A Comparison of Cooperative and Non-Cooperative Policies in a Strategic Multiplayer-Game

Carolin Dohmen

26.02.2017

Complex Adaptive Systems

Prof. Dr. Michael Köhler-Bußmeier

HAW Hamburg

Contents

Abstract

Examining the pros and cons of cooperative behavior in comparison to egoistic, non-cooperative strategies has been and still is of relevance for various scientific areas. Topics like collective behavior, affiliation and group formation are of great interest, not only for social science, but also for economic, political, biological and computer science research.

This article introduces a multi-agent model that has been created in order to simulate cooperative and non-cooperative behavior in a simple game scenario. Referring to the strategic board game *Auf Achse*, effects of both policies are investigated and set in context to behavioral strategies in the game. These effects will then be discussed.

1 Introduction

Effects of cooperation and competition have been investigated for centuries. Many researches out of widely varying fields have confronted themselves with related topics. In 1984, Robert Axelrod, a political scientist, addressed the following question:

> Under what conditions will cooperation emerge in a world of egoists without central authority? [Axelrod, 1984]

Axelrod had presented a game theoretic solution to give an answer to this question and had thus delivered a paradigm changing approach.

Agent-based computational models present a widely recognized and often used method to support game theory. Whereas the latter delivers mathematical models, agent-based models show many parallels to game theoretical aspects [Wooldridge, 2009, 10]. Agents hereby represent self-interested players and tasks are modeled as games between agents or agents and their environment [Wooldridge, 2009, 42].

In this article, an agent-based model will be used to compare the success of cooperative and egoistic policies. The two extremes, cooperation and egoism, will be manifested in two different groups of agents. One group will cooperate with other individuals of their kind in sharing rewards, the other group will act selfishly in keeping all rewards for themselves. The success of a policy will be measured by the population size of a group. Successful agents will reproduce, agents following non-effective strategies will be eliminated. Therefore the model also adapts evolutionary game theoretical aspects.

An expected outcome of this approach is to find advantages and disadvantages of both strategies, which can be deduced from the data created through simulation runs under varying environmental conditions. It is assumed that both populations show different reactions to those varying conditions. Also cooperative behavior is expected to have certain advantages over egoistic behavior, since cooperation has proved itself as an effective strategy within evolutionary processes.

2 Methods

2.1 Coalitional Game Theory

Whereas traditional game theory focuses on non-cooperative policies, *coalitional* or *cooperative game theory* examines cooperation within games, as the name might suggest. In [Shoham and Leyton-Brown, 2008], a coalitional game is defined as follows:

> **Definition 2.1 (Coalitional game with transferable utility)** *A coalitional game with transferable utility is a pair (N, v), where*
>
> - *N is a finite set of players, indexed by i; and*
> - *$v : 2^N \mapsto \mathbb{R}$ associates with each coalition $S \subseteq N$ a real-valued payoff $v(S)$ that the coalition's members can distribute among themselves. The function v is also called the* characteristic function*, and a coalition's payoff is also called function its* worth. *We assume that $v(\emptyset) = 0$.*
>
> [Shoham and Leyton-Brown, 2008, 384]

Likewise, two main questions regarding coalitional game theory are formulated:

> - Which coalition will form?
> - How should that coalition divide its payoff among its members?
>
> [Shoham and Leyton-Brown, 2008, 384]

In case of this article's model, the first question will be ignored since there is only one big coalition, namely the cooperative agents group. The second question will be handled with a simple approach. The characteristic function will be realized through reward values distributed in the environment. The corresponding payoff vector will be defined by

$$x \in \mathbb{R}^N : x_i = \frac{v(N)}{N},\ i \in N$$

with N being the population size of cooperative agents.

The main question now is: How do cooperative agents compete against egoistic agents?

2.2 Auf Achse

The game *Auf Achse* is a strategic board game possessing economic and logistic elements [Kramer, 1987]. Multiple players compete against each other whilst accomplishing delivery missions and bidding for new supply orders in order to increase their asset. The environment is a abstracted graph of Middle-European cities and roads. Players are represented by trucks that can be moved around on the map.

