



Agents' Strategy

**The Influence of homo- and heterogeneous Strategies in Resource Exchange Environments
on the approximation of Equilibria**

Max Eradus¹

Supervisor(s): Georgios Iosifidis¹, Livia Chatzieleftheriou¹

¹EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 21, 2026

Name of the student: Max Eradus

Final project course: CSE3000 Research Project

Thesis committee: Georgios Iosifidis, Livia Chatzieleftheriou, Rafa Bidarra

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Abstract

Sharing economies like BitTorrent are popular alternatives to traditional ownership. Users in these networks trade resources using different behavioural strategies. Past research only tested a few basic behaviours. We do not know how networks behave when complex strategies mix. If a platform fails to approximate an optimal state, users might feel dissatisfied and leave the platform. This paper investigates how 6 different strategies affect the stability of decentralized exchanges across 5 graph topologies. We simulated pure, homogeneous, and heterogeneous networks for 400 iterations over 30 repeats. Our results show that a homogeneous 50/50 mix of proportional and greedy strategies achieves the lowest average market loss. Introducing varied, heterogeneous weights between agents increases this loss. Conversely, petty and imitation strategies disrupt the network. These strategies consistently have the highest loss. Overall, homogeneous networks converge faster and perform better than heterogeneous networks.

1 Introduction

Sharing economies are popping up more and more often as an alternative to the ownership-based economy[20]. These online alternatives to the traditional platforms like BitTorrent, The Brooklyn Microgrid[15] and AI driven trading platforms [11] have made a resurgence largely because of the widespread adoption of the internet. In this paper, these platforms are modelled as undirected graphs. The nodes are the participants. The edges represent the option to exchange resources.

Participants in these economies use different strategies, which are the rules or principles used to make trading decisions. For example, a "greedy" strategy seeks maximum immediate return, while peer-to-peer file-sharing systems like BitTorrent often use a direct reciprocity rule [3], only giving resources to neighbours who shared with them in the past round.

In a recently written work Nongaillard et al.[16] propose several trading strategies. However, their

model includes negotiation between agents. Our model does not have negotiations. Instead, agents make their exchange decisions all at the same time, without negotiations.

Hellinga et al.[9] propose a centralised program to find a market equilibrium. A market equilibrium is a state in which no participant in the market has a reason to change their decision of how they exchanged their resources. They also show that greedy and proportional strategies approximate this equilibrium in homogeneous economies. Which is were all agents in the economy use the same exact strategy. However, they only evaluate these two strategies.

In real networks, humans use a wider range of strategies. Peer-to-peer file-sharing systems like BitTorrent use a direct reciprocity rule[3]. In those systems, nodes give resources only to neighbours who gave them resources in the past round. Other models of agent behaviour look at the imitation or the satisficing strategies[14; 1]. In these models, agents copy the most successful strategy or keep their current allocation if it meets their goal.

It is still unknown how sharing economies behave when participants who operate on combinations of these strategies are present. If combinations of these strategies do not approach an equilibrium, meaning participants in the sharing economy are not satisfied with their allocation. These resulting inefficiencies could cause participants to leave these real world exchanges[17].

To address this gap, our research asks:

How do homogeneous and heterogeneous combinations of two agent strategies affect the approximation of a theoretical optimum in a decentralized resource exchange?

This question has been subdivided into the following sub-questions:

- **SQ 1:** Which pure strategies approximate an equilibrium allocation?
- **SQ 2:** Which homogeneous combinations of two strategies approximate an equilibrium allocation?
- **SQ 3:** Which heterogeneous combinations of two strategies approximate an equilibrium allocation?
- **SQ 4:** How do different strategy combinations

compare to each other in terms of distance from an equilibrium allocation?

- **SQ 5:** How do different strategy combinations compare to each other in terms of number of iterations to get arbitrarily close to an equilibrium allocation?

We explore a larger set of strategies to answer this. First, we look at homogeneous environments, where all agents follow the exact same pure strategy. Then at homogeneous environments where all agents follow the exact same combination of strategies. For example, half of the agents decision is based on the greedy, and half on the proportional strategy. Finally, we look at heterogeneous environments where all agents use a different mix of strategies. Which is where each agent follows their own strategy mix. For example, agent 1's decision is half greedy and half proportional, but agent 2's decision is a quarter greedy and three quarters proportional. We investigate how these environments behave and perform.

The remainder of this paper is structured to investigate the stability of decentralized exchanges. First, the problem setting is described with the mathematical nomenclature that is necessary to understand the paper. Then the agent strategies are introduced. We will provide mathematical description for these strategies and examples how they interact with other agents. We then describe our simulation environment, used to validate previous findings on homogeneous and heterogeneous populations and investigate the novel combination of these strategies.

2 Sharing Economy Model

In this paper, the sharing economy will be modelled as follows. The prosumers, participants in the exchanges that both provide and consume resources, are abstracted to nodes or agents $\mathcal{N}$. The option for the prosumers to exchange resources with another prosumer is represented by an undirected edge $\mathcal{E}$ between the node representation of these prosumers. Using this we can model the whole as an undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$.

2.1 Decentralised Market View

The sharing economy modelled is a time-slotted market. Which means that there are discrete time-slots in which the agents can make trading decisions. For every time-slot t a prosumer i will have

some amount of resources to share with their neighbours $D_i(t)$. These distributable resources are normally distributed with the mean being the agent's endowment ε_i . Thus, $D_i(t) \sim N(\varepsilon_i, \sigma^2)$.

