Parallel Simulations of the Iterated n-Player Prisoner's Dilemma

Raviv Komem

Lior Wunsch

Paper by Queiroz Diego and Jaime Sichman

Introduction

The prisoner's dilemma is a standard example of a game analyzed in game theory that shows why two completely rational individuals might not cooperate, even if it appears that it is in their best interests to do so.

Rational behavior refers to a decision-making process that is based on making choices that result in the optimal level of benefit or utility for an individual.

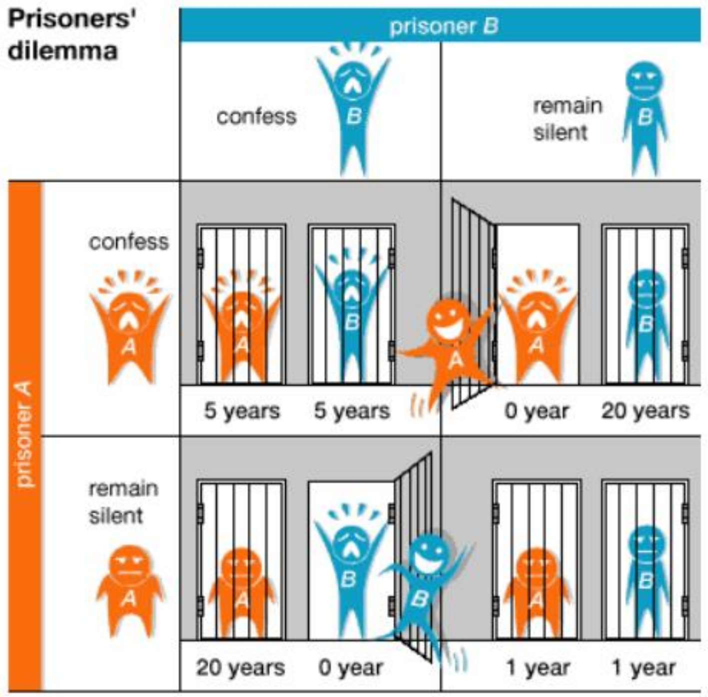
Why?

There are many real life instances that can be modeled as a N-PD game .

Example

Two prisoners committed a crime. They are arrested and taken into the investigation; they both receive two options.

Either cooperate with the other prisoner and remain silent or betray their accomplice and confess the crime.



Source: Prisoner’s Dilemma by Jana Vembunarayanan

2 Players Prisoner's Dilemma

The payoff to player A is shown with illustrative numerical values. It is symmetric between A and B.

The game is defined by the following rules:

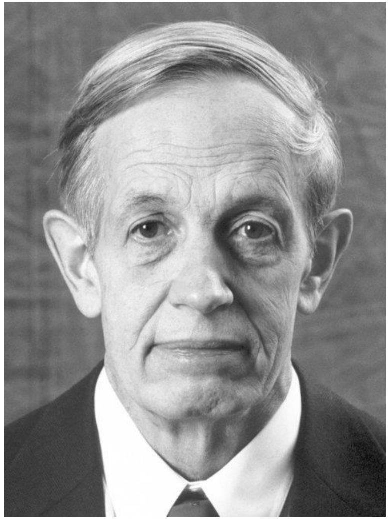
- $T > R > P > S$
- $R > \frac{S+T}{2}$

Player A \ Player B	C Cooperation	D Defection
C Cooperation	R=3 Reward for mutual cooperation	S=0 Sucker's payoff
D Defection	T=5 Temptation to defect	P=1 Punishment for mutual defection

Source: Reference [2]

Nash Equilibrium

Nash equilibrium in game theory is a situation in which a player will continue with their chosen strategy, having no incentive to deviate from it, after taking into consideration the opponent's strategy.



