Calculating the Price of Anarchy for Network Formation Games

Shaun Lichter, Christopher Griffin, and Terry Friesz

Abstract

There has been recent interest in showing that real networks, designed via optimization [7], may possess topological properties similar to those investigated by the Network Science community [2], [17], [6], [1]. This suggests that the Network Science community’s view that topological properties such as scale-freeness are not the result of some immutable physical laws, but in fact intentional optimization. In [13], it was shown that stable graphs with an arbitrary degree sequence may result from a stability point of a collaborative game. In this paper, we present an integer program (IP) whose solutions yield graphs with a degree sequence, that is closest to a given degree sequence in the Manhattan metric. Stable graphs to the game in [13] and solutions to the IP in this paper, may be non-unique. We relate graphical solutions of the given IP to stable collaboration networks via the price of anarchy which we can calculate exactly as the result of another integer program.

I. INTRODUCTION

The Network Science community has largely dedicated its efforts to the exposition and analysis of topological properties that occur in several real-world networks; e.g., scale-freeness [2], [17], [6], [1]. Recently, there has been interest in showing that these topological properties may arise as a result of optimization, rather than some immutable physical law [7]. As Doyle et al. [7] point out, various networks such as communications networks and the Internet are designed by engineers with some objectives and constraints. While it is true that there is often not a single designer in control of the entire network, the network does not naturally evolve without the influence of designers. In each application, the network structure must be feasible with respect to some physical constraints corresponding to the tolerances and specifications of the equipment used in the network. For example, in a system such as the world wide web, a single web-page
might have billions of connections, however it is not possible for a node to have such a degree in many other applications, such as collaboration or road networks. Certainly the structure of the network has a significant impact on its ability to function, its evolution, and its robustness. However network structures often arise as a result (locally) of optimized decision making among a single agent or multiple competitive or cooperative agents, who take network structure and function into account as a part of a collection of constraints and objectives. Recently, network formation has been modeled from a game theoretic perspective \[16\], \[12\], \[8\], \[9\], and in \[13\], it was shown that there exists games that result in the formation of a stable graph with an arbitrary degree sequence. In this paper we formulate an optimization problem to calculate the price of anarchy of stable graphs for this network formation game.

II. PRELIMINARY NOTATION AND DEFINITIONS

Let \( N = \{1, 2, \ldots n\} \) be the set of nodes in a graph. A graph \( g \) is a set of links (set of subsets of \( N \) of size two) and \( g^N \) is the complete set of all links. Denote \( G \) as the set of all graphs over the node set \( N \), that is, \( G = \{ g : g \subseteq g^N \} \). Let \( \eta_i(g) : G \rightarrow \mathbb{R} \) denote the degree of node \( i \) in graph \( g \) and \( \eta(g) : G \rightarrow \mathbb{R}^n \) be the degree sequence of the graph \( g \); i.e., \( \eta(g) = (\eta_1(g), \eta_2(g), \ldots, \eta_n(g)) \). Let \( [g]_\eta \) be the equivalence class of graphs with the same degree sequence as \( g \). A degree sequence \( d = (d_1, \ldots, d_n) \) on \( n \) nodes is graphical, if there exists a graph with \( n \) nodes and degree sequence \( d = (d_1, \ldots, d_n) \). The \( \ell_1 \)-norm between two degree sequences \( d = (d_1, \ldots, d_n) \) and \( k = (k_1, \ldots, k_n) \) is given as:

\[
\|d - k\|_1 = \sum_{i} |d_i - k_i|
\]

We define a graph \( g \) to be the closest in \( \ell_1 \) norm to degree sequence \( d \) if it is a graph in \( G \) with a degree sequence that is the minimum \( \ell_1 \) norm distance to \( d \) of all graphs in \( G \). Naturally, this graph may not be unique. That is, \( g \) is a closest graph to degree sequence \( d \) if:

\[
\|\eta(g) - d\|_1 = \min_{g' \in G} \|\eta(g') - d\|_1
\]

