



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Complex Social Systems: Modeling Agents, Learning, and Games
HS2022

Project Report

Overcoming Braess's Paradox: A Modelling based Approach

Nikhil Sethukumar, Vishnu Varadan

Zurich
December 20, 2022

Agreement for free-download

We hereby agree to make our source code for this project freely available for download from the web pages of the SOMS chair. Furthermore, we assure that all source code is written by ourselves and is not violating any copyright restrictions.

Nikhil Sethukumar

Vishnu Varadan



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

OVERCOMING BRAESS'S PARADOX: A MODELLING BASED APPROACH
(Project work for Complex Social Systems: Agent-based Modelling HS2022)

Authored by (in block letters):

For papers written by groups the names of all authors are required.

Name(s):

SETHUKUMAR

VARADAN

First name(s):

NIKHIL

VISHNU

With my signature I confirm that

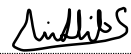
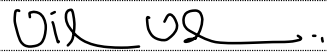
- I have committed none of the forms of plagiarism described in the '[Citation etiquette](#)' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date

Zurich, 20-12-2022

Signature(s)

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.

Contents

1	Abstract	5
2	Individual contributions	5
3	Introduction and Motivations	5
3.1	Mathematical Preliminaries of Game Theory	5
3.2	Braess's paradox	5
3.3	Relevance of Braess's paradox and Previous Works	6
4	Description of the Model	6
4.1	Problem setup	6
4.2	Setting up the game	6
4.3	Preliminary Agent Model	8
4.4	Centralized Scheduler	9
4.5	Dynamic Agent Model	10
4.6	Distributed Architecture	11
5	Simulation Results and Discussion	12
5.1	Centralized Scheduler	13
5.2	Distributed Architecture	13
6	Conclusion and Outlook	16

1 Abstract

The selection of route in a congestion game depends mainly on the density of traffic over the available routes. Selfish route choice does not always result in a lower time taken for an agent to get from the source to the destination when compared to strategies that are formed by cooperation by agents. Moreover, extension of the road network to redistribute the traffic flow can result in higher travel times for the agents leading to a paradoxical situation. In this work we investigate if an agent can be modelled such that the paradoxical effect can be mitigated in a distributed way with minimal coordination between the agents.

2 Individual contributions

Nikhil was responsible for designing the Game, code the models in an object oriented manner in Python, for the Presentation and Report. He was also responsible for maintaining the repository on GitHub.

Vishnu came up with the models to mitigate the effect of the paradox. He also wrote MATLAB Code and performed Data Collection and was also responsible for the Presentation and the Report.

3 Introduction and Motivations

3.1 Mathematical Preliminaries of Game Theory

Game theory provides a mathematical model for studying interactions between agents with a selfish interest [3]. We consider $i = 1 \dots n$ agents interacting, each with an action $a_i \in \mathcal{A}_i$. The cost incurred by agent i is $J_i(a_1 \dots a_n)$, which is what they try to minimize. This may be extended to multi-stage games by assuming action sets particular to a stage for each agent, and costs that can vary with the stage.

The game is said to be in a *Nash Equilibrium* when the set of actions of all agents $\{a_1^*, \dots, a_n^*\}$ are such that no one can unilaterally reduce their cost by changing their action. In other words, Equation (1) is satisfied.

$$\text{for each agent } i, \quad a_i^* = \underset{a_i}{\operatorname{argmin}} J_i(a_i, a_{-i}) \quad (1)$$

where a_{-i} represents the actions of the other agents $\{a_1 \dots a_{i-1}, a_{i+1} \dots a_n\}$.

Social cost is a number that indicates the welfare of all the agents. It could, for example, be the sum of costs of all agents. The *price of anarchy* indicates how worse off the Nash Equilibrium is as compared to the case with the least social cost ($\{a'_1, \dots, a'_n\}$). They can, for example be defined as in Equation (2).

$$\text{Social Cost} = \sum_{i=1}^n J_i(a_i, a_{-i}) \quad \text{and} \quad \text{PoA} = \frac{\sum_{i=1}^n J_i(a_i^*, a_{-i}^*)}{\sum_{i=1}^n J_i(a'_i, a'_{-i})} \quad (2)$$

3.2 Braess's paradox

Congestion games are a special class of games in Game Theory, where the cost for each agent depends on the resources it chooses, which in turn varies with the number of agents choosing that

resource. Traffic flow can be modelled as a congestion game where agents have to get from a start node to an end node. They can choose to pass through one of two intermediate nodes, with one leg of each path having travel time proportional to the number of agents using the leg. In this scenario, for specific choices of travel time over each connection, adding a connection of 0 travel time between the intermediate nodes can increase the Nash Equilibrium travel time. This is called Braess’s paradox [1].

3.3 Relevance of Braess’s paradox and Previous Works

Braess’s paradox is a counter-intuitive result and has occurred in real-life. One such instance is mentioned in a newspaper:

“ON Earth Day this year, New York City’s Transportation Commissioner decided to close 42d Street, which as every New Yorker knows is always congested. . . . But to everyone’s surprise, Earth Day generated no historic traffic jam. Traffic flow actually improved when 42d Street was closed.

