Multiple Agent Fashion Game

Master's Thesis — 2013

Zuzanna Stamirowska

Supervisor: Francis Bloch
Sciences Po Paris, Ecole Polytechnique, ENSAE

"There is no other way toward an understanding of social phenomena but through our understanding of individual actions directed toward other people and guided by their expected behaviour."
— F.A. von Hayek [12]

Abstract. The market of luxury exists thanks to the feeling of exclusiveness which it creates. Such specificity creates externalities. On the one hand, luxury firms want to make their product desirable to a big audience in order to maximise profits, but on the other hand they need to take into account the feeling of exclusivity, which is one of the most important features of their products. As a result, such firms face a tradeoff between delivering a large number of products to as many clients as possible, and preservation of exclusivity. In this paper we develop a model of Multiple Agent Fashion Game, which captures both conformist and non-conformist behaviour of agents (consumers) connected by a network, which signifies interactions between them. Non-conformism implies a repulsion of the same, a denial to conform to a rule or standard, whereas conformity implies imitation. In this paper the agents have two alternatives to choose from. We present efficient algorithms for maximisation of social utility in such a setting for special topologies of graphs. In order to solve the problem in general we present approximation algorithms which make use of semidefinite programming. We also provide some auxiliary results on Nash equilibria of the game and the Price of Anarchy.

Contents

1 Introduction

1.1 Conformity and Vanity

Targeting of marketing campaigns is now in the center of attention of firms delivering products to the market [13][25]. As the world has become more and more digital and social networks have become nearly formalised thanks to online services such as Facebook or Twitter, firms seek ways to exploit the network structure. For instance, Facebook offers targeting of adverts thanks to the inside knowledge of personal data of the users, such as topics on which they post, their education, age etc., but most importantly — Facebook also allows to reach the friends of targeted people, who are likely to imitate the behaviour of the latter [33]. The position of a given person in the network is known to have an important effect on how she can influence others [1]. For instance, by funding the most central person in the network and careful targeting of their adverts, marketing firms increase the efficiency of their marketing campaigns. An example of a strategy exploiting the features of a network would be to give out free samples of products to those who are the most central, and to ask them to spread word about this. This targeting technique is more efficient than a random distribution of free samples, and allows firms to reach quickly the customers who are really interested in their product. The imitation effect that we have described above can also be seen as conformity, which is defined in sociology as *human behavior which follows the established norms of a group or society. The bulk of human behavior is of a conforming nature as people accept and internalise the values of their culture or subculture.*[34]

Conformity appears in many contexts and can concern many aspects of life. This can take the form of imitating behaviour with respect to material products, as is the case with products whose possession is seen as customary for or associated with some social classes, like a castle owned by the aristocracy, or an expensive watch worn by a businessman (even if he prefers to check the time on his mobile). Goods can be a sign of belonging to a certain group and a symbol of social status. Conformity can also concern behaviour understood as attitudes, such as the effort put into studying [3], into searching for a job, the language and accent used, way of spending spare time, etc. Customs related to these aspects of life appeared as novelties at some point in time, and then had to be imitated by others in order to become a custom. One could ask: why did some novelties become customs, whereas others did not? The answer to this question is partially offered by anal-

ysis of connections between people and their relative importance. This does not necessarily follow from their financial wealth or fame, as shown by the example of the rise of the House of Medici in 15th century Florence [24]. Initially, they were neither the most notable, nor the wealthiest of families of Florence. All of a sudden, members of the House of Medici were promoted to the highest dignities, some became popes, one became the Queen of France, and so forth. By looking at the relationships of marriage and employment in Florence at the time, one can find that the Medici, regardless of their relative inferiority in terms of wealth, were the most central of the houses of Florence in the social network of the day. Intuitively, this means, that the other great families, in order to access one another to negotiate a deal, a marriage, etc., had to go through an intermediary — the House of Medici.

The mechanism of conformity is well studied in literature on social and economic networks. Young [31] analyses the creation of standards via conventions. Zenou et al. [26] look at network conformity effects in petty crimes. Some researchers direct their attention to both conformity and nonconformity effects. Shy [28] shows how conformity and nonconformity can be used to explain and simulate changes in the proportions of secular and religious people. Grilo, Shy, and Thisse [11] explore the question of how non-conformist behaviour affects price competition in a spacial duopoly setting, but they do not analyse the toplogy of the underlying network. To the best of the author's knowledge, there are only few studies that incorporate the heterogeneity of agents with respect to conformity, offering an analytical study of both conformity and non-conformist vanity on networks.

This paper aims to fill in this gap.

Nonconformity is the refusal to conform to accepted standards or conventions, whereas vanity is a sort of arrogant pride. We use these two words to simply describe a person who has negative taste for conformity. Identification of such heterogeneity is important for markets of innovation, luxury, or positional goods. A part of the definition of luxury is the feeling of vanity and uniqueness that follows ownership of such goods — exclusivity [2]. This elite feeling makes such products very desirable, which in turn appeals to the conformity effect. Some people like to be early adopters, or fashion leaders, and extract utility from being different than their friends or associates. If the producer is aware of the division of her customers into groups of non-conformists (who may also be described by the terms fashion leaders or early adopters) and conformists (those who want to imitate behaviour of

others), she can carefully schedule the launch of a new product so as to maximise profits from sales. For instance, she can first offer to sell a small quantity of a product only to early adopters, with a guarantee that for some time the general public will not have access to that good. After some time she can sell the product to the masses, who may be even more willing to buy the product because of the elitist image created in the first phase of the launch. The more central in the network the early adopters are, the stronger the elitist image will be. However, at the same time, the centrality of the early adopters results in a faster process of imitation by conformists, so the producer will have to make the first pre-launch phase longer in order to keep the early adopters happy. This is just an imaginary scenario, and probably only one of many possible applications of the presented model.

Conformity can be seen as a "keep up with the Joneses" scenario, in which the player's payoff depends on how different his choices are from those of his neighbours. This in some sense reflects the behaviour of the masses following fashion, who want to adopt the behaviour of the "elite" people who are for them a point of reference. Conformity is probably the most fundamental property of games on networks, mostly because it has direct applications, and it can be tested empirically. A representative example of a game with such a property is given by Jackson and Zenou [16], where the utility of a player depends, among others, on the difference between the average behaviour of the friends of the player and his own behaviour. This difference bears more weight for the player depending on his individual taste for conformity. In the game from [16], the agents are supposed to choose a level of effort, taking into account their preferences (preferred individual level of effort, and taste for conformity with their peers).

1.2 Our contribution

In this paper, we develop a Multiple Agent Fashion Game, where agents can either be conformists or non-conformists, thus receiving heterogeneous payoffs depending on the actions of their neighbours, their own interaction preference (whether they are conformists, or non-conformists), and their own choices. In our model the choice variable is binary, so agents can choose only between, say, red or green dresses. Conformity or nonconformity is captured by an integer value that is also binary, depending on whether a given vertex is a conformist or non-conformist. The vertices of the network represent agents, and each edge stands for a friendship

relationship between the two. Each individual choice in this game impacts directly the social utility function, which we define as the sum utilities associated with every edge in the graph. This formulation is equivalent to the sum of individual utilities.

The primary question that we seek to answer in this paper is how to maximise social utility in the Multiple Agent Fashion Game?

The paper is organised in following manner. Subsection 1.3 presents a detailed literature review. Section 2 presents the model and all basic lemmas that are needed in the following sections. In Section 3 we present and analyse algorithms for maximisation of social utility in the studied game. Subsection 3.1 provides algorithms for finding optimal solutions in special instances (rings, bipartite graphs, and complete graphs). Subsection 3.2 provides a brief introduction to the semidefinite programming technique in reference to the problem of maximum cut (Max-Cut) in undirected graphs. This is then used in Subsection 3.3, where we present a proof of NP-hardness of our problem by reduction from Max-Cut, and an approximation algorithm for the maximisation of social utility in the general case. In Subsection 3.6 we present simulations of performance of our approximation algorithm. In section 4 we provide auxiliary results on checking for existence of Nash equilibria and on Price of Anarchy in the game. We show that the problem of checking for Nash equilibria in trees can be solved efficiently. We also describe the behavior of the Price of Anarchy on the line, depending on the number of edges and arrangement of nodes. We conclude the paper with Section 5.

1.3 Overview of literature

Network Economics is an area that spans many scientific disciplines. This is why the present paper builds upon literature from different fields, such as economics, physics, and computer science. Translating classical game theoretical problems, present in economics for decades, into a network setting is now in the center of attention of both economists and computer scientists[9] [7]. The idea of Fashion Game is present in economics, however, to the best of the author's knowledge has never been translated into a network setting. Parts of related economic literature were presented in 1.1.

Within computer science, algorithmic graph theory is a particularly relevant subfield. Optimisation and complexity theory are very useful to be able to say whether

the problem we want to solve is even solvable in reasonable time. If it turns out that finding the global social optimum is a hard task (as is the case for the studied game), optimisation theory helps us to approximate the optimal solution. Physics proves useful when it comes to modeling complex and dynamic systems. The Ising model[14] for describing magnetic-type interactions in lattices (or more general networks) is the direct inspiration for the Multiple Agent Fashion Game.

1.3.1 Economics

The *Fashion Game* is a game mentioned by Peyton Young in [32] in order to illustrate a fictitious game with no-convergence property. Young refers to a game presented by Lloyd Shapley in 1964 [27], where two agents play a zero-sum game and each agent has three possible choices to make. Shapley presents a learning process, where each agent makes her choice of strategy based on the history of the game. In the case of only two alternatives, the game converges to a fixed probability of choosing one of these alternatives, i.e. to a mixed strategy equilibrium. The problem arises when there are three or more alternatives, in such a case the game need not converge. Young takes this example to show non-convergence and justify the idea of stochastic stability.

In the game they consider there are two agents. One of them wants to follow the trend, and for the sake of this paper she will be called the conformist (*fashion follower* in Young). The second agent wants to be different from others and she will be called the non-conformist (*fashion leader* in Young). Imagine, for example, that two ladies meet at parties in a country club. Each of them has three different available dresses: blue, red and green. Let us assume a gradation of preferences, so they both prefer green to red, and red to blue. This gradation is necessary because of the existence of three alternatives; in the two-option setting it is not needed. Young shows that even if the process does not converge, some states are more probable than others. In fact, the behaviour of agents will fall into an easily predictable cycle, where one alternative will fall out of the picture and we will be in the simple two-option setting. An example of the Fashion Game cycle can be seen in Figure 1. For the first party, the two ladies wear their favourite green dresses. The non-conformist is very unhappy with the fact that the conformist was wearing the same dress as hers, so for party number two she puts on her second best preference — the red dress. The conformist lady sees this and copies, wearing a red dress as well for party number three. Again, the non-conformist lady is

unhappy and she switches. She gets to choose between blue and green, but she clearly prefers green to blue, so she puts on a green dress for party number four. The conformist lady sees it, she copies, and party number five closes the cycle. The described states are the only possible states of this game.

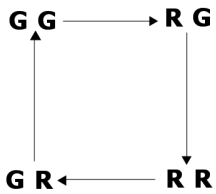

Figure 1: Fashion Game cycle

In the Multiple Agent Fashion Game model which we study here we will assume that each agent has only two alternatives, so there will be no need for gradation of preferences. In our setting the number of players will be arbitrary and the topology of their relations will matter. So, we could say that there will be multiple country clubs, and the each agent will get to meet all of her friends (and only her friends) in each turn, getting to choose only one dress for all these meetings.