Aside from each agent's endowment, every agent's resource also has a value v_i . This value stays constant through the time slots.

Every time-slot every agent chooses how much to give each of their neighbours $\mathcal{N}_i$. The allocation from agent i to agent j at time t is $x_{ij}(t)$. The whole allocation of agent i at t is denoted as its allocation vector $\vec{x}_i(t)$. The total allocation of all agents is denoted as the matrix $\mathbf{X}(t) = [\vec{x}_1(t) \dots \vec{x}_n(t)]$. Every agent cannot allocate more than its distributable resource that round, in particular $D_i(t) \geq \sum_{j \in \mathcal{N}_i} x_{ij}$.

The vector of received resources for agent i is denoted by $\vec{r}_i = [x_{ji}]_{j \in \mathcal{N}_i}$ and the total resources received is $R_i = \sum_{j \in \mathcal{N}_i} x_{ji}$.

Agents get benefits from the resources they receive from other agents, this is denoted by their utility. The received utility of agent i at t is:

$$U_i(t) = \sum_{j \in \mathcal{N}_i} x_{ji}(t) \cdot v_j$$

Agents base their allocation vector each timeslot on their strategy $s(t)$. Every agent i keeps track of their sharing ratio ρ_i , which is the ratio between the average amount of resources received from other agents multiplied with the value of that resource and the agent's own resource endowment multiplied with its resource value.

$$\rho_i(t) = \frac{\sum_{j \in \mathcal{N}_i} \bar{x}_{ji}(t) \cdot v_j}{\varepsilon_i \cdot v_i}$$

At the start of each round the sharing ratio is broadcast to neighbours.

Agents' goal is to maximize their total received utility $u_i(t) = \sum_{j \in \mathcal{N}_i} x_{ji}(t) \cdot v_j$.

2.2 Centralised Market View

The centralised market differs from the decentralised market in that one entity determines the optimal allocation for all participants in the market. This way all participants can be given a fair allocation of resources. In centralised markets, using the knowledge of edges, value and endowment of all agents it is possible to calculate an equilibrium allocation. The equilibrium idea comes from game

theory and economics[12], which describes a situation where there is no participant in the market that has a reason to change their chosen allocation.

For the resource sharing economy, we use the equilibrium definition from Hellings et al.[9]. Which states that an allocation matrix $\mathbf{X}$ is an equilibrium allocation matrix $\mathbf{X}^*$ if the following conditions hold:

- Market Clearance: all agents in the market allocate all their endowment, $\forall i \in \mathcal{N}, \sum_{j \in \mathcal{N}_i} x_{ij} = \varepsilon_i$.
- Utility Maximisation: the utility of all agents has been maximised considering the market constraints, $\forall i \in \mathcal{N}, \mathbf{X} \in \arg \max_{\mathbf{X}} u_i(\vec{r}_i)$

Hellings introduced a centralised program for computing this equilibrium[9]. This algorithm is based on the Eisenberg-Gale program[4]. It finds the equilibrium by solving the following equation:

$$\begin{aligned} \max_{\mathbf{X}} \sum_{i \in \mathcal{N}} & \left((v_i \cdot \varepsilon_i) \log \left(\sum_{j \in \mathcal{N}_i} v_j \cdot x_{ji} \right) \right) \\ \forall i \in \mathcal{N} : & \sum_{j \in \mathcal{N}_i} x_{ij} \leq \varepsilon_i \\ \forall i, j \in \mathcal{N} : & x_{ij} \geq 0 \end{aligned}$$

3 Agent Strategies

This section formally defines the collection of pure and mixed strategies evaluated in this paper. These strategies are selected to represent different types of decision-making. The Greedy and Proportional strategies act as a baseline. The other four strategies introduce social goals and human behaviours. The Egalitarian strategy focuses on fairness among neighbours rather than personal gain. The Imitation, Satisficing, and Petty strategies model bounded rationality. These choices represent copying successful peers, satisfying basic needs, and demanding strict reciprocity.

The first two strategies described are Greedy and Proportional strategies. The definitions have been taken from Hellings[8, p. 25-26]. The third strategy evaluated is the Egalitarian strategy. Which is based on the behaviour described in Endriss[5]. The fourth is the Imitation strategy. This is done by adapting the description defined in Hien Le et al.[14]. The fifth is the Satisficing strategy. This strategy is an adaptation of the strategy described in

Artinger et al.[1]. The final is the Petty behaviour strategy. Which is based on Kim[13].

3.1 Greedy Strategy

“A greedy strategy is an allocation strategy for which each agent i maximises the expected utility $\hat{U}_i$ of next timeslot: $\vec{x}_i(t) = \max_{\vec{x}_i} (\hat{U}_i(t+1))$ ” [8, p. 25].

Greedy agents use the weighted sharing ratio ρ to look for neighbours $\mathcal{N}_i$ with lowest ρ . If multiple neighbours have the lowest ρ , the endowment is split equally across these agents. The rationale behind this, is that this agent will reciprocate the sentiment and give more to the agent next round. A simple algorithm for this strategy can be found algorithm 1.