III. NETWORK GAME NOTATIONAL PRELIMINARIES

We follow the notational conventions common in this body of literature \[12\], \[8\], \[10\]. The value of a graph \( g \) is the total value produced by agents in the graph; we denote the value
of a graph as the function \( v : G \rightarrow \mathbb{R} \) and the set of all such value functions as \( V \). An allocation rule \( Y : V \times G \rightarrow \mathbb{R}^n \) distributes the value \( v(g) \) among the agents in \( g \). Denote the value allocated to agent \( i \) as \( Y_i(v,g) \). Since, the allocation rule must distribute the value of the network to all players, it must be balanced; i.e., \( \sum_i Y_i(v,g) = v(g) \) for all \( (v,g) \in V \times G \). The allocation rule governs how the value is distributed and thus makes a significant contribution to the model. Denote the game \( G = G(v,Y,N) \) as the game played with value function \( v \) and allocation rule \( Y \) over nodes \( N \). Jackson and Wolinsky use pairwise stability to model stable networks without the use of Nash equilibria [12].

**Definition 1.** A network \( g \) with value function \( v \) and allocation rule \( Y \) is pairwise stable if (and only if):

1. For all \( ij \in g \), \( Y_i(v,g) \geq Y_i(v,g-ij) \) and
2. For all \( ij \not\in g \), if \( Y_i(v,g+ij) > Y_i(v,g) \), then \( Y_j(v,g+ij) < Y_j(v,g) \)

Pairwise stability implies that in a stable network, for each link that exists, (1) both players must benefit from it and (2) if a link can provide benefit to both players, then it in fact must exist. Jackson notes that pairwise stability may be too weak because it does not allow groups of players to add or delete links, only pairs of players [10]. Deletion of multiple links simultaneously has been considered in Belleflamme and Bloche [3]. Previously, in [13] we extend work [11] and [8], showing that stable networks may be formed as a result of a link formation game with an arbitrary (desired) degree sequence.

**Theorem 2.** Let \( d = (d_1, \ldots, d_n) \) be a desired degree sequence for \( n \) players in the node set \( N \). Assume that Player \( i \) wishes to maximize objective \( Y_i(g) = -f_i(\eta_i(g)) = -f(\eta_i(g) - d_i) \) or minimize \( f_i(\eta_i(g)) \), where \( f : \mathbb{R} \rightarrow \mathbb{R} \) is a nonnegative \( (f(\eta) \geq 0) \) convex function that has a minimum at 0. Let \( v \) be the balanced value function induced from the allocation rule \( Y = (Y_1, \ldots, Y_n) \). If \( \eta^{-1}(d) \) is non-empty (i.e., \( d \) is graphical) then any graph \( g \) such that \( \eta_i(g) = d \) is pairwise stable for the the game \( G(v,Y,N) \).

**IV. Price of Anarchy - An Optimization Approach**

When networks form as a result of selfish competition among nodes, the resulting stable network may not, in fact, be system optimal. It is possible that a stable configuration is achieved
in which each node does worse than if a central planner had optimized the system. The resulting loss is called the \textit{price of anarchy}. We may calculate the \textit{price of anarchy} as the result of two optimization problems. Denote by $\rho$ the price of anarchy which may be defined as:

\[
\rho = \frac{\max_{g' \in G'} \sum_i Y_i(g')}{\min_{g \in G} \sum_i Y_i(g)} = \frac{\max_{g' \in G'} \sum_i Y_i(g')}{\sum_i Y_i(g^*)}
\]

where $G'$ is the set of stable graphs and $g^* \in G$ is a graph with the lowest possible total value.

We will formulate two optimization problems: (1) to find $\max_{g' \in G'} \sum_i Y_i(g')$ and (2) to find $\min_{g \in G} \sum_i Y_i(g)$.

\section{The Worst Stable Graph}

In this section, we define an integer program to find the stable graph with the worst total allocation for the players involved in the game. The feasible region of this integer program will be the set of stable graphs. Since each allocation function $Y_i(v, g) = -f_i(\eta_i(g))$ where $f_i : \mathbb{R} \to \mathbb{R}$ is convex with a minimum at $d_i$, we can define stability directly in terms of degree (i.e. rather than using the allocation function). For any graph $g \in G$ on nodes $N = \{1, \ldots, n\}$ we may define the binary variable $x_{ij}$ so that an edge exists between nodes $i$ and $j$ if and only if $x_{ij} = 1$. As a consequence, we may construct a graph $g$ corresponding to binary assignment of values to variables $x_{ij}$ ($i \in N$, $j \in N$). For the remainder of this paper, we will assume that $x_{ii} = 0$ for all $i \in N$ and thus $x_{ii}$ may be safely removed from all expressions. If $g$ is constructed in such a way from a value assignment to the variables $x_{ij}$ then we will say that $g$ is generated by these variables and their assigned values.

