Solving Influence Problems on the DeGroot Model with a Probabilistic Model Checking Tool

Sotirios Gyftopoulos Pavlos S. Efraimidis
Dept. of Electrical and Computer Engineering
Democritus University of Thrace
{sgyftopo,pefraimi}@ee.duth.gr

Panagiotis Katsaros
Departement of Informatics
Aristotle University of Thessaloniki
katsaros@csd.auth.gr

ABSTRACT

DeGroot learning is a model of opinion diffusion and formation in a social network of individuals. We examine the behavior of the DeGroot learning model when external strategic players that aim to bias the final consensus of the social network, are introduced to the model. More precisely, we consider the case of a single decision maker and the case of two competing external players, and a fixed number of possible influence actions on each individual. When studying the influence problems, we focus on the stochastic processes underlying the solution of DeGroot problems. In case of one decision maker, the analysis of the DeGroot model leads to the formation of a Markov Decision Process (MDP) and in the case of two external competing players the model is reduced to a Stochastic Game (SG). Since such models are heavily used in probabilistic model checking we apply tools of the field to solve them. Preliminary experimental results confirm the viability of our approach, which relies on the common mathematical foundations of the DeGroot problems and probabilistic model checking.

CCS Concepts

•Human-centered computing → Social networks; •Mathematics of computing → Markov processes; •Computing methodologies → Stochastic games; Model verification and validation; •Theory of computation → Algorithmic game theory;

Keywords

Social networks; Opinion dynamics; DeGroot model; Stochastic games; Probabilistic model checking

1. INTRODUCTION

Opinion dynamics is the research field of investigating the mechanisms of opinion diffusion among individuals, understanding their basic principles and proposing models that incorporate the basic rules of opinion formation. Many models have been introduced that imitate the underlying principles of opinion diffusion [7, 9, 10] and are being examined [12] in order to determine elaborate characteristics of the opinion formation process and ways to bias the diffusion process.

The DeGroot model was introduced by Morris H. DeGroot in 1974 [7] and suggests a simple mechanism of opinion propagation: every individual forms his opinion by averaging his own opinion with those of his friends. The process is repeated until all opinions converge. Even though the mechanism is simple, it models sufficiently opinion diffusion and incorporates elaborate characteristics of the process. The individuals form a social network with friendships and the location of each individual in this network is of vital importance for the prevalence of his opinion in the consensus. The averaging of opinion highlights the importance of centrality of individuals in the network and provides solid ground for experimentation on the opinion formation process.

Model checking is the scientific field of examining systems that are represented in a mathematical language and verifying their properties. Probabilistic model checking focuses on the analysis of systems that have probabilistic behavior and uses algorithms to examine the validity of probabilistic properties [12, 15]. Several software tools have been developed to analyze probabilistic models [16, 5, 3, 6] and verify their properties. We have chosen for our experiments the PRISM model checker [16] and the PRISM-games extension [3] due to their efficiency and wide range of functionalities.

In this work, we experiment on variations of the opinion formation mechanism. We extend the DeGroot model with the introduction of external influence to the opinion formation process. Our aim is to extract the strategies that external players could develop in order to interfere with the process and their bias in the final consensus. The common mathematical foundations of DeGroot model and the probabilistic model checking offer the opportunity to examine the behavior of the model under a new prism and, hence, we use software from the field.

The paper is organized as follows: Section 2 presents concise descriptions of the DeGroot model, stochastic games and the model checking problem, Section 3 describes the implementation of the models and demonstrates the experimental results, and Section 4 summarizes the findings and restrictions that were highlighted by our experiments.

PCI '16, November 10-12, 2016, Patras, Greece
© 2016 ACM. ISBN 978-1-4503-4789-1/16/11…$15.00
DOI: http://dx.doi.org/10.1145/3003733.3003780

2. PRELIMINARIES

2.1 The DeGroot Model

The core idea of the DeGroot model is that individuals tend to adopt the opinions of their friends. Each individual has an initial opinion and a set of friends that he shares his opinion with. The range of trust in his friends' beliefs may vary. After the opinions are exchanged, an update process is initiated by each individual that averages his opinion and the opinions of his friends. DeGroot proved that, under certain conditions, the opinions of all members converge after adequate number of iterations [7].