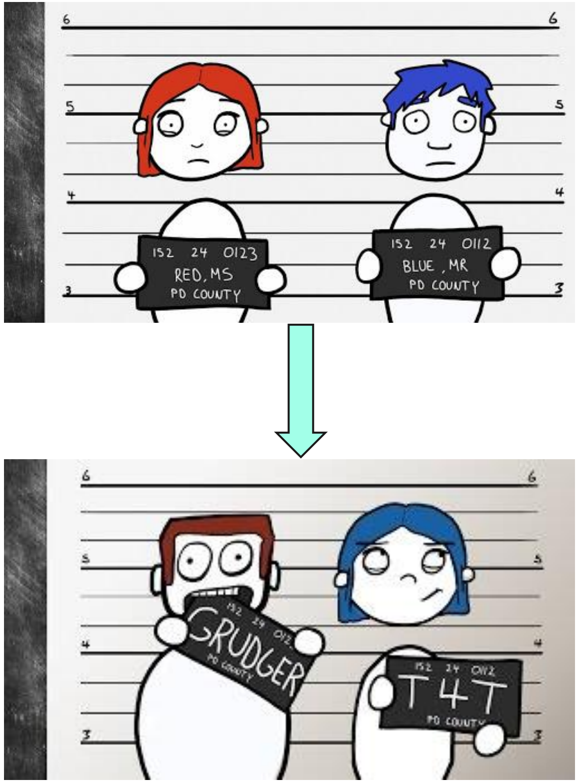
John Forbes Nash Jr.

Iterated 2 Players PD

The iterated PD is an extension of the general form except the game is repeatedly played by the same participants.

The game is played iteratively for a number of rounds until it is ended and the end is unknown to the participants.

The scores from each round are accumulated, so the object is to optimize the point score before reaching game over.



N-Players Prisoner's Dilemma

N-Players Prisoner’s Dilemma is a general case of the 2 players Prisoner’s Dilemma.

There are N players (where N is a natural number greater than one) and each player has two available strategies : C (cooperation) and D (Defection).

It is assumed that the payoff to a player is strictly a **function** of the strategy played and the number of other players who cooperate.

N-Players PD Definitions

Let $C_i(x)$ be the payoff to player i given that player i **cooperates** and x other players cooperate.

Let $D_i(x)$ be the payoff to player i given that player i **defects** and x other players cooperate.

These two functions must satisfy the following four assumptions for all $x, i, and\ j$:

- $C_i(x) = C_j(x)$ and $D_i(x) = D_j(x)$
- $D_i(x) \ > \ C_j(x)$
- $C_i(n-1) \ > D_i(0)$
- $D_i(x) \ \geq \ D_i(0)$

Iterated N-Players PD

An iterated game is a sequence of ordinary games: $g_1, g_2, \ldots$

So, iterated N-Player PD games can be defined by two sequences of functions:

$C_{i1}(x), C_{i2}(x), \ldots$; and $D_{i1}(x), D_{i2}(x), \ldots$;

Where each $C_{it}(x)$ is the payoff to player i in ordinary game g_t given that i and x others cooperate,

and each $D_{it}(x)$ is the payoff to player i in game g_t given that i defects and x others cooperate.

Experiment Context

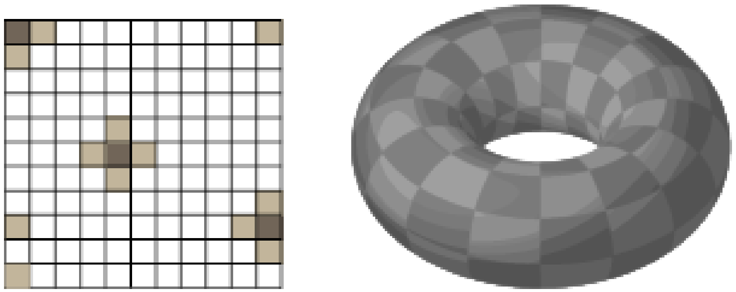
- Every test battery consists of $\boldsymbol{d\ x\ d}$ squared matrix, where each cell is a single agent.
- Every agent starts with a predefined **game strategy**.
- Every agent will play the Iterated PD for 150 iterations with his von Neumann Neighborhood.
- The battery will play for $\boldsymbol{g}$ generations, where in the end of every generation the agents may mutate with probability of $\boldsymbol{P}(\boldsymbol{m})$ and change their strategy.
- At every turn, the agent may deviate from his game strategy with probability $\boldsymbol{P}(\boldsymbol{e})$.

von Neumann Neighborhoods

This type of neighborhood applies on two-dimensional lattices, and it comprises all cells orthogonally surrounding one given cell.

