A Framework for Incentivizing Time-Inconsistent Planners to Produce Documentation in a Social Network

Neelan Coleman
coleman8@illinois.edu

Edward W Huang
ewhuang3@illinois.edu

Cassandra Jacobs
csjacob2@illinois.edu

ABSTRACT

Internal documentation is essential to the function of technology companies. Documents range, in complexity and importance, from goals spreadsheets to in-line code comments. While high-quality documentation is undoubtedly beneficial, many organizations struggle to produce and maintain such documentation.

There is a wide variety of tools and processes aimed to promote documentation production within technology companies (e.g., Google Docs Enterprise, Twiki, and GitHub Pages). Each of these solutions has been shown to have short-term, positive effects on documentation production. However, they fail to drive long-term behavioral changes within the organization.

In our paper, we construct three game-theoretical models that take into account agents who are time-inconsistent planners to best capture the real-world lack of documentation. The novelty of this work is in integrating these agents into diffusion models. Along with mathematical formulations, we provide simulations. Across each of our models, we show that in order to drive long-term behavioral changes, there is a lower bound on the required percentage of agents needed to give incentive.

For our first model, a time-based model, we show that the max margin social influence heuristic requires 26.3% of a network as seed nodes to diffuse good behavior. For our second model, a task-based model, we show that incentivizing agents in proportion to team size performs best: incentivizing 29.7% of all nodes diffuses good behavior. Lastly, for a role-based model, we find that by adding only 4% of agents, whose sole task is documentation, with a 1:10 ratio to other single-task workers, we diffuse the behavior of well-producing documentation.

1. INTRODUCTION

In the technology industry, internal documentation is essential to maintaining, recording, and succinctly describing each company's work [14]. Despite the importance of this, many organizations fail to provide high-quality documentation [6] [12]. To rectify the problem, several applications, such as Google Docs and wiki pages, have been developed to encourage teamwork, make the documentation process more fun, etc. However, in spite of these applications, it has been shown that documentation availability is lacking in many project settings [15].

Using a game theoretical framework, it is straightforward to see why a resource as important as documentation frequently fails to see development. In Akerlof's story [1], Akerlof ships packages that incur additional costs each day that he procrastinates. In the context of code documentation, the longer agents wait to document their code, the lower quality the documentation tends to be (imagine twenty years after the code is written, and the agent is then asked to document the code). Furthermore, the immediate cost of documenting code is calculable in terms of man hours. On the other hand, the amount of time documentation saves agents in the future is less apparent. Code writers thus display present bias and exhibit behaviors as in the quasi-hyperbolic discounting model [10]: procrastination, abandonment, and benefiting from choice reduction.

In our framework, we wish to implement choice reduction, which would be setting aside a mandatory time slot after a project is finished to document the code. The framework needs to increase the power of documentation relevance [7] as well as maintain documentation simplicity. Additionally, we wish to spread the behavior of good documentation habits throughout a social network. In practical settings, these social networks would be the employees in a company. Thus, *we wish to attempt to use social influence to spread a behavior that induces present bias in agents who are already constrained for time.* We attribute the adoption of good documentation habits to workplace culture, as confirmed by our *independent survey* results (Figure 1).

In this paper, we model a real-world company containing time-inconsistent planners. Thus, over time, the model will reveal that the agents tend to drift away from high quality documentation. To rectify this, we present three models: time-based, role-based, and task-based. Each of these models is given unique parameters, and each attempts to encourage agents to produce good documentation regularly. The time-based model tests different seed selection heuristics for finding agents to incentivize. The task-based model incorporates task difficulty and agent resource constraint to do the same. The role-based model tests agent assignments to discover if single-task working agents can incentivize multi-task working agents.

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

Our results show that there exists a lower bound for the number of agents to incentivize, as well as a dominant strategy for selecting these agents. Our work can be directly applied to real workplace environments to help solve the difficult problem of incentivizing employees to produce high quality documentation.

2. RELATED WORK

Akerlof's story [1] provided the foundation for our framework by introducing the game-theoretical reasoning behind procrastination. However, while time-inconsistent planners are excellent representatives of actual employees who write documentation, it only models agents, and is not enough for us to model the dynamics of the entire social network as a whole.

Thus, we integrate time-inconsistent planners into diffusion models. For behavior adoption, we propose a variation of the *multi-behavior diffusion model* [13] with a limited scope of two behaviors: to either document or not document. This work provides the basis for selecting seed nodes, but the problem changes when we add time-inconsistent agents. Additionally, agents do not display progressive behavior adoption, since in some of our models behaviors are adopted and forgotten on a per-iteration basis.