Extensions of the game include cooperative behavior between players. Single truckers or forwarding agencies can form syndicates and therefore divide missions into submissions in order to gain market advantages.

The link between *Auf Achse* and the model proposed in this article is the question, how successful cooperative strategies are in comparison to solely competitive conduct. To achieve this, the model captures selected aspects of the game, which are simplified in order to reduce complexity.

Movable agents represent players, rewards and step costs describe supply orders and behavioral strategies of different agent types simulate the formation of coalitions or a sheer competitive policy.

2.3 Modeling Environment

A wide range of multi-agent modeling and simulation tools has been developed in the last decades. Many of those frameworks do not offer a graphical simulation environment and also require deeper knowledge about the syntax of respective modeling languages. For the purpose of creating simple models in a short period of time, the framework *NetLogo* [Wilensky, 1999] constitutes a good choice. NetLogo is a discrete-event agent modeling and simulation environment, combining a graphical interface and a code editor in a single program. This way it is possible to see changes in simulation scenarios immediately after editing the model code.

The language used in this framework is a dialect of the programming language Logo [Tisue and Wilensky, 2004]. In NetLogo, there are four kinds of predefined agents, which can be explained described as follows [Railsback and Grimm, 2011]:

- *Turtles* depict mobile individuals that have a position and are able to move around in their environment. In many cases, turtles are used to represent spacial agents in a static environment. The standard graphical presentation of turtles is an arrow, as it can be seen in screen shots of the model.
- *Patches* are static agents represented by squares arranged in a grid. Often patches represent the environment that spacial agents respective turtles are placed in.
- The *Observer* is a special kind of agent. Its appearance is unique within a simulation. The observer has power over other agents, it can give instructions as well as deliver information about state variables of all other agents.
- *Links* are agents that are specifically designed to represent connections between agents. Links can be directed or undirected and are graphically depicted as lines between the agents. In this model, no links are used.

3 Model Description

In order to describe the model, a standardized description model will be used, which has already proven its value for the design and conception phase of building multi-agent models. The *ODD Protocol* has been developed to give a concise overview on the model configuration [Grimm et al., 2006]. For the following model description, an updated version of the protocol is being used [Grimm et al., 2010].