1.3.2 Computer Science

Graph theory contributed greatly to the development of Multiple Agent Fashion Game model. In order to study games on networks it was necessary to study properties of different classes of graphs that were used for the proofs in this paper. A short introduction to graph theory in a social network context is given by Jackson [15], where the author presents the most classical notions and problems of graph theory, such as Chinese postman, Hamiltionian cycle, centrality measures, etc. This introduction lacks, however, computational properties of different classes of graphs. A deeper discussion of these issues can be found in [30]. For a more complete understanding of discrete optimisation and algorithmic computer science, the reader is referred to [5] and [6]. In this paper we will exploit properties of bipartite and complete graphs, together with basic lemmas allowing to compute the number of edges or nodes in a given graph.

Models of diffusion in social networks became important for this research as an inspiration for the optimisation algorithm (Algorithm 1). A nice explanation of such

models is given in Jackson's book mentioned above, and in Easley, Kleinberg [8]. The two main and most up-to-date models have been studied by Kleinberg an Tardos in their two papers entitled *Maximising the Spread of Influence Trought a Social Network* [18], and *Influential Nodes in a Diffusion Model for Social Networks* [19]. The two papers present a rigorous study of the Treshold and Cascade models, which was needed to address the question: which nodes in a social network should be targeted in order to maximise the diffusion effect?

In order to prove NP-hardness in the general case and to find approximation algorithms to solve for optimal payoff we will use complexity theory [17]. The solution for the general case is largely based upon Goemans and Williamson's paper [10], which provides the best known approximation algorithm for the Max-Cut problem. The algorithm uses the technique of semidefinite programming, which will be presented and explained in detail in Section 3.2.

Price of Anarchy is the ratio of the social optimum outcome over the worst Nash equilibrium. The concept was developed by Koutsoupias and Papadimitriou in [20]. In this study we will use a very simple version of Price of Anarchy, for deterministic settings and lines only.

1.3.3 Physics

The biggest building block for the Multiple Agent Fashion Game model that will be discussed in this paper is the Ising model, which is a mathematical model in statistical mechanics [14]. The Ising model enables to study phase transitions in magnetism, thanks the characterisation of local interactions between the particles of different spins [21]. The interactions can be either ferromagnetic or antiferromagnetic, which is for us the most important feature of the Ising model, that will be incorporated into the model which will be studied in this paper. This is not the first attempt to incorporate the Ising model into social sciences [22], it is however (to the best of the author's knowledge) the first attempt to incorporate it together with the kinds of interactions.

The model that we build up in this paper differs substantially from the interpretations of the Ising model that have infiltrated social sciences. Current literature in social sciences uses this model in order to describe "yes" or "no" choices, so activation or non-activation of a given vertex in the network, where all agents are homogeneous, except for preferences for one or another option (say, political

parties A or B). The interactions are typically based on homophily, which brings Ising model very close to the class of problems widely studied in computer science, like matching or diffusion problems.

In our basic setting agents are heterogeneous. They are of two kinds (conformists and non-conformists), they have two different alternatives to choose from (say spins). In fact, our model is much closer to the initial ferromagnetic and antiferromagnetic interactions idea stated by Ising, than the simple diffusion with adoptation or nonadoptation settings, presened in the Sznajd's model [29] or the diffusion models studied by Kleinberg and Tardos.

The type of the agent is modeled by the value J_{ij} that she assigns to the interactions with her neighbours, which enters the integral of interaction forces towards a given neigbour. We assume that each agent adopts the same behaviour towards all her neighbours. It follows then that if $J_{ij} > 0$ the interaction is a ferromagnetic coupling, so the two agents want to adopt the same option - imitate the neighbour, and if $J_{ij} < 0$ the interaction is antiferromagnetic coupling, so the agents want to adopt different option.

The adopted options are being modeled by σ_i and σ_j, which described spins in the original Ising model (-1 or $+1$). In our setting spins will mean a choice of product.

In the original Ising model the variables of the Hamiltonian (total energy equation) are interpreted in the following manner:

$$H(\sigma) = -\sum_{<i\ j>} J_{ij}\sigma_i\sigma_j - \sum_j h_j\sigma_j$$

where J_{ij} describes the nature of interaction between two neighbouring particles. This can be either repulsion, or attraction.

σ_i describes the spin, either +1 or -1

$\sum_j h_j\sigma_j$ describes interactions of the given particle with the external field. In social sciences this part of equation is understood as a trend, that is independent of agents' friendships. In our fashion game model we leave out the external field that appears in the Ising model ($h_j = 0$).

Another crucial insight of physics for our model is that every system in nature prefers states having the lowest possible energy, likewise the Ising model. In our

model, the lowest possible energy can be interpreted as the highest social utility level. This intuition will be used in construction of Algorithm (1). The Hamiltonian of discrete systems, used in Ising model, will be incorporated into Multiple Agent Fashion Game model as the social utility function.

For a network of n agents we have 2^n possible configurations of choices $y = 1$ or $y = -1$, where y is the choice variable (corresponding to σ in the Ising model), and 2^n possible distributions of conformists and non-conformists.

Our model will describe individual optimisation, where each agent optimises her utility by taking into account her own characteristics (whether she is a conformist or a non-conformist) and the choices of her direct neighbours.

2 Preliminaries

2.1 Model

As described in the introduction, we aim to characterise a game on a network of an arbitrarily chosen structure with heterogeneous agents distributed on that network in an arbitrary manner. The agents differ in their taste for conformity, which can be positive or negative, and their choices, which are conditioned by the choices of their direct neighbours and the individual taste for conformity (which we shall call now on the interaction variable J).

Formally, the setting is described as follows. We are given a graph (network) G. The entire set of vertices V is divided into two subsets V_c and V_n, standing for conformists and non-conformists respectively, $V_c \cup V_n = V$. Each individual vertex can be therefore $v_c \in V_c$ or $v_n \in V_n$. Vertices are connected by edges representing social relationships, and the set of all edges is denoted by E.

Each vertex v has a choice variable y_v, where $y_v \in \{-1, 1\}$. Each vertex in V receives different payoff depending on its own choice, the choices of its neighbours, and interaction variable J that depends on whether the vertex belongs to V_c or V_n. The difference between vertices is the kind of interaction they prefer, either attraction or repulsion. The interaction variable J for a vertex v is $J_v \in \{-1, 1\}$; we have $J_v = 1$ for $v \in V_c$, and $J_v = -1$ for $v \in V_n$. The global payoff is given

by

$$H = \sum_{i,j \in V} J_{ij} y_i y_j$$

where $J_{ij} = J_i$ if i and j are connected by an edge, and $J_{ij} = 0$ otherwise. Note that variable J_{ij} describes kind of interaction preffered by vertex i only — it does not say anything about j's preference (the preference of j is taken into account in the term J_{ji}).

2.2 Important Lemmas

In order to proceed we will need a series of lemmas.

First of all, observe that the expression for the payoff H can be decomposed into payoffs associated with individual edges. For a pair of nodes i, j connected by an edge, the payoff associated with this edge will be given as $J_{ij} y_i y_j + J_{ji} y_j y_i$. Each of the two elements in this sum is either equal to -1 or to $+1$. Thus, the payoff associated with an edge may be equal to -2, 0, or $+2$. We will speak of edges with a negative payoff (-2) as conflicts (marked in red in all figures), with a zero payoff as neutral (marked in yellow). Edges which contribute $+2$ to the global payoff are marked in green.

Lemma 1. *Every edge between a node $v_c \in V_c$ and a node $v_n \in V_n$ is neutral for (i.e., contributes 0 to) the global payoff H, therefore it can be omitted.*

Proof. Consider an edge between a pair of nodes $v_c \in V_c$ and $v_n \in V_n$. Since interaction variable J takes values of opposite sign for conformists and non-conformists, $J_{v_c} = 1$, while $J_{v_n} = -1$, and the product of choice variables is same for both vertices, we have $J_{v_c v_n} y_{v_c} y_{v_n} + J_{v_n v_c} y_{v_n} y_{v_c} = (J_{v_c v_n} + J_{v_n v_c}) y_{v_n} y_{v_c} = 0$, and so the contributions to the global payoff H associated with the edge between v_c and v_n always cancel-out.

Therefore these edges bring nothing to the global payoff H and they may be omitted. □

Lemma 1 will allow us to to treat the problem from the perspective of homogeneous subgraphs, where by a homogeneous graph we mean a situation when $V = V_c$ or $V = V_n$. Finding the optimal solution for the network boils down to optimising the choices of vertices in each homogeneous subgraph.

Lemma 2. *If $V = V_c$, then the payoff is maximised, regardless of the network's topology, if and only if all vertices choose the same value of y.*

Proof. The interaction variable for conformists equals $J_c = 1$. So,

$$H = \sum_{\{i,j\} \in E} y_i y_j.$$

When all choices y_i are the same, we have $H = 2|E|$.

On the other hand, in order to ensure that $H = 2|E|$, we cannot have a negative term $y_i \cdot y_j$. This means that for any two adjacent vertices i and j with $\{i, j\} \in E$, we have $y_i = y_j$. Since G is a connected network, this means that all vertices in G must make the same choice. □

Lemma 2 shows us that optimising the graphs of conformists is straightforward and does not depend on topology. It is enough to assign the same value of y to all conformists in order to maximise their utility. As we know from Lemma 1, the edges between conformists and non-conformists always cancel out, therefore the overall gain in utility can be extracted only from optimising choices between vertices with the same value of the interaction variable. Non-conformists are more tricky than conformists because of possible conflicts arising from *repulsion* property ($J = -1$). In this case, topology will matter a lot for the final payoff.

Lemma 3. *If $V = V_n$, and the graph is bipartite, the payoff of that network is maximised if and only if every pair $v_i, v_j \in V_n$ of adjacent vertices chooses opposite values of y.*

Proof. Consider a network where $V = V_n$ and recall the global payoff equation with $J_n = -1$ which takes the form

$$H = -\sum_{\{i,j\} \in E} y_i y_j.$$

The product $y_i \cdot y_j$ is negative when the two vertices i, j choose opposite values of y. To maximise the global payoff, we would like as many pairs of adjacent vertices as possible to make opposite choices.

Since the network is bipartite, it can be coloured with two colours so that adjacent vertices get different colours. If we make choice $y = 1$ for all vertices having one colour, and choice $y = -1$ for all vertices having the other colour, two-colourability

implies that that there will be no conflict and $y_i \cdot y_j = -1$ for all pairs of adjacent vertices. Then, we have $H = 2|E|$.

On the other hand, if some pair of adjacent vertices makes the same choice, then for this pair we have $y_i \cdot y_j = 1$. This means that $H < 2|E|$. □

Now, instead of creating a bipartite structure, imagine the set of non-conformists creates a complete graph of size 3, which happens to be the 3-vertex ring. In this case the graph is not bipartite and at least one pair of vertices will be in conflict. Now imagine adding the fourth vertex. We can get a path, a ring, or a sort of blossom (a ring of length 3 with one additional node attached to one of those in the ring). The two first possibilities are bipartite and thus bring no conflicts. However the latter one is not and brings at least one unavoidable conflict. From this reasoning we can conclude that topology influences the type of solution obtained for the considered problem of maximising payoff.

In fact, our problem for non-conformists boils down to the Max-Cut problem that is known to be NP-hard in general case. We will develop on this later in the section on general case, but first we will treat special instances and we will present ways to achieve optimal payoffs for them. Lemma 4 concerns complete graphs. We recall that a complete graph is a connected, undirected graph in which every pair of vertices is connected by an unique edge. If the number of vertices of such a graph is n, then the degree of each vertex is equal to $n - 1$.