We consider in our work that the agents in the edges of the grid have neighbors on the opposite sides of the lattice.

The space where agents are arranged can be better understood as a discrete torus.



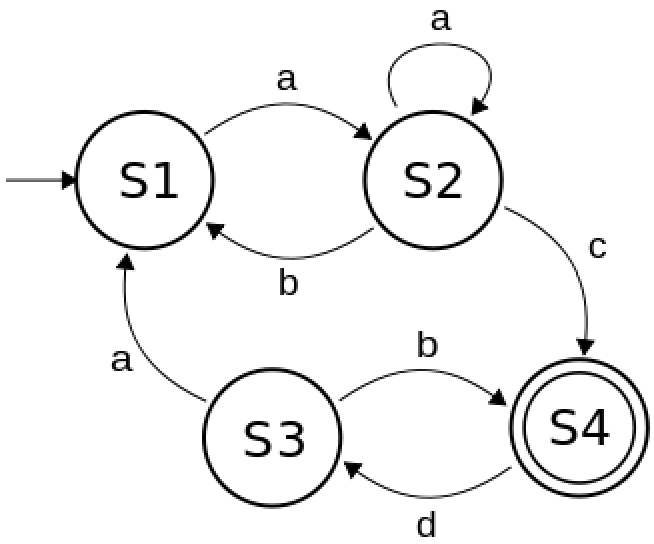
Neighborhood on the lattice edge, defining a torus.
Source: Reference [4]

Game Strategy Representation

There are many ways to represent the agent game strategy, however in this paper we decided to use two methods:

- Automata Finite (AF)
- Automata Adaptive (AA)

Each test battery will be performed twice using AF and AA models.



Automata Finite Example

Research Process

In the research they used several parameters for their experiments. And each set of experiments used to evaluate the impact of changes in those parameters on the system outcome.

- $\boldsymbol{k}$ is the number of processors to run the simulation.
- $\boldsymbol{d}$ is the dimension of the lattice.
- $\boldsymbol{g}$ is the number of generations.
- $\boldsymbol{P(m)}$ is the probability of mutation.
- $\boldsymbol{P(e)}$ is the error rate.

We will compare the average utility.

$$\text{Utility}\,(\mathbb{U}) = \frac{\sum_{a_{ij} \in \mathbb{U}} V_{a_{ij}}}{|\mathbb{U}|}$$

Wilcoxon Test

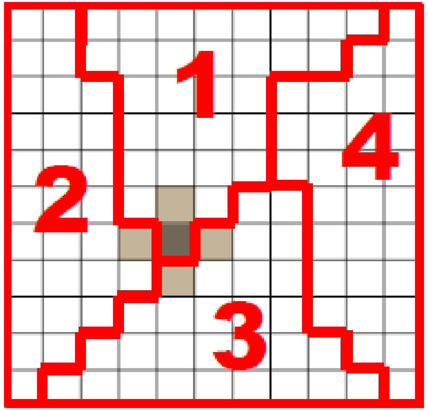
In the following experiments Wilcoxon test was used to accept or to reject the statistical hypothesis.

This is a non-parametric statistical hypothesis test used to compare two repeated measurements on a single sample to assess whether their population mean ranks differ.

Parallelization

Parallel computation of simulations involving multiple agents can be a hard task to accomplish due to the high dependence that may reside in their interactions.

When dividing agents in groups to run in parallel, it is possible that some agents may need to be accessible by more than one group.



Example of lattice division.
Source: Reference [4]

Experiment - Number of Processors

A first set of experiments was designed with the unique purpose of evaluating the performance gain achieved by distributing the agents' interactions in the cluster.

