Radboud University Nijmegen

Bachelor thesis

On the evolution of cooperation, trust and diversity

Jeroen Lappenschaar *(s0513458)*
JLappenschaar@gmail.com

Supervisor: Franc Grootjen

Abstract

Cooperation and trust have been well studied themes in the scientific world. How they relate is however unclear. This research investigates the effect that trust has on a cooperative population with the means of an evolutionary algorithm. This is done by comparing a cooperative model without trust to one with trust. The results of the models are compared at four things: the amount of cooperation, the conditions at which cooperation will occur, the stability of cooperation and the amount of diversity in a population. Results show how cooperation can explain only a small amount of diversity in a population, that trust explains unstable cooperation levels and that trust could explain different times of 'peace' and 'war'.

Nijmegen, November 14, 2010

Contents

Chapter 1

Introduction

This thesis focuses on two well researched aspects in the scientific world: cooperation and trust. Here, in the first part the most relevant background research on each of these two aspects is introduced, followed by the introduction of the research question of this thesis.

1.1 Cooperation and trust

1.1.1 Cooperation

Cooperation is an important aspect of our society. We can observe cooperation everywhere, both in the human world as in the animal kingdom. Humans and animals use cooperation to bundle their strength and solve tasks more efficiently.

How cooperation could have evolved is still under debate. In the words of Nowak [14]: "Evolution is based on a fierce competition between individuals and should therefore reward only selfish behavior. Every gene, every cell, and every organism should be designed to promote its own evolutionary success at the expense of its competitors". Different theories have been designed that could explain the evolution of cooperation, like kin selection, direct reciprocity, indirect reciprocity, network reciprocity and group selection. All these mechanisms can explain cooperation but all in a very different way and under different conditions [14]. How nature has evolved cooperation is still unclear, that evolution has evolved is, however, not under debate.

An interesting side effect of cooperation is growth of specialized individuals in a population. If a task is complex enough, certain individuals might specialize in specific skills so that they can work together more efficiently [4]. There has yet only been a small amount of research done after this specialization. Quinn et al. [16] showed an example of this with robots evolving functionally distinct roles so that each individual would have its own specialization and the assignment could be efficiently solved as a team. Koldijk [10] showed with the help of an evolutionary algorithm that individuals would specialize a lot when there is enough cooperation in a population.

1.1.2 Trust

An important aspect of cooperation is trust. Trust plays a serious role in daily life. Friendship and cooperation are build on trust and trust is at the basis of our society. Trust is an important measurement of how smart it is to cooperate with someone. Cooperating with someone you do not trust has a high probability of you being scammed.

How trust exactly plays a role in our world is still under debate in the scientific world. Scientist like Francis Fukuyama [9], Robert Putnam [15] and others say that trust is essential for large scale cooperation in societies. Other, however, like Cook, Hardin and Levi [6] say that trust only plays a part in small groups of individuals and that cooperation in bigger groups originates from rules and regulations made by state and other institutions.

That trust exists, however, is just like cooperation not under debate. Over the years, scientists in different fields of research have tried to make computational models with individuals being able to learn to trust one another. To name some:

- Axelrod and Hamilton [3] showed in 'The evolution of cooperation' that both cooperation as well as trust can evolve. They used an evolutionary algorithm in the iterated prisoners dilemma.
- Macy and Skvoretz [12] proved that widespread trust within a population can be established when the prisoners dilemma is played between strangers. Trust was able to evolve by giving individuals the possibility to not play the game. This comes with a small cost to an individual but the individuals show that this cost is more than acceptable against the risk that would otherwise been taken.
- Axelrod [2] showed that cooperation in the Prisoners Dilemma is possible when there is the foresight of more encounters with the same individual. This so called 'shadow of the future' however, needs to have an unlimited number of encounters, otherwise the individuals would not find it worth the risk to start trusting each other.
- Bravo and Tamburino [5] showed with a variant on the prisoners dilemma, the trust game, that trust might not be so stable. It also showed that the amount of cooperation switches between periods of high and low values.

1.1.3 Research question

The previous sections have shown that both cooperation and trust have been widely studied. However, the link between cooperation and trust itself has not. What is the difference between cooperation without a trust mechanism and cooperation with a trust mechanism? What is it exactly that trust adds to cooperation? To answer this, the research question of this thesis will be:

How does trust affect the evolution of cooperation?

The research question will be answered with the help of two subquestions:

- How does cooperation evolve without a trust mechanism?
- How does cooperation evolve with a trust mechanism?

First of all a computational model without a trust mechanisme will be used to answer the first subquestion. Then, this model will be extended with a trust mechanism to answer the second subquestion. Finally the results from both models will be compared to answer the research question.