To mathematicians, this may be a real-world example of Braess’s paradox, a statistical theorem that holds that when a network of streets is already jammed with vehicles, adding a new street can make traffic flow even more slowly.” [4]

In a more theoretical aspect, the papers [8] and [6] explore the conditions in which the paradoxical situation might arise. Previous works have outlined the negatives of selfish routing and highlighted the importance of cooperative behaviour among the agents [7]. The work carried out in [2] especially focuses on the emergence of cooperation between agents in the selection of routes in a repeated game setup. The paper [5] introduces a currency like mechanism called as “reputation” to overcome a similar problem that arises in a multistage tragedy of commons game setup.

4 Description of the Model

4.1 Problem setup

Let us begin with the basic problem formulation and then choose numbers that satisfy our conditions. Let us consider the case of crossing the Grand Canal in Venice. Let us assume that the Canal has dried up a bit beneath the Rialto and people who want to cross it would have to either take a ferry before the Rialto and then walk via an alley, or walk via an alley and then take a ferry after the Rialto, or walk via alleys on both banks and take the Rialto to cross the Canal. This is illustrated in Figure 1.

A diagram for the paths available is shown in Figure 2. **A** is the start node and **D** is the end node. The nodes in between are **B** and **C**. We have named the paths too, for convenience: **AB** and **CD** are **ferry** paths, and **BD** and **AC** are **alley** paths. **BC** is the **bridge** path which is connected later and causes an increase in time taken for the Nash Equilibrium.

4.2 Setting up the game

We need to choose the parameters in Table 1 such that the all the agents take the alley-bridge-alley path at the Nash Equilibrium, and this is higher than the Nash Equilibrium cost for the case without the bridge. Therefore, the following costs need to be in descending order:

1. Time taken on the **ferry-bridge-ferry** route (2α)
2. Time taken on the **ferry-bridge-alley** route ($\alpha + \beta + \delta n_{\text{BD}}, \alpha + \beta + \delta n_{\text{AC}}$)

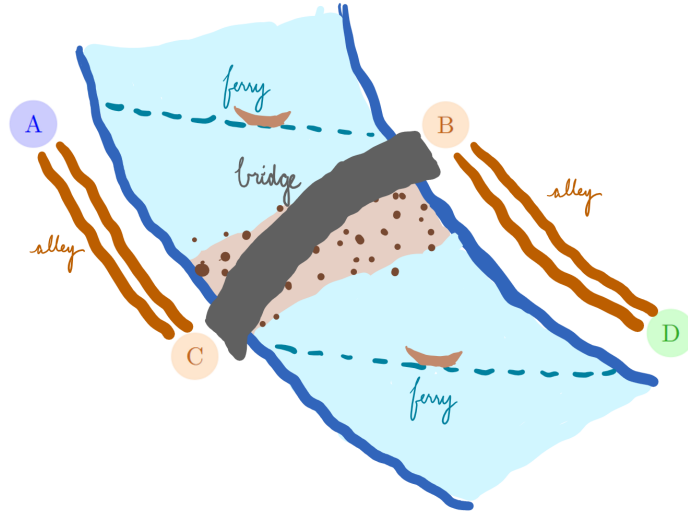


Figure 1: Illustration of paths available to cross the Grand Canal

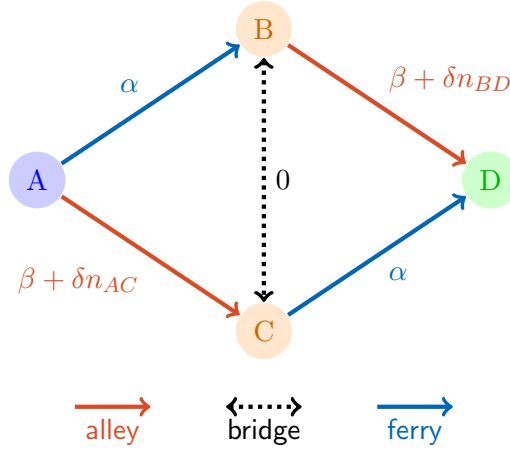


Figure 2: Basic framework for illustrating Braess's paradox

α	Time taken on the ferry
β	Time taken on the alley
δ	Extra time taken on the alley for every person on the alley
n	Number of agents in the game

Table 1: Parameters of the Braess's paradox game

3. Time taken on the **alley-bridge-alley** route ($2\beta + \delta(n_{\text{BD}} + n_{\text{AC}})$)
4. Time taken on the **alley-ferry** route in the case without the **bridge** connected

In the case without the bridge connected, let us consider unequal numbers of agents taking either of the routes **ABD** or **ACD**. Then the number of people on a particular route will be higher, increasing the time taken on that route. None of the agents would want to change routes when the cost on either route is equal, that is, the number taking either route is the same $= n/2$.