Algorithm 1: Algorithm implementing the greedy strategy

```

1 for  $t = 1, 2, \dots$  do
2   foreach  $i \in \mathcal{N}$  do
3     Agent  $i$  announces to its neighbours
       the weighted sharing ratio;
4      $\rho_i(t) =$ 
        $\left( \sum_{j \in \mathcal{N}_i} v_j \cdot \bar{x}_{ji}(t) \right) / (v_i \cdot \varepsilon_i);$ 
5   end
6   foreach  $i \in \mathcal{N}$  do
7     Agent  $i$  distributes its resources
        $D_i(t)$  evenly over its neighbours
        $j \in \mathcal{N}_i$  having the smallest
       weighted sharing ratio  $\rho_j(t)$  in set
        $\mathcal{N}_i$ ;
8   end
9 end

```

3.2 Proportional Strategy

We will be using the same definition for the proportional strategy as was used in Hellings[8, p. 26]:

“for each agent i , the proportional contribution to its utility that it received from each neighbour j :

$$x_{ij}(t+1) = D_i(t+1) \cdot \frac{v_i \cdot x_{ji}(t)}{\sum_{k \in \mathcal{N}_i} v_k \cdot x_{ki}(t)}$$

“

This means that for the next round $t+1$ agent i will give to agent j proportional to the value i has

receives in the current round t from j . The upper part of the fraction is how much i receives from j . The bottom part is the total utility agent i receives in the current round.

This mathematical description has been extended to a algorithm in algorithm 2.

3.3 Egalitarian

The egalitarian strategy seeks to minimize inequality among an agent's neighbours. Rather than maximizing personal utility, the agent distributes resources proportionally to neighbour deficits to equalize their weighted sharing ratios. This is based on the definition of egalitarian agents from Endriss[5, p. 332]

This behaviour can be formalized by having the egalitarian agent take the largest weighted sharing ratio, and then allocating resources to its neighbours proportional to their distance from the highest weighted sharing ratio neighbour.

$$x_{ij}(t+1) \propto \left(\max_{k \in \mathcal{N}_i} \rho_k(t) - \rho_j(t) \right)$$

This algorithm is written in algorithm 3.

3.4 Imitation

The imitation strategy stems from the real world phenomenon where people try to imitate what they see is working for others. This can be found in people's conscious attempts to copy other peoples behaviour or choices (e.g. the copy trading platforms eToro, Pepperstone). It has also been shown that we copy people we are close to Rizzolatti et al.[18].

In previous papers the imitator strategy would copy the *decision* from the most successful agent[14]. This is not possible for this game setup since the different agents could be connected to different neighbours.

The imitation agent has been adapted such that for the next time slot $t+1$ the imitation agent will check the set of strategies with the highest $\rho(t)$ from the current round $\mathcal{M}(t)$ to see if its current strategy $s_i(t)$ is among the best performing. If this is the case it keeps the current strategy. If this is not the case it randomly picks a strategy $s^{max}(t)$ from the set of best performing strategies.

$$s_i(t+1) = \begin{cases} s^{max}(t) & \text{if } s_i(t) \notin \mathcal{M}(t) \\ s_i(t) & \text{if } s_i(t) \in \mathcal{M}(t) \end{cases}$$

This algorithm is written in algorithm 4.

3.5 Satisficing

The definition of satisficing that is most applicable for this problem consists of 2 parts. Part one is the aspiration level, "the minimum level of a given scale of interest that is acceptable to the agent in order to choose an alternative" Artinger et al.[1, p. 603]. In other words, each agent has an amount of utility to which it strives, but after reaching this it does not try to optimise its utility further.

Part 2 describes what actually happens when the agent's aspiration level has not been met. This is the search part. When the agent is not satisfied it searches for a alternative decision that does satisfy.

This behaviour can be described more formally as an agent maintains its current allocation if its performance (e.g., weighted sharing ratio ρ_i) meets or exceeds a predefined aspiration level α_i . If the performance falls below this threshold, the agent searches for a new allocation.

$$\vec{x}_i(t+1) = \begin{cases} \vec{x}_i(t) & \text{if } \rho_i(t) \geq \alpha_i \\ \vec{x}_i^{greedy} & \text{if } \rho_i(t) < \alpha_i \end{cases}$$

For the experiments in this paper, the aspiration level of satisficing agents has been set to $\alpha_i = 1.0$. This is done with the idea that agents in networks are generally satisfied if their contribution is at least reciprocated.

This algorithm is written in algorithm 5.

3.6 Petty

Pettiness is described as being unnecessary exact and unforgiving in transactions. For example, you owe someone €14.99 and when paying them back the money you hand them exactly €14.99. Nowadays this behaviour is observed more often due to the prevalence of digital payment systems[13].

This strategy has been modelled for the agents as follows.

$$x_{ij}(t+1) = x_{ji}(t)$$

This algorithm is written in algorithm 6.

3.7 Mixed strategy agents

Aside from the pure strategies, there exist also interesting cases for the mix of multiple strategies in a single agent. This could show more similarities to human behaviour, who do not act always perfectly as a greedy or proportional agent would.

How strategies are mixed is taken from Hellenga[8], where the same philosophy is used.