Lemma 4. *If G is complete and $V = V_n$, the payoff is optimal if $n/2$ of vertices choose $y = 1$, and the other $n/2$ choose $y = -1$, for n even. If n is odd, the payoff is optimal if we put $y = 1$ for $(n+1)/2$ vertices, and $y = -1$ for $(n-1)/2$ vertices.*

Proof. Suppose that exactly k vertices make a choice of $y = 1$, and the other $n-k$ vertices make a choice of $y = -1$. Since all vertices are homogeneous within the complete graph, which vertices make which choice does not affect the payoff.

For each of the k vertices with a choice of 1, exactly $k - 1$ of its neighbours make the same choice (are in conflict), and $n - k$ of its neighbours make the opposite choice. Thus, such a node contributes $-(k-1)+(n-k) = n-2k+1$ to the global payoff.

For each of the $n-k$ vertices with a choice of -1, exactly $n-k-1$ of its neighbours make the same choice (are in conflict), and k of its neighbours make the opposite

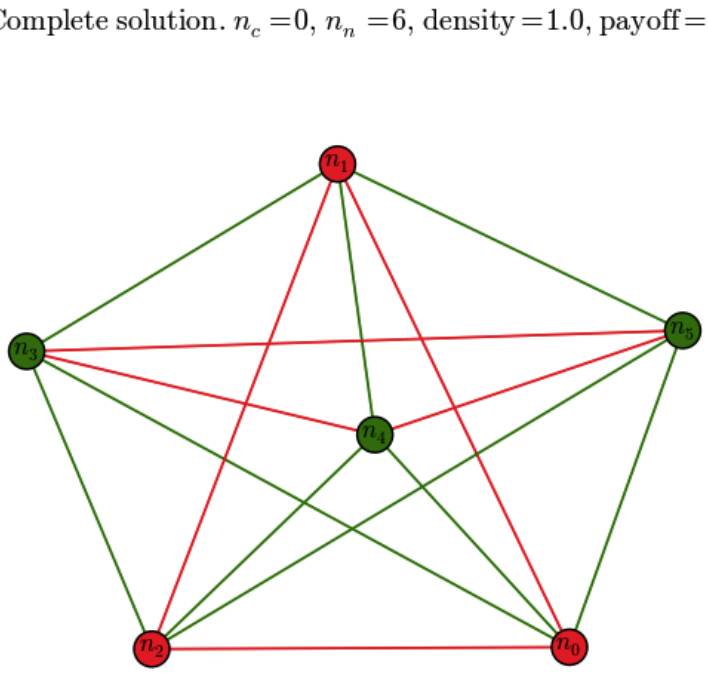

Figure 2: Complete graph of non-conformists

choice. Thus, such a node contributes $-(n-k-1)+k = 2k+1-n$ to the global payoff.

The overall payoff is now given as:

$$H = k(n-2k+1)+(n-k)(2k+1-n) = 4kn-4k^2-n^2+n = 4k(n-k)-n(n-1).$$

This payoff is maximised when k is as close to $n-k$ as possible. So, we choose $k = n/2$ for even n, and $k = (n+1)/2$ for odd n, obtaining the claim.

We remark that the value of the optimal total payoff in the complete graph given by the above formula is n for n even, and $n-1$ for an odd number of vertices. □

The optimal assignment is illustrated in Figure 2 for $n = 6$. We use the formula: $n/2 - 1$ to compute the number of conflicts $6/2 - 1 = 3 - 1 = 2$. We note that each vertex has 5 neighbours and has conflicts with two of them. The two conflicts will be offset by two green edges. Hence, in the overall balance we are left with exactly one edge per vertex that adds to the positive payoff. The overall payoff is therefore equal to $n = 6$.

3 Maximising Social Utility in Multiple Agent Fashion Game

In this section we will present algorithms for maximisation of social utility in the Multiple Agent Fashion Game. The social utility here is understood as the sum of all individual utilities in the graph, but for computational reasons we will sum the payoffs associated with each edge in the graph. The presented algorithms describe the ways in which the choices of $y = \{-1, +1\}$ should be assigned to vertices in order to maximise social utility. Algorithms (1) and (2) offer such solution for special classes of graphs, which are bipartite graphs and rings for Algorithm (1), and complete graphs for Algorithm (2). These classes of graphs exhibit some particular properties which enable us to present efficient polynomial time algorithms for maximising social utility, even when the distribution of conformists and non-conformists on these graphs is still given in an arbitrary way. In the general case the topology of the underlying graph is also arbitrary and does not exhibit any special properties that could facilitate the search for optimal solution. We will prove that there can be no polynomial time algorithm solving our maximisation problem in the general case (unless P=NP), in other words, we will prove NP-hardness of our problem. Then we will present an approximation algorithm that uses semidefinite programming. In order to facilitate the reception of our results we will first provide the reader with an introduction to semidefinite programming, before proceeding with the general case.

3.1 Algorithms for special cases

Now we are ready to present algorithms for optimal assignment of y in special instances, which are bipartite graphs, rings, and complete graphs. A bipartite graphs is any graph that is two colourable, which means that it can be coloured with two colours in such a way that there will be no edge linking vertices of the same colour. This is an important class of graphs, which includes, among others, trees and lines. Any subgraph of a bipartite graph is also bipartite. A ring is a connected, undirected graph, where every vertex has degree equal to 2. A complete graph is a connected, undirected graph in which every pair of vertices is connected by an unique edge, so if the number of vertices of such graph is n, then the degree of each vertex is equal to $n-1$. In all presented instances the size of the graph (n)

is arbitrary and so is the distribution of conformists and non-conformists.

3.1.1 Algorithm (1)

Algorithm (1) is a local greedy algorithm which maximises utility of agents in bipartite graphs or rings. In this algorithm the non-conformists take decisions one by one, individually and locally, only by observing the choices of their neighbours. It is very close to a diffusion process. We will prove that such individual, greedy decisions of the agents lead to maximisation of social utility in bipartite graphs. First, let us present the algorithm.

Algorithm 1 Social optimum in bipartite graphs and rings

1. Pick one vertex arbitrarily.
2. Assign to this vertex a choice of $y = +1$ or $y = -1$, e.g., by flipping a coin.
3. **While** there exists an unactivated vertex v in the network with an activated neighbour **do:**
   - **If** v is a conformist
     then assign to v the same value of y you chose for the first vertex in the network.
   - **If** v is a non-conformist
     then assign to v the opposite value to the one assigned to her already activated neighbour.

Theorem 1. *Algorithm (1) maximises social utility of the network if it is bipartite or a ring.*

The intuition for the algorithm is simple. The subgraph of conformists is known to be optimal if all of them choose same value of y. Then, a non-conformist, after observing the choice of his neighbour, and thanks to bipartition property chooses his favourite y. Note that all subgraphs of a bipartite graphs remain bipartite, thus even if we "take away" some arbitrary vertices (conformists), the remaining subgraphs of non-confomists will be also bipartite. The non-conformists choose their favourite value of y one, by one, so that everybody has a chance to observe choices of neighbours and pick their value accordingly, thus providing a solution without conflicts and maximising utility of the network.

Proof. In bipartite graphs, the algorithm assigns choices to vertices so that any two vertices which are connected by an edge make opposite choices if they are both

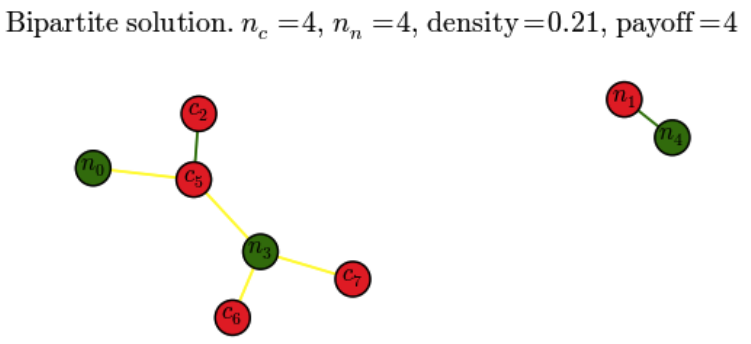

Figure 3: Result of Algorithm (1) on a bipartite graph

non-conformists, and make the same choice if they are both conformists. Thus, the optimality of the algorithm for bipartite graphs follows directly from Lemmas 1, 2, and 3.

For rings that are not bipartite (that is for n odd, and so not two-colourable) where $V = V_n$ the algorithm will result in exactly one unavoidable conflict, i.e., exactly one edge which brings a payoff of -2. □

3.1.2 Simulations of Algorithm (1)

We implement Algorithm (1) on bipartite random graphs. A red vertex stands for a vertex that chose $y = -1$, whereas green vertex stands for a choice of $y = +1$. The letters n and c stand for non-conformists and conformists, respectively. Green edges mean that the edge brings $+2$ to the global payoff, whereas yellow edges mean edges that cancel out (recall lemma 1). In Figure 3 we present an example of a randomly picked bipartite graph of size 8, note that the graph is not fully connected. This fact is not a problem for our algorithm. In Figure 4 the graph is larger ($n = 20$) and is fully connected. After running Algorithm (1) on 100 uniformly randomly picked bipartite graphs we have found no conflicting edges, which is consistent with our theoretical result.

3.1.3 Algorithm (2)

Unlike Algorithm (1), Algorithm (2) for complete graphs does not propagate locally, but is global. It splits the vertices in half in an "authoritarian" way in order

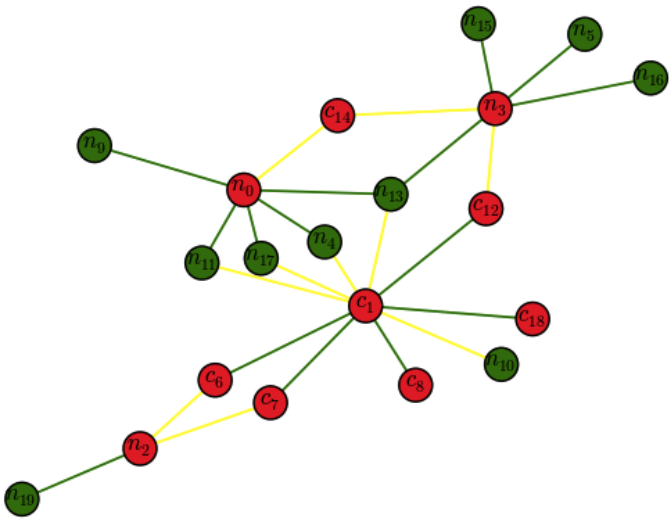

Figure 4: Result of Algorithm (1) on a bipartite graph

to maximise social utility. [1]

Algorithm 2 Social optimum in complete graphs

1. Pick $y = 1$ or $y = -1$ arbitrarily.
2. Assign this value to all the conformists.
3. Split the non-conformist vertices into two equal sets (or two sets differing in size by 1, if the number of non-conformists is odd).
4. Assign to one set the same value as to conformists.
5. Assign to the other set the opposite value.

The intuition for Algorithm (2) is the following. First, from lemma 2 we know that graphs of conformists give optimal payoff when they all choose the same y, and from Lemma 1 that we can forget about edges between conformists and non-conformists. Therefore the only remaining problem is a complete graph of non-conformists. Second, from Lemma 4 we know that we need to split the non-conformists in half to get the optimal payoff. In Step 3 we subtract or not one vertex just to deal with possibly odd number of vertices.