Furthermore, we extended the model by having the quality of documentation decay over time. We derived this component from the idea of document recentness discussed in the software engineering survey from Lethbridge *et al* [11]. Lethbridge conducted a survey and independently observed software engineers at a telecommunications company. The study aimed to answer the question as to how software engineers perceived documentation relevance. The results of the survey showed that documentation remains valuable despite being out of date, but its accuracy depends largely on its recentness. We framed this concept in a model where documentation accuracy depends linearly on the time since it was created.

Jackson and Yariv 2011 worked on time streams for individuals with time inconsistency to model collective dynamic choice [8]. This work showed that appointing a dictator was the only method of aggregating unanimous preferences. Furthermore, they showed that time inconsistency exists in all heterogeneous groups who incorporate different temporal motives. However, they do not take into account network factors, which is crucial to our attempts to model a company setting.

Contractor and DeChurch 2016 researched the identification of opinion leaders in order to diffuse scientific fact and influence [5]. This corresponds closely with our time-based and task-based models, which attempt to discover seed nodes to diffuse good behavior. However, time-inconsistent agents are not considered in their model. and agents do not drift away from participating in diffusing scientific opinion.

2.1 Survey Results

We conducted an independent survey to gather feedback regarding documentation habits and the effect of social influence on the behavior of documenting. The survey was given to software engineers at various technology companies. We received 67 responses to the following questions:

1. **Q1**. After completing a task, what percentage of the time would you say that you document?
2. **Q2**. Does the difficulty of the task affect your decision to document?.
3. **Q3**. What percentage of the time would you say that you make your documentation available for others to see?
4. **Q4**. How often do you revisit documentation that you've written (approximately how many times in a given week)?
5. **Q5**. How often do you view documentation that a fellow co-worker has written (approximately how many times in a given week)?
6. **Q6**. What would encourage you to spend more effort on documentation? (choose any that apply)

Table 1: Documentation Survey

Questions	
Q1	41.0%
Q2	**68.7% YES**
Q3	69.0%
Q4	1-3
Q5	1-3
Q6	**knowing documentation is used by others**

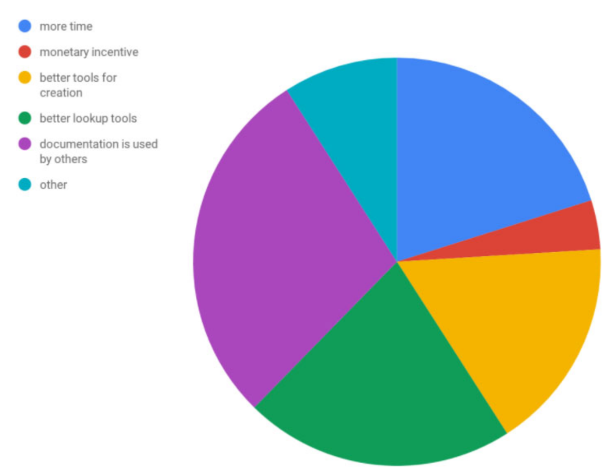

Figure 1: Q6. Incentivizing documentation efforts. Monetary incentive ranked the lowest (4%) while Documentation used by others ranked the highest (29%).

Question six (Q6) also had an "other" option. From this field, we collected the following responses that leaned towards social influence and pressure:

- Tools to build more easily on existing documentation
- A better culture of documentation, encouraged and bought into by management
- Existing documentation already done to high standards

- Better tools for working in a collaborative environment, such as the ability to track edits, versioning, publishing, searching, long-term maintenance, and having a single source of truth

- A culture that encouraged team contributions and effort

- Well-written source material from which the documentation is being created from, which would make the task easier in the first place

The takeaway from the survey is that, for many engineers, documentation likelihood correlates with task difficulty, confirming previous studies [3]. Additionally, the overall tone of the question six (Q6) and the open-ended responses indicated that an established culture of documentation can encourage employees to produce high quality documentation, providing a strong basis for applying social influence in our model.

3. MODEL DESCRIPTION

3.1 Network Generation

The baseline social network should ideally model those that we have observed in previous, real-world studies: agents in the network should be inclined to not document their code. We elected to use two methods of network generation. For two of our models, we used the Barabási-Albert preferential attachment model [2] to simulate the existence of project leaders. The addition of a new node to the network is akin to a newly hired agent who is assigned to a team. For one of our models, the task-based model, we modeled the network as a Random Graph [4] to add variance to team sizes. The network probabilistically forms edges, which results in clusters of nodes. While the addition of nodes to a network varies among our three proposed models, the average degree of nodes across model simulations was between five and eight.