Furthermore, from the symmetry of the game, let us assume that the number of people taking **ferry-bridge-alley** and **alley-bridge-ferry** are the same, and therefore,

$$n_{\text{ABD}} = n_{\text{ACD}} \implies n_{\text{ABD}} + n_{\text{ACBD}} = n_{\text{ACD}} + n_{\text{ACBD}} \implies n_{\text{BD}} = n_{\text{AC}} = n'$$

Therefore we need the following inequality to be satisfied:

$$2\alpha > \alpha + \beta + \delta n' > 2\beta + 2\delta n' > \alpha + \beta + \delta \frac{n}{2}$$

which simplifies to

$$n' > \frac{n}{2} \ \& \ \alpha > \beta + \delta n' \ \& \ 2n' - \frac{n}{2} > \frac{\alpha - \beta}{\delta}$$

now allowing all the agents to take the **alley** routes (setting $n' = n$) we have

$$n < \frac{\alpha - \beta}{\delta} < \frac{3}{2}n$$

A set of parameters that obeys this rule is ($\alpha = 15, \beta = 2, \delta = 0.2, n = 60$). These parameters make sense in our setting of the Grand Canal crossing in Venice. We have also considered the parameter set ($\alpha = 40, \beta = 15, \delta = 0.1, n = 200$). The costs and price of anarchy due to this setup is shown in Table 2.

	$(\alpha = 15, \beta = 2, \delta = 0.2, n = 60)$	$(\alpha = 40, \beta = 15, \delta = 0.1, n = 200)$
Cost at NE without bridge (ABD , ACD)	(23, 23)	(65, 65)
Cost at NE (ABD , ACD , ABCD , ACBD)	(29, 29, 30, 28)	(75, 75, 80, 70)
Price of Anarchy	$\frac{28}{22.96} = 121.95\%$	$\frac{70}{64.375} = 108.74\%$

Table 2: Costs and Price of Anarchy with different parameter choices

We consider the infinite repetition of the above described game in which the route selection is posed infinitely over multiple stages. For simulation purposes and developing the algorithm, we consider a high number K that represents a finitely large number of stages.

4.3 Preliminary Agent Model

Initially we model all the agents to be identical and possess two different parameters namely,

- **Selfishness** γ : $\gamma \in [0, 1]$ defines how selfish an agent is. $\gamma = 0$ indicates that the agent is completely selfish while $\gamma = 1$ indicates that the agent is completely benevolent.
- **Connectivity** $|\mathcal{N}|$: Connectivity defines how well-connected an agent is. \mathcal{N} is a set that comprises of all the the “neighbours” of the agent. The higher the number of neighbours, the more well-connected the agent is considered to be.

We can model the cost that each agent is trying to minimize at stage s of the game to be,

$$J_i(s, a_1, \dots a_n) = (1 - \gamma)C_i(s, a_1, \dots a_n) + \gamma \sum_{l \in \mathcal{N}} C_l(s, a_1, \dots a_n) \quad (3)$$

where $C_i(s, a_1, \dots a_n)$ represents the time taken by an agent to reach the destination from the source, given the actions of all the agents $a_1, \dots a_n$.

4.4 Centralized Scheduler

The Centralized Scheduler (CS) has been modelled as an entity, different from the agents that play the game, to regulate the state of affairs. The two major responsibilities of the Centralized Scheduler are

- to decide the γ and $|\mathcal{N}|$ values for the agents
- to schedule/prioritize the agents based on the cost they incur at every stage of the game

Figure 3a represents how the communication would happen in the presence of a Centralized Scheduler. The agents would all communicate with the CS and there would be no inter-agent communication. Figure 3b on the other hand represents the information that is actually exchanged between the agent and the CS. While the CS transmits the selfishness, connectivity and the time that all the neighbours of an agent would collectively take in order to reach the destination to the agents, each agent would in turn transmit the cost it incurs to the CS.

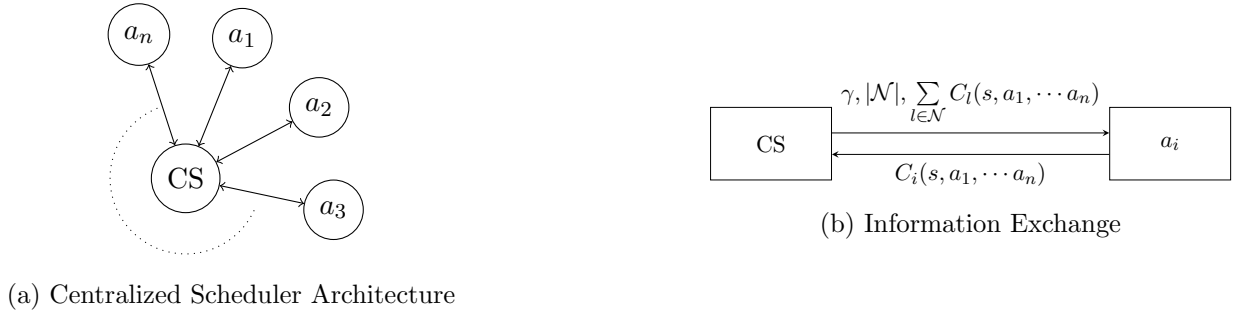


Figure 3: Architecture and information exchange for the Centralized Scheduler

Algorithm 1 $C(a_1, \dots, a_n) = \text{CentralizedSchedulerArchitecture}(\gamma, |\mathcal{N}|, \sum_{l \in \mathcal{N}} C_l(a_1, \dots, a_n))$

```

1: Receive  $\gamma$  and  $\mathcal{N}$  from CS
2: for  $s = 1 \dots K$  do
3:    $prev_{a_i} \leftarrow \text{None}, \forall i \in [n]$ 
4:    $a_i \leftarrow \text{ABD}, \forall i \in [n]$ 
5:   while  $prev_{a_i} \neq a_i$  do
6:     for  $i = 1 \dots n$  do
7:        $prev_{a_i} \leftarrow a_i$ 
8:        $a_i = \underset{a_i}{\operatorname{argmin}} (1 - \gamma)C_i(s, a_1, \dots, a_n) + \gamma \sum_{l \in \mathcal{N}} C_l(s, a_1, \dots, a_n)$ 
9:     end for
10:  end while
11:   $C(a_1, \dots, a_n)[s][i] \leftarrow C_i(s, a_1, \dots, a_n)$ 
12:  CS sorts  $C_i(s, a_1, \dots, a_n)$  values in descending order
13:  CS assigns  $i$  based on the sorted order
14: end for