The mixed strategy ψ will consist of 2 separate strategies π and ϕ . Each agent i will have a weight value w_i , which decides how much each strategy influences the allocation vector. A weight of 1 means that the agents follows purely the π strategy and 0 means purely the ϕ strategy.

$$\vec{x}_i^\psi(t) = w_i \cdot \vec{x}_i^\pi(t) \cdot (1 - w_i) * \vec{x}_i^\phi(t)$$

This definition can be extended to mix n strategies. A unit vector $\vec{w}_i$ is used to hold the weights for each strategy for each agent i . With $\vec{w}_{i,1}$ being the weight of the first strategy $\vec{x}^{s_1}$ and so on until $\vec{w}_{i,n}$ as the weight for strategy $\vec{x}^{s_n}$.

$$\vec{x}_i^\psi(t) = \vec{w}_{i,1} \cdot \vec{x}^{s_1}(t) + \vec{w}_{i,2} \cdot \vec{x}^{s_2}(t) + \dots + \vec{w}_{i,n} \cdot \vec{x}^{s_n}(t)$$

An example of this for agent i using 3 strategies and weights $\vec{w}_i = [0.2, 0.5, 0.3]$. The allocation vector for strategy 1 is $\vec{x}_i^{s_1} = [1, 1, 1]$, strategy 2 is $\vec{x}_i^{s_2} = [3, 0, 0]$ and strategy 3 is $\vec{x}_i^{s_3} = [0, 1, 2]$. This results in a final allocation vector $\vec{x}_i = [1 \cdot 0.2 + 3 \cdot 0.5 + 0 \cdot 0.3; 1 \cdot 0.2 + 0 \cdot 0.5 + 1 \cdot 0.3; 1 \cdot 0.2 + 0 \cdot 0.5 + 2 \cdot 0.3] = [1.7, 0.5, 0.8]$

4 Experimental Setup

This chapter will describe the experimental setup, which metrics and parameters are used to obtain the results.

4.1 Metrics

Sharing Ratios Weighted sharing ratios are monitored to evaluate agent behaviour, as these ratios do not necessarily approach to a fixed value and are highly dependent on the equilibrium of the underlying market structure.

Market Loss The market loss is defined as the Euclidean distance from the current market state to the equilibrium solution. The current market state is the average received utilities from all agents $\bar{U}(t)$. Using this, the market loss is defined as:

$$l^{eq}(t) = \sqrt{\sum_{i=1}^n (\bar{U}_i(t) - U_i^*)^2}$$

This is the same loss function that was used in Hellenga[8]. As in this paper, the loss is normalized to correct for different lengths of the utility vector. This is done by: $l_{norm}^{eq} = l^{eq} / |U^*|$.

4.2 General Experiment Parameters

Because this paper seeks to expand the understanding of agent strategies in exchange markets, our experimental parameters closely mirror those used in Hellenga[8]. This is done to eliminate the differences caused by these parameters.

Each experiment will run for 400 time-steps. Each experiment will have a generated network of size n , which is specified for each experiment. To minimize the influence of outliers in these randomly generated networks, each experiment will be performed 30 times.

The type of network can have a large influence on the performance of the strategies. To simulate the diverse networks of sharing economies, multiple network structures are used: complete, random, grid, scale-free, and small-world. These networks reflect what is observed in the real world.

The parameters for the network generation methods are taken from Hellenga [8, p. 40] and can be found in section B.

4.3 Experiment specific Parameters

Aside from the general experiment parameters there is also the experiment specific parameter that will vary between each experiment. The variable that will be changed is the weight vector of each agent w_i .

- **Pure isolated strategies:** For this simulation, the whole network will only contain agents with a weight vector w_i which is 0 for all entries except for one active strategy.
- **Pure strategy mixed agent networks:** This simulation will contain all pairs of pure agent strategies in a 1 to 1 proportion (e.g. 10 greedy agents and 10 egalitarian agents)
- **Homogeneous strategy agent networks:** Experiments are done with different weight vectors w_i , but each agent will use the same weight vector for the simulation.
- **Heterogeneous strategy agent networks:** Finally the w_i parameter will be varied between agents in a network, so each agent will use a different proportion of strategies.

4.4 Simulation Software

The simulation environment used is made in Python. The package *networkx* is used to generate

Strategies	Final Loss
Proportional	0.004567
Greedy	0.006581
Egalitarian	0.059991
Satisficing	0.201142
Petty	0.487316
Imitation	0.508763

Table 1: Experiment 1 final aggregated loss of the pure agent strategies ordered from minimal loss to maximal loss

the graphs. And to optimize the program that finds a market equilibrium state *cvxpy* is used.

5 Results

Pure Strategies

Some pure strategies approximate an equilibrium a lot better than others. In fig. 1 the aggregated results of these experiments are visible. The proportional and greedy agents approach an equilibrium very quickly. The satisficing and egalitarian agents are very dependent on graph structure. We think this is because the agents strive toward a sharing ratio ρ of 1.0. However, in some graph structures, like the random (fig. 3), an equilibrium does not mean that every agent's ρ is 1.0. Meanwhile, in graphs where an equilibrium does work out to $\rho = 1.0$ for each agent these strategies do approach an equilibrium in a similar manner to the greedy and proportional strategies as seen in fig. 2.

The imitation agent does not approximate an equilibrium since there is no strategy to copy and thus randomly allocates its endowment each round.

The petty agent can be seen to start with approaching an equilibrium in the very first round. However, the network of petty agents immediately starts to approach some value of loss for the network.