3.2 Node Attributes

Each node, or agent, in the network is characterized by three attributes.

1. **Resource**. For some node v, there is a resource variable, $r(v)$, indicating time in our model. It is obtained independently from a uniformly distributed random variable $U(0, 1)$, as in previous work.

2. **Documentation state**. For some node v, there is a state variable, $s(v) \in [0, 1]$, indicating the quality of their documentation. This variable is also obtained independently from a uniformly distributed random variable $U(0, 1)$, to reflect employees who regularly produce documentation to varying degrees while they code.

3. **Social influence threshold**. For some node v, there is a social influence threshold $\theta(v)$, also obtained independently from a uniformly distributed random variable $U(0, 1)$, to reflect the minimum amount of neighboring pressure it takes for the agent to adopt a certain behavior.

3.3 Network Mutation

The social network mutates over discrete iterations. The time-based and role-based models undergo the same transformations, while the task-based model undergoes a variant of the transformation.

3.3.1 *Time-Based and Role-Based*

At the start of each iteration, each node is given a choice of two behaviors: whether to work on documentation or not. Each behavior is associated with a utility function, and the agent selects the behavior that yields more utility. In the case of a tie, the agent will choose to not work. We now discuss the benefits and costs of each action.

1. **The agent works**. The agent accrues a cost of 0.1 (time), and converts this to documentation quality, increasing its state by 0.1. However, due to present bias, the agent will not immediately observe the value of documentation until much further in the future. Thus, we scale the benefit by a factor of β.

2. **The agent does not work**. Instead of accruing a cost of 0.1, the agent gets that time it would have spent on documentation for other things, and thus gains a benefit of 0.1. However, in our model, we decrease the quality of documentation by a factor of δ. The reasoning behind this is that empirically, the longer an agent waits to write documentation, the less the agent will remember details of the code.

In equation form, the utility functions of each behavior is as follows.

$$U_{\text{work}} = \beta \times 0.1 - 0.1 \tag{1}$$

$$U_{\text{don't work}} = 0.1 - \delta \times \text{documentation state} \tag{2}$$

Each node computes its two utility functions, and publicly declares a behavior it will adopt during the day.

Social Influence

Even if a node declares that it will not work for the current iteration, social influence may still change the node's behavior. Here, we define the social signal from a node u to a neighboring node v to be proportional to $\frac{1}{N(v)}$, where $N(v)$ is the degree of node v. Like the Linear Threshold Model [9], a node v will choose to work if the fraction of working neighbors exceeds its internal threshold attribute.

3.3.2 *Task-Based*

The task-based model takes the signal from question two (Q2)1 of the independent survey results. At the start of each iteration, a task with difficulty d, where $d \in [0, 1]$, is generated and assigned to a node in the network chosen independently and at random. The node updates its state which represents its adoption of good documenting behavior.

1. **State Update**. The state is updated based on a utility function that compares the cost of documenting with the perceived immediate reward. In equation form:

$$state(i+1) = state(i) + \gamma \times Utility(i) \quad \begin{cases} 0, & \text{if state} \leq 0 \\ 1, & \text{if state} \geq 1 \end{cases} \tag{3}$$

Here γ is the time inconsistent discounting factor. Essentially as time goes on (tasks get completed) the agent becomes less likely to change its state in either direction.

2. **Cost and Reward**
The perceived reward that an agent attains during a given iteration is directly related to the social influence of its peers combined with the difficulty of the task it completed that iteration. The reward is discounted by the available resource the agent has. The corresponding influence is the average state of the agents neighbors. The resource and difficulty are selected independently and at random. The cost of documenting is fixed to a constant c. In our task-based model, the constant was tuned in order to get the desired behavior that the average state of the network, absent of external factors, should remain constant throughout the simulation.

$$\text{Utility}(i) = \text{reward}(i) - c(i) \tag{4}$$

$$\text{reward}(i) = \text{influence}(i) + \text{difficulty}(i) \times \text{resource}(n) \tag{5}$$

$$\text{influence}(i) = \sum_n \text{state}_n(i)/n \tag{6}$$

$$\text{difficulty}(i) \in [0,1] \tag{7}$$

$$\text{resource}(i) \in [0,1] \tag{8}$$

$$c(i) = .54 \tag{9}$$

4. PROBLEM STATEMENT

Here, we describe our three models, how they attempt to solve the problem of diffusing good documentation behavior, and discuss their differences[1].