[1] It is plausible that the same result can be obtained with a "local" algorithm following a self avoiding random walk. The equivalence of this solution compared to Algorithm (1) will be analysed in the online appendix.

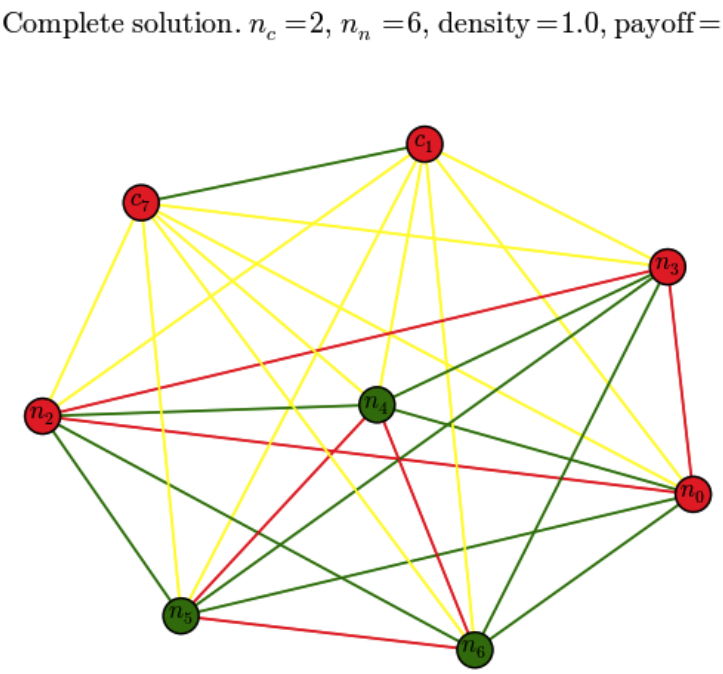

Figure 5: Result of Algorithm 2 on a complete heterogeneous graph

Theorem 2. *If G is complete Algorithm (2) maximises social utility on that graph.*

Proof. The proof follows directly from Lemmas 1, 2 and 4. □

3.1.4 Simulations of Algorithm (2)

Just like in for bipartite graphs, we generate complete graphs with random distribution of conformists and non-conformists. In Figure 5 and 6 we present some solutions delivered by our algorithm. Note that in Figure 6 we get a complete subgraph of non-conformists of odd number of vertices.

3.2 Introduction to General Case: Semidefinite Programming Applied to Max-Cut

Before we proceed with the solution concept for the general case of Multiple Agent Fashion Game, we will present to the reader the technique of semidefinite programming, which will be used in order to approximate the optimal solution in our game.

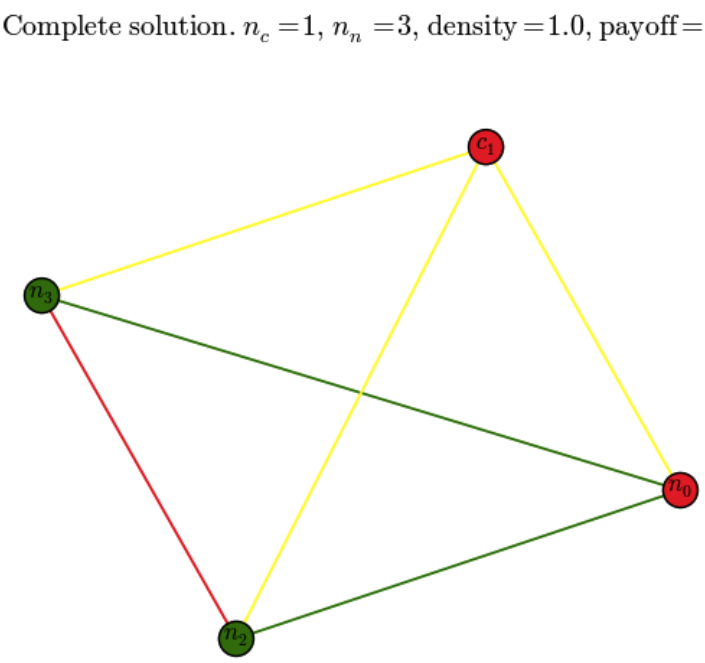

Figure 6: Result of Algorithm 2 on complete heterogeneous graph

This introductions is based on [4][23] and also presents a rounding technique first observed by [10].

Semidefinite programming is a generalisation of linear programming and enables us to set some special non-linear constraints.

Semidefinite programming takes its name from positive semidefinite matrices, which appear as coefficients of the solved problem. A matrix $A \in R^{n \times n}$ is called positive semidefinite ($A \succeq 0$) if it is symmetric and all eigenvalues of A are non-negative.

Any semidefinite program needs to contain the following four components:

- a set of variables y_{ij}
- a linear objective function to maximise (or minimise)
- a set of linear constraints over y_{ij}
- a semidefinite constraint $Y = [y_{ij}] \succeq 0$

The difference between linear and semidefinite programming arises thanks to the last, semidefinite constraint. A classical linear programming gives us constraints

in the form of a hyperplane (with flat faces) which gives the feasible region of the LP. In the case of SDP we have one more constraint (the semidefinite one) which gives us a non-flat face restraining the feasible region of a given SDP. This face (or constraint) can be interpreted as an infinite number of linear constraints.

A well established result in SDP is that we can achieve a $(1+\epsilon)$ approximation to an SDP in time polynomial to n and $1/\epsilon$, where n is the size of the program[4].

Semidefinite programming is equivalent to so-called vector programming, which enables us to work on dot products (equivalent to a scalar product of vectors). The dot product is directly related to the cosine of the angle between two vectors in Euclidean space. Thus, in the Euclidean space we have that for any two vectors

$$\vec{A} \cdot \vec{B} = \sum_i a_i b_i = \|\vec{A}\| \|\vec{B}\| \cos\theta$$

where a_i and b_i represent the i-th coordinate of vectors $\vec{A}$ and $\vec{B}$ in Euclidean space, and θ is the angle between the vectors.

Semidefinite programming can be applied as a method of solving integer programming problems, which cannot be properly approximated by (relaxed to) by a set of linear constraints. An illustration of such use of semidefinite programming is the Max-Cut problem, where one wants to find a subset of vertices $S \subseteq V$ of the graph in order to maximise the number of edges between S and the complementary subset $V \setminus S$. First, we need to write a problem as an integer quadratic program for each pair i and j. Let $y = -1$ stand for the fact that vertex i is on the left of the cut and $y = 1$ if the vertex falls on the right of the cut.

$$\text{Maximise} \sum_{\{i,j\} \in E} \frac{(1 - y_i \cdot y_j)}{2}$$

subject to

$$\forall i \in V : \; y_i \in \{-1, 1\}$$

Note that $y_i \cdot y_j$ is equal to -1 if the edge between the two vertices i, j crosses the cut, and 1 otherwise. Thus, the term $\frac{(1-y_i \cdot y_j)}{2}$ contributes 1 to the maximisation if the edge crosses the cut, and 0 otherwise. The maximised expression precisely describes the number of edges crossing the cut.

The above constraints define a quadratic problem, i.e., there appear products of two variables y, with y constrained to a small set of integers. Now we want to

relax the quadratic integer program and get a vector program, which is a linear program over dot products. A relaxation means that we will allow the variables y to belong to a larger vector space. Thus, vector programs have a larger set of feasible solutions than the first one, and so include in particular the integral solutions offered by the integer program. This guarantees that the solution to our vector program will not be worse than the solution to the original integer one. Now, we will replace the integers in our program by n-dimensional real vectors. For the Max-Cut problem, we use unit-length vectors $\vec{g_i}$ on the n-dimensional unit sphere.

$$\text{Maximise} \sum_{\{i,j\}\in E} \frac{(1 - \vec{g_i} \cdot \vec{g_j})}{2}$$
$$\text{subject to} : \forall_{i\in V} : \vec{g_i} \in R^n, \|\vec{g_i}\| = 1$$

We recall that such a vector program can be reformulated as a semidefinite program, and be solved equally efficiently [4].

Observe that by constraining the space of vectors to only two different types of vectors shown schematically in Figure 7, i.e., when any two vectors $\vec{g_i}$, $\vec{g_j}$ are either pointed in the same direction, or in exactly opposite directions, we can represent one of these orientations as -1, the other by 1, and so we return to the original integral formulation with $y \in \{-1, 1\}$. The beauty of our relaxation allows the vectors to form any angle in the unit sphere. So, we have for the optimal solutions:

$$\text{Optimal Vector} \geq \text{Optimal Max-Cut}.$$

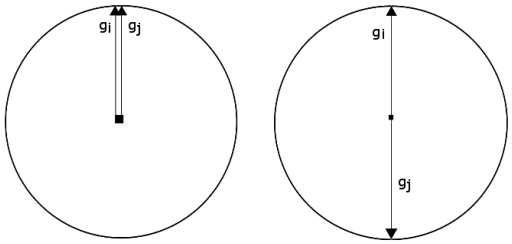

Figure 7: Integers programme

Since all vectors are of length 1, they all lie on the unit sphere around the origin. We now want to round the solution of the vector program back to an integral

solution, such that the total value of our obtained integral solution is within a constant factor $\alpha > 0$ of the optimum vector solution. Thus, we want to have:

$$\text{Rounded Solution} \geq \alpha \text{Optimal Vector.}$$

Combining the above relations will then give us

$$\text{Rounded Solution} \geq \alpha \text{Optimal Max-Cut.}$$

and so we will have solved Max-Cut within a α-factor of optimality.

The way of rounding the vector solution is by choosing a cut of the sphere with a hyperplane, which will place some of the vectors $\vec{g_i}$ above or below it. We will assign a value of $y_i = -1$ to those vertices for which $\vec{g_i}$ lies under the chosen hyperplane, and $y = 1$ for those vertices where $\vec{g_i}$ is over the hyperplane. Our payoff is given by the edges that cross the cut.

In order to establish the cut and choose the hyperplane we draw a vector $\vec{r}$ uniformly at random from the unit sphere. (The hyperplane of the cut is that orthogonal to vector $\vec{r}$.) This vector r is chosen once, and applied for rounding at all vertices. As we mentioned earlier, the dot product of vectors is related to the cosine of the angle θ between them, and is in fact equal to this cosine for unit length vectors. The sign of the product of the random vector $\vec{r}$ with our vectors $\vec{g_i}$ or $\vec{g_j}$ enables us to decide whether a given vector lies above or below the cut.

Formally, for all $i \in V$, we put $y_i = 1$ if $\vec{r} \cdot \vec{g_i} > 0$, and we put $y_i = -1$ if $\vec{r} \cdot \vec{g_i} < 0$. From Figure 8, we see that for a pair of vertices i, j, we have $y_i \neq y_j$ only if vector $\vec{r}$ falls into the dashed area of the sphere. For a uniform sampling of vectors $\vec{r}$, the probability of this happening is proportional to the angle between the hyperplanes orthogonal to vectors $\vec{g_i}$ and $\vec{g_j}$, and so given as $(2\theta_{ij})/(2\pi)$, where θ_{ij} is also the angle between vectors $\vec{g_i}$ and $\vec{g_j}$ ($\vec{g_i} \cdot \vec{g_j} = \cos\theta$).