\textbf{Lemma 3.} A graph $g$ generated by a set of binary variables $\mathbf{x} = \langle x_{ij} \rangle$ with value function $v$ and allocation rule $Y_i(v, g) = -f_i(\eta_i(g))$ where $f_i$ is convex and has a minimum at $d_i$ is pairwise stable if (and only if):

1) for all $x_{ij} = 1$, $\sum_{l \neq i} x_{il} < d_i$ and $\sum_{l \neq j} x_{lj} < d_j$

2) for all $x_{ij} = 0$, if $\sum_{l \neq i} x_{il} < d_i$, then $\sum_{l \neq j} x_{lj} = d_j$

\textbf{Proof:} This definition is immediate from inspection.

The conditions for stability can be rewritten as:

1) $\sum_{l \neq i} x_{il} < d_i$ and $\sum_{l \neq j} x_{lj} < d_j \Rightarrow x_{ij} = 1$
Lemma 4. For degree sequence \( \mathbf{d} \), a graph \( \mathbf{x} = \langle x_{ij} \rangle \) and \( s, z \) invoked via constraints 4, 6, 7, 8, and 9 is not stable if and only if constraint 5 is violated.

\[
\sum_{j \neq i} x_{ij} + s_i = d_i \quad \text{for all } i \tag{4}
\]

\[
z_i + z_j - 1 \leq x_{ij} \quad \text{for all } ij \tag{5}
\]

\[
z_i \leq s_i \quad \text{for all } i \tag{6}
\]

\[
s_i \leq d_i z_i \quad \text{for all } i \tag{7}
\]

\[
s_i \geq 0 \quad \text{for all } i \tag{8}
\]

\[
z_i \in \{0, 1\} \tag{9}
\]

Proof: For a graph \( \mathbf{x} = \langle x_{ij} \rangle \) and degree sequence \( \mathbf{d} \), the slack vector \( s \) and its binary counterpart \( z \) will be invoked by constraints 4, 6, 7, 8, and 9.

Suppose the forward implication is not true. That is, suppose \( \mathbf{x} = \langle x_{ij} \rangle \) is not stable, but constraint 5 holds. The graph \( \mathbf{x} = \langle x_{ij} \rangle \) may not be stable in two ways:

1) A link \( ij \) exists, which node \( i \) or \( j \) have an incentive to veto

2) A link \( ij \) does not exist, but node \( i \) and \( j \) have an incentive to form it

First, suppose link \( ij \) exists, so \( x_{ij} = 1 \). Without loss of generality, suppose it is \( i \) with an incentive to veto this link. This implies that \( \sum_{l \neq i} x_{il} > d_i \). Constraint 4 implies \( s_i < 0 \), which violates constraint 8. Hence, the assumption of the Lemma does not even hold for this case.

Second, suppose link \( ij \) does not exist, so \( x_{ij} = 0 \). Since node \( i \) and \( j \) have an incentive to form this link, this implies that \( \sum_{l \neq i} x_{il} < d_i \) and \( \sum_{l \neq j} x_{jl} < d_j \). Constraint 4 implies \( s_i < 0 \) and \( s_j < 0 \), implying \( z_i = 1 \) and \( z_j = 1 \) by constraint 7. Now, since \( z_i = 1 \) and \( z_j = 1 \), but \( x_{ij} = 0 \), constraint 5 is violated and this is a contradiction.

Now suppose the backward implication is true. That is, constraint 5 is violated but graph \( \mathbf{x} = \langle x_{ij} \rangle \) is stable. The violation of constraint 5 implies that there exists an \( i \) and \( j \) such \( x_{ij} = 0, z_i = 1, \) and \( z_j = 1 \). The definition of stability requires that if \( x_{ij} = 0 \) then either \( \sum_{l \neq i} x_{il} = d_i \) or \( \sum_{l \neq j} x_{lj} = d_j \). However, \( \sum_{l \neq i} x_{il} = d_i \) implies that \( s_i = 0 \) and then \( z_i = 0 \).
Similarly, $\sum_{l \neq j} x_{lj} = d_j$ implies that $s_j = 0$ and then $z_j = 0$. Since, $z_i = 1$ and $z_j = 1$, this is a contradiction.