Pure Strategy Mixed Agent Network

Most of the pure strategy mixes of agents approximate an equilibrium slower or not at all in these environments as can be seen in fig. 4 and in table 2. The greedy and egalitarian strategy combination approximates an equilibrium the best out of all the strategies. However, it can be seen bottoming out

Pair	Final Loss
Greedy vs Egalitarian	0.025833
Proportional vs Egalitarian	0.044953
Proportional vs Greedy	0.051009
Greedy vs Satisficing	0.096235
Greedy vs Imitation	0.104808
Proportional vs Satisficing	0.121246
Egalitarian vs Satisficing	0.130750
Egalitarian vs Imitation	0.176027
Imitation vs Satisficing	0.219229
Proportional vs Imitation	0.239025
Egalitarian vs Petty	0.306018
Greedy vs Petty	0.320772
Proportional vs Petty	0.388870
Petty vs Satisficing	0.410688
Petty vs Imitation	0.588874

Table 2: Experiment 2 final aggregated loss of simulations containing 10 agents of each strategy ordered from lowest to highest

already in fig. 4. We reason that this means all strategy combinations in experiment 2 do not approximate an equilibrium.

Homogeneous Weighted Strategies

Aside from the proportional/greedy combination, other homogeneous strategy mixes do not approximate an equilibrium. The egalitarian performs fine in combination with the greedy or proportional strategy. Other strategy mixes can be seen not to approximate 0 loss in table 3 and in fig. 6.

The combined strategy of proportional and greedy has less loss after 400 iterations than the purely proportional or greedy strategies have. A network containing homogeneous agents who all operate on the same mix of convergent strategies has been proven to converge toward an equilibrium by Hellinga[9]. However, here we can see that the 50/50 mix of strategies also converges faster averaged over these 5 topologies.

The 50/50 mix can be seen to still have less loss after 1000 iterations than either greedy or proportional strategies in all of the examined topologies except for the scale-free topology. These results can be found in table 4 and in fig. 5.

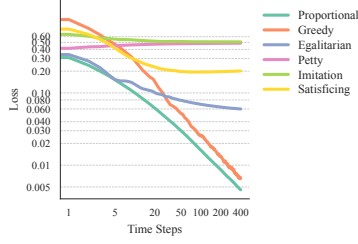


Figure 1: Experiment 1: Aggregated results of pure agent strategies

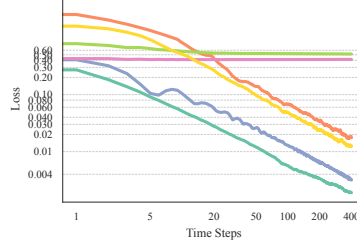


Figure 2: Experiment 1, network loss for the complete graph topology

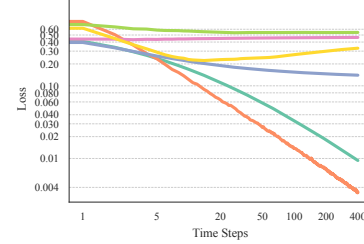


Figure 3: Experiment 1, network loss for the random graph topology

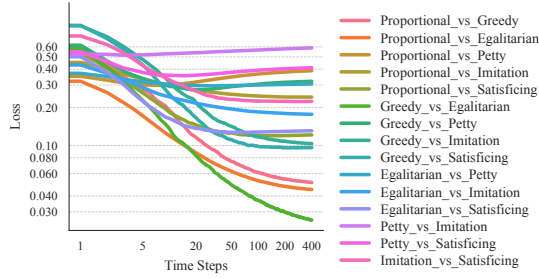


Figure 4: Experiment 2: Aggregated result of the environment containing 10 agents with the first strategy and 10 agents with the second strategy

Heterogeneous Weighted Strategies

Consistent with the results from the homogeneous weighted networks, the pairing of the proportional and greedy strategies achieves the lowest average market loss (0.021579) as visible in table 6 and fig. 7. Combinations involving the egalitarian strategy (paired with greedy or proportional) also continue to perform relatively well, having the second and third lowest loss values.

It does seem like there are no strategy pairs who approximate an equilibrium. Even the best performing strategy seems to be bottoming out when we look at 5000 iterations in fig. 8, since the loss is still 0.018386.

Furthermore, as expected from previous experiments, any combination involving the petty or imitation strategies performs poorly. The pairing of petty and imitation yields the highest overall average loss (0.541367). This shows that strategies that do not converge toward a market equilibrium, still do not converge even in mixes with other strategies.

Pair	Final Loss
Proportional 50 and Greedy 50	0.003628
Greedy 50 and Egalitarian 50	0.024935
Greedy 50 and Imitation 50	0.025170
Proportional 50 and Egalitarian 50	0.045911
Proportional 50 and Imitation 50	0.052685
Greedy 50 and Satisficing 50	0.056905
Egalitarian 50 and Imitation 50	0.071753
Egalitarian 50 and Satisficing 50	0.081001
Proportional 50 and Satisficing 50	0.106776
Egalitarian 50 and Petty 50	0.156855
Proportional 50 and Petty 50	0.182447
Imitation 50 and Satisficing 50	0.208047
Greedy 50 and Petty 50	0.268334
Petty 50 and Satisficing 50	0.294068
Petty 50 and Imitation 50	0.559428

Table 3: Experiment 3.1 final aggregated loss of agents with weight vector with 0.5 for the first strategy and 0.5 for the second strategy

6 Responsible Research

Through this work it has been shown that certain behaviours in sharing economies could lead these environments to not approach an equilibrium. This could mean that administrators could choose to restrict participants behaviour. Admins could might also be able to detect this behaviour and remove users showing this behaviour. This might lead to discrimination against certain groups in these networks.