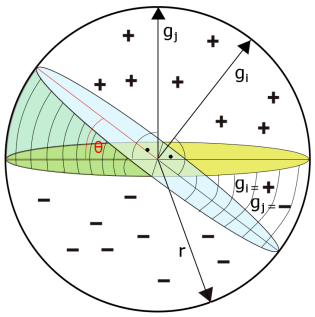

Figure 8: Rounding of the vector program

The coefficient α of approximation of the obtained solution follows from the relation between an angle θ and its cosine. This is discussed in more detail further on.

We now proceed to consider our fashion game problem in general graphs.

3.3 General Case

In this section we will consider optimal and approximated solutions to the Fashion Game. First we will prove the NP-hardness of the problem by showing that if we were able to solve the Multiple Agent Fashion game in general, we would be able to solve Max-Cut problem in general, which is known to be NP-hard. Then we will use semidifinite programming to obtain an approximation algorithm for maximimising the social utility, based on the approach of Goemans & Williamson [10]. The section will conclude with implementation of the algorithm.

3.4 NP-hardness of Multiple Agent Fashion Game for general graphs

Max-Cut is a well-known problem in Computer Science that is known to be NP-hard from the original Karp's paper[17]. We recall that in the Max-Cut problem we want to find a subset of vertices (S) in order to maximise the number of edges between S and the complementary subset.

Theorem 3. *The unweighted Max-Cut problem can be reduced to the Multiple Agent Fashion Game, therefore Multiple Agent Fashion Game is NP-hard in general.*

Proof. Suppose we are asked to solve the Max-Cut problem for some graph G. Consider the Multiple Agent Fashion Game on the same graph, in which all of the vertices are non-conformists ($V_n = V$). For a graph having non-conformists only, the social optimum is given by the payoff $H = -\sum_{\{i,j\}\in E} y_i y_j$ (see proof of Lemma 3). Let S be the vertices who chose $y = 1$. Maximising H boils down directly to maximising the number of edges between the subset of vertices choosing $y = 1$, say S, and those who choose $y = -1$, say $V \setminus S$. If the number of edges in the cut between S and $V \setminus S$ is a, then we associate a payoff of $+2$ with each of these a edges, and a payoff of -2 with each of the other $|E| - a$ edges. Thus, the total

payoff is $H = 4a - 2|E|$. Observe that H is maximised if and only if a is maximised. But a is then the optimal solution to Max-Cut. Since Max-Cut is known to be NP-hard, the Multiple Agent Fashion game is, in general, NP-hard. □

The above theorem means that the optimal solution to Multiple Agent Fashion Game can in the general case only be approximated in reasonable time (unless P=NP).

3.5 Approximated Solution

In general, the Multiple Agent Fashion Game can be reduced to the Max-Cut problem thanks to the removal of edges linking V_c and V_n (Lemma 1. Once we take them away, we are left with possibly multiple (disconnected) networks built by conformists or non-conformists only. From Lemma 2 we know that networks where $V = V_c$ are optimal only when all vertices adopt the same value of choice y. Thus, the parts of the network with conformists can be easily solved, and do not affect the non-conformist part of the network.

In the following, we only need to consider the solution to our problem for networks of non-conformists.

Graphs consisting of non-conformists can be optimally solved only if all the edges in that graph link vertices who have chosen opposite values of y (see Lemma 3), and in general the payoff is maximised when the number of edges between the subset of vertices choosing $y = 1$, say S, and those who choose $y = -1$, say $V_n \setminus S$, is as large as possible.

We observe that in a network of non-conformists, the payoff associated with an edge between vertices $i, j \in V$, which make choices y_i and y_j, is -2 if $y_i = y_j$, and $+2$ if $y_i \neq y_j$. More compactly, we can write this payoff as $-2y_iy_j$.

Note that the above payoff can sometimes be negative. To allow for a reasonable definition of approximation algorithms, we will modify our payoffs associated with edges.

First, recall that our global payoff function was defined as

$$H = \sum_{\{i,j\} \in E} J_{ij} y_i y_j$$

From this we can simply get the contribution of vertex j to the payoff of vertex i (H_{ij}), which takes the following form:

$$H_{ij} = J_{ij} y_i y_j$$

So our global payoff H is equal to the sum of all $H_{ij} \in \{-1, 1\}$

$$H = \sum_{\{i,j\} \in E} H_{ij}$$

Now, we perform the rescaling of our payoff range so that our initial H_{ij} is being augmented by 1

$$H'_{ij} = H_{ij} + 1.$$

Note that $H'_{ij} \in \{0, 2\}$.

We obtain the modified payoff

$$H' = \sum_{\{i,j\} \in E} H'_{ij}$$

The social optimum for the Fashion Game with payoffs H' is of the same form as for the Fashion Game with payoffs H.

Now, for the modified payoff function, the payoff associated with an edge between vertices $i, j \in V$, which make choices y_i and y_j, is 0 if $y_i = y_j$, and +4 if $y_i \neq y_j$. More compactly, we can write this payoff as $4(1 - y_i y_j)$.

We have the following integer quadratic program for each pair i and j, where $i, j \in V_n$

$$\text{Maximise} \sum_{\{i,j\} \in E} 4(1 - y_i \cdot y_j)$$

The above program resembles the one which appears when solving Max-Cut (see the previous subsection). Since solving this problem is NP-hard, we will as in the case of Max-Cut relax some assumptions of it, by replacing y_i with a multidimensional vector $\vec{g_i}$ belonging to a unit-sphere S_n. The relaxation (P) will take the form:

$$\begin{aligned} &\text{Maximise} \sum_{\{i,j\} \in E} 4(1 - \vec{g_i} \cdot \vec{g_j}) \\ &\text{subject to} : \vec{g_i} \in S_n, \forall i \in V_n \end{aligned} \tag{1}$$

Now we are ready to present our general approximation algorithm for the Fashion Game problem.

Algorithm 3 Social optimum in general

1. Assign to all $i \in V_c$ same value of y (choose $y = 1$ or $y = -1$ for all).
2. On the subgraph induced by non-conformists (V_n), solve (P) using semidefinite programming, to obtain a $(1 - \epsilon)$-approximation of the optimal set of vectors $\vec{g_i}$ (for arbitrarily small chosen $\epsilon > 0$)
3. Let $\vec{r}$ be a vector picked uniformly at random from the unit sphere S_n.
4. Set $S = \{i | \vec{g_i} \cdot r \geq 0\}$.
5. Put $y = 1$ for all non-conformists belonging to S, and $y = -1$ for all non-conformists belonging to $V \setminus S$.

We denote the payoffs of subgraphs of conformists by c. The set of edges in a graph where $V = V_c$ is equal to the the sum of interaction variable J in that graph. We need to multiply it by 4 in order to make it coherent with our rescaling.

$$c = 4 \sum_{i,j \in V_c, \{i,j\} \in E} J_{ij}^c = 4\{\{i,j\} \in E | i,j \in V_c\}$$

The value brought by conformists to the global payoff is null if there were no conformists in initial network. By H' we denote the global payoff of graph G after rescaling. However, for the sake of computational simplicity we will operate on A, which will denote the payoff of subgraph built by non-conformists only, and then we will simply add c, the payoff arising from the subgraph of conformists, as a deterministic component. $E[A]$ is the expectation of the payoff of the non-conformist graph. Following the analysis outlined in the previous subsection, we obtain the following

Claim 1.

$$E[A] = \sum_{i,j \in V_n, \{i,j\} \in E} 4 \frac{\arccos(\vec{g_i} \cdot \vec{g_j})}{\pi} \tag{2}$$

The proof that follows is given in Goemans & Williamson paper[10]. We are adapting it for our rescaled case and present it for the sake of completeness.

Proof. Given a vector $\vec{r}$ drawn uniformly from the unit sphere S_n, the contribution of the edge $\{i,j\}$ to the payoff H' is 4 with probability $[sgn(\vec{g_i} \cdot \vec{r}) \neq sgn(g_j \cdot \vec{r})]$, and 0 otherwise. (Here, $sgn(x) = 1$ if $x \geq 0$, and -1 otherwise.) Hence, the expected value of this contribution is $4[sgn(\vec{g_i} \cdot \vec{r}) \neq sgn(g_j \cdot \vec{r})]$. We have by linearity of expectations that

$$E[A] = \sum_{i,j \in V_n, \{i,j\} \in E} 4Pr[sgn(\vec{g_i} \cdot \vec{r}) \neq sgn(g_j \cdot \vec{r})]$$

Next, we have

$$Pr[sgn(\vec{g_i} \cdot \vec{r}) \neq sgn(\vec{g_j} \cdot \vec{r})] = \frac{\arccos(\vec{g_i} \cdot \vec{g_j})}{\pi}$$

which follows from the fact that the probability the random hyperplane separates the two vectors is given as θ/π, where θ is the angle between the two vectors (see the previous subsection, Figure 8), and

$$\vec{g_i} \cdot \vec{g_j} = \cos\theta$$

$$\theta = \arccos(\vec{g_i} \cdot \vec{g_j})$$

Introducing the above expression for $Pr[sgn(\vec{g_i} \cdot \vec{r}) \neq sgn(\vec{g_j} \cdot \vec{r})]$ into the sum gives the claim. □

Now we will show that the algorithm gives a performance guarantee. Compare the expressions for the payoff $E[A]$ achieved by the algorithm in expectation (2), and the upper bound on the payoff given by (1). Let $Z_P^* = \sum_{i,j \in V_n, \{i,j\} \in E} 4(1 - \vec{g_i} \cdot \vec{g_j})$. Comparing the elements of the sum in Z_P^* and $E[A]$, we have the following

Claim 2.

$$E[A] \geq \alpha \cdot Z_P^*$$

where

$$\alpha = min_{0 \leq \theta \leq \pi} \frac{2}{\pi} \frac{\theta}{1 - \cos\theta}$$

It remains to bound the value of α

Lemma 5.

$$\alpha > 0.87856$$

Proof. With a use of simple calculus we can find that α achieves its value for $\theta = 2.331122...$, the nonzero root of $\cos\theta + \theta \sin\theta = 1$. To formally prove the bound of 0.87856, we first observe that

$$\frac{2}{\pi} \frac{\theta}{1 - \cos\theta} \geq 1$$

$$\text{for } 0 < \theta \leq \pi/2$$

The concavity of

$$f(\theta) = 1 - \cos\theta \quad \text{in the range} \quad \pi/2 \leq \theta \leq \pi$$

implies that for any θ_0 we have

$$f(\theta) \leq f(\theta_0) + (\theta - \theta_0) f'(\theta) \text{ or}$$

$$1 - \cos\theta \leq 1 - \cos\theta_0 + (\theta - \theta_0)\sin\theta_0 = \theta\sin\theta_0 + (1 - \cos\theta_0 - \theta_0\sin\theta_0)$$

Choosing $\theta_0 = 2.331122$ for which $1-\cos\theta_0-\theta_0\sin\theta_0 < 0$, we derive that $1-\cos\theta < \theta\sin\theta_0$, implying that

$$\alpha > \frac{2}{\pi\sin\theta_0} > 0.87856$$

□

Hence, following the methodology of Goemans & Williamson [10], we have obtained that the approximation ratio of the semidefinite programming approach for our payoff function is $\alpha > 0.878$.