4.1 Time-Based Model

We introduce a variation of the Total Participation Maximization problem introduced in Sarkar and Sundaram 2016 [13]. The idea is that with a fixed amount of money, we wish to promote a set of employees (seed nodes) in the network so as to maximize the quality of documentation through diffusion. Furthermore, we exclude the resource attribute in this model. The defining characteristic of the time-based model is that we treat each iteration as a "day" in a set documentation period, and all nodes update in the same iteration.

We ran experiments 100 times for each network size in $[100, 200, 300, 400, 500]$. We evaluate a method by finding the minimum number of employees, k, who we need to encourage to work to induce good behavior in the network. Here, good behavior is arbitrarily defined as 75% of the network with a documentation state of 90% or more. This is a variation of total participation. Next, we present a few heuristics we used to select seed nodes.

4.1.1 Naïve Node Degree

We ranked nodes by their degree, and forced the top k nodes who are not already working to work on documentation every day.

[1]Code can be found on github at:
https://github.com/uiucEnterpriseDocumentation/simulator

4.1.2 Ranked Degree and Threshold

We took into account degree and threshold when selecting seed nodes. For each node, we computed the number of neighbors with below-average threshold (recall that thresholds are drawn from $U(0,1)$. The seed nodes are the top k nodes who are not already working with the most below-average threshold neighbors.

4.1.3 Rank-Based Social Influence Weight

This heuristic computes an influence weight measure for each node. The social influence of a node v is proportional to $\sum_{u \in N(v)} \frac{1}{|N(u)|}$, where $N(u)$ is the number of neighbors for a node u. Again, we pick the top k nodes who are not already working.

4.1.4 Max Margin Social Influence Weight

This heuristic computes social influence the same way as in the rank-based heuristic, but instead iteratively builds the seed set by selecting a new seed node that maximizes the marginal increase of influence weight.

4.1.5 Expected Immediate Adoption

We incrementally build the seed set based on the Expected Immediate Adoption (EIA) value, as computed in previous work [13]. The probability of a node working is computed by the fraction of its neighbors working. A node's EIA is computed by summing up the probability of all of its neighbors working.

4.2 Task-Based Model

We solve the identical problem [13] as with our time-based model. We attempt to maximize the quality of documentation through diffusion. A fundamental difference between the task-based and time-based model mentioned above is threefold: (1) we include the resource attribute for each node (2) we include task difficulty. Meaning the more difficult a task, the more perceived benefit an agent has to document. (3) Each iteration represents a single task. Only one agent completes the task and updates its state in a given iteration.

We ran experiments 100 times for a network size of 200 and $200,000$ tasks were inserted into the simulation *(i.e. 200,000 iterations)*. Here, again, we wanted to find the minimum percent of employees, k, that we need to give incentive at initialization in order to induce good behavior in the network. We define the average state of the network at initialization to .3 as a low estimate of the signal from question one (Q1) of the private survey (Figure 1). Here, like above, good behavior is arbitrarily defined as 75% of the network with a documentation state of 90% or more. this corresponds to a network with an average documentation state of 90% Next, we present a few heuristics we used to select k nodes to give incentive.

4.2.1 Random

We selected k nodes uniformly and at random to give incentive at initialization.

4.2.2 Sparse

We took into account neighboring nodes up to 2 degrees of separation. At initialization we counted the number of neighbors and extended neighbors with a high documentation state. If the number of neighbors and extended neighbors did not exceed the allowance threshold, this node was given incentive. The allowance threshold was a parameter that we tuned via experiment.

4.2.3 Weakest

This heuristic computes the average state of all nodes in the network and prioritizes giving incentive to agents within clusters with the lowest state. Essentially give incentive to agents with the poorest documentation behavior.

4.2.4 Strongest

Similarly, this heuristic computes the average state of all nodes in the network and prioritizes giving incentive to agents within clusters with the highest state.

4.2.5 X Per Cluster

This heuristic sought to distribute the k seed nodes amongst clusters evenly. We distributed x nodes per cluster. x being a parameter tuned based on network size.

4.2.6 Smallest Cluster Priority

This heuristic computes the size of each cluster and gives priority to selecting the k seed nodes from smaller clusters

4.2.7 Largest Cluster Priority

Similarly, this heuristic computes the size of each cluster and gives priority to selecting the k seed nodes proportional to cluster size.

4.3 Role-Based Model

In our survey results and discussions from various sources, research indicates that there are jobs where the agent's tasks are solely to create documentation for others. In a scenario such as this, documentation is no longer a resource-intensive task that must be taken on by an agent whose primary role is in other areas. This model changes the parameters of the simulation as both resource and state are no longer contributing attributes to the agent.