```

The above Algorithm 1 illustrates the working of the CS based setup. Initially, the CS decides the parameters for the agents that are fixed for all stages of the game. Then, at each stage of the game, the agents initialize to random actions and try to choose the action that minimizes their costs while constantly exchanging information with the CS. Once all the agents iterate and finalize their actions, they incur a cost which they communicate to the CS. The CS then orders the agents such that at the next stage of the game $s + 1$, the agents with a high cost at the current stage s get the opportunity to decide their action first.

4.5 Dynamic Agent Model

Although the Centralized Scheduler and the Preliminary Agent Model work in overcoming the paradox on average as we would see in Section 5, certain characteristics of this model make it undesirable. For instance, all the decisions are controlled by one entity, which is the CS in the previous model. Moreover, the agents are not allowed to act independently and choose their own actions. Finally, the agents may simply not follow the directions obtained from the CS and thus the model might not work practically. For this reason, we desire a distributed model that overcomes all the said shortcomings, while still trying to overcome the paradox which is our primary goal.

We propose the Dynamic Agent Model in which the agents have full control over the parameters introduced in Section 4.3. In the absence of the Centralized Scheduler, the aim of this model is for the agents to update their selfishness $\gamma_i(t)$ and connectivity parameters $\mathcal{N}_i(t)$ over the stages of the game in a meaningful way, while also making sure that all the agents get a fair chance to overcome the paradoxical situation.

Intuitively, it is easy to see that if an agent i experiences a “high” cost at stage $s - 1$, then it indicates that the agent has not been “selfish” enough in $s - 1$. Thus, for the agent to be benefited at stage s , it needs to reduce its selfishness and connectivity parameters. While on the other hand if the agent incurs only a “low” cost at stage $s - 1$, then the agent has been more selfish in that stage and in order to give a fair chance to all the other agents, the agent should decide to be more benevolent and more connected to its neighbours in the upcoming stage than the current stage.

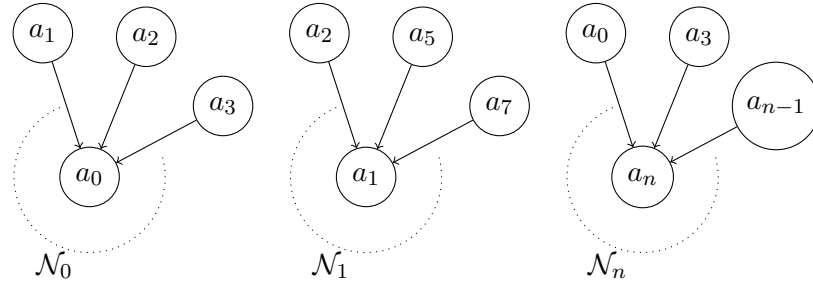
A working heuristic for the dynamics of the parameters is

$$\begin{aligned}
&\text{If } J_i(t-1) > \tau \\
&\quad \gamma_i(t) = \max(0, \gamma_i(t-1)/4) \\
&\quad |\mathcal{N}_i(t)| = \max(0, |\mathcal{N}_i(t-1)|/4) \\
&\text{Else} \\
&\quad \gamma_i(t) = \min(1, 2\gamma_i(t-1)) \\
&\quad |\mathcal{N}_i(t)| = \min(n, 2|\mathcal{N}_i(t-1)|)
\end{aligned} \tag{4}$$

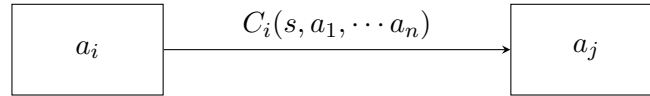
where τ is the cost threshold that indicates if a cost incurred is high or low. An “informed” decision of τ can be made by the agents based on the NE cost they used to incur prior to the construction of the bridge.

4.6 Distributed Architecture

As remarked in Section 4.5, the centralized scheduler and approach have some limitations to be addressed, leading to a distributed approach. As illustrated in Figure 4 in this architecture, the an agent i receives information from its neighbours in \mathcal{N}_i directly, while transmitting the time they would take to reach the destination to all the agents that consider agent i to be their neighbour. It is to be noted that an agent i need not necessarily consider another agent j to be its neighbour, just because agent j considers agent i to be its neighbour.