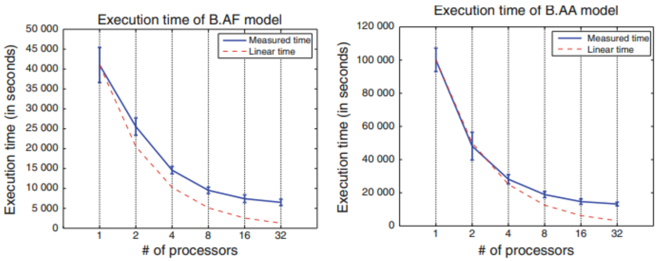
The analysis is based on the collection of the simulation run times in order to calculate the algorithm's efficiency.

Battery	k	d	g	$P\,(\mathrm{m})$	$P\,(\mathrm{e})$	Strategy
01	1	50×50	5 000	1 %	1 %	TT5
02	2	50×50	5 000	1 %	1 %	TT5
03	4	50×50	5 000	1 %	1 %	TT5
04	8	50×50	5 000	1 %	1 %	TT5
05	16	50×50	5 000	1 %	1 %	TT5
06	32	50×50	5 000	1 %	1 %	TT5

Results - Number of Processors

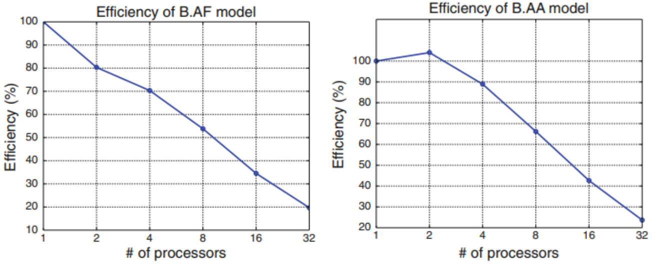
In both models, AA and AF, the speedup in execution time increased as more processors were added, reaching an upper limit of around 85%.

However, the efficiency decreased as more processors were added and the cluster resources were shared between more users.



(a) B. AF (b) B. AA

Fig. 5. Batteries 01–06: Average execution time for B.AF and B.AA models



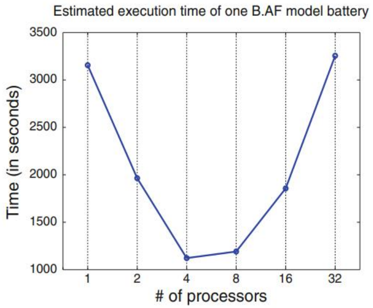
(a) B. AF (b) B. AA

Results - Number of Processors

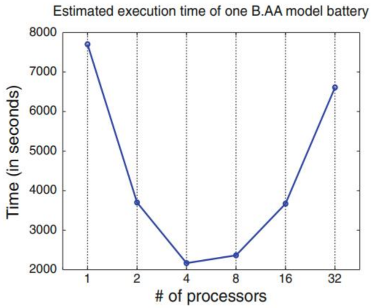
The system also allows each user to not hold more than 13 tasks running simultaneously, regardless of amount of resources allocated.

This graph shows the total execution time of each battery, composed by the execution of 32 different simulations, considering both the availability and how resources are allocated on the server.

The use of 4 processors results in the shortest time.



(a) B. AF



(b) B. AA

Experiments - Number of Generations

This first experiment aimed to determine how much the outcomes vary with respect to the system evolution.

The average utility was compared every 5,000 generations.

Hypothesis A:

Agent's utility grows in higher generations.

Battery	k	d	g	$P\,(\mathrm{m})$	$P\,(\mathrm{e})$	Strategy
07	4	50×50	$\boxed{20{,}000}$	1 %	1 %	TT5

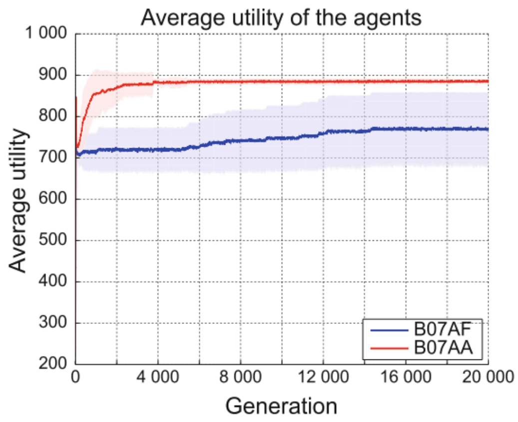
Results - Number of Generations

In the AA model the system stabilizes after the generation 4,000. However, in the AF model, we can observe a subtle growth until generation 16,000.