4.3.1 Agent Type, Resource and State Constraints

Agents were assigned a new attribute, Type, which can be one of three variations:

1. **Documenter (D)**. An Agent whose only task is to create documents. Their state variable is always set to 1 from $s(v) \in [0, 1]$, representing good documentation at all times.

2. **Engineer (E)**. An agent whose only task is to work at their assigned job and not create documentation. Their work is being supported by the Documenter agent instead, so their state is also set to 1 from $s(v) \in [0, 1]$, representing good behavior in the simulation.

3. **Combo (C)**. The default agent who performs both their assigned task as well as documentation. Their state is a random value from $s(v) \in [0, 1]$, with a randomly assigned resource $U(0, 1)$.

4.3.2 Assigned Ratios of Agents

Parameters were added to allocate the number of Documenter and Engineer agents to the nodes in the graph.

1. **MAX_NUM_DOCS** represents the maximum number of Documenter agents that will be assigned to graph nodes.

2. **ENG_TO_DOC** represents the ratio of Engineer agents to Documenter agents.

When assigning agent types, the algorithm calculates the number of Documenter agents and assigns the allocated ratio of Engineer agents based on the ENG_TO_DOC parameter. Once all D and E type agents are assigned, any remaining unassigned agents are labeled Combo type. This assignment is based on the parameter value NUM_POINTS. A realistic assignment scenario should allocate types where

$$D < E < C \tag{10}$$

and

$$NUM\_POINTS = D + E + C \tag{11}$$

4.3.3 Working State, Decay and Social Influence

Like the base model, agents still have the option to work or not work during each iteration with the following restrictions:

1. Documenter and Engineer agents are always working. These agents have assigned tasks that don't require extra resource or incur a work or lazy utility, so they cannot change status to not working. We can assume the Documenter and Engineer agents are "happy" in their roles.

2. Combo agents behave the same as the base model, where they can either work or not work, decided by the utility function.

Combo agents $lazy\_utility$ attribute is influenced by decay as they gravitate towards a not working state. As Documenter and Engineer agents cannot change state, they are not influenced by decay. The agent $work\_utility$ attribute is positively influenced by the social influence threshold. This can change a Combo agent's state from "not working" to "working", or keep it from switching to "not working" from "working". All agent types contribute to the social influence threshold by being in a working state.

5. RESULTS AND DISCUSSION

5.1 Time-Based Model

For each heuristic in selecting a seed set for the time-based model, we recorded the minimum k required to induce good behavior for varying network sizes, averaged over 100 trials. For all methods, we found a linear relationship between the two axes. For naïve node degree, k was approximately equal to 27.6% of the number of nodes (Figure 2). In other words, we need to force the 27.6% most popular, but lazy, employees to write documentation each day to promote a culture of documentation. For the ranked degree and threshold heuristic, the minimum k was approximately 27.8% of the network size (Figure 3). For the rank-based social influence weight heuristic, k was approximately equal to 26.7% of all nodes (Figure 4). For the max margin social influence weight heuristic, we found k to be approximately 26.3% of all nodes (Figure 5). For the EIA heuristic, we found that k is approximately 26.8% (Figure 6).

We see that the max margin social influence weight heuristic requires the fewest seed nodes to diffuse good behavior. Using an unpaired t-test, this heuristic performed better than any runner-up heuristic with a p-value < 0.01.

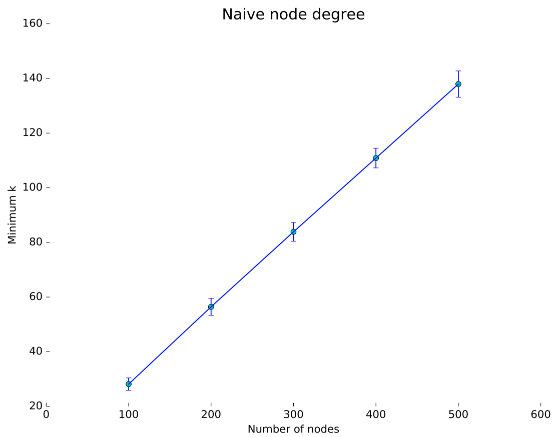

Figure 2: Naïve node degree heuristic. Minimum number of seed nodes to diffuse good behavior vs. network size.