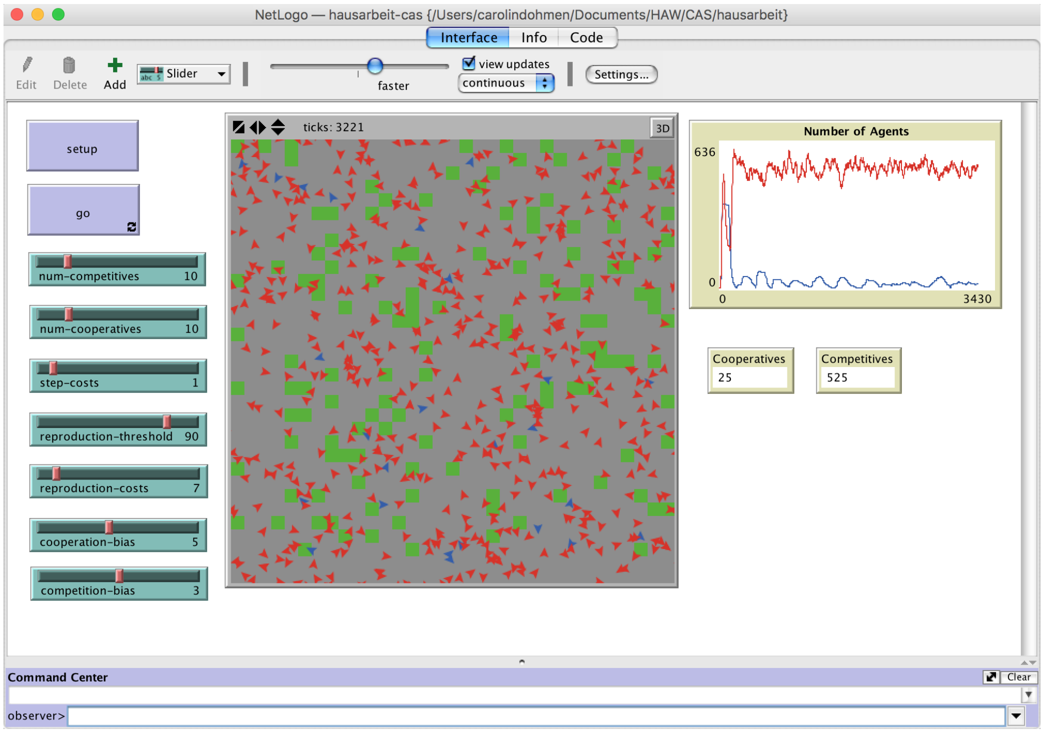

Figure 1: Screenshot of the simulation interface

3.1 Overview

Purpose: This model has been implemented in order to investigate the impacts of cooperative and non-cooperative behavior on agent population sizes. Therefore success rates of different strategies are being measured in a simple game scenario.

Entities, state variables and scales: The main actors in this model are primitive spacial agents. Every agent has an inner representation of its overall fitness and a definite position in its environment. When initialized, agents receive a start value of 80 fitness points. Furthermore, their position is determined by random.

A second general type of agents are represented by the so called patches. These entities act stationary, whereby all patches as a compound form the environment of mobile agents.

A grid of patches in shape of a torus has been chosen to represent the agents' environment. The state of patches is determined by a reward value, which is zero points by default. Within simulation, rewards are distributed randomly over the grid, scaling from one to 50 points.

There are two kinds of agents: One breed shows cooperative behavior, the other one is designed to be egoistic. A further explanation on behavioral traits will be given in the next paragraph.

In addition to that, there are seven global variables that can be defined by controllers before starting a simulation. The first two are the initial numbers of cooperative and non-cooperative agents in the scenario. The other controllers determine step costs, a reproduction threshold and reproduction costs. Also, there are bias factors regarding cooperative and competitive behavior. The role of these values will be explained in the following paragraph.

Process overview and scheduling: Within simulation, time is modeled in form of discrete steps. According to NetLogo as a discrete event simulation environment, agents perform their actions in every *tick*, which represents a single step. There is a fixed scheme of processes that are being performed in a tick. Figure 2 gives an overview on this process outline. Variations of individual agents' behavior will become apparent within single processes and are described in subsection 3.3.

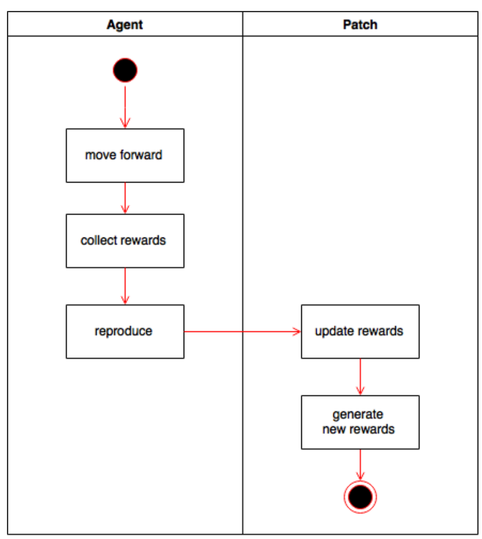

Figure 2: Diagram of activities in a single tick

3.2 Design Concepts

Basic Principles: The actual underlying principles can be described as co-operative and non-cooperative behavior in an agent-based model simulating a simple game scenario. These behavioral strategies are manifested in how agents proceed after having found a reward in their environment. Cooperative agents share the sum and therefore make sure that other agents profit as well. In contrast to that, egoistic agents do not share and only profit from the rewards themselves, yet they are only able to collect small and medium size rewards. These two principles and their comparison portray the main idea behind the model and determine its overall purpose.