To answer the research question and the subquestions the focus will be on four different aspects that evolve in the models:

- The amount of cooperation.
- The conditions when cooperation will occur.
- The stability of the cooperation.
- The amount of diversity between the individuals.

Answering the research question can give new insight in to what role trust exactly plays in cooperation and in our society. Furthermore it continues on the work from Koldijk [10] to see how cooperation affects the diversity of a population.

My hypothesis is that cooperation will evolve at the same moments in a model with trust as without trust. The amount of cooperation will in the model with the trust mechanism however, probably be a little lower due to the fact that not all individuals will trust each other. Previous research showed mostly stable results of cooperation and a diverse population.

1.2 Procedure

The fact that evolution can lead to cooperation was also shown by Koldijk [10]. Koldijk created a model to answer the research question: "To what extend can cooperation mechanisms (in combination with certain task types) explain the evolution of cooperative and diverse populations?". The cooperation mechanisms Koldijk talks about are described in Sachs et al [17] where different mechanisms are explained that should make it possible for created individuals to work together. The mechanism Koldijk chose to use in her model is the 'two way byproduct benefits', where the helping individual is rewarded with a small part of the total reward. This reward should be higher than any costs the helper makes, which should stimulate the helper to actually help.

Koldijk showed that the individuals would, with the help of the evolutionary algorithm and under the right circumstances, learn to cooperate. Further more, she showed how diverse populations arise with individuals specialized in different tasks. This diversity seemed to be a direct result of the high amount of cooperation. However, in this model the individuals did not have the possibility to cheat on the other individual and trust was not needed to cooperate. For the individuals the only thing that mattered was if they would get (more) profit by working together.

The model from Koldijk is a well designed model and ideal to answer the first subquestion: "How does cooperation evolve without a trust mechanism?". The model will therefore be used to answer this question. To investigate the aspect of trust I will extend the model from Koldijk with a trust mechanism. The results from this extended model will be used to answer the second subquestion: "How does cooperation evolve with a trust mechanism?". Finally, I will compare the results from the different models to answer the research question: "How does trust affect the evolution of cooperation?". The models will be compared on the amount of cooperation that evolves, the conditions when evolution evolves, the stability of the cooperation and the diversity of the population.

1.3 New results model Koldijk

The programming code from the original model was made available to me by Koldijk. While performing my research I stumbled on a programming error made in the model. I will discuss this error in more detail in Section 2.9. When the error was fixed different results were obtained from the model. Therefore I will also present these new results in this paper, which will also be used for the comparison between this model and the new one.

1.4 Thesis structure

This thesis is devided in two methods and results sections. The methods to answer the first subquestion are explained in Chapter 2 and the related results are shown in Chapter 3. Secondly, the methods of the second subquestion are explained in Chapter 4 and Chapter 5 shows these results.

Concludingly, some ideas for further research are given in Chapter 6 and this thesis will end in Chapter 7 with the conclusion.

Chapter 2

Methods model without cooperation mechanism

This Chapter will describe the methods that are used to answer the first subquestion of this research: 'How does cooperation evolve without a trust mechanism?'. As said, the model from Koldijk will be used to investigate the evolution of a population without a trust mechanism. In this Chapter the model is described as it was designed by Koldijk.

The model is an evolutionary algorithm with individuals that are each allocated with an assignment. This Chapter is ordered as following. First the model-specific settings are explained: the population (Section 2.1), the assignments (Section 2.2), the cooperation mechanism (Section 2.3) and the settings of the evolutionary algorithm (Section 2.4). Next an overview of these settings (Section 2.5) are given together with an example for clarification (Section 2.6), followed by the implementation (Section 2.7) and the experimental set-up (Section 2.8).

2.1 The population

The model as designed by Koldijk has a population of 100 individuals. The goal of each individual is, as is in nature, to survive. This is done by gaining as much fitness[1] as possible. In the model, individuals can obtain fitness by solving assignments. Every individual will be assigned multiple assignments in each generation. The more efficiently they solve these assignments the better their fitness-score will be.

Individuals can solve assignments with the help of their own skills. Every individual is identified by three skills: skill A, skill B and skill C. These skills determine how well the individuals perform at the execution of a specific task. The skills are figures between 0.1 and 1. The higher the skill, the better the individual is in solving the task. An individual can never totally lack a skill and therefore the minimum of a skill is 0.1.

The skills are the genes that are being evolved using the evolutionary algorithm. Koldijk chose to lay a restriction on the skills; the sum of the skills should be equal to 1. This way the individuals have to choose how they divide the skills. Koldijk defends this restriction by referring to trade-offs that also occur often in nature: "... an individual specialized on prey A loses the ability to efficiently capture prey B". This also solves the problem that all individuals would convert all their skills to 1, which is the optimal solution. At the beginning of the first generation the individuals start with a random value for every skill and through a normalization function it is guaranteed that the sum of the skills equals to 1.