An effective representation of a social group is a social graph. Every individual of the group is represented as a node in the graph and his friendships to other members are depicted as adjacent edges to the corresponding nodes. Figure 1 is a social graph consisting of six members. Each node has the depicted adjacencies according to its outgoing edges. The weights attached to the edges represent the range of trust of a member to the opinions of his neighbors and are used in the opinion update process. Their sum for each node equals to one. Hence, the updated opinion of a node is the weighted average of the opinions of its neighbors. We should notice that every node is neighbor with itself, i.e. he takes his own opinion into account in the averaging process.

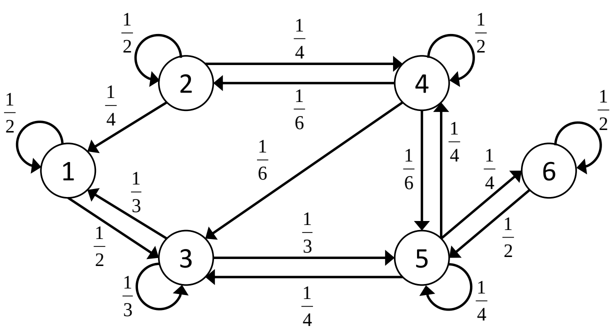

Figure 1: A social network with six members.

In terms of linear algebra, the graph of Figure 1 is represented as a 6×6 adjacency matrix P. Each item p_{ij} of P is the averaging factor of node i to node j. If F_0 is the vector of initial opinions of the nodes, then the averaging process of DeGroot model is described in Equation 1.

$$F_n = PF_{n-1} = P^n F_0 \tag{1}$$

F_n and F_{n-1} represent the opinion vectors after n and $n-1$ iterations of the averaging process respectively.

DeGroot remarked that P can be interpreted as the one-step transition matrix of a Markov chain and applied the standard limit theorems of Markov chains theory to determine its convergence. In Theorem 2 he states: "*if all the recurrent states of the Markov chain communicate with each other and are aperiodic, then a consensus is reached*" [7] (recurrent states have a finite hitting time with probability 1).

In [14], Jackson uses the definitions of *strongly connected* and *closed groups* to formulate the conditions under which a consensus is reached in a *social network*. A *strongly connected group of agents* is a group of nodes S that can reach each other via a set of consecutive directed edges and a *closed group of nodes* is a set C such that there is no directed link from an agent in C to an agent outside C. Jackson states in Proposition 8.3.1 that *"Under P, any strongly connected and closed group of individuals reaches a consensus for every initial vector of beliefs if and only if it is aperiodic"*. Golub and Jackson used the definitions of simple cycles in [11] to examine the aperiodicity of matrix P. A *simple cycle* is a set of consecutive directed edges that start from and lead to the same node which is the only node that appears twice in the traversal of the edges. Golub and Jackson state in Definition 2 that *"the matrix P is aperiodic if the greatest common divisor of the lengths of its simple cycles is 1"*.

2.2 Stochastic Games

Stochastic games were introduced by Shapley [19] and form a mathematical model which incorporates *actions* and *payments* for two *strategic players* interacting on a finite set of N *game states*. At each state k, the two players choose their actions i and j from the sets of available actions M_k and N_k and the next state l of the game is determined probabilistically depending on their actions. The probability of reaching state l from state k is p_{ij}^{kl}. Shapley also introduced a stopping factor at each stage, i.e. a probability that the game stops after the players make their moves, defined as $s_{ij}^k > 0 \ \forall i, j, k$. Since s_{ij}^k is positive for every i, j and k, it is certain that the game will end after a finite number of steps because the probability of an infinite game is $p^\infty = \prod_{t=1}^{\infty}(1 - s_{ij}^t) \to 0$ where s_{ij}^t is the stopping factor at step t when players choose their *ith* and *jth* alternatives.

The mathematical model of stochastic games includes *payoffs* for the players in the form of *payments* of the second player to the first at each state k. The payment a_{ij}^k is determined based on the *ith* and *jth* actions of the players at each state k. Both players have incentives to choose their actions properly in order to maximise their long-term payoff in the game developing, thus, *strategies* that indicate their moves in every state of the game. Shapley defined the *value* `val` of a stochastic game Γ as the minimum expected payoff that the first player expects to receive regardless of the second player's strategy. He proved the existence of the value `val` of a stochastic game and of optimal strategies for the players that achieve this value. Moreover, the optimal strategies are stationary, i.e. the formation of the strategies depend only on the current state of the game and any history of previous states is of no use.