Therefore, the chosen number of generations is 5,000. This choice was made due to two reasons:

(i) In this generation the AA model is already stabilized.

(ii) Computational constraints given the number of resources available, and the number of experiments proposed for this work.



Experiments - Grid Size

The following batteries were designed to analyze how the grid dimension influences the strategies used by the agents.

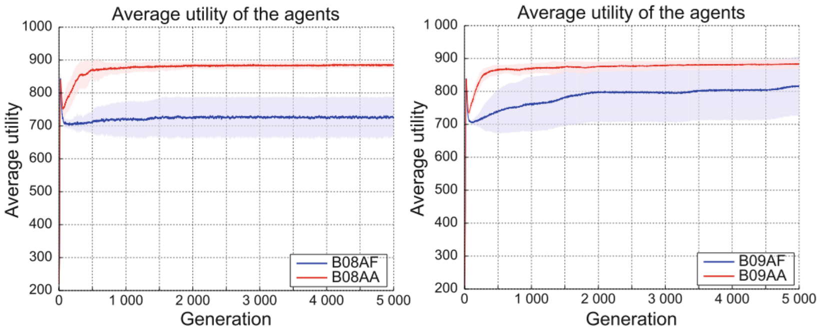
Hypothesis B:

Agent's utility grows in grids with higher dimension.

Battery	k	d	g	$P(\mathrm{m})$	$P(\mathrm{e})$	Strategy
08	4	50×50	5 000	1 %	1 %	TT5
09	8	128×128	5 000	1 %	1 %	TT5

Results - Grid Size

In the case of the AA model, there is a significative increase in the average utility of the agents in the larger lattice, possibly due to the diversity of strategies produced by the experiment.



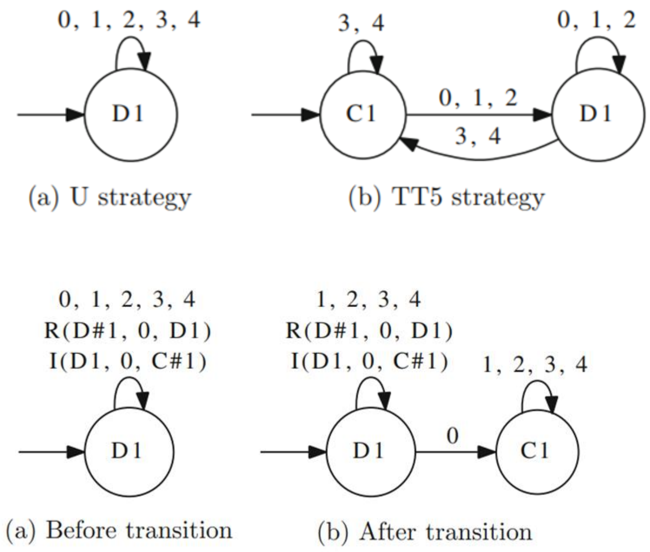
(a) Battery 08 (b) Battery 09

Game Strategies

U: The agent start playing defect and maintain this play, independently of the other players.

TT5: The agent start playing cooperate and then follows the last play of most of the players (Tit for Tat 5 Players).

CCD: The agent starts playing cooperate and maintains this play until another player defects. Then his behavior will be adapted according to other players' strategies.



Strategies used in the simulation.
Source: Reference [1]

Experiments - Initial Strategy

In this experiment we evaluate the three different initial strategies mentioned before using both AA and AF to describe the behavior of the agents:

- **U** (Always Defect)
- **TT5** (Tit For Tat 5 Players)
- **CCD** (Adaptive Strategy)

Hypothesis C:

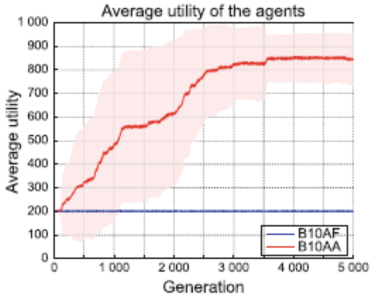
A variation in the initial strategy increases the agent's utility.