Emergence: The most remarkable aspect that changes with varying parameters is the different population sizes that emerge during the simulation. First, the maximum amount of living agents of one kind is a value that is interesting for simulation analysis. Second, a ratio of the number of competitive in comparison to non-competitive agents is a value that may vary significantly.

Adaptation: In this model, agent strategies are deterministic. The breed of an agent constitutes its behavior in the simulation, so there is no adaptive behavior. Success of different strategies becomes apparent in population sizes, not in individual behavior leading to success.

Objectives: The main goal of all agents is to increase their fitness points in order to stay alive and reproduce. There are no alternatives or decision possibilities. An agent's fitness is captured by its energy points. Reproduction success of course shows itself in the number of same breed agents living on the grid.

Learning: As stated above, agents do not show adaptive behavior and therefore are not equipped with the ability to learn.

Prediction: Cooperative as well as non-cooperative agents do not make assumptions on the success of their policy. Both policies are deterministic, therefore all agents are bound to their breed-specific behavior strategy.

Sensing: In every tick, agents inspect the patch they are currently residing on, more specifically they check the patches' reward value. Based on this, rewards are distributed within the agent populations. Another more passive form of sensing becomes apparent in the behavior of cooperative agents. When having found a reward, it is divided and distributed within the cooperative agent group. This assumes that agents have knowledge about their group, so that other members can be provided with partial sums.

Interaction: The degree and complexity of interaction in this model is relatively low. The only interaction taking place during the ticks is the distribution of rewards within the cooperative agent group. While performing this action, agents have access to fitness points of agents of their kind, so that this value can be increased.

Stochasticity: There are a few processes including random elements. During simulation setup as well as in the simulation ticks, rewards are distributed randomly over the grid environment, meaning that their position is chosen by random and the amount of the reward is randomly determined within a certain scope.

Listing 1: Code of reward distribution

```
ask n-of 50 patches [
  [...]
  set reward random 50
]
```

Furthermore, the movement of the agents is modeled non-deterministic. On one hand, agents turn at a degree between one and 360. On the other hand, they move forward a number of steps that is also decided by random.

Listing 2: Code of agent movement

```
to walk
  rt random 360
  let stride-length random 5
  fd stride-length
  ifelse breed = cooperatives [
    set energy energy - stride-length * step-costs
  ] [
    set energy energy - competition-bias
                * step-costs * stride-length
  ]
  [...]
end
```

Finally, another element is located in the reproduction process. If an agent's energy level is above the reproduction threshold, a new agent is generated with a probability that is equal to one divided by the energy level value.

Listing 3: Code of agent reproduction

```
to reproduce
  if energy > reproduction-threshold
      and random 100 < energy [
    [...]
    hatch 1
  ]
end
```

Collectives: A main feature of this model is the cooperating group of agents and their behavior. These agents have knowledge about other cooperating individuals, furthermore they can actively influence the others' state. Cooperative traits are determined by the agent breed, which cannot be changed during simulation.

Observation: The most interesting data that can be collected in a simulation are the numbers of agents changing over time. Therefore, there are two monitors observing the current size of both agent groups as well as a diagram showing the number of agents of both groups plotted against the number of executed ticks.

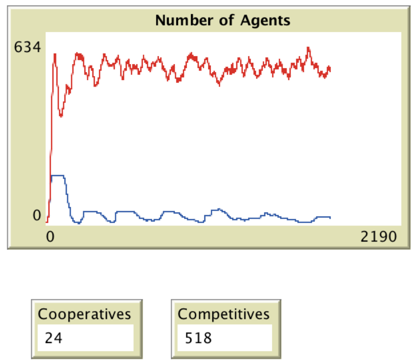

Figure 3: Screenshot of the agent monitor

3.3 Details

Initialization: There are seven values that have to be set during the initialization process. First, the user has to state the initial numbers of cooperative as well as competitive agents. In addition to that, step costs, a reproduction threshold and reproduction costs have to be defined. Finally, bias factors against competitive and in favor of cooperative behavior have to be added. Except for the first two, none of the values are subject to change.