Theorem 4. *Algorithm (3) gives in expectation at least 0.878 of optimal global payoff H'*

Proof. Since $H' = A + c$, from Lemma 2 we know that subgraphs of conformists are necessary optimal if we assign to all the vertices same value of y, and from lemma 1 we know that all the edges between conformists and non-conformists can be omitted, our worst case scenario is when $c = 0$, so the payoff is in expectation $(1-\epsilon)A$. (Note that we do not solve semidefinite programming quickly and exactly, but can apply an arbitrarily good approximation.) Therefore approximation of A gives us the lower bound on H', which is $\alpha(1 - \epsilon) > 0.878$, for a sufficiently small chosen ϵ. □

3.6 Simulations

In this section we will present simulations of performance of our approximated algorithm alone, and compared to the two algorithms for special instances. All the codes were written in Python and are available online on the website [35]. The applied solver was CVXOPT, which provided a good approximation for semidefinite programming ($\epsilon < 10^{-6}$). We encourage the reader to play with the parameters and to compare different algorithms.

In Figure 9 we present a randomly chosen graph with $n = 10$, on which we have run our approximation algorithm. The colours of the nodes stand for values of

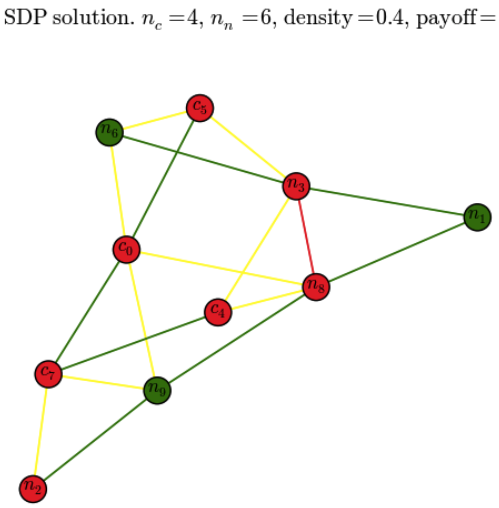

Figure 9: Example: Algorithm (3) run on a random network

choice variable, red means that $y = -1$ and green means that $y = 1$. The edges are coloured depending on the payoff (H) they bring, green edges bring payoff equal to +2, yellow equal to 0, and red signifies conflicts (payoff=-2). Note that all the payoffs returned by our program are the non-rescaled ones. (Rescaling was only purposeful for computing approximation.)

To perform tests, we have implemented the approximated algorithm on random graphs of $n = 100$. We have run it a thousand times for four different specifications, the results are presented in Figures 10, 11, 12, and 13.

In each of the specifications we have defined the number of vertices $n = 100$. We also defined either the number of conformists (in Figures 10, 11) or the probability with which a vertex is a conformist (in Figures 12, 13), and the density of the graph (the number of edges relative to the complete graph — 0 if the graph is empty, 1 if it is complete). We also set the number of iterations (different graphs for which performed the tests), and the number of different allocations of conformists/non-conformists which we chose randomly and tested for each graph. We considered either 1 such allocation per graph, or 1000 different allocations per graph, depending on the specification. For each specification we report the parameters and the payoffs on the histograms.

In Figures 10 and 12 we draw only one graph, on which we pick the distribution of conformists and non-conformists 1000 times. The difference between the two is

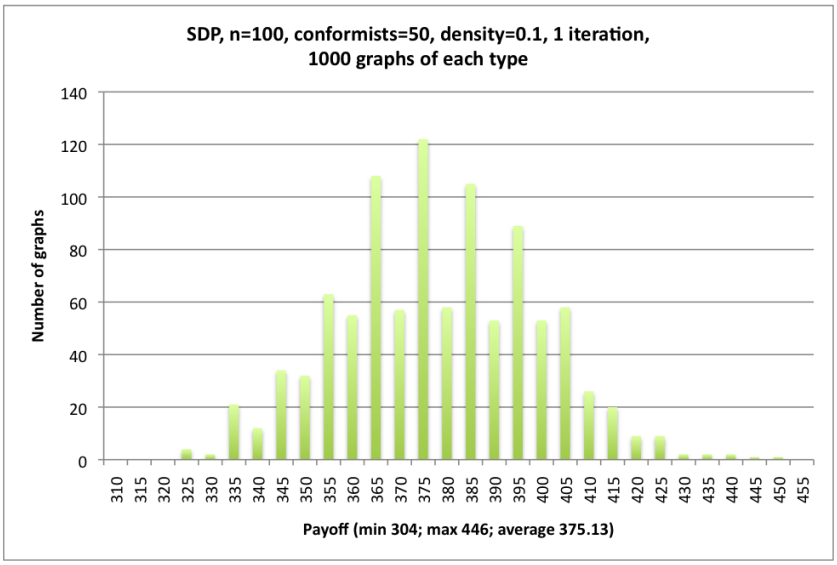

Figure 10: Algorithm (3)'s performance on one graph with 1000 distributions of conformists, deterministic

the fact that in the case of Figure 10 we have a fixed number of conformists $(n/2)$, and in Figure 12 every vertex can be a conformist with probability equal to 0.5. The spread between the maximum and the minimum payoff is much larger in the randomised specification (equal to 270) than in the deterministic one (spread equal to 142), whereas the average payoffs are relatively close, with only 4.85 difference (we recall that $n = 100$).

In Figures 11 and 13 we draw 1000 random graphs, each with one allocation of conformists and non-conformists. In Figure 11 we have a fixed number of conformists $(n/2)$, and in Figure 13 every vertex can be a conformist with probability equal to 0.5. Once again we can observe larger spread in the case of the randomised specification of the number of conformists, but the average payoffs are not that close.

The randomised approximated algorithm does very well for bipartite graphs. We have run hundred iterations on graphs of $n = 100$ with randomised specifications, and in each case the payoff given by the randomised algorithm was equal to the payoff given by Algorithm (1) on the very same graph. This implies that in our simulations the randomised approximated algorithm always delivers the optimal solution for bipartite graphs, which is very reassuring. The comparison can be seen on 15 and 14, 17 and 16. The colouring is always different, but leads to the

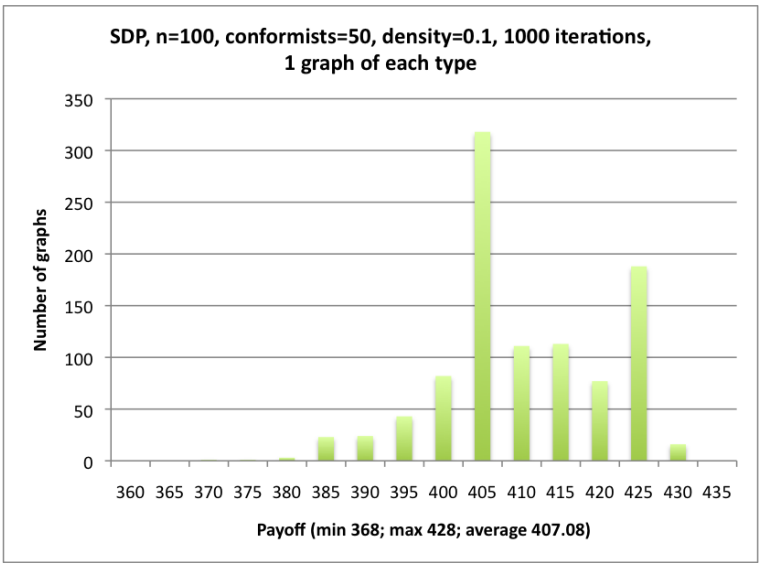

Figure 11: Algorithm (3)'s performance on 1000 random graphs, deterministc

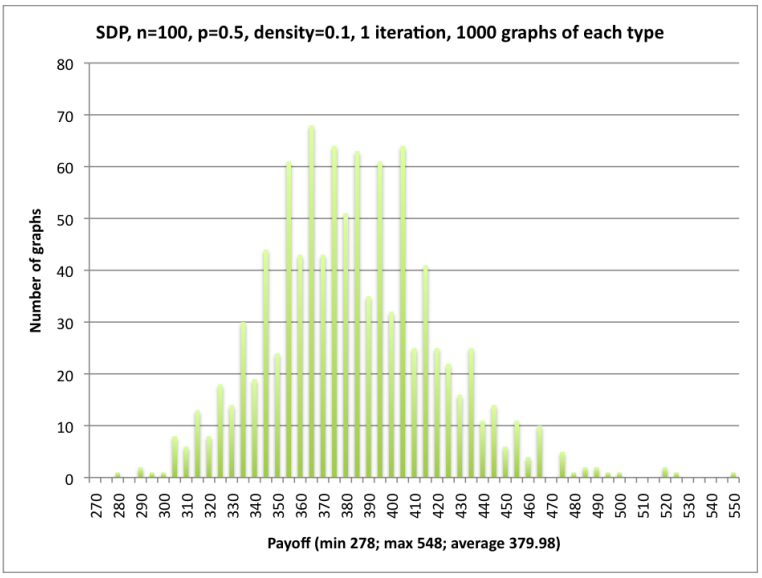

Figure 12: Algorithm (3)'s performance on one graph with 1000 distributions of conformists, random

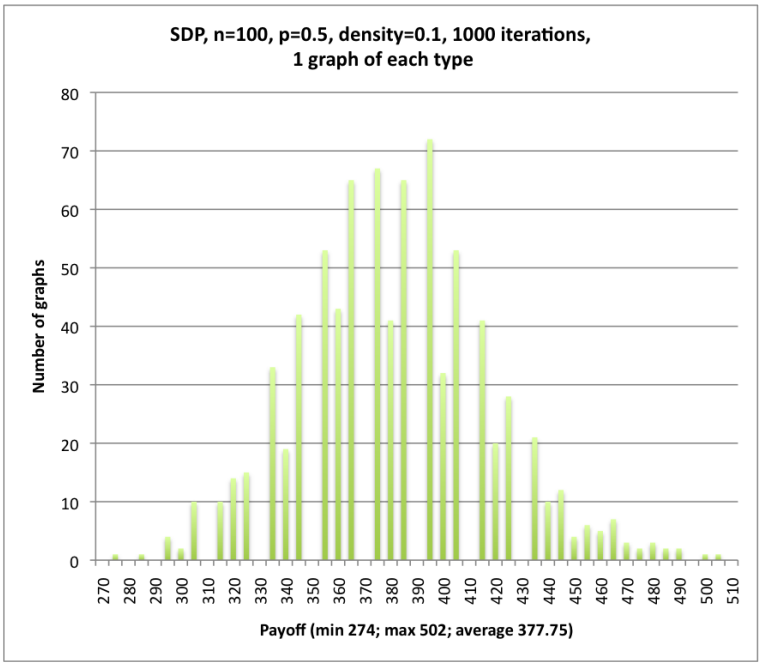

Figure 13: Algorithm (3)'s performance on one graph with 1000 distributions of conformists, random

same payoff.

In the case of complete graphs the randomised approximation algorithm does worse than for the bipartite graphs (sometimes achieving suboptimal solutions), but still very well on average. Just like for bipartite graphs, we have generated 100 graphs on which we have run both the randomised algorithm and Algorithm (2). Note that throughout the simulation we operate on our original payoff range H, and not on the rescaled H'. This means that if we take the ratio of the randomised solution over the optimal solution we are likely do deal with some negative terms, which will spoil our ratio, making it lower than it should be. However, out of curiosity, we have computed the average payoff for 100 graphs solved with Algorithm (3), and 100 respective graphs solved with Algorithm (2). The ratio of the two is equal to 0.988. Given that this result is underestimated, it follows that our approximation algorithm does on average very well for complete graphs. In some rare cases, however, it can fail to deliver the optimal solution, just like presented in figures 18 and 19, but please note that the algorithm is randomised. A positive example, where payoffs delivered by both algorithms are equal is shown in figures 20 and 21.

