} \) are defined in the previous section, while the symbol ‘’ denotes elementwise multiplication of the two vectors.

Initial population: As initial population for the genetic algorithm we consider a specific number of feasible random perturbations of the initial topology \( x_0 \). That is we produce a number of chromosomes of the form \( x_0 + x \cdot S_{x_0} \), which satisfy the constraint (25), where \( x \) are randomly derived 0-1 sequences. The feasibility check, which is described below, is applied on these chromosomes in order to verify which of them are satisfying the constraint (25) and reject the others from the initial population.

Fitness function: The fitness function of the genetic algorithm coincides with the objective function of the problem (23)-(25), so it has the following form

\[
\text{fitness}(\text{chromosome}) = \| \text{chromosome} - x_0 \parallel_1 = \| x_0 + x \cdot S_{x_0} - x_0 \parallel_1 = \| x \parallel_1.
\]

Selection: For the choice of a portion of the population for the breeding of the next generation we use a simple truncation selection criterion. We choose the 50% fittest part of the population in the case the size population exceeds a specific lower bound. The reason for this is to avoid the diminishment of the population in the case that many new offsprings are rejected because they do not satisfy the constraints. The next generation of the population is initialised by the selected part of the previous population.

The truncation selection has the drawback that it may lead to elitism, and thus the algorithm may converge to a local minimum of the optimisation problem, but the convergence speed of the algorithm if we use another selection procedure, such as fitness proportionate selection, is much more slow, so we have kept this simple method for our experimental simulations. Moreover, by choosing our initial conditions relatively close to the optimum - we initialise the algorithm with perturbations of the initial infeasible topology which are adequately close to it and feasible - we enhance our chances to find the global optimum even with this selection procedure.

Of course, in cases of practical interest where great accuracy is needed and with sufficient computing power available, we can easily replace this subroutine by one applying fitness proportionate selection.

Crossover: The crossover operator considered here chooses randomly two parents form the selected portion of the population and chooses also randomly a crossover point between \( 1, \ldots, \frac{n(n-1)}{2} \) and produces two offsprings form the two possible combinations of the parent chromosomes around this point.

Mutation: The mutation operator applied to an offspring changes each of its bits with probability \( p_m = \frac{2}{n(n-1)} \), resulting on an average change of one bit per chromosome.

Feasibility check: After the production of the new offsprings with the application of the genetic operators, each offspring is checked for the feasibility of the constraint (25).

For this we use an LMI solver, which uses a projective method algorithm, to examine the existence of a matrix \( P > 0 \) which satisfies the LMI (25), where the matrices \( A(x) \) and \( G_u(x) \) have the fixed values corresponding to the vector \( x \) of the offspring’s chromosome \( x_0 + x \cdot S_{x_0} \). If this LMI is found feasible the new chromosome is added to the next generation, else it is rejected.

Termination criterion: The genetic algorithm terminates after a specified number of generations \( N \). The fittest chromosome of the last generation is returned as solution for our topology design problem.

![Genetic algorithm flowchart](image)

**Figure 8: Genetic algorithm flowchart.**

C. Simulations of the results of the genetic algorithm

In the following simulations we consider a network of \( n = 20 \) agents participating in a repeated social choice procedure for \( T = 300 \) times. The parameters \( c_i \) are chosen randomly from the interval \([10, 100]\). The parameters \( g_i \) are randomly chosen form the interval \([0, 10]\). The function \( \Phi \) which maps the opinions to the desired actions is considered to be \( \Phi(\theta) = 10tanh(\theta/10) \), which is both continuous and locally Lipschitz.

The initial opinions \( \theta_i(0) \) are randomly chosen from the interval \([0, 10]\) and the initial actions are the ones corresponding to these opinions \( u_i(0) = \phi(\theta_i(0)) \). All the aforementioned parameters remain the same in both simulations.

The initial graph topology is the realisation of a random graph with edge probability \( p = 0.2 \) shown in figure Fig[7].

The resulting opinion and action dynamics are shown in figure Fig[10] where we can see that the action dynamics are unstable.