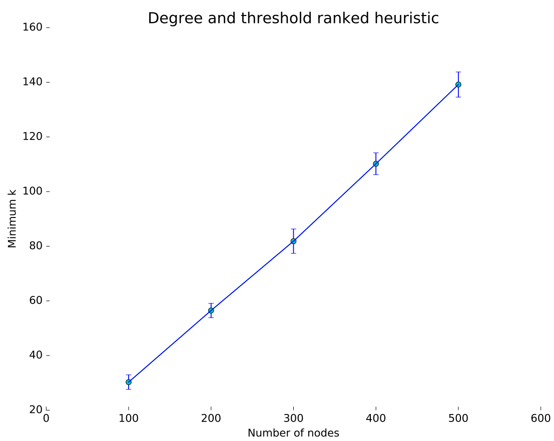

Figure 3: Ranked degree and threshold heuristic. Minimum number of seed nodes to diffuse good behavior vs. network size.

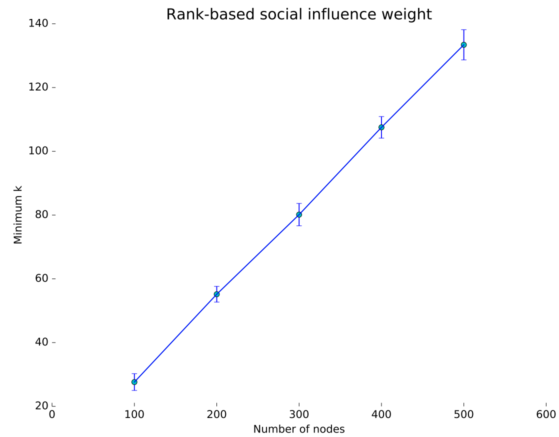

Figure 4: Rank-based social influence weight heuristic. Minimum number of seed nodes to diffuse good behavior vs. network size.

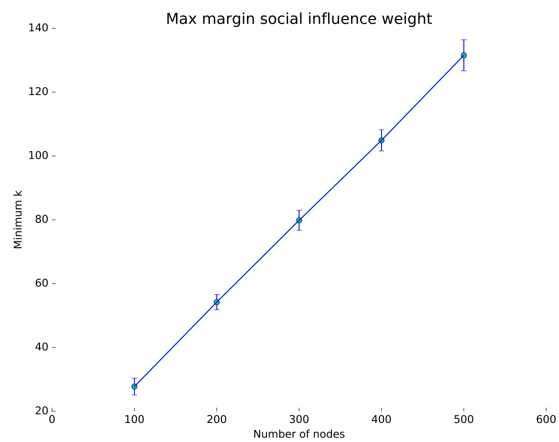

Figure 5: Max margin social influence weight heuristic. Minimum number of seed nodes to diffuse good behavior vs. network size.

5.2 Task-Based Model

For each heuristic described for selecting seed nodes in the task-based model, we recorded the minimum k required to induce good behavior for a network size of 200, averaged over 100 trials. The results are twofold: (1). The average number of employees required to give incentive is 41.6% across heuristics, with the best performance of 29.7%(Figure 7. This would mean that a company wishing to maintain long term behavior change towards good documentation should expect to invest in at least 29.7% of employees. (2). The heuristic of selecting seed nodes based on cluster size, giving priority to larger teams requires the least amount of seed nodes. (Figure 8). This would mean, for example, that a company could hold a training and utilize this heuristic in selecting which employees to invite to attend the training in order to get the best results.

5.3 Role-Based Model

The goal in the simulation is to socially influence the Combo agents in a positive manner against their $work\_utility$ property in order to have them producing documentation in a good manner. As Documenters and Engineers are set to a static state of 1, they are already producing at a good state but without enough of them in the network, they still cannot positively influence the Combo agents, or influence enough to get the entire network running well. In initial tests, with only 2.5% of the agents assigned to the Documenter type, productivity of the network did not rise over 50%.

Multiple simulations were run while adjusting the both the number of Documenters and the Engineer to Documenter ratio. The goal was to find a balance of both types that would have enough positive influence in the network but where the total would not outnumber the Combo agents. A simulation of 8 Documenters with a Engineer to Documenter ratio of 10:1 in a network size of 200, leaving 112 Combo agents, produced a productivity output of 73%.

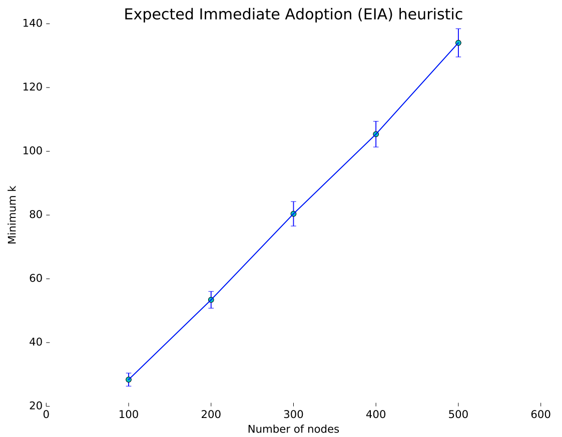

Figure 6: Expected immediate adoption heuristic. Minimum number of seed nodes to diffuse good behavior vs. network size.