(a) Distributed Architecture



(b) Information Exchange in Distributed Architecture

Figure 4: Details of the Distributed Architecture

Algorithm 2 $C(a_1, \dots, a_n) = \text{DistributedArchitecture}(\tau)$

```
1: Initialize  $\gamma_i(1) \sim \mathcal{U}(0, 1), \forall i \in [n]$ 
2: Initialize  $|\mathcal{N}_i(1)| \sim \mathcal{U}(0, n), \forall i \in [n]$ 
3: for  $s = 1 \dots K$  do
4:    $prev_{a_i} \leftarrow \text{None}, \forall i \in [n]$ 
5:    $a_i \leftarrow \text{ABD}, \forall i \in [n]$ 
6:   if  $s > 1$  then
7:     if  $J_i(s-1, a_1, \dots, a_n) > \tau$  then
8:        $\gamma_i(t) = \max(0, \gamma_i(t-1)/4)$ 
9:        $|\mathcal{N}_i(t)| = \max(0, |\mathcal{N}_i(t-1)|/4)$ 
10:    else
11:       $\gamma_i(t) = \min(1, 2\gamma_i(t-1))$ 
12:       $|\mathcal{N}_i(t)| = \min(n, 2|\mathcal{N}_i(t-1)|)$ 
13:    end if
14:  end if
15:  while  $prev_{a_i} \neq a_i$  do
16:    for  $i = 1 \dots n$  do
17:       $prev_{a_i} \leftarrow a_i$ 
18:       $a_i = \underset{a_i}{\operatorname{argmin}} (1 - \gamma)C_i(s, a_1, \dots, a_n) + \gamma \sum_{l \in \mathcal{N}} C_l(s, a_1, \dots, a_n)$ 
19:    end for
20:  end while
21:   $C(a_1, \dots, a_n)[s][i] \leftarrow C_i(s, a_1, \dots, a_n)$ 
22: end for
```

The above Algorithm 2 depicts the simulation of the system based on the dynamic agent model and the distributed architecture. We initialize the selfishness and connectivity parameters to a random value sampled from a uniform distribution. Then at each stage of the game, the agents update their parameters based on the dynamics described in equation 4; followed by which the agents iterate over their best responses until they converge to an action.

5 Simulation Results and Discussion

In this section we look at and compare the results obtained from the Centralized Scheduler and the Distributed Architecture. The parameters that we have used for the simulation are $(\alpha = 40, \beta = 15, \delta = 0.1, n = 200)$ with an NE cost of 65 without the bridge and 70 with the bridge, amounting to a Price of Anarchy of 108.74% as stated in Table 2. We simulate the multistage game for $K = 1000$ stages.

The two major values that we consider in our results are J_{avg} and $J_{i,avg}$. J_{avg} is the average cost incurred per agent per stage of the game. $J_{i,avg}$ is the average cost incurred by an agent i over the stages of the game. J_{avg} indicates the overall welfare of all the agents, whereas $J_{i,avg}$ indicates the welfare of each agent over the stages of the game. We would need the agents to have similar welfare and not favour any particular group of agents.

5.1 Centralized Scheduler

We study the results of the CS Architecture Algorithm depicted in Algorithm 1. Figure 5 shows the variation of J_{avg} with respect to γ and $|\mathcal{N}|$ values selected by the CS. It is clear that for low values of γ and $|\mathcal{N}|$, the average cost per agent per stage J_{avg} remains high at 70, which is the NE cost value. However the lowest value obtained is 64.375 for $\gamma = 1$ and $|\mathcal{N}| = 200$. This is lower than the value of 65 which is the NE value prior to the construction of the bridge. This result reaffirms the fact that if agents do not play selfishly, but rather coordinate with other agents they can overcome the paradox.

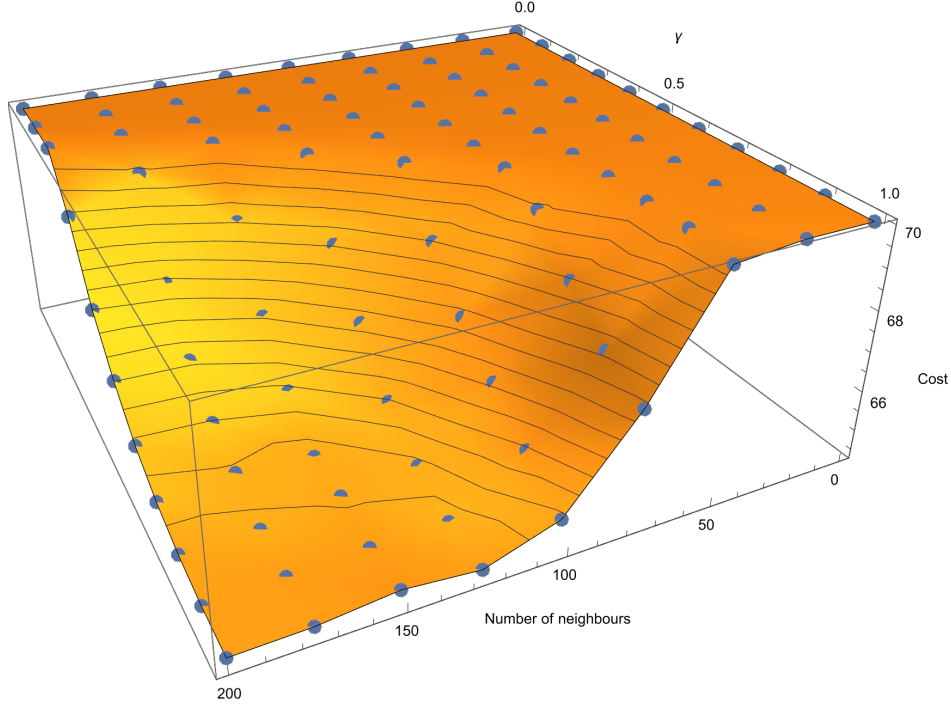
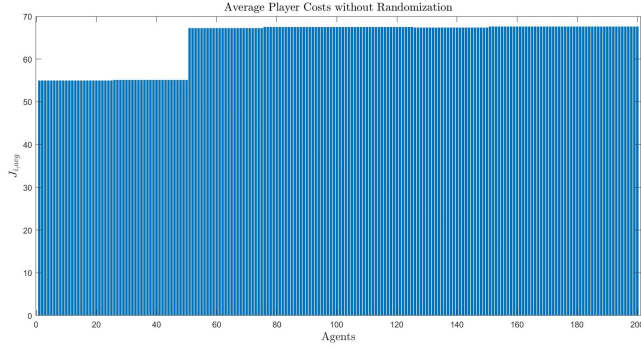