We have taken steps to ensure the reproducibility of our results. Aside from the in described simulation environment and parameters used in the simulations. The implementation of this simulation environment and the experiments is also available

Strategy Mix	Topology					Average
	Complete	Grid	Random	Scale-free	Small-world	
50/50 Greedy/Proportional	0.003266	0.000812	0.000897	0.000780	0.001475	0.001446
Purely Proportional	0.001123	0.003013	0.003818	0.000980	0.001206	0.002028
Purely Greedy	0.006557	0.001134	0.001424	0.000867	0.002867	0.002570

Table 4: Experiment 3.2 final aggregated loss after 1000 iterations comparing the networks containing only proportional, only greedy and the 50/50 weighted proportional/greedy strategies

online[6]. These experiments were build on top of the framework that was already build. No results or data has been used from other experiments. The only components that were reused, were the basic market setup methods, the method to calculate the market equilibrium, and the already present implementations for the greedy and the proportional strategies. The exact experiments conducted can be found on the `strategies-max` branch in the `experiments/max_eradus` folder as Jupyter notebooks and the strategies are in `src/ant/decentralised/direct.py` and `decentralised/special_strategies.py`. All components that use randomness can be provided with a seed. For reproducibility, the seed used for the experiments is provided with each experiment.

AI has been used responsibly in this research. Tools such as Scholar Labs has been used to find relevant articles and papers. The content of this paper was written by the author, but AI tools have been used to check grammar, to rewrite sentences to better convey meaning, and for inspiration for the structure of sections. No experimental results or graphs have been generated by AI. Code for the strategies and the experiments has been assisted by AI.

7 Conclusions and discussion

This paper investigated how different agent strategies behave in decentralized sharing economies. Specifically, we examined whether diverse strategy combinations converge to a market equilibrium.

Our results show that strategy choice greatly impacts market loss. Proportional and greedy strategies perform the best both in the pure environments and as a pair in the homogeneous environment. In contrast, petty and imitation strategies disrupt the network and fail to reach an equilibrium. And the satisficing and egalitarian strategies are some-

where in-between. They are very much dependent on graph structure, performing well on for example complete graphs but very poorly on random graphs.

We also added a new finding: a homogeneous mix of these strategies converges even faster than simulations of the pure strategies alone when averaging over the 5 topologies investigated. All other strategies seem to get stuck in sub optimal market states in both the homogeneous and the heterogeneous market setting.

For the heterogeneous simulations it seems like even the strategies that previously did approximated an equilibrium, do not any more.

By mixing a larger set of strategies than previous studies, we saw how destructive certain behaviours can be. Petty and imitation strategies perform poorly across all network topologies. They cause inefficiencies that could cause participant to leave the market.

Our experimental setup provided a strong baseline. We used 30 repeats across five graph topologies. This ensures our results are not just random chance. However, our simulation has limits. We modelled extreme cases to clearly see the differences. Real-world platforms are often much messier than our models.

These limits leave several open issues for future research. First, our networks were static. Real platforms are dynamic, unhappy users leave, and new users join. Future simulations should let agents and edges appear or disappear. Second, in our experiments it seems like the heterogeneous mix of proportional and greedy agents does not approximate an equilibrium. For future research, doing a formal proof of for this might be a promising direction. Finally, this paper only looked at pairs of strategies. Testing complex mixes of three or more strategies is a promising direction for future work.