Table 2: Minimum k as a fraction of network size for each seed selection heuristic.

Seed Selection Heuristic	Minimum k
Naïve node degree	27.6%
Ranked degree and threshold	27.8%
Rank-based social influence weight	26.7%
Max margin social influence weight	**26.3%**
Expected immediate adoption	26.8%

Decreasing the edges between agents from 8 to 6 only slightly affected the social influence threshold, which is calculated based on neighbor nodes. Removing more edges, thus isolating agents from their network would continue to negatively impact their productivity.

Similarly, decreasing the ratio of Engineers to Documenters also decreased the productivity, thus increasing the overall Combo agents. When Engineer to Documenter ratio was decreased to 5:1, the productivity dropped by 57%. The takeaway from this result is that the more Engineers a single Documenter can support, the better the network will run, but a realistic threshold needs to be found for a balanced ratio. In the real world as well, a company may place an upper limit on how many Documenters they can employ while still requiring Combo agents to maintain both tasks.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we showed three different methods of diffusing good behavior in a network. In the time-based model, we showed that max margin social influence weight is the best heuristic for selecting seed nodes to work, selecting 26.3% of the network. For the task-based model, we showed that selecting seed nodes based on team size is best, requiring 29.7% of nodes. The role-based model shows that for a small percentage of dedicated documenting agents, approximately 4%, and a portion of agents who are able to be influenced, we achieve the desired company wide documentation behavior. All three models can potentially guide network administrators desiring to diffuse good documentation behavior.

In the future, we plan to experiment with other networks in addition to the current Barabási-Albert model. For the time-based model, methods more sophisticated than heuristics can be used to discover better ways to diffuse behavior. Additionally, we plan to run simulations to determine the performance for varied average team sizes. For the task-based model, we hope to explore the question as to what is the expected time it will take for the network to adopt good documentation behavior. We also hope to test the heuristic in a real world company setting. For the role-based model, we plan to explore the ratio of combo agents to others agent types.

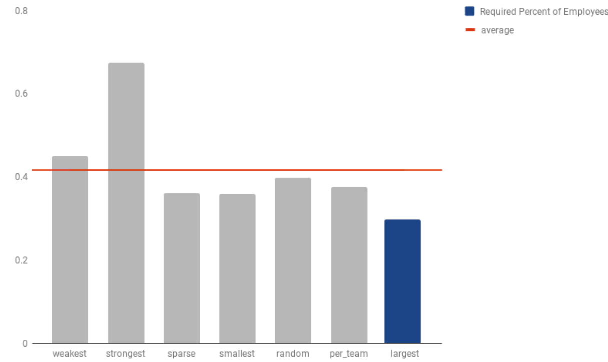

Figure 7: Largest Cluster Priority heuristic requires the minimum % of company to be given incentive to document.

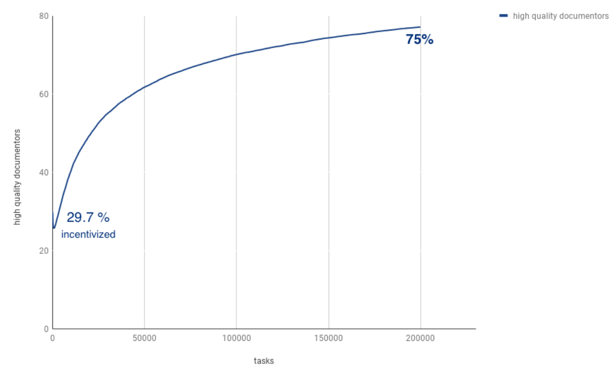

Figure 8: Largest Cluster Priority heuristic at each iteration

7. REFERENCES

[1] G. A. Akerlof. Procrastination and obedience. *The American Economic Review*, 81(2):1–19, 1991.

[2] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002.

[3] R. S. Baron, J. A. Vandello, and B. Brunsman. The forgotten variable in conformity research: Impact of task importance on social influence. *Journal of Personality and Social Psychology*, 71(5):915, 1996.