**Theorem 5.** The stable graph $x = \langle x_{ij} \rangle$ with degree sequence furthest from $d$ in $\ell_1$ norm can be found using the integer program:

$$\max \sum_i s_i$$

subject to:

\begin{align*}
\sum_{j \neq i} x_{ij} + s_i &= d_i \quad \text{for all } i \\
z_i + z_j - 1 &\leq x_{ij} \quad \text{for all } ij \\
z_i &\leq s_i \quad \text{for all } i \\
s_i &\leq d_i z_i \quad \text{for all } i \\
s_i &\geq 0 \quad \text{for all } i \\
z_i &\in \{0, 1\}
\end{align*}

(10)

**Proof:** Since a graph that violates the constraints is not stable, by Lemma 4, it is almost immediate that any graph must either:

1) violate the constraints and not be stable
2) obey the constraints and have an objective less than the optimal solution (worst graph), so it is not the furthest from $d$
3) obey the constraints and have an objective equal to the optimal solution (worst graph) and be one of the non-unique optimal solution graphs (worst graphs)

The theorem states that a graph that satisfies the constraints (a graph that is stable) and maximizes the objective (has degree sequence furthest from $d$ in $\ell_1$ norm), is optimal (is the solution to the math program). This statement is immediate.

Since the graph $g$ is represented as $x = \langle x_{ij} \rangle$, the degree of node $i$ can be interchangeably defined using the equivalence $\eta_i(g) = \sum_{j \neq i} x_{ij}$. Constraint 4 implies that $s_i = \sum_{j \neq i} x_{ij} - d_i = \eta_i(g) - d_i$. Hence, $f_i(\eta_i(g)) = f(\eta_i(g) - d_i) = f(s_i)$. The worst graph is hence the graph that maximizes $\sum_i f(s_i)$.

**Corollary 6.** The stable graph $x = \langle x_{ij} \rangle$ with the worst total allocation among players can be
found using the integer program:

\[
\begin{align*}
\text{max} & \quad \sum_i f(s_i) \\
\text{s.t.} & \quad \sum_{j \neq i} x_{ij} + s_i = d_i \quad \text{for all } i \\
& \quad z_i + z_j - 1 \leq x_{ij} \quad \text{for all } ij \\
& \quad z_i \leq s_i \quad \text{for all } i \\
& \quad s_i \leq d_i z_i \quad \text{for all } i \\
& \quad s_i \geq 0 \quad \text{for all } i \\
& \quad z_i \in \{0, 1\}
\end{align*}
\]  

(11)

**Proof:** After incorporating the payoff for each player as a function of their slack \( f(s_i) \), the integer program (11) follows directly from the integer program (10).

**Remark 7.** Since \( f(s_i) \) is convex with a minimum at \( s_i = 0 \), the minimizer of \( f_i(s_i) \) is equivalent to the minimizer of \( s_i \). Note that there may be different ways to define the worst graph, here we assume all players are equally weighted. As a result, the integer program (10) should prove to be more useful as it measures the distance between the graphs and not how that is valued by the players (which can be added to the objective afterward).

**B. The Best Graph**

In this section, we define an integer program to find the graph with the best total allocation for the players involved in the game. Note that we are not necessarily looking for a stable graph, so the feasible region is the set of all graphs. This will provide a baseline to evaluate the worst price that may be paid for selfish competition (e.g. the Price of Anarchy).

Previously, there has been work on generating graphs with an arbitrary graphical degree sequence [4], [15], [5], [14]. However this literature is mainly concerned with the algorithms to generate a graph for a graphical degree sequence. Here we seek to find the closest graph to any degree sequence (graphical or not) and we use an optimization formulation to do this.