Figure 5: Variation of J_{avg}

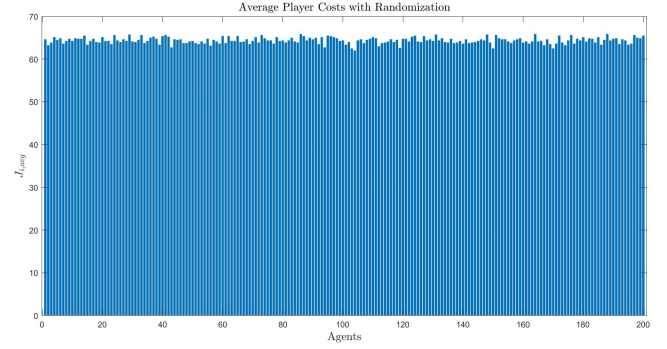
We then plot the average costs incurred by the agents over the stages of the game. Figure 6a shows the case when the CS inadvertently favours a group of agents while assigning the priority values when the agents incur the same cost. This effect can be mitigated by introducing a randomization when the CS assigns priority to agents that incur the same cost in the current stage of the game. Figure 6b shows that the agents incur similar average costs over the stages and as K increases, the difference in costs between the agents would reduce.

5.2 Distributed Architecture

In this subsection we study the results of the Distributed Architecture based algorithm that is outlined in Algorithm 2. After conducting Monte Carlo Study with 200 trials, Figure 7 shows the value of J_{avg} over the trials. It can be observed that $J_{avg} \sim \mathcal{N}(64.515, 0.016)$. The mean over the trials is 64.515 is lower than 65 which is the NE value prior to the construction of the bridge. This indicates that the paradoxical situation can be avoided with the proposed dynamic agent model on average over different stages of the game. We would also like to look at how the average agent costs is distributed over the trials. Clearly, from Figure 8, it can be inferred that the costs of agents are



(a) Without randomized priority in scheduling



(b) With randomized priority in scheduling

Figure 6: Variation in $J_{i,\text{avg}}$ based on scheduling priority

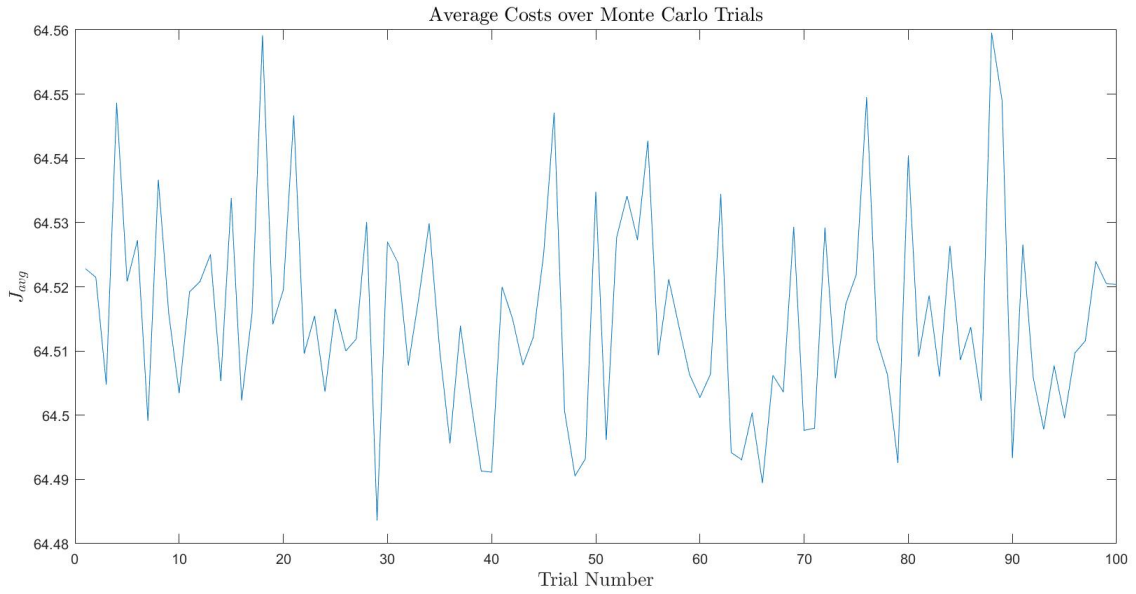


Figure 7: Average Costs - Results of Monte Carlo Trials with $\gamma_i(0) \sim \mathcal{U}(0, 1)$ and $|\mathcal{N}_i(0)| \sim \mathcal{U}(0, 200)$

distributed more or less uniformly except for some anomalies that arise due to the approximation to a finite number of stages. It would also be important to look into the dynamics of the agent

