Design of Multi-Agent Systems - B20
Gossip Simulations with Semi-Random Strategies

Arjan Jawahier (s2762161)
Roeland Lindhout (s2954524)
Xabi Krant (s2955156)
Niek Naber (s2515970)

Beta version, October 13, 2019

Abstract

The abstract should briefly summarize your project in 150–250 words.

1 Introduction

In this paper we will consider communication problems between agents. Several problems can be found in the literature, such as the marble problem [Lan54], or the gossip problem [HMS72]. Both problems are the same in essence, but in this paper we will focus on the latter, because of its simplicity.

1.1 Problem

The gossip problem is described as the following:

There are n ladies, and each of them knows some item of scandal which is not known to any of the others. They communicate by telephone, and whenever two ladies make a call, they pass on to each other, as much scandal as they know at that time. How many calls are needed before all ladies know all the scandal

We know that the minimum time needed when only 1 call can be made per time step, is $t(n) = 2n - 4$ [HMS72]. We will however consider a situation where multiple calls can be made per time step.

1.2 State of the art

The minimum completion time needed by a group of agents to complete this problem is τ where τ is the integer satisfying:

$$log_{r+1}n \leq \tau < 1 + log_{r+1}n \tag{1}$$

[Lan54]. In this formula n is the number of agents, and r stands for the number of concurrent connections they can make per time step. This formula considers one-way parallel communication per time-step. For $n > 4$, Landau could however not mention a set of rules so that the completion time of the task reduces to this minimal completion time. It is important to notice that this completion time can only be reached in the centralized version of the gossip problem, "where the protocols tell the agents whom they have to call" [Coo+19]. The opposite of this is the distributed version, where "individual agents, on the basis of their own information, decide which other agent to call" [Coo+19].

1.3 New idea

This paper will take a look at the parallel version of the gossip problem, with r (the number of messages per time step per agent) being one, and two-way communication. Two-way communication can be thought of in this case as one-way communication with the simple extra restriction that *if any agent a calls agent b to tell its secrets, agent b must also call agent a in that time-step to tell his secrets.* The new ideas proposed in this project are the introduction of semi-random strategies. Landau mentions in his paper that the minimum completion time is attainable by using the optimal strategy [Lan54]. This means that whenever the optimal strategy is not used, there is a large probability that the minimum completion time will not be reached. The optimal strategy Landau speaks of however, is the strategy where each agent knows which agent to speak to in order to reach that optimal time, and thus the centralized version of this problem. In real life, this sort of knowledge is usually unattainable. This paper will therefore investigate the distributed version of the gossip problem. The fact that r is chosen to be one, and communication is chosen to be two-way, resembles real life the best. When we consider agents to be humans or processors for example, only one connection can be made per time step, and information can easily be exchanged between two agents, making the type of communication two-way. Distributed strategies that decrease completion time of the gossip problem, can be used in systems like bitcoin to be more time efficient.
The methods proposed in this paper have to do with the introduction of semi-randomness in the strategies the agents make use of. In these strategies, agents use randomness as well as a set of rules when they consider whom to call. The effect of these new strategies, and some existing strategies, will be studied by measuring the time needed for the completion of the task, and comparing these times with the theoretical minimal completion time for one-way communication (equation 1). We can compare optimal completion times of two-way and one-way communication problems because they are very similar, as explained before in this section. In our two-way version, there is only one connection per agent possible. This makes it in essence possible to create the same communication graphs with one-way communication, as we create with two-way communications.

2 Method

2.1 Simulation model

The gossiping task can be represented by a graph containing nodes and edges. The nodes represent the agents and the edges represent the possible lines of communication between the agents. Landau talks about different kinds of graphs, where the number of edges is not equal for each node (1954). In this paper, only a fully connected graph structure is considered (see Figure 1). Each agent is then able to communicate with every other agent. This change was made because it seemed to model real world applications better (e.g spies in a spy network running around to pass on information to other spies).

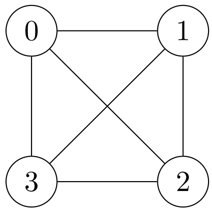

Figure 1: An example of a fully connected network of agents ($n = 4$). Each node represents an agent in the network. Each edge signifies a possible connection between agents, with which they can communicate their secrets to each other.

In Landau's work, computer simulations were performed using only 3 to 6 agents [Lan54]. The reason for this is that the computers of those times were not nearly as fast as the computers in our time (2019). This project will also handle increased numbers of agents.