Stochastic games can be perceived as a generalization of *Markov Decision Processes (MDP)*. MDPs represent mathematically the behaviour of one strategic entity (the *decision maker*) in a specific environment. The introduction of a second strategic entity in the environment transforms the mathematical model to a stochastic game. Several algorithms have been proposed that solve MDPs [1, 13]. Markov chains can be perceived as stochastic games where both players have no alternatives in each state of the game.

2.3 Model Checking

Probabilistic model checking is an automated formal verification technique for analyzing systems or processes with probabilistic behavior. Model-checking tools like PRISM [16] combine graph-theoretic algorithms for reachability analysis with iterative numerical solvers. They can evaluate properties of the form $\mathcal{P}_{\bowtie q}(\psi)$ with $\bowtie \in \{<, \leq, \geq, >\}$, $q \in \mathbb{Q} \cap [0, 1]$ ($\mathbb{Q}$ is the set of rational numbers) or $\mathcal{P}_{=?}(\psi)$, which compute the probability that a path satisfies ψ. The path formula ψ is interpreted over the paths of the probabilistic model, which can be a Discrete Time Markov Chain (DTMC), a Continuous Time Markov Chain (CTMC), or an MDP.

A PRISM model is a parallel composition of modules, whose state is determined by a set of variables. A module consists of a collection of guarded commands. In an MDP or DTMC, each such command consists of a guard g (i.e., a state predicate) and one or more updates for the module's variables, where each update u_i is labeled by the probability λ_i with which u_i occurs from a state satisfying the guard g. When a module has a command whose guard is satisfied in the current model state, it can update its variables probabilistically, according to the specified updates. In a DTMC or a CTMC model, only one command can be enabled in a state. In MDPs, multiple commands (actions) may be enabled simultaneously, thus representing a nondeterministic choice between multiple discrete probability distributions over successor states.

PRISM-games [3] is an extension of PRISM that supports the formulation and the analysis of turn based stochastic multi-player games. Stochastic games are described in a modeling language similar to the one for PRISM. A model is composed of modules, whose behavior is specified by a set of guarded commands, each of which contains an (optional) action label, a guard, and a probabilistic update for the module's variables. For action-labeled commands, multiple modules execute updates synchronously, if all their guards are satisfied. Each probabilistic transition in the model is thus associated with either an action label or a single module. A model also defines players, each of which is assigned modules, as well as a disjoint subset of the model's synchronizing action labels. A module can be part of multiple players as long as each enclosed command is assigned to exactly one player. Thus, each probabilistic transition is assigned to one player. All possible probabilistic transitions on a state must belong to the same player; PRISM-games detects and disallows concurrent actions. All players must be divided into two groups. These groups of players (coalitions) act as adversaries of each other, thus taking into account their competitive behavior.

PRISM-games supports an extension of rPATL (Probabilistic Alternating-time Temporal Logic with Rewards), a CTL-style branching-time temporal logic that can be used to express quantitative properties of stochastic games with rewards. rPATL allows one to write coalition-based properties that identify a strategy that maximizes or minimizes either the expected probability of a path or the expected value of the accumulated reward while reaching a set of states.

3. DEGROOT MODEL AND EXTENSIONS AS STOCHASTIC PROCESSES

In this work, we present the formulation of the DeGroot model as a stochastic process (*DeGroot Problem DP*) along with two extensions where external influence is wielded by strategic entities: the *DeGroot Influence Problem (DIP)* and the *DeGroot Game (DG)*. The software that we used to construct the DeGroot model and our extensions' models is PRISM and PRISM-games.

3.1 The DeGroot Problem (DP)

3.1.1 Definition of DP

The *DeGroot Problem* DP consists of a tuple $(G\langle N,P\rangle, O_0)$ where G is the graph of the model, $N = \{1,...,n\}$ is a set of nodes, P is a real-valued $n \times n$ matrix with non-negative elements and $O_0 = \{o_{0,1},...,o_{0,n}\}$ is a vector of initial opinions of nodes N. The element $p_{i,j}$ of matrix P represents the link weight of node i to node j that is used in the opinion update process of node i. The matrix P is also referred to as the *adjacency matrix* as it defines the links of the nodes to each other [14]. In case two nodes are not linked, the corresponding element of matrix P is zero. The sum of each row i of matrix P is equal to 1 as its elements represent the factors of the weighted averaging process of node i in the DeGroot model. The opinions $o_{0,i}$ of vector O_0 are real values that represent the belief of the corresponding node on a specific matter of interest. Their range is from 0 to 1.