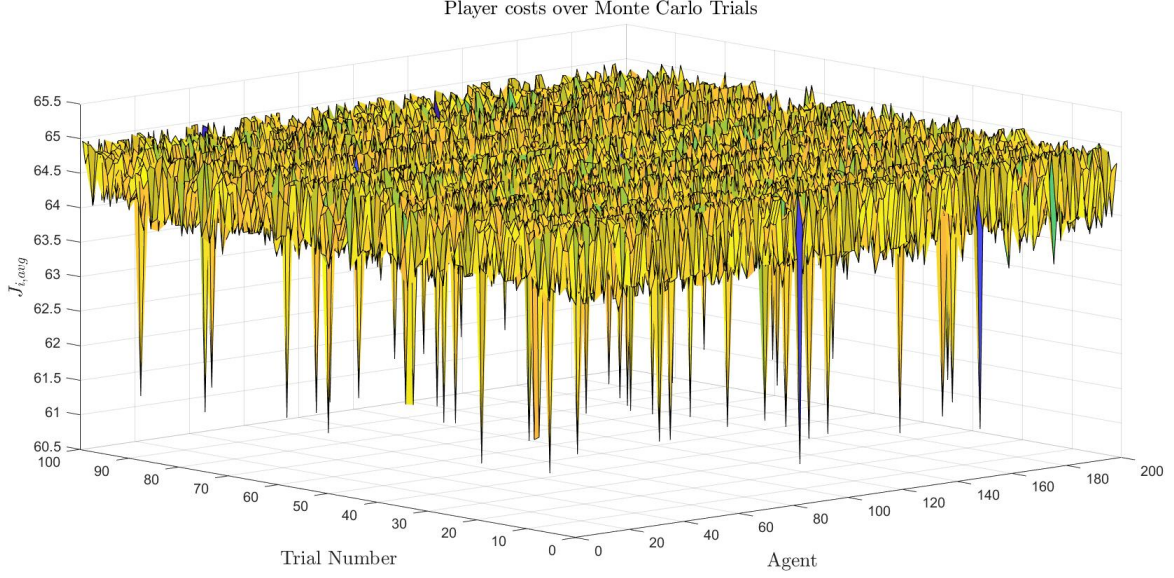
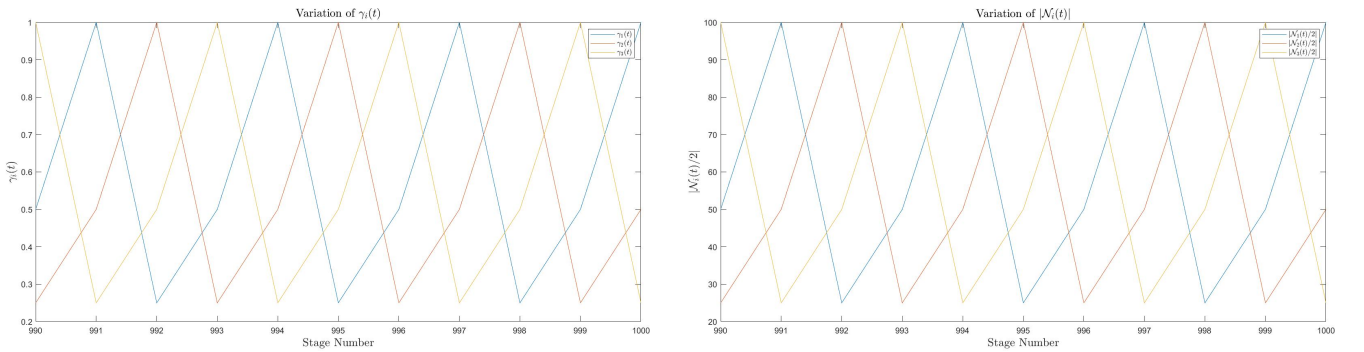


Figure 8: Average agent Costs - Results of Monte Carlo Trials with $\gamma_i(0) \sim \mathcal{U}(0, 1)$ and $|\mathcal{N}_i(0)| \sim \mathcal{U}(0, 200)$

parameters $\gamma_i(s)$ and $|\mathcal{N}_i(s)|$ and how it evolves over stages since they are specific to each agent. It is inferred from the observations that after a number of stages of the game has passed, most of the agents can be grouped together based on their parameter dynamics. Figure 9a shows the evolution of $\gamma_i(t)$ for three such groups. It can be seen that the values oscillate between 0.25, 0.5 and 1 and it converges to a “limit cycle” like behaviour. Similar is the case with the dynamics of $|\mathcal{N}_i(s)|$ depicted in Figure 9b. The final 11 stages are depicted in the figures for clarity.