Applying the genetic algorithm presented in the previous section to the initial graph topology we obtain the graph topology presented in figure Fig[11] which differs from the initial one only on three edges.

The resulting opinion and action dynamics are shown in figure Fig[12] where we can see that the action dynamics are stable over the designed graph topology.
Figure 9: The initial graph topology, derived as a random graph with edge probability $p = 0.2$.

Figure 10: Unstable action dynamics on the initial graph topology.

Figure 11: The designed graph topology by the genetic algorithm.

Figure 12: Stable action dynamics on the designed graph topology.

Comments: As we can easily observe from the simulations above the graph topology that derived from the genetic algorithm results in stable action dynamics, so it is a feasible point of our optimisation problem. Moreover, with respect to its optimality, we have already pointed that the designed topology differs from the initial one on just 3 edges (specifically 1 edge has been removed and 2 new edges have been added), meaning that $\|x\|_1 = 3$ which is very small. It may be a suboptimal solution (even if it seems unlikely to find a topology even closer to the initial one resulting in stable dynamics), but in most cases it might be an acceptable design. Finally, compared with the heuristic approaches developed in section 4.3 it outperforms them by far, since the best we had achieved there was a difference of 8 on the amount, not on the exact location, of the existing edges of the two topologies, while now we achieved a difference of 3 on the exact location of the edges of the two topologies.

D. Simulations over Special Structured Initial Topologies

In the following simulations we consider a network of $n = 20$ agents and we check just the structure of the resulting topologies after the implementation of the genetic algorithm on several special structured initial topologies. The parameters $g_i$ indicating the manipulative tendencies of the agents are chosen accordingly in each case in order to make the initial topology resulting in unstable dynamics.

Figure 13: Initial ring topology

Figure 14: Designed unconnected topology from a ring (optimal)

1) Ring: For the ring topology Fig[13] the parameters $g_i$ indicating the manipulative tendencies of the agents are chosen randomly from the interval $[0, 10]$. The ring is a very sparse structure for a connected one. It has only 20 edges while 19 are needed in order to be connected. Furthermore, its stability properties are not very enhanced - even small manipulative parameters result in instabilities. So a connected stable topology differs a lot from the initial one. That’s why our algorithm returns an unconnected topology as the optimal solution Fig[14]. This topology has 5 edges and differs from the initial one on 15 edges. The unconnected designed topology
is stable, since the isolation of the agents pauses their social interactions and results in the preservation of their initial opinions and actions, which are stable by the time they are not increasing.

Even if it is mathematically acceptable, the isolation of the agents is a bit unrealistic and in many cases undesirable design. Subsequently, we add a simple linear constraint in the topology design problem demanding the designed topology to have at least 19 edges - the minimum edges needed to be connected. Interestingly, we obtain a connected topology Fig[15] which has 39 edges and differs from the initial one on 20 edges.

2) 4-lattice: For the 4-lattice Fig[16] the parameters \( g_i \) indicating the manipulative tendencies of the agents are chosen randomly from the interval \([0, 20]\). This increase in the manipulation parameters shows from the beginning that the lattices have enhanced stability properties in comparison with the ring, as it is expected since they are more dense and well connected topologies. The 4-lattice depicted in Fig[16] has 40 edges. Our design results in the topology Fig[17] which has 43 edges and differs from the initial one on 5 edges.

3) Star: For the star topology Fig[18] the parameters \( g_i \) indicating the manipulative tendencies of the agents are chosen randomly from the interval \([0, 20]\), except the one of the central node which is chosen much larger (here \( g(1) = 70 \)). That is because the star structure is a very robust one with respect to its stability properties, since the central node is very difficult to manipulate and to be manipulated as she has the most neighbors she could have. So the parameters should be chosen large enough in order to arise instabilities on this initial topology. Moreover, the star graph has the least possible edges needed to be connected (19 edges), so it seems to be a very robust design for the number of its edges. That is the reason why our algorithm converges to an unconnected topology Fig[19] which is closer to the star topology than any connected stable one. It has only 3 edges and it differs from the initial topology on 18 edges. It shall be noted here that, as in the case of the ring, the unconnected designed topology is stable.