Bipartite solution. $n_c = 6$, $n_n = 4$, density $= 0.29$, payoff $= 18$

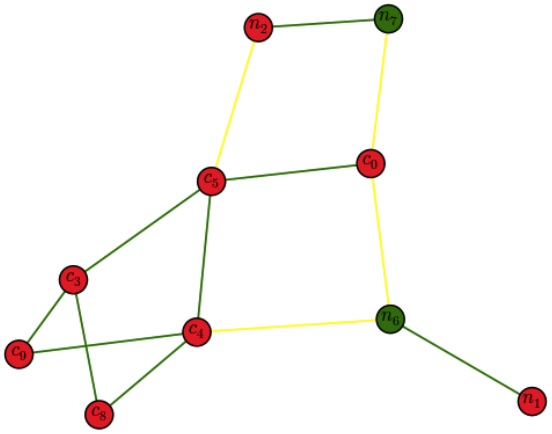

Figure 14: Algorithm (1)

SDP solution. $n_c = 6$, $n_n = 4$, density $= 0.29$, payoff $= 18$

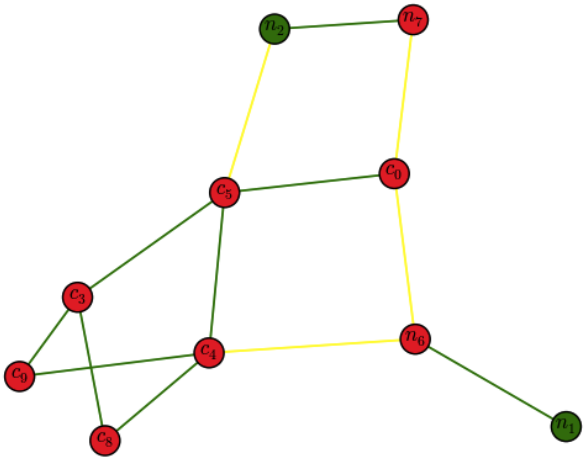

Figure 15: Algorithm (3)

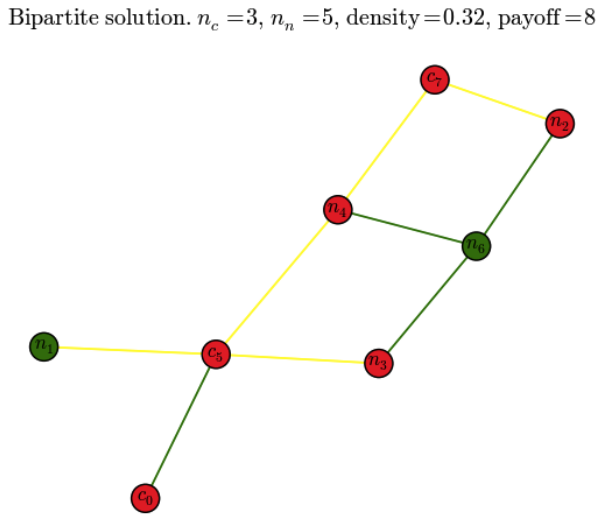

Figure 16: Algorithm (1)

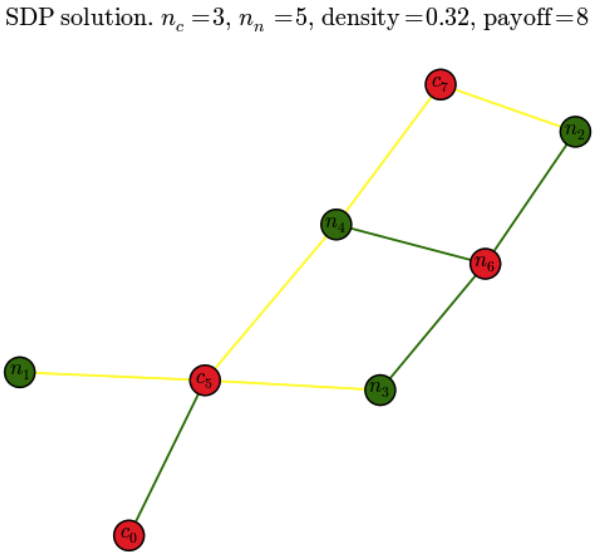

Figure 17: Algorithm (3)

Complete solution. $n_c = 2$, $n_n = 4$, density $= 1.0$, payoff $= 6$

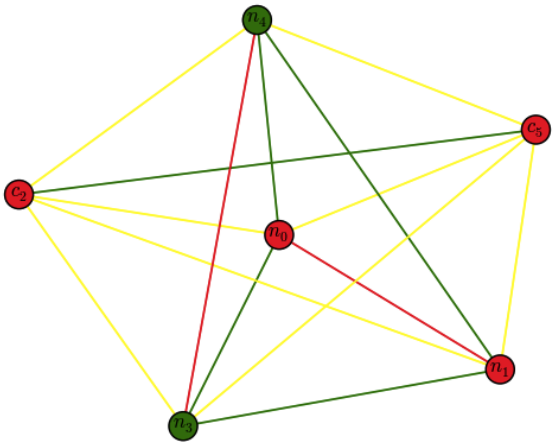

Figure 18: Algorithm (2)

SDP solution. $n_c = 2$, $n_n = 4$, density $= 1.0$, payoff $= 2$

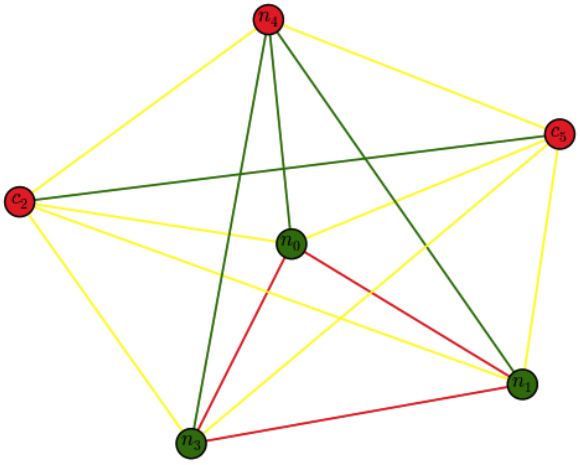

Figure 19: Algorithm (3)

Complete solution. $n_c = 3$, $n_n = 3$, density $= 1.0$, payoff $= 8$

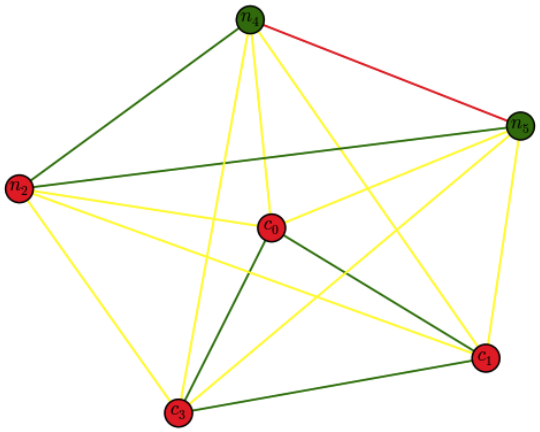

Figure 20: Algorithm (2)

SDP solution. $n_c = 3$, $n_n = 3$, density $= 1.0$, payoff $= 8$

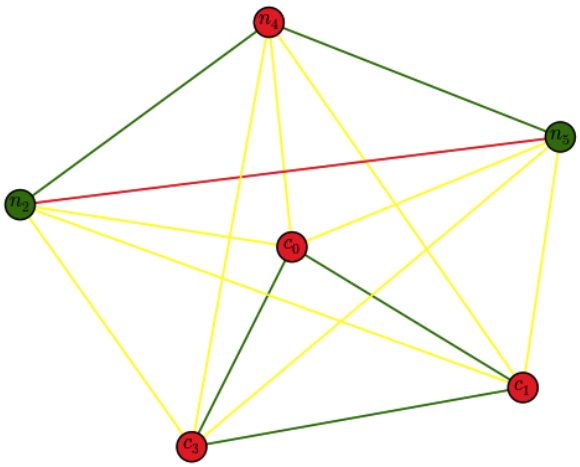

Figure 21: Algorithm (3)

4 Auxiliary Results: Nash Equilibria of the Multiple Agent Fahion Game

Nash's concept of equilibrium in a non-cooperative setting is probably the most important and widely used solution concept in game theory. A Nash equilibrium is a set of strategies in which each player has no incentive to deviate from his strategy, given the strategies of other players. In the network setting the topology of the underlying network defines who interacts with whom. Not all the agents interact with each other directly, but they can be affected indirectly by the choices made by friends of their friends etc. We consider pure Nash equilibria only.

4.1 Pure Nash Equilibria on trees

Our game, in general, does not guarantee the Existence of a pure Nash Equilibrium. For instance we can see the example from the introduction, in which two ladies (one conformists and one non-conformist) choose dresses as a path of size 2, which has no pure Nash Equilibrium. We attempt to characterise for which graphs an equilibrium exists, considering the special case of trees.

Trees are a special class of graphs which has a crucial property of not allowing for any cycles in the graph. This property makes possibile induction from the leaves to the origin of the network, which in case of different network structures would be hard, or even impossible. In this section we will exploit the properties of trees to present a simple algorithm that allows us to prove the existence of a pure Nash equilibrium on an arbitrary tree in the Multiple-Agent Fashion game context. The algorithm can only prove of disprove existence of a pure Nash equilibrium on a given tree. It does not give us any insights about it's unicity.

Framework: In the Multiple-Agent Fashion game we have two kinds of vertices: conformists and non-conformists who differ in terms of interaction variable $J \in \{-1, 1\}$. This gives us 2^n possible networks of size n(configurations), or 2^{n-1}, if we exclude symmetric solutions. As a reminder, n is the total number of vertices in a graph (here it is a tree).

We look at the nature of edges linking each pair of vertices. They can be either *same*, linking vertices that choose the same value of y (i.e. $y_i = 1$ and $y_j = 1$, or

$y_i = -1$ and $y_j = -1$) or they can be *different*, linking vertices of opposite values (i.e. $y_i = 1$ and $y_j = -1$).

We consider a vertex *stable* if at least a half of the edges originating from her is: *same*, for conformists, and *different*, for non-conformists. The game reaches a pure Nash equilibrium if none of the vertices has an incentive to deviate, so to say that all the vertices are stable. Our algorithm, which has to be understood more as a method, enables us to verify whether each individual vertex can ever be stable, given the topology and distribution of conformists and non-conformists.

Our solution is based on induction an recurrence. Intuitively, we will run the algorithm from the leaves of a tree and then reccurently it will climb up to the origin, folding, at every level, the results from previous one. More formally, we will look at the subtrees $T(a)$ of a given tree T, where a is the root vertex of a subtree (the vertex that links $T(a)$ to T). The tree T will be decomposed in a following manner. First, we will look at the leaves only (the vertices of degree equal to 1), then we will "go up" to their parents, then parents of their parents etc. until we reach the origin of the network. So first a will be just a leaf, then it will be the parent of the initial leaf and so on.

We will check for possible stability of the vertices with a use of predicates:

- $N1(T(a))$ - The sub-tree is at pure Nash equilibrium if the vertex above the vertex a adopts same value of y as a did (the edge linking a to the rest of the graph is *same*)
- $N2(T(a))$ - The sub-tree is stable if the vertex above the vertex a chooses opposite to a value of y (the edge linking a to the rest of the graph is *different*)

First, consider a case in which a is just a leaf.