Input data: No external sources are being used as input. All of the observable effects emerge from the agents' actions during simulation time.

Submodels: Several procedures can be identified as submodels. This description refers to the illustration in Figure 2.

- *move forward* : All agents begin with a random movement on the grid. After turning at a random degree and moving one step forward, step costs are subtracted from their fitness points. If after this one agent's energy level is below zero, it is removed from the simulation.

- *collect reward*: The next action marks an essential difference between the two agent breeds. When visiting a patch on the grid containing a reward greater than zero, cooperative individuals divide the sum and share it with other individuals of the same kind. On the other hand, egoistic agents keep the reward wholly for themselves, but with the restriction that they can only utilize rewards less than 30 points. After being visited by an agent, the reward value is set to zero again.

- *reproduce*: If after this the energy level is above a certain threshold value, agents can reproduce. The probability for this event is determined by its fitness points. First, reproduction costs are subtracted from its energy level, then a new agent is spawned. The new agent's initial energy level is equal to its parent's value.
- *update rewards*: The next actions are performed by the patches. If the reward value of a patch is equal to zero, it is colored gray.
- *generate new rewards*: Finally, fifty patches are chosen randomly from the grid and equipped with a reward in the scope of one to fifty. Again, the color is changed to green in order to identify them.

The next actions are performed by the patches. First, their color is adjusted according to their reward value. If the reward is equal to zero, patches are being colored gray. Finally, twenty randomly chosen patches are equipped with a reward in the scope of one to fifty and again are colored green.

4 Results

In the following section, effects caused by variations of different initial parameters will be shown. Every subsection focuses on a certain type of parameter. Except for the part explicitly describing the bias factors, these values will mostly be assigned with 2, if nothing else is mentioned. The reason is that like this it is easier to describe influences of other parameters, as populations seem to be more stable being influenced by these biases.

4.1 Number of Agents

Varying the initial numbers of agents of both groups, cooperative and non-cooperative populations show differences in growth and stability. With a high number of cooperatives, the competitive group is more likely to collapse during the first period of simulation. With less than 25 individuals, the probability of extinction is getting higher. In 1000 simulation runs with initial numbers of 50 cooperatives and 5 competitives, the extinction share of competitive populations within the first 500 ticks has reached a value of 51%. In contrast to that, none of the cooperative populations ceased within the same experiment setup, but 50 competitives and 5 cooperatives instead.

In scenarios including equal sized initial agent populations, the initial size did not lead to significant differences in group expansion as shown in Table 1.

Population Size	*Average Competitives*	*Average Cooperatives*
5	41,55584726	17,15140811
50	41,48141968	17,1766739

Table 1: Average population sizes under varying initial number of individuals

4.2 Step Costs

Because of the competitive-bias factor, competitive agents are very sensitive to an increase of step costs. With a bias factor of 2.0 and step costs greater than 1.0, the competitive group will always collapse. In contrast to this, cooperative agents are more resistant to variations in step costs.