Subsequently, as in the case of the ring topology, we add an extra constraint for the topology to enhance a connected
design and we derive the final topology depicted in Fig. 21. It has 47 edges and it differs from the initial one on 42 edges.

Figure 21: Initial random graph topology for identical agents

Figure 22: Designed topology for identical agents

4) Identical Agents: In the final simulation the agents are considered to be identical with respect to their subjective characteristics. So, the parameters $g_i$ indicating their manipulative tendencies are chosen to be the same and equal to 10. The parameters $c(i)$ affecting the convergence of the opinions are also chosen to be the same and equal to 100 (this affects only the rate of convergence of the dynamics not their stability). As initial topology a random graph with edge probability $p_0 = 0.3$ and 34 edges is considered Fig. 21. Our design results in the topology of Fig. 22 which has 52 edges and differs from the initial on 20 edges.

We note here that since the agents are identical and the stability of the action dynamics does not depend on their initial conditions any permutation of the agents on the designed structure is also stable. The resulting depicted position of them in Fig. 22 is the closest design to the initial topology Fig. 21 according to our agent based design. It is interesting to observe that even if the agents are identical the designed topology is not symmetric. Surely, there exist symmetric topologies that are stable, for example see remark 11.3 but they are not close to the initial topology with respect to our metric. Moreover, we should point that in this case of identical agents, if permutations are permitted i.e. we do not care who is who, our agent based criterion is not a proper metric for the distance between graph topologies and there may exist another topology more similar to the initial unstable one with respect to some proper graph similarity metric, which our algorithm cannot find.

Comments: From the study of these special structures we deduce several interesting conclusions. At first, the isolation of some agents form the rest network is sometimes optimal as it effectively stops their manipulative activity. The fact is that such a design will not be acceptable by these agents nor it may be socially acceptable. So we add more constraints which do not affect the difficulty of the problem in order to avoid a design which may be optimal but inapplicable in social networks. Fortunately, since the increase of the agent’s friends leads to the decrease of her ability to manipulate each one of them, as we deduce from the sufficient condition 17, it is guaranteed that there exists another topology with more edges than the initial, which satisfies the stability constraints and it is in fact a local minimum of our optimisation problem. We can also design this topology to be connected by adding more edges and not affecting its stability.

Secondly, we observe the stability properties of the dynamics induced by some very simple and useful topologies as the ring and the star. It worthy to be mentioned that these two topologies, despite having almost the same number of edges, the induced dynamics have totally different stability properties, since in the ring just a small amount of manipulative behavior can lead to instabilities, while the star topology is extremely robust with respect to the manipulative behavior needed to raise instabilities.

Finally, these special cases are useful as test cases for the functionality our algorithm and indeed the last one with the identical agents indicates a vulnerability of our agent based formulation and the respective graph distance in the case where the agents are identical and their names do not matter. Nevertheless, in most existing social networks each node stands for a specific agent who chooses his neighborhood and thus this anonymity and position insignificancy case does not occur.

V. CONTRIBUTIONS AND FUTURE EXTENSIONS

In this work we considered a social choice procedure as a repeated Nash game between social agents communicating over a social network. A contribution of this work is an enrichment of the model for social choice procedures proposed in 16 by considering dynamically changing opinions and thus resulting in second order dynamics. However, our basic novelty is a new approach for the stabilization of these dynamics through the graph topology design, which results in an integer programming problem with nonconvex constraints. Finally, we designed a proper genetic algorithm for this problem.

However, there exist several possible alternatives to deal with this problem, such as simulated annealing or particle swarm optimization algorithms, which may be part of our future work. Moreover, the study of some special cases of the topology design problem, restricted on some subset of admissible topologies where the detection of an exact solution may become possible, is an interesting future direction of our research. Except for that direction, it would be interesting to verify or reject our model’s features and assumptions by the use of real data from some social experiments or polls and evolve the model so as to describe better the specific social phenomena.