Our goal in the DP problem is to evaluate using PRISM the final opinion in state of consensus given all the parameters $(G\langle N,P\rangle, O_0)$. Solving the underlying DeGroot model is a problem of polynomial complexity. The influence of each node of the social network is determined by the eigenvector centrality of the node. Algorithms for such problems have been studied especially in the context of the popular PageRank centrality [17]. The exact complexity depends on the specific parameters of the problem [2].

3.1.2 Solving DP

As a start point, we modelled the social graph of an example $DP^* = (G^*\langle N^*,P^*\rangle, O_0^*)$ as a DTMC in PRISM. Figure 2(a) depicts the graph G^* of model DP^* used in our experiment. The graph is strongly connected and the greatest common divisor of all simple cycles is 1. Therefore, matrix P^* converges and the underlying DeGroot model reaches a consensus. The DTMC model consists of six states that correspond to the six nodes of our graph. The transitions between the states are defined by matrix P^*. The initial opinions of all nodes were set to 0.5 except node's 1 and node's 6 opinions which were set to 0 and 1 respectively.

Our aim was to extract the *stationary probability vector* π as DeGroot defined it in [7]. The stationary probability vector π is the solution of Equation 2 and its components are non negative numbers whose sum is 1.

$$\pi P^* = \pi \tag{2}$$

Vector π can be used to calculate the opinion in the state of consensus of our social network. From Equations 1 and 2 we can deduce that o_c^*, the opinion in consensus, can be computed by multiplying the transpose π vector with the initial opinions of the nodes, as described in Equation 3.

$$o_c^* = \pi O_0^* \tag{3}$$

The extraction of the stationary probability vector π was achieved using the available functionalities of PRISM, which allows the computation of the *steady state probabilities* of the DTMC corresponding to the factors of vector π when the DeGroot model converges.

The factors of vector π and the opinion in state of consensus o_c^* of the DP^* retrieved by our PRISM model are presented in Table 1.

3.2 The DeGroot Influence Problem (DIP)

3.2.1 Definition of DIP

We extend the DeGroot Problem DP with the introduction of a strategic entity D that aims to tamper with the consensus formation process in order to bias the final opinion of the social graph towards a preferred one.

The *DeGroot Influence Problem (DIP)* consists of a tuple $(G\langle N,P\rangle, O_0, D\langle A,t\rangle)$ where G is the graph of the model,

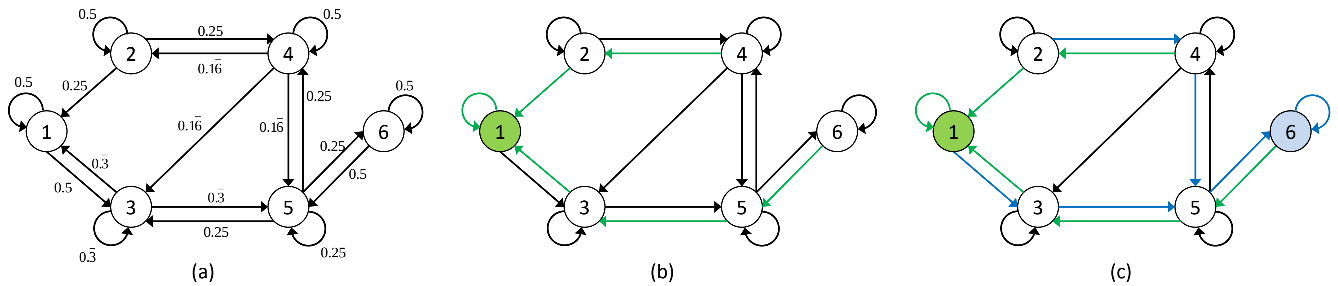

Figure 2: (a) Graph G^* of DP^* and the factors of P^* (b) Graph G^* of DIP^* and strategy σ (green arrows) of the decision maker (c) Graph G^* of DG^* and strategies σ_1 and σ_2 (green and blue arrows) of players D_1, D_2.