The gene distribution of a start population is shown in Figure 2.1. On the x-axis skill A is plotted and on the y-axis skill B. Skill C can be derived since the total of the genes sums up to one.

[1]The fitness-score is the score used in evolutionary algorithms to measure the performance of the individual. At the hand of this score the chance of survival of this individual will be determined.

A point at the coordinates (0.2, 0.3) represents an individual with the skill levels: {0.2, 0.3, 0.5}. One can see that the genes of the individuals are random divided and a heterogeneous population has emerged.

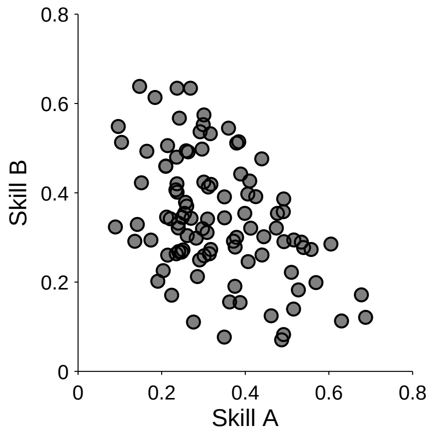


Figure 2.1: *Random initialization of the population. The horizontal axis shows the skill level of A, the vertical axis that of skill B. Skill C can be derived since the total of the skills adds up to exactly one. The Figure shows a heterogeneous population.*

2.2 The assignment

Every individual gets per generation ten assignments that have to be solved. An assignment is abstract and is only defined by the tasks that are needed to solve it. There are three different tasks each corresponding to a specific skill:

- Task A: is solved with skill A.
- Task B: is solved with skill B.
- Task C: is solved with skill C.

An assignment can consist of one or two tasks. An assignment where only task A is needed is notated as {A}, an assignment where task B is needed two times is notated as {B, B}. With these restrictions in mind there are three different kinds of assignments:

- Assignments consisting of one task

  A simple assignment, consisting of only one task: {A}, {B} or {C}

- Difficult assignments consisting of two tasks

  A different assignment, consisting of two different tasks: {A, B}, {A, C} or {B, C}.

  A same assignment, consisting of two same tasks: {A, A}, {B, B} or {C, C}.

An individual is obligatory to solve the assignment. Koldijk chose for this restriction since most of the real life assignments are. It also keeps the experimental setup simpler. An individual can solve an assignment by solving all of the containing tasks.

By solving the assignment the individual will get a reward. However, using the skills to solve a task also means making costs. How the reward and the costs are exactly defined will be explained in Section 2.4.1.

2.3 Cooperation

As in real life one can always solve a simple assignment alone. However, when an assignments gets bigger or more complicated it might be more efficient to work together. This also holds for the individuals in the model. They will always solve simple assignments alone, but for difficult assignments (assignments with multiple tasks) they will always check the possibility for cooperation.

For the cooperation the assignment will be split up in the different tasks. Koldijk gives a nice metaphor to show how an assignment can be split up in tasks. She gives an example of a situation where an individual has to drive a car to a specific location:

> The skills are map reading and car driving. You might be good in map reading, but have no drivers' license. You can ask someone to help you to fulfill the task by driving. Maybe you have a drivers' license and also have the skill of map reading. But nevertheless it will probably be better and more efficient to fulfill the tasks with a partner, so that one person can focus on driving and the other on map reading.

To solve the assignment cooperative three individuals will be randomly chosen from the population. The individual can choose one of these three individuals to help. Koldijk chose to be able to pick out of only three random picked individuals since in real life you would also only have "a limited number of people that are around to help you". Koldijk chose to keep this setting fixed, she showed that small variations in this value showed no difference in the results.

The cooperation is a split up of the two tasks that are part of the assignment, each individual takes care of one task. The helper that gives the individual the highest fitness-reward is chosen to help. The distribution of the two tasks is chosen by picking the distribution where the overall cost is lowest.

A helper will however only help when its reward is bigger than the costs it makes. This reward is a percentage of the reward of the assignment and is one of the parameters in the model, the so called proportion reward (pr). The reward that the individual and the helper obtain is their contribution to the fitness-score. Conclusively, both individuals carry out their assigned task. The cost that they make with their task is for their own account. At the end the individual gets the reward minus the percentage that goes to the helper.

To sum up the restrictions that are laid upon the cooperation:

- Cooperation can only occur for difficult assignments, where the tasks can be split.
- An individual will, if it is cheaper of cooperating than working alone, always cooperate.
- A helper is, if its reward is higher than the cost, always willing to help.
- If none of the helpers will help the individual is obligatory to solve the assignment on its own.

2.4 Evolutionary algorithm

In the previous part the definition of the individuals, the assignment and the solving of the assignments have been discussed. In this Section the structure of the evolutionary algorithm is explained together with the cost and reward functions.