References

- [1] Florian M. Artinger, Gerd Gigerenzer, and Perke Jacobs. Satisficing: Integrating two traditions. *Journal of Economic Literature*, 60(2):598 – 635, 2022. Cited by: 55; All Open Access, Green Open Access.
- [2] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, October 1999.
- [3] Bram Cohen et al. Incentives build robustness in bittorrent. In *Workshop on Economics of Peer-to-Peer systems*, volume 6, pages 68–72, 2003.
- [4] Edmund Eisenberg and David Gale. Consensus of subjective probabilities: The parimutuel method. *The Annals of Mathematical Statistics*, 30(1):165–168, 1959.
- [5] Ulle Endriss, Nicolas Maudet, Fariba Sadri, and Francesca Toni. Negotiating socially optimal allocations of resources. *Journal of Artificial Intelligence Research*, 25:315 – 348, 2006. Cited by: 114; All Open Access, Gold Open Access, Green Open Access.
- [6] Max Eradus and Joeri Geuzinge. Source code for agent strategy comparisons. <https://github.com/Joeri-G/Ant/tree/strategies-max>, 2026. Accessed: 2026-06-20.
- [7] Paul Erdős and Alfréd Rényi. On random graphs i. *Publicationes Mathematicae Debrecen*, 6:290–297, 1959.
- [8] Rixt Hellinga. Algorithms for the sharing economy: An economic modelling perspective on distributed exchange. Master’s thesis, Delft University of Technology, Delft, Netherlands, July 2025. Publicly defended on Thursday July 17, 2025.
- [9] Rixt Hellinga, George Iosifidis, and Neil Yorke-Smith. Towards strengthening decentralised exchange. In *Proceedings of the 25th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS ’26, page 3128–3130, Richland, SC, 2026. International Foundation for Autonomous Agents and Multiagent Systems.
- [10] M.D Humphries, K Gurney, and T.J Prescott. The brainstem reticular formation is a small-world, not scale-free, network. *Proceedings of the Royal Society B: Biological Sciences*, 273(1585):503–511, 11 2005.
- [11] Yusuf Izmirliloglu, Loc Pham, Tran Cao Son, and Enrico Pontelli. A survey of multi-agent systems for smartgrids. *Energies*, 17(15), 2024.
- [12] Mamoru Kaneko and Kenjiro Nakamura. The nash social welfare function. *Econometrica*, 47(2):423–435, 1979.
- [13] Tami Kim, Ting Zhang, and Michael I. Norton. Pettiness in social exchange. *Journal of Experimental Psychology: General*, 148(2):361 – 373, 2019. Cited by: 15.
- [14] Hien Le and Pouria Ramazi. Heterogeneous mixed populations of best-responders and imitators: Equilibrium convergence and stability. *IEEE Transactions on Automatic Control*, 66(8):3475–3488, 2021.
- [15] Esther Mengelkamp, Johannes Gärtner, Kerstin Rock, Scott Kessler, Lawrence Orsini, and Christof Weinhardt. Designing microgrid energy markets: A case study: The brooklyn microgrid. *Applied Energy*, 210:870–880, 2018.
- [16] Antoine Nongaiard and Philippe Mathieu. Resource allocation problems on networks: Maximizing social welfare using an agent-based approach. volume 2, page 206 – 211, 2011. Cited by: 0.
- [17] Adrien Querbes. Banned from the sharing economy: an agent-based model of a peer-to-peer marketplace for consumer goods and services. *Journal of Evolutionary Economics*, 28(3):633–665, 2018.
- [18] Giacomo Rizzolatti and Laila Craighero. The mirror-neuron system. *Annual Review of Neuroscience*, 27:169 – 192, 2004. Cited by: 6460.
- [19] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of ‘small-world’ networks. *Nature*, 393(6684):440–442, 1998.
- [20] Xiaoxi Zhu and Kai Liu. A systematic review and future directions of the sharing economy: business models, operational insights and environment-based utilities. *Journal of Cleaner Production*, 290:125209, 2021.

A Algorithms for Agent's Strategies

Algorithm 2: Algorithm implementing the proportional strategy

```

1 for  $t = 1, 2, \dots$  do
2   foreach  $i \in \mathcal{N}$  do
3     Agent  $i$  announces to its neighbours
       the weighted sharing ratio;;
4      $\rho_i(t) =$ 
        $\left( \sum_{j \in \mathcal{N}_i} v_j \cdot \overline{x_{ji}}(t) \right) / (v_i \cdot \varepsilon_i);$ 
5   end
6   foreach  $i \in \mathcal{N}$  do
7     foreach  $j \in \mathcal{N}_i$  do
8        $x_{ij}(t+1) =$ 
          $D_i(t+1) * \frac{v_i * x_{ji}(t)}{\sum_{k \rightarrow \mathcal{N}_i} v_k * x_{ki}(t)};$ 
9     end
10  end
11 end

```

Algorithm 3: Algorithm implementing the egalitarian strategy

```

1 for  $t = 1, 2, \dots$  do
2   foreach  $i \in \mathcal{N}$  do
3     Agent  $i$  announces to its neighbours
       the weighted sharing ratio;;
4      $\rho_i(t) =$ 
        $\left( \sum_{j \in \mathcal{N}_i} v_j \cdot \overline{x_{ji}}(t) \right) / (v_i \cdot \varepsilon_i);$ 
5   end
6   foreach  $i \in \mathcal{N}$  do
7     Agent  $i$  distributes its resources
        $D_i(t)$  among its neighbours  $j \in \mathcal{N}_i$ 
       proportionally to their deficit
       relative to the highest-earning
       neighbour, such that  $x_{ij}(t+1) \propto$ 
        $(\max_{k \in \mathcal{N}_i} \rho_k(t) - \rho_j(t));$ 
8   end
9 end

```

Algorithm 4: Algorithm implementing the imitation strategy

```

1 for  $t = 1, 2, \dots$  do
2   foreach  $i \in \mathcal{N}$  do
3     Agent  $i$  announces its weighted
       sharing ratio to the network;;
4      $\rho_i(t) =$ 
        $\left( \sum_{j \in \mathcal{N}_i} v_j \cdot \overline{x_{ji}}(t) \right) / (v_i \cdot \varepsilon_i);$ 
5   end
6   Determine the set of highest-earning
       strategies from its neighbours  $\mathcal{M}(t)$ 
       and select the best strategy  $s^{max}(t)$ ;
7   foreach  $i \in \mathcal{N}$  do
8     Agent  $i$  looks at its neighbours to
       compare its strategy against the
       most successful ones;;
9     if  $s_i(t) \notin \mathcal{M}(t)$  then
10      Agent  $i$  copies the strategy of the
        highest-performing node:
         $s_i(t+1) = s^{max}(t);$ 
11    else
12      Agent  $i$  maintains its current
        strategy:  $s_i(t+1) = s_i(t);$ 
13    end
14  end
15 end