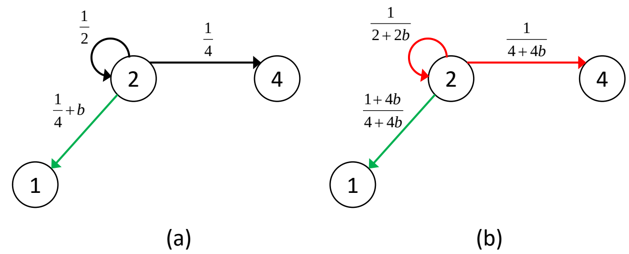

Figure 3: (a) The decision maker D chooses action $a_{21} = b$ (b) The distribution of all $p_{2j}, j \in N$ is reformed after the normalization process.

$N = \{1, ..., n\}$ is a set of nodes, P is a real-valued $n \times n$ matrix with non-negative elements, $O_0 = \{o_{0,1}, ..., o_{0,n}\}$ is a vector of initial opinions of nodes N, D is the strategic entity (i.e. the decision maker) that interferes with the consensus formation process, A is the set of *actions* available to D and t is the *target opinion* of D. The elements G, N, P, O_0 comply with the same restrictions as in DP. The set of actions A consists of real-valued elements a_{ij} representing action j that D can undertake on node i. Each a_{ij} corresponds to an alteration of the value p_{ij} of matrix P. Set A contains actions for the non-zero elements of P.

Figure 3 illustrates the undergoing changes forced by an action a_{ij} chosen by D. The decision maker chooses action $a_{21} = b$. The element $p_{21} = \frac{1}{4}$ is increased by a factor b and D enhances, thus, the opinion of node 1 in the update process of node 2 (Figure 3a). The modified value of $p_{21} = \frac{1}{4} + b$ forces a normalization process for all $p_{2j}, j \in N$ in order to comply with the weighted averaging process of the DeGroot model of the graph G. The normalization reforms the distribution of node's 2 factors (Figure 3b). p_{21} is increased while p_{22} and p_{24} are decreased. This example demonstrates that a single action on element p_{ij} of G triggers the alteration of other elements and, consequently, the alteration of the stationary probability vector π of the whole graph.

D is urged to select the proper actions a_{ij} in order to bias the consensus of G towards t. Hence, D aims to construct a strategy $\sigma = \{a_{ij} | a_{ij} \in A\}$ that would alter the stationary probability vector π into π' to promote the prevalence of the opinions of those nodes that are closest to t. We should note that the decision maker can only choose one action for every node of the graph, i.e. $\forall a_{ij}, a_{kl} \in \sigma, i = k \Rightarrow j = l$.

3.2.2 Solving DIP

We implemented an example $DIP^* = (G^* \langle N^*, P^* \rangle, O_0^*, D^* \langle A^*, t^* \rangle)$ as a MDP in PRISM. The graph G^* and the opinion vector O_0^* is equivalent to the graph of the DP^* implementation of Section 3.1.2. In our experiment, the decision maker D^* was provided with a set of actions A^* consisting of actions a_{ij}^* with values set to 0.25 for all the existing links of graph G^* and the target opinion t^* was set to 0. The MDPs can be solved in polynomial time using dynamic or linear programming approaches [21, 18].

The aim of our implementation was to force the decision maker to construct a strategy σ^* in PRISM, that would promote his target opinion t^* hosted by node 1, and extract it in order to examine it. With the use of a suitable property expression, PRISM exported the decision maker's strategy in the form of a DTMC eliminating, thus, the non-used actions of the MDP. The extracted DTMC that incorporated strategy σ^* was subsequently imported in PRISM in order to compute the reformed stationary probability vector π' imposed by the strategy's actions. Vector π' and the graphs consensus was computed accurately as in the case of DP^*.

The reformed stationary probability vector π' of our experiment is presented in Table 1 and the actions of the constructed strategy σ^* are depicted in Figure 2(b) as green arrows. It is evident that the decision maker's strategy selected these actions that promoted the diffusion of node's 1 opinion. The same deduction can be made by examining the factors of π': the factor of node 1 is significantly increased and this enhancement is also clear in its neighbouring node 3. The factors of the rest of the nodes were decreased and the opinion in state of consensus was decreased as it was intended by the decision maker's strategy.

3.3 The DeGroot Game (DG)

3.3.1 Definition of DG

A further extension of the DeGroot Problem DP is the introduction of a second strategic entity to the DeGroot Influence Problem DIP that aims to bias the formation of consensus towards his favored opinion.