(a) Variation of γ with stage

(b) Variation of $|\mathcal{N}|$ with stage

Figure 9: Variation of agent parameters with stage

Finally, we look at costs incurred by the three largest groups and we can see that they incur pretty low costs (lesser than 65) for two stages and one high cost (80) due to the agents being modelled to be reasonable. Clearly, we can see that the average costs are lower than 65 on average, thus reducing the time taken on average to get to the destination from the source after the construction

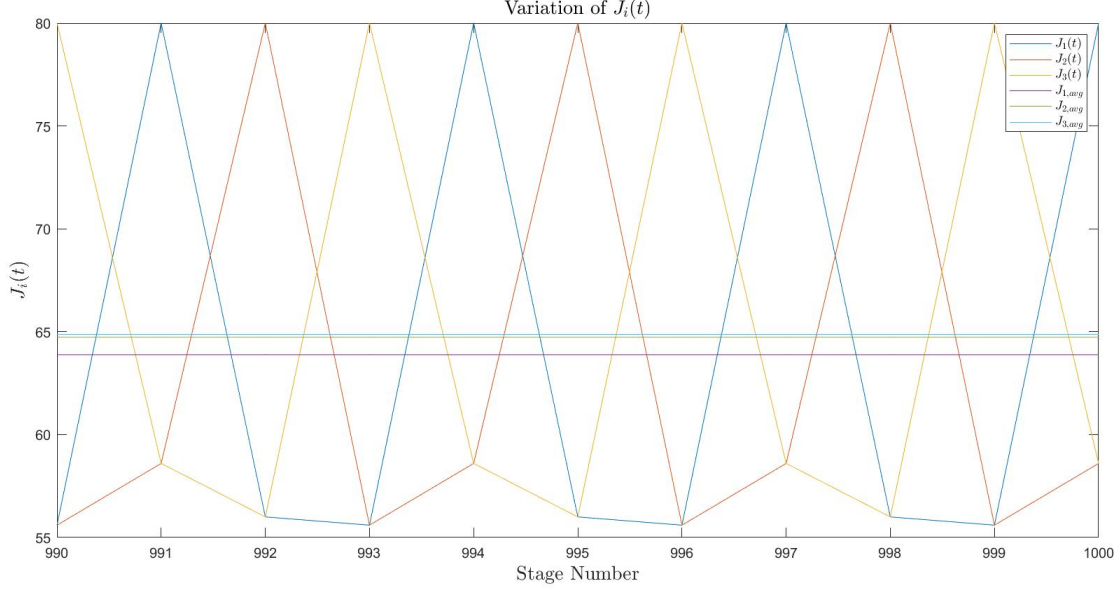


Figure 10: Individual agent Cost Behaviour with $\gamma_i(0) \sim \mathcal{U}(0, 1)$ and $|\mathcal{N}_i(0)| \sim \mathcal{U}(0, 200)$

of the bridge.

6 Conclusion and Outlook

It is a known fact that selecting a route selfishly might not always lead to the minimum possible cost that can be achieved and from our simulations we are clearly able to reiterate the fact. A Centralized Scheduler based Architecture might be able to mitigate the effect and overcome the paradox under the assumption that the Scheduler always acts in a fair manner and all the agents abide by the rules set by the Centralized Scheduler. The shortcomings however with this method outweigh the advantages which makes us investigate as to how the agents can themselves take rational decisions independently to overcome the paradox. The proposed Dynamic Agent Model along with the heuristics for the dynamics of the agent parameters achieves this as outlined in the results. Thus, modelling the agents reasonably, **helps overcome Braess's Paradox on average in an infinitely repeated game.**

Further directions of investigations would be to take practical considerations into account. Human beings might not be able to fit into the proposed Dynamic Agent Model; however smartphone apps that perform the route selection on behalf of the human beings could be modelled in this way, thus promoting cooperation and resulting in overall reduced costs. Other directions to explore would be to induce an 'equilibrium' like situation thus cutting off the incentive for an agent to turn selfish from the reasonable model. Trying out other different heuristics and training the agents to be intelligent using Machine Learning and Neural Network concepts are potential directions for further research.

References

- [1] Dietrich Braess, Anna Nagurney, and Tina Wakolbinger. On a paradox of traffic planning. *Transportation Science*, 39(4):446–450, 2005.
- [2] Dirk Helbing, Martin Schönhof, Hans-Ulrich Stark, and Janusz A Hołyst. How individuals learn to take turns: Emergence of alternating cooperation in a congestion game and the prisoner’s dilemma. *Advances in Complex Systems*, 8(01):87–116, 2005.
- [3] João P. Hespanha. *Noncooperative Game Theory: An Introduction for Engineers and Computer Scientists*. Princeton University Press, 2017.
- [4] Gina Kolata. What if they closed 42d street and nobody noticed? *The New York Times*, 1990.
- [5] Manfred Milinski, Dirk Semmann, and Hans-Jürgen Krambeck. Reputation helps solve the ‘tragedy of the commons’. *Nature*, 415(6870):424–426, 2002.
- [6] Eric I. Pas and Shari L. Principio. Braess’ paradox: Some new insights. *Transportation Research Part B: Methodological*, 31(3):265–276, 1997.
- [7] Tim Roughgarden and Éva Tardos. How bad is selfish routing? *J. ACM*, 49(2):236–259, mar 2002.
- [8] Richard Steinberg and Willard I. Zangwill. The prevalence of braess’ paradox. *Transportation Science*, 17(3):301–318, 1983.