```

Algorithm 5: Algorithm implementing the satisfying strategy

```

1 for  $t = 1, 2, \dots$  do
2   foreach  $i \in \mathcal{N}$  do
3     Agent  $i$  announces to its neighbours
       the weighted sharing ratio;;
4      $\rho_i(t) =$ 
        $\left( \sum_{j \in \mathcal{N}_i} v_j \cdot \bar{x}_{ji}(t) \right) / (v_i \cdot \varepsilon_i);$ 
5   end
6   foreach  $i \in \mathcal{N}$  do
7     Agent  $i$  evaluates its performance
       against its predefined aspiration
       level  $\alpha_i$ ;
8     if  $\rho_i(t) \geq \alpha_i$  then
9       Agent  $i$  is satisfied and maintains
       its current allocation vector:
        $\bar{x}_i(t+1) = \bar{x}_i(t);$ 
10    else
11      Agent  $i$  is unsatisfied, searches
       for, and adopts a new allocation
       vector using the greedy
       strategy:  $\bar{x}_i(t+1) = \bar{x}_i^{greedy};$ 
12    end
13  end
14 end

```

Algorithm 6: Algorithm implementing the petty behaviour strategy

```

1 for  $t = 1, 2, \dots$  do
2   foreach  $i \in \mathcal{N}$  do
3     Agent  $i$  announces to its neighbours
       the weighted sharing ratio;;
4      $\rho_i(t) =$ 
        $\left( \sum_{j \in \mathcal{N}_i} v_j \cdot \bar{x}_{ji}(t) \right) / (v_i \cdot \varepsilon_i);$ 
5   end
6   foreach  $i \in \mathcal{N}$  do
7     Agent  $i$  distributes resources to each
       neighbour  $j \in \mathcal{N}_i$  focusing on exact
       reciprocity, returning the amount
       received in the previous timestep
       subject to a sensitivity parameter  $\delta$ ;
8      $x_{ij}(t+1) = x_{ji}(t);$ 
9   end
10 end

```

B Graph Settings

C Simulation Results

C.1 1000 Iterations Comparison

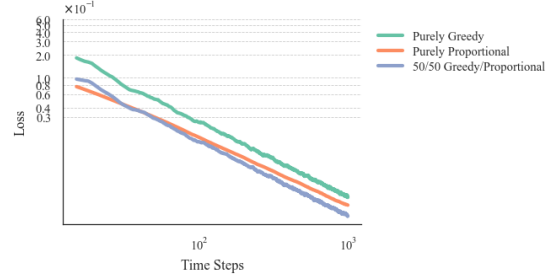


Figure 5: experiment 3.2 1000 iterations

C.2 Experiment 3

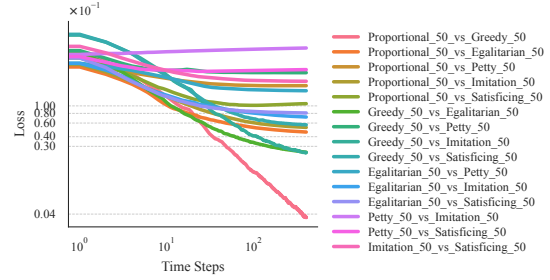


Figure 6: Experiment 3 aggregated results of homogeneous agent strategies

C.3 Experiment 4

	Method	Parameter	Values
complete	-	-	-
random network	Erdős-Rényi [7]	probability of edge creation	q 0.2
grid network	-	dimensions	$m \times l$ $\frac{n}{4} \times 4$
scale-free network	Barabasi-Albert [2]	number of edges to attach from a new agent to an existing one	m 1
small-world network	Watts-Strogatz [19]	mean degree	k $n/2$
		probability to rewire an edge	q 0.05
		clustering coefficient [10]	- > 1

Table 5: The methods and parameters used to generate networks for simulations

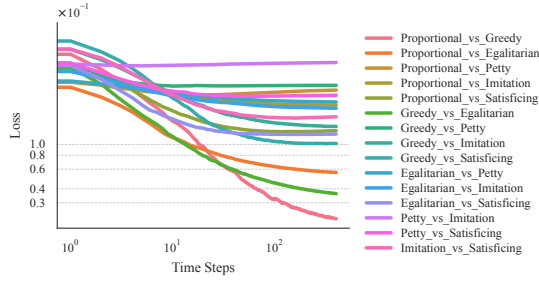


Figure 7: Experiment 4 aggregated result of heterogeneous agent strategies

Pair	Final Loss
Proportional and Greedy	0.021579
Greedy and Egalitarian	0.036170
Proportional and Egalitarian	0.056031
Greedy and Satisficing	0.102001
Egalitarian and Satisficing	0.123944
Proportional and Satisficing	0.132706
Greedy and Imitation	0.145086
Imitation and Satisficing	0.182675
Proportional and Imitation	0.210972
Egalitarian and Imitation	0.224798
Egalitarian and Petty	0.240059
Proportional and Petty	0.275914
Petty and Satisficing	0.306587
Greedy and Petty	0.338512
Petty and Imitation	0.541367

Table 6: Experiment 4 final aggregated loss of environment where each agent has its own uniformly random distributed weight vector

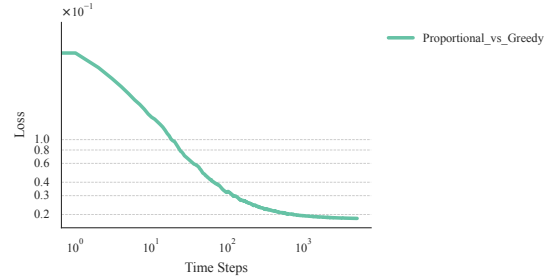


Figure 8: Experiment 4.1 heterogeneous mix of proportional and greedy, 0.018386 loss after 5000 iterations