Battery	k	d	g	P (m)	P (e)	Strategy
10	4	50×50	5 000	1 %	1 %	U
11	4	50×50	5 000	1 %	1 %	TT5
12	4	50×50	5 000	1 %	1 %	CCD*

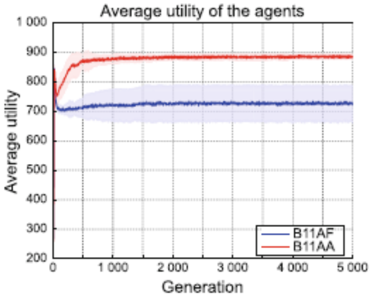
⋆ Only applicable to the B.AA model.

Results - Initial Strategy

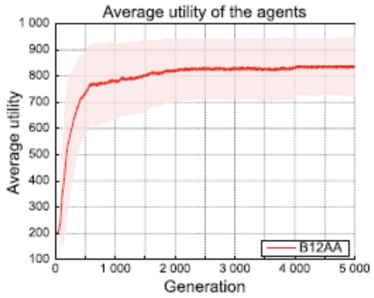
The experiments using TT5 (B11) as the initial strategy had the best performance, and those using U (B10) had the worst performance.



(a) Battery 10



(b) Battery 11



(c) Battery 12

Experiments - Mutation Rate

In order to increase the diversity of the population and to observe its effect, some batteries were included varying the probability of mutation $P(m)$.

Since new strategies are generated only by mutation, this parameter becomes important in our study.

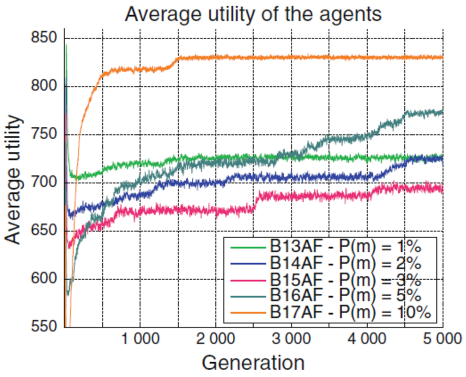
Hypothesis D:

If we increase the probability of mutation, the average utility increases.

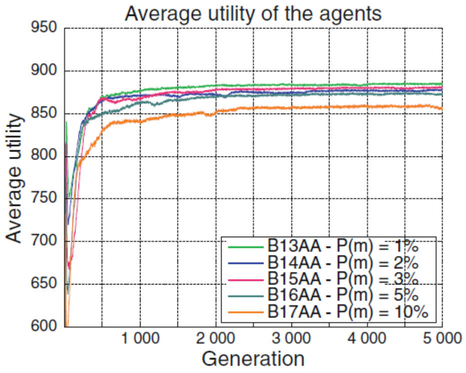
Battery	k	d	g	$P(\mathrm{m})$	$P(\mathrm{e})$	Strategy
13	4	50×50	5 000	1 %	1 %	TT5
14	4	50×50	5 000	2 %	1 %	TT5
15	4	50×50	5 000	3 %	1 %	TT5
16	4	50×50	5 000	5 %	1 %	TT5
17	4	50×50	5 000	10 %	1 %	TT5

Results - Mutation Rate

The test results are shown. We observe that our hypothesis was confirmed in the AA model; on the other hand, we could not confirm it when using the AF model.



(a) B. AF



(b) B. AA

Experiments - Error Rate

In these batteries, the effect of the error rate, i.e the probability that the player chooses a play not consistent with his strategy, is evaluated.

Error rate of 0% corresponds to the situation where the agents always play according to their strategy.