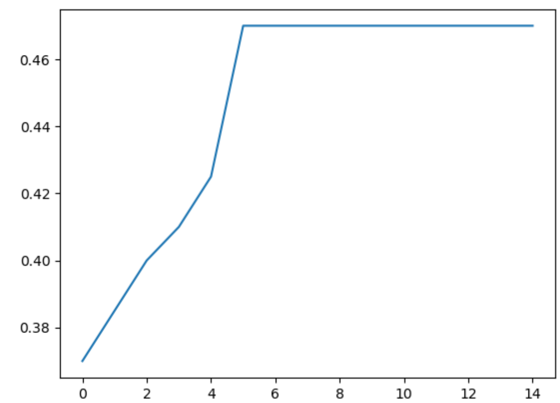

Figure 9: Role-based model with productivity output at only 47%. Documenters = 5, Engineer to Documenter ratio 10:1, in a 200 agent simulation.

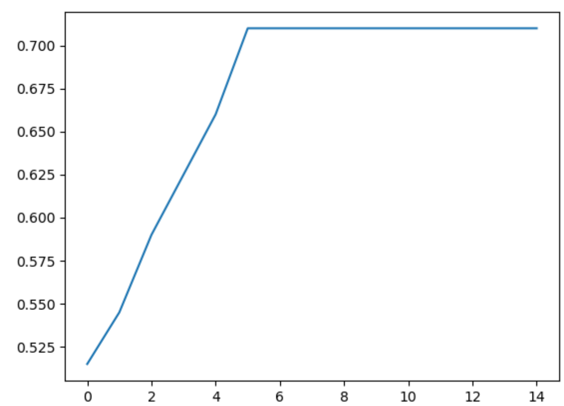

Figure 10: Role-based model with productivity output at 73%. Documenters = 8, Engineer to Documenter ratio 10:1, in a 200 agent simulation.

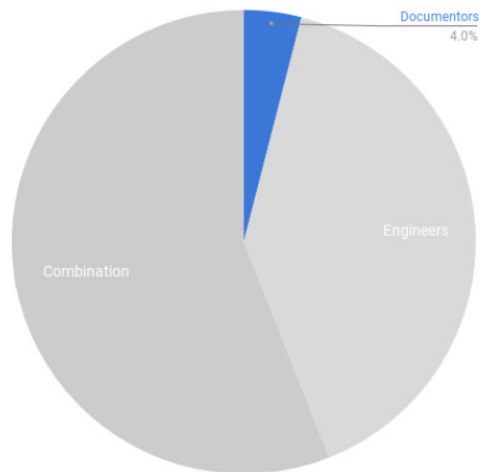

Figure 11: Division of well producing output at 73% with 4% Documenters, 40% Engineers and 56% Combo agents.

[4] B. Bollobás. Random graphs. In *Modern Graph Theory*, pages 215–252. Springer, 1998.

[5] N. S. Contractor and L. A. DeChurch. Integrating social networks and human social motives to achieve social influence at scale. *Proceedings of the National Academy of Sciences*, 111(Supplement 4):13650–13657, 2014.

[6] B. Dagenais and M. P. Robillard. Creating and evolving developer documentation: Understanding the decisions of open source contributors. In *Proceedings of the Eighteenth ACM SIGSOFT International Symposium on Foundations of Software Engineering*, FSE '10, pages 127–136, New York, NY, USA, 2010. ACM.

[7] A. Forward and T. C. Lethbridge. The relevance of software documentation, tools and technologies: a survey. In *Proceedings of the 2002 ACM symposium on Document engineering*, pages 26–33. ACM, 2002.

[8] M. O. Jackson and L. Yariv. Collective dynamic choice: the necessity of time inconsistency. *American Economic Journal: Microeconomics*, 7(4):150–178, 2015.

[9] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003.

[10] J. Kleinberg and S. Oren. Time-inconsistent planning: a computational problem in behavioral economics. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pages 547–564. ACM, 2014.

[11] T. C. Lethbridge, J. Singer, and A. Forward. How software engineers use documentation: The state of the practice. *IEEE software*, 20(6):35–39, 2003.

[12] M. L. Markus and C. Tanis. The enterprise systems experience-from adoption to success. *Framing the domains of IT research: Glimpsing the future through the past*, 173:207–173, 2000.

[13] K. Sarkar and H. Sundaram. Influencing busy people in a social network. *PloS one*, 11(10):e0162014, 2016.

[14] D. Spinellis. Code documentation. *IEEE software*, 27(4):18–19, 2010.

[15] C. J. Stettina and W. Heijstek. Necessary and neglected?: an empirical study of internal documentation in agile software development teams. In *Proceedings of the 29th ACM international conference on Design of communication*, pages 159–166. ACM, 2011.