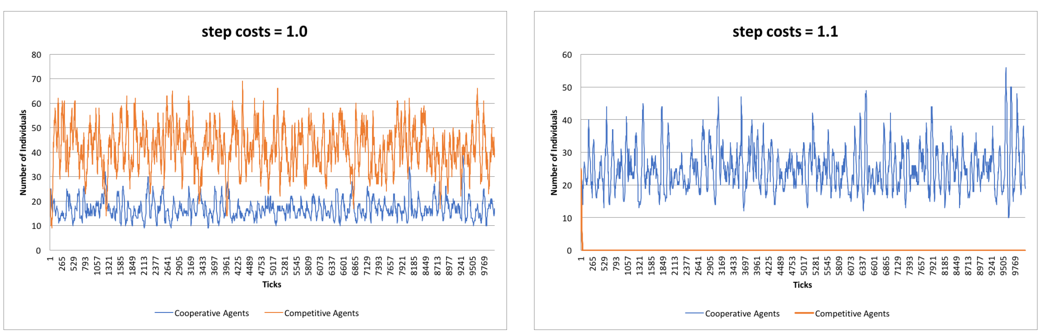

Figure 4: Effects of varying step costs on population sizes

Even if the factor is just a little bit higher than 1.0, the competitive agent group is not able to survive, as seen in Figure 4. Even though, if the competition bias is assigned with a value of 1.0 and the cooperation bias factor is increased, both populations can sustain higher step costs (Figure 5).

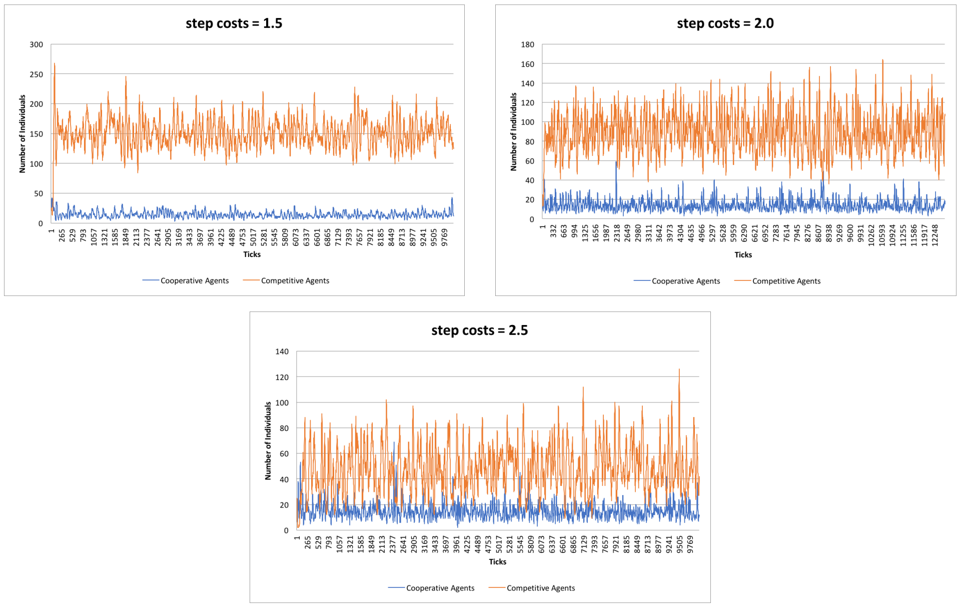

Figure 5: Population sizes with cooperation-bias = 4, competitive-bias = 1

4.3 Reproduction

The two parameters concerning the reproduction process are a threshold value and reproduction costs. Depending on the combination of both parameters (high: r-threshold = 95, r-costs = 30 / low: r-threshold = 55, r-costs = 5), population dynamics change significantly. Table 2 shows standard deviations within population sizes during one simulation run. If both parameters are assigned with low values, the deviation is remarkably higher. Also, dynamics are much less predictable than in other cases. An example for this can be seen in Figure 6.

Repr.-Threshold / Repr.-Costs	*STD Cooperatives*	*STD Competitives*
low / low	22,38234077	108,1873375
low / high	3,953337092	0,971843551
high / low	8,225156387	23,03696982
high / high	2,597843271	5,423607446

Table 2: Standard deviations of agent populations under varying reproduction parameter values