The *DeGroot Game (DG)* consists of a tuple $(G \langle N, P \rangle, O_0, D_1 \langle A_1, t_1 \rangle, D_2 \langle A_2, t_2 \rangle)$ where G is the graph of the model, $N = \{1, ..., n\}$ is a set of nodes, P is a real-valued $n \times n$ matrix with non-negative elements, $O_0 = \{o_{0,1}, ..., o_{0,n}\}$ is a vector of initial opinions of nodes N, D_1 and D_2 are the two strategic entities (i.e. players) that interfere with the consensus formation process, A_1 and A_2 is the set of *actions* available to D_1 and D_2 respectively and t_0 and t_1 are their *target opinions*. The sets A_1 and A_2 consist of real-valued elements $a_{1,ij}$ and $a_{2,ij}$ respectively representing the

Table 1: **Vectors π, π' and π'' and the final opinions.**

Node	Opinion	Factors of vector π	Factors of vector π'	Factors of vector π''
1	0	0.2045	0.3943	0.2232
2	0.5	0.0454	0.0343	0.0322
3	0.5	0.2727	0.3086	0.2594
4	0.5	0.1363	0.0617	0.0773
5	0.5	0.2272	0.1509	0.2447
6	1	0.1136	0.0503	0.1631
Final opinion		0.45454	0.32800	0.46994

jth action of each player on node i. Each $a_{1,ij}$ and $a_{2,ij}$ corresponds to an alteration of the value p_{ij} of matrix P.

The mechanism of the players' actions is equivalent to the mechanism described in the case of DIP: when the two players choose their actions $a_{1,ij}$ and $a_{2,iz}$ they alter the values p_{ij} and p_{iz} of P that correspond to the weights of node's i links to nodes j and z. Each alteration represents an enhancement of the leading node's opinion. After the weights of the links are updated, a normalization process for all $p_{ik}, k \in N$ is necessary in order to apply the weighted averaging process of the DeGroot model. Consequently, all the elements p_{ik} are influenced by actions $a_{1,ij}$ and $a_{2,iz}$ and the stationary probability vector π of the graph G is reformed to π'' as the result of the players' actions.

Players D_1 and D_2 are urged to select the proper actions in order to influence the consensus formation process towards their target opinions t_1 and t_2. The players develop their strategies $\sigma_1 = \{a_{1,ij} | a_{1,ij} \in A_1\}$ and $\sigma_2 = \{a_{2,ij} | a_{2,ij} \in A_2\}$ and the chosen actions of their strategies contribute to the formation of a new stationary probability vector π'' of the graph G. The restrictions of the decision maker's strategy in DIP apply also to strategies σ_1 and σ_2 of the DG's players: each player can choose only one action for each node.

3.3.2 Solving DG

We implemented an example $GD^* = (G^* \langle N^*, P^* \rangle, O_0^*, D_1^* \langle A_1^*, t_1^* \rangle, D_2^* \langle A_2^*, t_2^* \rangle)$ as a stochastic game in PRISM-games. The graph G^* and the opinion vector O_0^* is equivalent to the graph of the DP^* of Section 3.1.2. The sets A_1^* and A_2^* consisted of actions $a_{1,ij}^*$ and $a_{2,ij}^*$ with values set to 0.25 for all the existing links of graph G^* and the target opinions t_1^* and t_2^* were set to 0 and 1 respectively imposing, thus, a competitive relation between the players. The stochastic game developed in PRISM-games was a stopping game with a stopping factor $s_f = 0.05$.

The question whether at least certain classes of stochastic games can be solved in polynomial time is still open. In [4] Condon showed that simple stochastic games are in $\mathsf{NP} \cup \mathsf{coNP}$. The stochastic game used in our extended model is at least as hard as simple stochastic games. Several complexity results are provided in [20]. Consequently, for influence games in this class one has to carefully design the model or make compromises to keep the computational demand for the solution at an acceptable level.

The aim of our experiment was to urge the players to develop competing strategies in the PRISM-games framework and retrieve these strategies. Player's D_1^* strategy should promote his target opinion t_1 hosted by node 1 while player's D_2^* strategy should be aiming for the prevalence of his target opinion t_2 hosted by node 6. The proper incentives for the construction of such strategies were expressed with the use of rPATL and the desired strategies were developed. The strategies were extracted with the use of PRISM-games simulation functionality [8]. The results of the strategies construction's process were imported in a separate DTMC model of PRISM-games and both the stationary probability vector π'' and the consensus were computed.