We formulate a math program by defining the feasible region as the set of all graphs and then minimizing the distance between the arbitrary degree sequence \( d = \{d_1, \ldots, d_n\} \) and the degree...
sequence of a graph in the feasible region.

\[
\min \sum_i \left| \sum_{j \neq i} x_{ij} - d_i \right|
\]
\[
s.t. \quad x_{ij} - x_{ji} = 0 \quad \forall i < j
\]
\[
x_{ij} \in \{0, 1\} \quad \forall i, j
\]

This non-linear optimization problem may be reformulated as an integer linear programming problem:

\[
\min \sum_i e_i
\]
\[
s.t. \quad \sum_{j \neq i} x_{ij} - d_i \leq e_i \quad \forall i
\]
\[
- \sum_{j \neq i} x_{ij} + d_i \leq e_i \quad \forall i
\]
\[
x_{ij} - x_{ji} = 0 \quad \forall i < j
\]
\[
x_{ij} \in \{0, 1\} \quad \forall i, j
\]

**Theorem 8.** The graph generated by an optimal solution to the integer program:

\[
\min \sum_i e_i
\]
\[
s.t. \quad \sum_{j \neq i} x_{ij} - d_i \leq e_i \quad \forall i
\]
\[
- \sum_{j \neq i} x_{ij} + d_i \leq e_i \quad \forall i
\]
\[
x_{ij} - x_{ji} = 0 \quad \forall i < j
\]
\[
x_{ij} \in \{0, 1\} \quad \forall i, j
\]

is a closest graph under the $\ell_1$-norm to a graph with degree sequence $d = \{d_1, \ldots, d_n\}$.

**Proof:** Let $g$ be the graph generated by an optimal solution $x^* = \langle x^*_{ij} \rangle$ to Problem (13) and suppose further there is another graph $g'$ on $n$ nodes so that

\[
\|\eta(g) - d\|_1 > \|\eta(g') - d\|_1
\]
Now construct another assignment of values to the variables in Problem (13), which we denote by \( x = \langle x_{ij} \rangle \). This implies that

\[
\sum_i \sum_{j \neq i} x_{ij} - d_i < \sum_i \sum_{j \neq i} x^*_{ij} - d_i \leq \sum_i e^*_i
\]

where the left inequality is an implication of (14) and the right inequality is an implication of the \( x^* \) being a solution to (13). Finally, (15) contradicts our assumption that \( x^* = \langle x^*_{ij} \rangle \) was an optimal solution to Problem (13).

Now, the price of anarchy is simply the ratio of the objective function value from the worst graph (Problem (10)) to the best graph (Problem (13)).

V. NUMERICAL EXAMPLE

Suppose that we want the degree sequence of a stable graph that results from playing the game described in Theorem 2 to have a power law degree distribution. We embed this into the objectives of the players, so the resulting graph has the proper distribution. Let \( n = 35 \) players attempt to minimize their cost function

\[
f_i(\eta_i(g)) = f(\eta_i(g) - k_i) = (\eta_i(g) - k_i)^2 + \psi
\]

where \( \psi = 2 \) and the parameters of each player (\( k_i \)) are given in the following table:

<table>
<thead>
<tr>
<th>Node(s)</th>
<th>( k_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-29</td>
<td>1</td>
</tr>
<tr>
<td>30-33</td>
<td>2</td>
</tr>
<tr>
<td>34</td>
<td>3</td>
</tr>
<tr>
<td>35</td>
<td>4</td>
</tr>
</tbody>
</table>

The distribution of values of \( k_i \) forms an approximate (with rounding to integers) power law distribution as illustrated in Figure 1.

We can now solve the integer program (11) to find the graph with the worst allocation among players that will go into the numerator of the Price of Anarchy. The resulting graph is illustrated in Figure 2. Figure 3 compares the degree distribution of this worst graph to the \( k_i \) values. Next, we will solve the integer program (13) to find the graph with the lowest overall allocation. The resulting graphic solution is shown in Figure 4. Now, we can calculate the price of anarchy:
Fig. 1. The empirical desired distribution of the degrees of the players. This degree distribution follows an approximate power law distribution.

Fig. 2. Worst graph

1) Calculate the objective value of the Worst Graph.
2) Calculate the objective value of Best Graph.
3) Calculate Price of Anarchy as the ratio.

Fig. 3. Worst graph
In the worst graph, all nodes have their targeted degree other than two. That is $\sum_{j \neq i} x_{ij} = k_i$ for all nodes $i$ other than $i = 32$ and $i = 35$. Node $i = 32$ has degree $\sum_{j \neq i} x_{ij} = 1$ but $k_i = 2$ and node $i = 35$ has degree $\sum_{j \neq i} x_{ij} = 1$ but $k_i = 4$. Hence, for all nodes $i$ other than $i = 32$ and $i = 35$, $f(\eta_i(x)) = (\eta_i(x) - k_i)^2 + \psi = (0)^2 + \psi = \psi = 2$. However, for node $i = 32$ has $f(\eta_i(x)) = (\eta_i(x) - k_i)^2 + \psi = (2 - 1)^2 + 2 = 3$ and node $i = 35$ has $f(\eta_i(x)) = (\eta_i(x) - k_i)^2 + \psi = (4 - 1)^2 + 2 = 11$.