Hypothesis E:

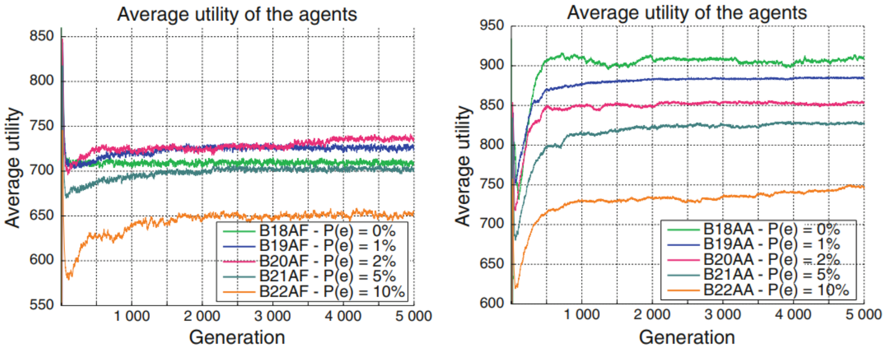
If we increase the error rate, the average utility increases.

Battery	k	d	g	$P\,(\mathrm{m})$	$P\,(\mathrm{e})$	Strategy
18	4	50×50	5 000	1 %	0 %	TT5
19	4	50×50	5 000	1 %	1 %	TT5
20	4	50×50	5 000	1 %	2 %	TT5
21	4	50×50	5 000	1 %	5 %	TT5
22	4	50×50	5 000	1 %	10 %	TT5

Results - Error Rate

In the AF model, the 2% value was the one that generated the greatest average utility.

Regarding the AA model, the agent's average utility was also reduced when the error rate was increased. The higher average utility was obtained with a value of 0%.



(a) B. AF (b) B. AA

Experiments - Model Expressivity

All the results obtained in the second set of experiments, which comprises the batteries 07 to 22, are used to verify the influence of the complexity of the strategy representation formalism on the simulation outcomes.

Hypothesis F:

Using a more expressive language to represent agents' strategies (AA model) increases the average utility when compared to a less expressive formalism (AF model).

Results - Model Expressivity

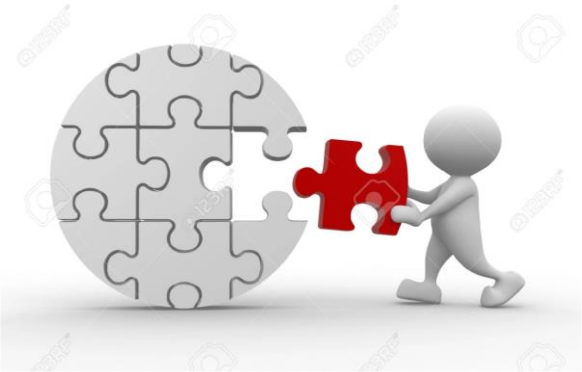
The use of the AA model to represent the agents ’strategies resulted in an increase in the population’s utility in all experiments.

This can lead us to conclude that the use of more complex mechanisms for the representation of strategies influences positively the gain of society.

Conclusions

These extended results were made possible by running the simulation in a cluster, applying parallelization techniques.

By using such techniques, we could better study the effect of several additional parameters on the simulation outcome, such as the grid dimension and the error rate.



Thanks For Listening!

References

1. Queiroz, Diego, and Jaime Sichman. "Parallel Simulations of the Iterated n-Player Prisoner’s Dilemma." International Workshop on Multi-Agent Systems and Agent-Based Simulation. Springer, Cham, 2015.
2. Axelrod, R.: The Evolution of Cooperation. Basic Books, New York (1985).
3. Carroll, John W. "Iterated N-player prisoner's dilemma games." Philosophical studies 53.3 (1988): 411-415.
4. Queiroz Macedo, Diego, and Jaime Simao Sichman. "Analysis of von neumann neighborhoods in parallel multi-agent simulations." 2010 Second Brazilian Workshop on Social Simulation. IEEE, 2010.
5. Andreï-Bogdan Sterescu. The Prisoner’s Dilemma in Environmental Politics: One Model to Rule Them All?