Figure 2(c) presents the results of our experiment for the DeGroot game GD^*. The green arrows represent the strategy developed by player D_1^* and the blue arrows represent player's D_2^* strategy. Player D_1^* wields his influence on the edges that maximizes the probability of reaching node 1 while player D_2^* chooses to affect the edges leading to node 6. The reformed stationary distribution vector π'' is presented in Table 1. The factors of nodes 1 and 6 are increased compared to the initial factors of π although node's 1 increase is not as significant as in the DIP^* (vector π') because of the intervention of player's D_2^* strategy. Node's 5 factor is also increased as a consequence of player's D_2^* strategy: his actions affect the edges leading to node 6 and, as node 5 is the only node connecting it with the social graph, its factor is also affected. Nodes' 2, 3 and 4 factors are decreased compared to those derived from DP^* (vector π). The opinion in state of consensus is increased compared to the final opinions of both DP^* and DIP^*. This is due to D_2^*'s strategy which increased node's 6 factor in GD^*.

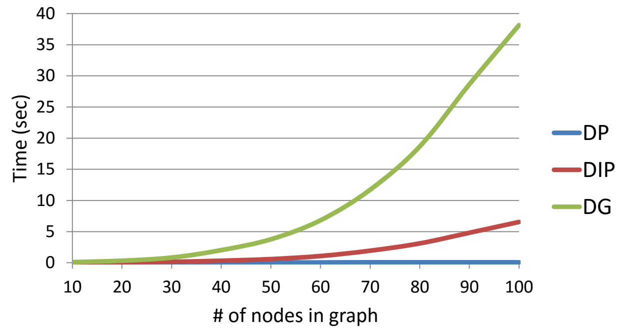

Figure 4: Measured execution times of our experiments

3.4 Measurement of execution time

In the final stage of our work we developed a set of experiments consisting of large models for DP, DIP and DG in order to examine the performance of the software in terms of execution time. We automated the process of transcribing large social graphs into PRISM language representing settings of DP, DIP and DG and executed the resulting problems using PRISM-games' *explicit engine* . The social graphs of our experiments were generated based on scale-free graphs and the links between the nodes formed an exponential distribution. The restriction of the generated graphs for strong connectivity was satisfied with the addition of proper edges that formed a cycle containing all the nodes. Each node of the graph had an edge leading to itself and, therefore, the common divisor of all simple cycles was 1. Hence, all the requirements for the reach of consensus were met.

The set of experiments consisted of ten categories depending on the number of the graph's nodes and the sample set for each category consisted of 50 auto-generated models. Figure 4 illustrates the average execution times for each category. In case of the DP experiments, the measurement times correspond to the process of building the model and evaluating the stationary probability vector π of the graph. It is evident that the execution times exhibit linear behaviour and were rapidly computed even for larger models (the average time for a graph with 100 nodes is 0.0679 seconds). In case of DIP, the experiments contained the process of building the model and exporting the decision maker's strategy through the evaluation of the suitable property. The measurements revealed a significant increase in the complexity of the calculations (the average time for a graph with 100 nodes is 6.5455 seconds). In case of DG, the measured times correspond to the process of building the model and generating the strategies for both players. The results indicate an exponential-like increase in the complexity (the average time for a graph with 100 nodes is 38.1375 seconds).

4. DISCUSSION

The DeGroot model shares common mathematical foundations with Markov chains. When external influence is wielded in DeGroot model, it corresponds to an intervention to the underlying Markov chain. This type of interventions are extensively studied in the field of model checking. Therefore, in this work we used software from the field of model checking in our study of DeGroot model and our extensions.

The successful modelling of the DeGroot model, as DeGroot Problem DP, and its extensions with the use of software for model checking highlights the strong underlying coherence of the models to stochastic processes. The DeGroot model is based on Markov chains, its extension with one decision maker is reduced to a MDP and the extension with two players is essentially a stochastic game. Figure 5 presents schematically the coherence of our experiments' models to stochastic processes.