Simulations will be run using $n = 10$, $n = 100$ and $n = 500$ agents. Different communication strategies will be implemented in each agent for different simulations. There exist a few general rules which remain the same in all simulations:

- Agents can make only $r = 1$ connection per time step to another agent.
- Agents always use two-way communication
- If an agent is already called by another agent, it loses the ability to call another agent that time-step, because it already has a connection.
- To decide which agent can make the first call in a time-step, the order of a list of agents is randomized. The first agent in this randomized list makes a connection with the agent of his choice. Then these two agents that now have a connection cannot call again in this time-step. Now the next agent in the randomized list can make a connection. This process repeats itself until all agents (or in the case of an odd number of agents: all agents but one) have a connection with another agent. It should be noted that some strategies might restrain agents from calling other agents. In a case such as this, not all agents will have a connection with another agent during a time-step.
- All agents are initialized with an id. An agent can be called by this id. Id's go from 0 to n-1 where n is the total number of agents.

2.1.1 Strategies

Multiple strategies will be used in the experiments. We use already existing strategies to compare to our devised found strategies. Existing strategies that are used can be found in Table 1. Table 2 shows all new strategies that were implemented.

Some strategies use the number of secrets that an other agents knows. This number is an estimation based on what agents pass along. When an agent communicates with another agent they know the number of secrets they both know. This is stored in their memory. When the agent then communicates with another new agent, they can also pass along this number.

Abbreviation	Name	Explanation Agent a calls...
R	Random	a completely random agent b
CO	Call me once	a random agent b iff he has not called b before
LNS	Learn new secrets	a random agent b of which he does not know the secret yet
TOK	Token	random agent b iff agent a has a token. The token is given to agent b.
SPI	Spider	random agent b iff agent a has a token. The token of agent b is taken by agent a.

Table 1: All existing strategies with their abbreviation and explanation

The maximum number of secrets for a specific agent is taken, as the number of secrets can not decrease.

2.2 Implementation details

The simulations are done using a program written in Python 3.7. The libraries used are: NetworkX, to generate graphs easily, Dash to view and update the graph with ease and pandas to store the results of the experiments in an easy to use DataFrame format.

2.3 Experiment design

For every strategy, 1000 simulations are run with $n = 500$, $n = 100$, $n = 50$ and $n = 10$ agents. In each set of simulations every agent will have the same strategy. Completion time is measured in time-steps taken.

To complete the gossiping task the fastest, agents should talk to many different agents. If each agent were to keep connecting to the same set of agents, no new secrets would be communicated. The *Random* strategy is already expected to be performing quite well. It is also expected to be hard to improve upon this random strategy. Essentially, the *Random* strategy is one of the most unrestrained strategies in the list of strategies, given in Tables 1 and 2.

3 Results

3.1 Experiment findings

The results of all strategies that we used can be found in Table 3. The comparison between the strategies and the optimal number of time-steps, calculated from equation 1 [Lan54], can be found in Table 4. Figure 2 shows all results from Table 3 together with τ_{opt} for different sizes of n. Figure 3 shows the results from the best 6 strategies in Table 3 together with τ_{opt} for different sizes of n.

3.2 Interpretation of findings

If we consider all of the simulated strategies, the ratio between τ and τ_{opt} varies from 142 to over 2000 percent for $n = 500$. There are 5 strategies that result in a $\frac{\tau}{\tau_{opt}}$ ratio below 1.5:

Abbreviation	Name	Explanation
TOKI	Token improved	a variation on **TOK**, wherein agents that know all secrets are not called again
SPII	Spider improved	a variation on **SPI**, wherein agents that know all secrets are not called again
MAT	Math	agent a calls an agent b with id corresponding to formula $callID = oId * (t+1) \ \% \ n$. If however agent already has another connection, agent a will increment the id of the agent he wants to call by one until he finds an available agent.
BUB	Bubble	In the bubble method groups are formed in which each agent knows the same secret. This bubble is expanded in each time step. Agents that do not fit into a bubble call a random available agent. The agents call the agent with their id plus $2^{timestep}$. This results in bubbles of $2^{timestep}$ agents, which know all secrets of the agents in their timestep.
CMINS	Call Min Secrets	agent a calls agent b with the lowest amount of secrets, that has not been called yet
CMAXS	Call Max Secrets	agent a calls agent b with the lowest amount of secrets, that has not been called yet
CBS	Call Best Secrets	agent a calls agent b if agent b has the most secrets and has not been called in the past 5 timesteps. If agent a knows all secrets, agent a calls agent b with the fewest secrets

Table 2: All newly implemented strategies with their abbreviation and explanation