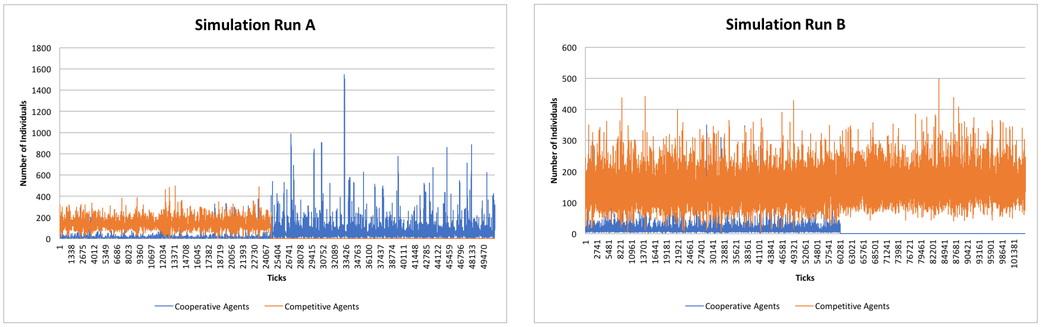

Figure 6: Two simulation runs with low reproduction threshold and reproduction costs

4.4 Biases

The bias factors are of special significance for the expansion and growth of both agent populations. As it can be seen in Figure 7, competitive agents are slightly able to survive with a bias factor greater than 2. With an increasing cooperative bias, the average population size decreases as well. Only if the cooperative bias is very low, competitive agents can sustain a bias factor of 2.

Figure 7: Competitive population sizes under varying bias factors

As expected, the maximum of average cooperative group size comes with a high cooperative bias and a low competitive bias factor. But in contrast to the other group, the population is not as sensitive to a change of bias than the competitive group. Still it is remarkable that with a low cooperative bias, average population sizes are very low.

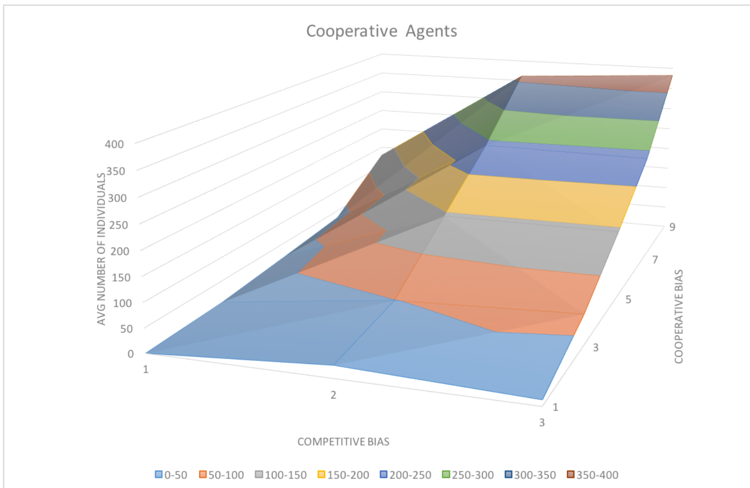

Figure 8: Cooperative population sizes under varying bias factors

5 Conclusion

5.1 Interpretation

The data collected from simulation runs under various conditions do not state a certain overall winner. Both groups have their strengths and weaknesses and none of them is able to succeed in any situation. More likely it is shown, that cooperative and competitive agent groups react differently to different changes of environmental factors. All in all, two main insights can be gathered from the experiments:

1. Competitive agents have a bigger chance of developing large populations. While the cooperative agents tend to form smaller groups, competitive agents are more successful in gaining fitness points. Due to the egoistic approach of keeping the whole payoff for themselves, their energy level rises quickly, so that the reproduction rate is relatively

high.
Cooperative agents were not even able to survive without implementing biases to advantage cooperation and disadvantage egoism. This shows that in order to evoke cooperative behavior, rewards have to be accordingly higher than just a partial sum of the mere payoff. Disadvantaging egoistic policies is not sufficient for the success of the cooperative individuals, as the results show.
Connection to Auf Achse: If building coalitions should constitute an attractive policy for agents, the benefits from this behavior should be considerably high. A mere division of the payoff is not sufficient for success. If fast success and a quickly gained high asset are the main goals, egoistic behavior will be the best strategy.