<table>
<thead>
<tr>
<th>Node(s) $i \not\in {32, 35}$</th>
<th>$f(\eta_i(x))$</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>3</td>
</tr>
<tr>
<td>35</td>
<td>11</td>
</tr>
</tbody>
</table>

Alternatively, in the best graph, all nodes have their targeted degree. That is $\sum_{j \neq i} x_{ij} = k_i$ and $f(\eta_i(x)) = (\eta_i(x) - k_i)^2 + \psi = (0)^2 + \psi = \psi = 2$.

The price of anarchy is calculated as the total objective of the worst graph $33 \text{ nodes} \times (2) + 1 \text{ node} \times (3) + 1 \text{ node} \times (11) = 80$ over the total objective of the best graph $35 \text{ nodes} \times (2) = 70$, which is $\frac{80}{70} = \frac{8}{7} \approx 1.14$

VI. Conclusion and Future Directions

In this paper we have formulated an optimization problem to generate graphs with an arbitrary degree sequence. The solution of this problem will determine the feasibility of a graph.
with a particular degree sequence. Future research may include how to embed other network characteristics, such as graph diameter or connectedness, into the optimization formulation, to generate graphs with particular measures of these characteristics.

VII. APPENDIX 1

In this appendix, we give a detailed explanation of the construction of the Integer Program formulated to find the worst stable graph. The degree of a node $i$ in a graphic solution $\mathbf{x} = \langle x_{ij} \rangle$ is related to the targeted degree $d_i$:

$$\sum_{j \neq i} x_{ij} + s_i = d_i \quad \text{for all } i$$

$$s_i \geq 0 \quad \text{for all } i$$

$$s_i \quad \text{integer for all } i$$

where $s_i$ is a non-negative integer slack variable. Note that negative slack ($s_i < 0$) would imply $\sum_{j \neq i} x_{ij} > d_i$, which is not stable because node $i$ could increase their payoff by dropping at least one link. To code the slack variables as binary, we use binary variable $z_i$:

$$z_i \leq s_i \leq M_i z_i \quad (16)$$

where $M_i$ is a large parameter. Constraint $(16)$ enforces Conditions $(17)$:

$$s_i > 0 \iff z_i = 1$$

$$s_i = 0 \iff z_i = 0 \quad (17)$$

Note that when $z_i = 1$, Constraint $(16)$ implies $1 \leq s_i \leq M_i$ and since $s_i$ is an integer, $s_i > 0$ is equivalent to $s_i \geq 1$. The first condition for stability $(1)$ requires each pair of nodes that have a degree less than targeted degree to be linked, which implies that if $s_i > 0 (z_i = 1)$ and $s_j > 0 (z_j = 1)$ then $x_{ij} = 1$:

$$z_i + z_j - 1 \leq x_{ij} \quad (18)$$

The second condition for stability requires that for each pair of nodes that are not linked together, one of the nodes must have their targeted degree $d_i$. This implies that if $x_{ij} = 0$ then $s_i = 0 (z_i = 0)$ or $s_j = 0 (z_j = 0)$. This condition is simultaneously satisfied by the exact same
constraint (18). Since $M_i$ is bounding $s_i$, using $M_i = d_i$ is a tight bound because the degree of $i$ cannot be negative. The consolidated set of constraints is:

\[
\sum_{j \neq i} x_{ij} + s_i = d_i \quad \text{for all } i
\]

\[
z_i + z_j - 1 \leq x_{ij} \quad \text{for all } i,j
\]

\[
z_i \leq s_i \quad \text{for all } i
\]

\[
s_i \leq d_i z_i \quad \text{for all } i
\]

\[
s_i \geq 0 \quad \text{for all } i
\]

\[
z_i \in \{0, 1\}
\]

Now the stable graph with degree sequence furthest from $d$ in $\ell_1$ norm can be found using the integer program (19).

\[
(19)
\]

REFERENCES