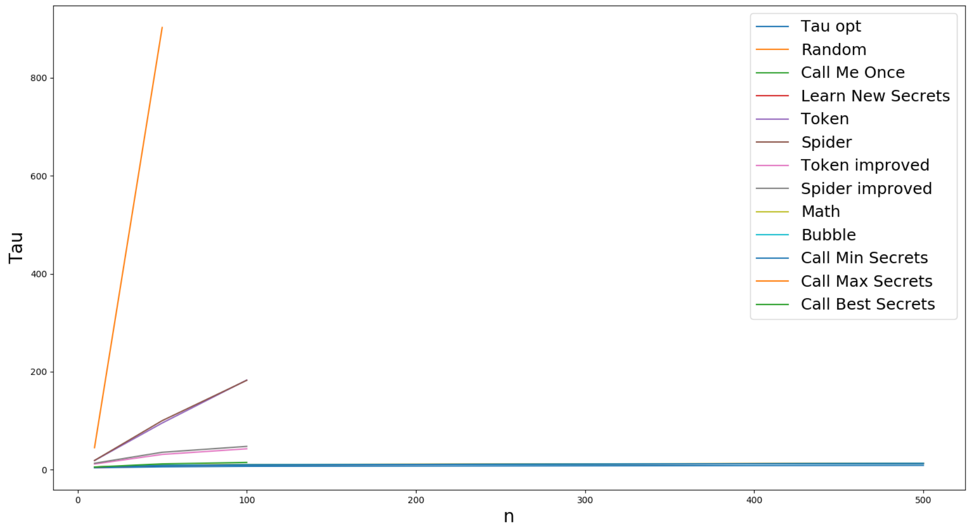

Figure 2: Average completion time for strategies, for different numbers of n. For every point, 1000 simulations were run. The first line in the graph is the theoretical optimal completion time.

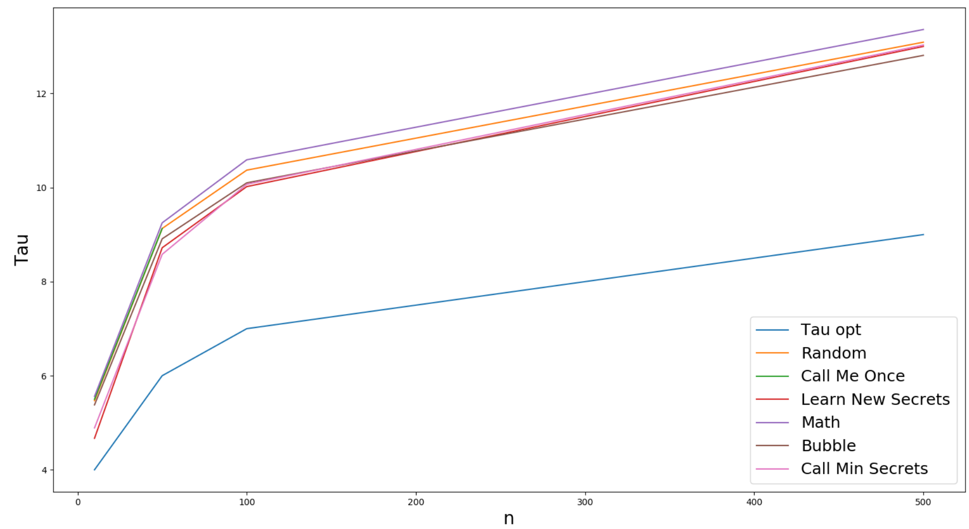

Figure 3: Average completion time for top 6 performing strategies, for different numbers of n. For every point, 1000 simulations were run. The first line in the graph is the theoretical optimal completion time.

Strategy	mean ± sd τ n = 10	mean ± sd *tau* n = 50	mean ± sd τ n = 100	mean ± sd τ n = 500
Random	5.47 ± 0.82	9.13 ± 0.48	10.37 ± 0.5	13.09 ± 0.29
Call Me Once	5.50 ± 0.82	9.13 ± 0.46	10.37 ± 0.49	TBD
Learn New Secrets	4.67 ± 0.51	8.72 ± 0.45	10.02 ± 0.13	13.0 ± 0.045
Token	18.76 ± 9.71	95.35 ± 54.98	183.01 ± 112.76	TBD
Spider	18.96 ± 10.07	100.19 ± 59.45	182.54 ± 112.27	TBD
Token improved	11.63 ± 2.43	31.28 ± 4.68	42.77 ± 5.15	TBD
Spider improved	13.29 ± 3.97	35.65 ± 7.1	47.69 ± 6.74	TBD
Math	5.56 ± 0.91	9.25 ± 0.57	10.59 ± 0.69	13.36 ± 0.58
Bubble	5.38 ± 0.74	8.91 ± 0.53	10.1 ± 0.38	12.81 ± 0.41
Call-Min-Secrets	4.89 ± 0.57	8.58 ± 0.51	10.07 ± 0.27	13.03 ± 0.17
Call-Max-Secrets	45.05 ± 30.37	902.46 ± 375.85	TBD	TBD
Call-Best-Secrets	5.59 ± 0.84	12.02 ± 0.87	14.94 ± 0.78	21.56 ± 0.66

Table 3: Table that displays the mean completion times of all strategies, shown for $n = 10$, $n = 50$, $n = 100$ and $n = 500$.