2. Competitive behavior is not a guarantee for success. Whereas egoistic agents are successful in gaining fitness points, populations are more sensitive to aggravating factors. With higher step costs, competitive agents are more likely to fail, as they do not have the backup of other agents working for their supply of energy points. Also, population dynamics are less predictable and underly higher fluctuations.
In contrast to that, cooperative agents are more resilient to those changes. In most of the cases population sizes stay at a low level, but are nevertheless relatively stable. As an increasing group size leads to lower single payoffs for every member, population sizes stay at a moderate level.
Connection to Auf Achse: Building coalitions is a suitable strategy if environmental factors often change. If one individual is being hindered to complete a mission and collect rewards, its chance of surviving is higher due to its coalition partners sharing their rewards. Therefore cooperative policies will be attractive for players seeking secure options and not wanting to take a high risk on pursuing the highest asset.

5.2 Criticism / Outlook

Running simulations with this model has led to discoveries that can be helpful for choosing between cooperative and non-cooperative strategies in a game context. However, the design of this model, including agents and environment, occupies a strong level of abstraction.

One problem of this approach is the fact that cooperative aspects are reduced to a minimum and do not depict the reality of coalitions. The model is an abstraction and therefore does not offer definitive answers to general questions concerning cooperation and egoism. Cooperative behavior is cut down to sharing rewards - no negotiation and further communication takes place between agents, therefore cooperating individuals are not capable of applying strategies more "clever" and successful than the one implemented in the model. This might be one reason for the small population sizes of cooperative agents and the lack of success in the competition between cooperative and non-cooperative agent groups.

Another point to criticize is the distribution of payoffs among cooperating agents. The formula used in this model is a very simple one, based on the assumption that the profit for each individual is high enough, if it just gets rewarded every time another group member receives a payoff. Looking at the simulation results, this is not the case, as cooperative agents were not as successful as expected. This indicates that in order to encourage cooperation, these policies will have to be rewarded higher than in this model.

Finally, the model does not cover learning and adaptive behavior. Agents do not have the possibility to choose between different actions and therefore are not able to develop different strategies and adapt to different situations.

Future versions of the model could implement the agents' freedom of choice for cooperative or non-cooperative behavior. A communication protocol could be included, so that individuals would be able to negotiate over supply orders and their fulfillment. If more than one group could exist at the same time, agents could choose which group to join. Furthermore, the characteristic function as well as the payoff vector could be modified, so that biases might not be necessary any more.

References

[Axelrod, 1984] Axelrod, R. (1984). The evolution of cooperation.

[Grimm et al., 2006] Grimm, V., Berger, U., Bastiansen, F., Eliassen, S., Ginot, V., Giske, J., Goss-Custard, J., Grand, T., Heinz, S. K., Huse, G., et al. (2006). A standard protocol for describing individual-based and agent-based models. *Ecological modelling*, 198(1):115–126.

[Grimm et al., 2010] Grimm, V., Berger, U., DeAngelis, D. L., Polhill, J. G., Giske, J., and Railsback, S. F. (2010). The odd protocol: a review and first update. *Ecological modelling*, 221(23):2760–2768.

[Kramer, 1987] Kramer, W. (1987). Auf achse. Schmidt Spiele.

[Railsback and Grimm, 2011] Railsback, S. F. and Grimm, V. (2011). *Agent-based and individual-based modeling: a practical introduction*. Princeton university press.

[Shoham and Leyton-Brown, 2008] Shoham, Y. and Leyton-Brown, K. (2008). *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press.

[Tisue and Wilensky, 2004] Tisue, S. and Wilensky, U. (2004). Netlogo: Design and implementation of a multi-agent modeling environment. In *Proceedings of agent*, volume 2004, pages 7–9.

[Wilensky, 1999] Wilensky, U. (1999). Netlogo.

[Wooldridge, 2009] Wooldridge, M. (2009). *An introduction to multiagent systems*. John Wiley & Sons.