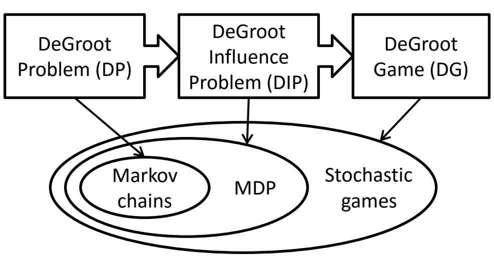

Figure 5: Coherence of models to stochastic processes.

Eigenvector centrality is the dominant factor in our experiments. In case of the DP, the stationary probability vector π extracted by the corresponding Markov chain defines the opinion in state of consensus. When external influence is wielded, the players aim to bias vector π in order to manipulate the final opinion in stable state. In our experiments we demonstrated the influence of external strategic entities on the reformation of vector π: in the case of DIP the reformed vector π' promoted the opinion of the decision maker's favourite node while in the case of DG the vector π'' exhibited the influence of the players' strategies towards the nodes hosting their preferred opinions.

Our research unveiled the restrictions of modelling the DeGroot model to stochastic processes. More specifically in DP, the formulation of DeGroot Model as a Markov chain and of DIP as a MDP can be effectively accomplished while in the case of DG, the execution of the stochastic game would exhibit exponential increase of complexity due the multiple alternatives offered to the players of the game.

5. REFERENCES

[1] R. Bellman. A markovian decision process. Technical report, DTIC Document, 1957.

[2] M. Bianchini, M. Gori, and F. Scarselli. Inside pagerank. *ACM TOIT*, 5(1):92–128, 2005.

[3] T. Chen, V. Forejt, M. Kwiatkowska, D. Parker, and A. Simaitis. PRISM-games: A model checker for stochastic multi-player games. In *Proc. of TACAS'13*, volume 7795 of *LNCS*, pages 185–191, 2013.

[4] A. Condon. The complexity of stochastic games. *Information and Computation*, 96(2):203–224, 1992.

[5] S. A. W. Cordwell. pymdptoolbox 4.0-b3. https://pypi.python.org/pypi/pymdptoolbox/, 1 2015.

[6] M.-J. Cros. Markov Decision Processes (MDP) Toolbox. MathWorks, 11 2009.

[7] M. H. DeGroot. Reaching a consensus. *Jour. of the Amer. Stat. Assoc.*, 69(345):118–121, 1974.

[8] T. Deshpande, P. Katsaros, S. A. Smolka, and S. D. Stoller. Stochastic game-based analysis of the dns bandwidth amplification attack using probabilistic model checking. In *Dependable Computing Conf., 2014 Tenth European*, pages 226–237. IEEE, 2014.

[9] N. E. Friedkin and E. C. Johnsen. Social influence and opinions. *J. of Math. Sociology*, 15(3-4):193–206, 1990.

[10] N. E. Friedkin and E. C. Johnsen. Social influence networks and opinion change. *Advances in group processes*, 16(1):1–29, 1999.

[11] B. Golub and M. O. Jackson. Naive learning in social networks and the wisdom of crowds. *Amer. Econ. Jour.: Microeconomics*, pages 112–149, 2010.

[12] R. Hegselmann, U. Krause, et al. Opinion dynamics and bounded confidence models, analysis, and simulation. *JASSS*, 5(3), 2002.

[13] R. A. Howard. Dynamic programming and markov processes. 1960.

[14] M. O. Jackson. *Social and Economic Networks.* Princeton University Press, 2008.

[15] M. Kwiatkowska, G. Norman, and D. Parker. Stochastic model checking. In *SFM'07*, LNCS 4486, pages 220–270. Springer, 2007.

[16] M. Kwiatkowska, G. Norman, and D. Parker. PRISM 4.0: Verification of probabilistic real-time systems. In *CAV'11*, LNCS 6806, pages 585–591. Springer, 2011.

[17] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: bringing order to the web. 1999.

[18] C. H. Papadimitriou and J. N. Tsitsiklis. The complexity of markov decision processes. *Math of Oper Res*, 12(3):441–450, 1987.

[19] L. S. Shapley. Stochastic games. *Proc. of the National Academy of Sciences*, 39(10):1095–1100, 1953.

[20] A. Simaitis. *Automatic verification of competitive stochastic systems.* PhD thesis, Univ of Oxford, 2014.

[21] J. N. Tsitsiklis. Computational complexity in markov decision theory. *HERMIS-An Intern. Jour. of Comp. Math. and its Applications*, 9(1):45–54, 2007.