Random, Learn New Secrets, Math, Bubble, and *Call Min Secrets* (these strategies can be found in the graph in Figure 3. The random strategy is also in the top 5, and compared to this strategy, there was a significant 3 percent gain possible in the $\frac{\tau}{\tau_{opt}}$ ratio (p ¡ 0.001) for the *Bubble* strategy. As mentioned in section 2.3, it was expected that the *Random* strategy is a good strategy, which is hard to improve upon. The newly devised strategies, as well as the already existing strategies that were used in this experiment all aim to fan out the agents to which a single agent connects as much as possible. The *Random* strategy apparently already does a good job at this.

4 Conclusion

4.1 Discussion

The strategies in this experiment turn out not to have a big advantage over the *Random* strategy. The best performance was the *Bubble* strategy which had a 3 % $\frac{\tau}{\tau_{opt}}$ ratio increase. It is suspected that the top 5 strategies resemble the *Random* strategy too much. In order to create a strategy that would increase performance even more, one would have to think about characteristics not only based on dividing connections in different time-steps amongst agents. If only this is considered, as it was in the strategies in this experiment, the strategy is similar to the *Random* strategy and has a similar performance. We can also conclude that the *Random* strategy is already a pretty good strategy, which is of course the least intensive computationally.

4.2 Relevance

The findings in this paper are relevant for systems like blockchain, where several nodes or agents need to communicate with each other to share knowledge. If, for such a system, a strategy needs to be chosen where computation time must be kept as low as possible, random strategy

Strategy	τ/τ_{opt} n = 10	τ/τ_{opt} n = 50	τ/τ_{opt} n = 100	τ/τ_{opt} n = 500
Random	136.75%	152.17%	148.14%	145.44%
Call Me Once	137.5%	152.17%	TBD	TBD
Learn New Secrets	116.75%	145.33%	143.14%	144.44%
Token	469%	1589.17%	2614.43%	TBD
Spider	474%	1669.83%	2607.71%	TBD
Token improved	290.75%	521.33%	611%	TBD
Spider improved	332.25%	594.17%	681.29%	TBD
Math	131.5%	153.83%	149.71%	148.44
Bubble	134.5%	148.5%	144.29%	142.33
Call-Min-Secrets	122.25%	143%	143.86%	144.78%
Call-Max-Secrets	150.41%	10518.18%	TBD	TBD
Call-Best-Secrets	139.75%	200.33%	207%	239.56%

Table 4: Table that displays the ratio of the average completion time (τ) over the optimal completion time (τ_{opt}), shown for n = 10, n = 50, n = 100 and n = 500. τ_{opt} = 4, 6, 7 and 9 for respectively n = 10, 50, 100 and 500. n stands for the number of agents in the simulation.

should be chosen, because it is almost the best one, and requires almost no computational resources, whereas the rest of the strategies needs to have some sort of representation of secrets of others, and/or computation about which agent to connect with.
On the other hand we suspect that there are strategies that would improve the completion time of the random strategy more than 3 %, and maybe even approximate the minimal completion time. Different strategies need to be thought of, that are using techniques other than techniques that resemble random picking and dividing connections. This should be determined by future research.

4.3 Team Work

References

[Coo+19] Martin C. Cooper, Andreas Herzig, Faustine Maffre, Frédéric Maris, and Pierre Régnier. “The epistemic gossip problem”. In: *Discrete Mathematics 342 654-663* (2019).

[HMS72] A. Hajnal, E. C. Milner, and E. Szemerédi. “A Cure for the Telephone Disease”. In: *Canadian Mathematical Bulletin (447-450* 15.3 (1972). DOI: 10.4153/CMB-1972-081-0.

[Lan54] G. H. Landau. “The distribution of completion times for random communication in a task-oriented group”. In: *The bulletin of mathematical biophysics v16 n3 (195409): 187-201* (1954). URL: https://link-springer-com.proxy-ub.rug.nl/content/pdf/10.1007%5C%2FBF02478413.pdf.