- $N1(T(a))$
  - True if a is a conformist
  - False if a is a non-conformist
- $N2(T(a))$
  - True if a is a non-conformist
  - False if a is a conformist

Next, consider that $T(a)$ is no longer a single leaf, that a has x children and that:

- $x1$ of her children report that $N1$ is true for them, but $N2$ is false
- $x2$ of her children report that $N1$ is false for them, but $N2$ is true
- $x3$ of her children report that both $N1$ and $N2$ are true for them
- $x4$ of her children report that both $N1$ and $N2$ are false for them

Now, if a is a conformist:

- $N1(T(a)) = True$ iff $x4 = 0$ and $x1 + x3 + 1 \geq (1/2)(x + 1)$
- $N2(T(a)) = True$ iff $x4 = 0$ and $x1 + x3 \geq (1/2)(x + 1)$

If a is an non-conformist:

- $N1(T(a)) = True$ iff $x4 = 0$ and $x2 + x3 \geq (1/2)(x + 1)$
- $N2(T(a)) = True$ iff $x4 = 0$ and $x2 + x3 + 1 \geq (1/2)(x + 1)$

Now "climb up the tree", when you reach the root of T we need a new predicate.

- $N3(T(a))$ - The graph is stable

If the root is a conformist:

- $N3(T(a)) = True$ iff $x4 = 0$ and $x1 + x3 \geq (1/2)x$

If the root is a non-conformist:

- $N3(T(a)) = True$ iff $x4 = 0$ and $x2 + x3 \geq (1/2)x$

There is no Nash equilibrium on a given tree if at any stage the answer to the predicated $N1$ and $N2$, or to the predicate $N3$ is "false".

We thus have the following theorem.

Theorem 5. *There exists a fast procedure to decide if a given tree, with an arrangement of conformists and non-conformists, admits a (at least one) Nash equilibrium.* □

4.2 Line: the Best and the Worst Possible Nash Equilibria

In this section we will prove that the worst possible Nash Equilibrium in Multiple Agent Fashion Game played on a line gives payoff equal to 2. Then, thanks to proven optimality of solution delivered by Algorithm (1), we will be able to analyse the Price of Anarchy on the line and conclude that it is decreasing with the number of conformist-non-conformist edges.

As mentioned in the introduction, the Price of Anarchy (PoA) is a concept first proposed by Elias Koutsoupias and Christos Papadimitriou [20]. In a system in which noncooperative agents share a common resource, they proposed the ratio between the worst possible Nash equilibrium and the social optimum as a measure of the effectiveness of the system.

Before we present our theorem we need the following lemma:

Lemma 6. *The worst possible payoff of a line at equilibrium in Multiple Agent Fashion Game is equal to* 2*.*

Proof. Every line consists of exactly two vertices of degree 1 and $n-2$ vertices of degree 2. In order to remain stable the vertices of degree 2 need to have at least 0 payoff, whereas the vertices of degree 2 need to have a payoff equal to 1. Since there are two vertices of degree 1 the worse possible equilibrium payoff of a line is equal to 2 and the lemma holds. □

Now remember that in our game there are conformists and non-conformist vertices. Thanks to the lemma 1 we know that any vertex between conformist and non-conformist vertices brings 0 to social utility no matter the choices of the vertices. We will keep the notions of *different* and *same* edges to describe edges linking v_c with v_n, and edges linking v_c with v_c or v_n with v_n respectively. From theorem 1 we also know that the solution delivered by Algorithm (1) is the social optimum for bipartite graphs (line is bipartite). In order to compute the Price of Anarchy we need to take the ratio of the worst possible Nash equilibrium and the social optimum.

Theorem 6. *For a given path, depending on the allocation of conformists and non-conformists, Price of Anarchy is decreasing in number of edges that are* different*, and increasing in number of edges that are* same*.*

Proof. The fact that PoA is decreasing in the number of *different* edges follows

directly from lemma 1, these edges always cancell out in the social optimum, therefore diminishing it and making the PoA smaller. The edges that are *same* however, always bring $+2$ to social optimum, but can bring even -2 to the "anarchy" equilibrium payoff, which makes the PoA bigger. □

5 Conclusions

In the present paper we have developed an original model of the Multiple Agent Fashion Game, in which agents get to choose between two different products. The properties of the game capture conformity and non-conformity interactions between agents distributed on a network. In our paper we asked the following question: how to maximise social utility in such game? In order to answer this question we studied carefully special classes of graphs and developed two efficient algorithms for computing the value of social optimum for them.

Algorithm (1) is a local greedy algorithm for computing social optimum in bipartite graphs and rings. The fact that it is local is a very nice feature, because it captures, at least to some extend, real life diffusion process and individual, selfish behaviour of the agents. We proved formally optimality of the solution delivered by Algorithm (1) and we perform simulations on randomly picked bipartite graphs.

Unlike Algorithm (1), our Algorithm (2) is global and delivers the optimal solutions in an "authoritarian" way. We have proven that the solution delivered by Algorithm (2) is always optimal for any complete graph, with distribution of conformists and non-conformists randomly picked. Just like for Algorithm (1) we performed simulations and found them consistent with our theoretical result.

In order to solve the problem in general we needed to define complexity of our problem. We found that our problem is NP-hard. We proved the complexity by showing that if we were able to solve the Multiple Agent Fashion Game, we would be able to solve the Max-Cut problem, which would imply that P=NP. Armed with the knowledge that it is impossible to solve our maximisation problem in polynomial time we developed a randomised approximation algorithm, which uses semidefinite programming in order to approximate tightly the value of social optimum in general case. In the first part of our Algorithm (3) we deal with conformists vertices by assigning to all of them the same value of choice, which, as we have proven, guarantees optimality of such subgraphs. This allowed us to

reduce the problem to the Max-Cut problems in following steps. In the second part of our Algorithm (3) we use semidefinite programming in order to establish the best cut and maximise utility of non-conformists agents. In order to do that we adapted Goemans and Williamson's approximation algorithm to our game and to the rescaled payoff range. We proved the lower bound on the expected rescaled payoff to be equal to 0.878.

We implemented all the algorithms on Python and performed comparative simulations on large number of randomly picked graphs. We found that for bipartite graphs Algorithm (3) always delivers the optimal solution indicated by Algorithm (1). In the case of complete graphs we found some rare discrepancies between the solutions delivered by Algorithms (2) and (3). In some cases our approximation algorithm failed to deliver the optimal solution for complete graphs. However on average it performed very well and close to the solutions delivered by Algorithm (2).

We provided also auxiliary results, which include an algorithm for proving existence of a Nash equilibrium in an arbitrary tree and we proved a theorem stating that the Price of Anarchy for a given line, depending on the allocation of conformists and non-conformists, is decreasing in number of edges between conformists and non-conformists, and increasing in number of edges between conformists and conformists or non-conformists and non-conformists.

Given that our social optimum is probably equivalent to profit maximisation of firms delivering products to the luxury market, the present paper is a starting point for models of optimal scheduling of a launch of products in luxury and an important tool for analysis of customers of the luxury sector. The first step for future research would be analyse local algorithms for maximising social utility, try to show their equivalence to algorithms whose solutions are proven to be a social optimum. If the local algorithms cannot reach the social optimum, how close can we get to that? These questions are especially interesting from the economic perspective, because the local algorithms will show the ways in which firms can control diffusion of their products on the market, while maximising their profits. An avenue for future research would be to develop a dynamic model, where agents can change their decision after observing choices of their neighbours. In the present paper we characterise the social optima, leaving full characterisation of the pure Nash equilibria of the game for future studies.

References

[1] C. Ballester, A. Calvo-Armengol, Y. Zenou, Who's who in networks. Wanted: the key player, Econometrica, 2006

[2] C. Berry, The idea of luxury: a conceptual and historical investigation, Cambridge University Pres, 1994

[3] A. Calvo-Armengol, E. Patacchini, Y. Zenou, Peer effects and social networks in education, Review of Economic Studies, 2009.

[4] S. Chawla, Semidefinite Programming, Advanced Algorithms Lecture Materials, 2013

[5] T. H. Cormen et al., Introduction to Algorithms, MIT Press, 2009

[6] D.-Z. Du, P.M. Pardalos, Handbook of Combinatorial Optimization, Springer 1998

[7] S. Durlauf, P. Young, Social Dynamics, Brookings Institution Press, 2001

[8] D. Easley, J. Kleinberg, Networks, Crowds and Markets, Cambridge University Press, 2010

[9] K. Georgiou, G. Karakostas, J. Koenemann, Z. Stamirowska, Social Exchange Networks With Distant Bargaining, COCOON 2013

[10] M. Goemans, D. Williamson, Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming, Journal of the ACM (JACM), 1995

[11] I. Grilo, O. Shy, JF. Thisse, Price competition when consumer behavior is characterized by conformity or vanity, Journal of Public Economics, 2001

[12] F.A. von Hayek, Individualism and Economic Order, 1948, page 6

[13] S. Hill, F. Provost, C. Volinsky, Network Based Marketing: Identifying Likely Adopters via Consumer Networks, 2006

[14] E. Ising, Beitrag zur Theorie des Ferromagnetismus, Z. Phys., 31 (1925) pp. 253–258

[15] M. Jackson, Social and Economic Networks, Princeton University Press, 2008

[16] M. Jackson, Y. Zenou, Games on Networks, Handbook of Game Theory Vol. 4, Amsterdam: Elsevier Publisher, forthcoming.

[17] R. Karp, Reducibility among combinatorial problems, Complexity of Computer Computations, 1972

[18] D. Kempe, J. Kleinberg, E. Tardos. Maximizing the Spread of Influence through a Social Network. Proc. 9th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining, 2003

[19] D. Kempe, J. Kleinberg, E. Tardos. Influential Nodes in a Diffusion Model for Social Networks. Proc. 32nd International Colloquium on Automata, Languages and Programming (ICALP), 2005.

[20] E. Koutsoupias and C. H., Worst-case equilibria, STACS 1999

[21] K. Kulakowski, Cellular Automata, OEN AGH, Krakow 2000

[22] C. Laciana, S. Rovere, Ising-like agent-based technology diffusion model: Adoption patterns vs. seeding strategies, Physica A, 2010

[23] M. Laurent, F. Rendl, Semidefinite Programming and Integer Programming, 2002

[24] Padgett J. F., Ansell, Christopher K., Robust Action and the Rise of the Medici, 1400-1434, The American Journal of Sociology 1998

[25] A. Palmer, N. Koenig-Lewis, An experiential, social network-based approach to direct marketing, Direct Marketing: An International Journal, 2009

[26] E. Patacchini, Y. Zenou, Juvenile deliquency and conformism, Journal of Law, Economics, and Organization 28, 1-31, 2012

[27] L.S. Shapley, Some Topics in Two-Person Games, Rand Corporation, 1964

[28] O. Shy, Dynamic models of religious conformity and conversion: Theory and calibrations, European Economic Review 2007

[29] K. Sznajd-Weron, J. Sznajd, Opinion evolution in closed community, Int. J. Mod. Phys. C 11, No.6 2000

[30] R.J. Wilson, Introduction to Graph Theory, 4th edition, Prentice Hall, 1998

[31] P. Young, The economics of convention, Journal of Economic Perspectives, 1996

[32] P. Young, Individual Strategy and Social Structure. An Evolutionary Theory of Institutions, Princeton University Press, 1998

[33] Facebook advertising. Consulted on 20/05/2013. https://www.facebook.com/advertising/how-it-works

[34] Sociology Dictionnary. Consulted on 20/05/2013. http://sociology.socialsciencedictionary.com/Sociology-Dictionary/CONFORMITY

[35] Online Appendix https://sites.google.com/site/zuzannastamirowska