2.4.1 Fitness

The fitness of a individual is zero at the start of every generation. For every assignment that an individual solves, or for every assignment where it helps, it gets a reward which is added to its fitness score, the so called fitness contribution score (FC). The FC of an individual when it works alone is defined as follows:

$$FC_{ind} = Reward(assignment) - Cost(assignment, skills) \tag{2.1}$$

The reward function when there is going to be cooperated is:

$$FC_{ind} = ((1 - pr) \times Reward(assignment)) - Cost(task_{ind}, skills_{ind}) \tag{2.2}$$

$$FC_{helper} = (pr \times Reward(assignment)) - Cost(task_{helper}, skills_{helper}) \tag{2.3}$$

As said, pr is the proportion reward, the percentage of the reward that is for the helper.

The reward is defined as:

$$Reward(assignment) = 10 \times Sum(tasks) \tag{2.4}$$

This basically means that for a simple assignment the reward equals to 10 and for a difficult assignment, which consists of two tasks, to 20.

The cost function that comes with solving an assignment is defined as:

$$Cost(assignment, skills) = \frac{taskA}{skillLevelA} + \frac{taskB}{skillLevelB} + \frac{taskC}{skillLevelC} \tag{2.5}$$

Here a task is represented by a 0 or 1, depending on whether it has to be solved by the individual or not. This means that if an individual has to solve a task, the costs for it will be 1 divided by the corresponding skill-level of the individual. The costs of an assignment for an individual than equals to the sum of the costs for every task the individual has to solve.

Solving a difficult assignment alone has extra costs. This is because solving an assignment alone can be done less efficiently. Therefor the costs of a difficult assignment that is solved alone is multiplied by a factor of 1.3. Koldijk chose this value with the help of pilot studies and to prevent the experiment from becoming more complex the parameter was fixed.

2.4.2 Mutation

At the start of a new generation every individual has a chance of 1/3 for every gene that it is mutated. For the mutation a Gaussian distribution is used with $\mu = 0$ and $\sigma = 0.05$. It is assured that a gene is at least 0.1 and at most 1.0, next the genes are normalized so that the sum equals to one.

Koldijk used in her genetic algorithm no cross-over because mutation takes care of enough variation since there are only three genes.

2.4.3 Parent selection

At the end of each generation only the best individuals survive. This is done with a tournament selection. From the hundred individuals three individuals are randomly chosen, from these three the best one (the one with the highest fitness) is selected for the next generation. This is done a hundred times, an individual can therefore be chosen several times. This way, the best individuals will form a big part of the new generation while the less good individuals will only form a small part of the population.

2.4.4 Generations

Koldijk chose to set the model to run for 200 generations. Her results and my pilot experiments showed stable results far before these 200 generations. Therefore I chose to reduce the number of generations to 100. This way the effect of the generations is better visible in the results and the results are still stable far before these 100 generations.

2.5 Overview

For clarification, the most important aspects of the model are summarized in this Section. An overview of the different assignments and tasks is given in Table 2.1.

Assignments	Simple, different and same assignments
Tasks	task A, task B, task C
Skills	skill A, skill B, skill C
Value skills	minimal 0.1, maximal 1; Sum(skillA, skillB, skillC) = 1

Table 2.1: *An overview of the assignments and tasks in the model from Koldijk.*

The most important settings for the evolutionary algorithm are shown in Table 2.2.

Population size	100
Length of genomes	3
Minimum value of genes	0.1
Maximum value of genes	1
Tournament selection method Tournament selection with size	3
Recombination	-
Mutation probability	1/3
Mutation type Gaussian	$\mu = 0$, $\sigma = 0.05$
Termination	After 200 Generations

Table 2.2: *An overview of the evolutionary settings as used by Koldijk. Table from Koldijk [10]. The same settings were used in the model without the cooperation mechanism, except for the number of generations which was reduced to 100.*

In Figure 2.2 the model from Koldijk is visualized in a scheme. It shows the steps taken by every individual every generation. One can clearly see the three possible routes to fulfilling an assignment; solving it alone when there is only one task, solving it alone when there are multiple tasks and solving the assignment together.

2.6 Example

In this Section an example is given for clarification. The model is adjusted for different assignments and the proportion reward is set to 0.25. Imagine individual A, who has the genes: (0.4, 0.3, 0.3), gets the assignment: $\{A, B\}$. There are multiple tasks so the individual asks for help, three individuals are randomly picked from the population:

- Individual X (0.8, 0.1, 0.1)
- Individual Y (0.1, 0.5, 0.4)
- Individual Z (0.4, 0.4, 0.2)

If individual A would cooperate with individual X it would mean that individual A performs task B (even though it cannot use its best quality for this) and individual X task A. This is because the distribution of tasks is determined by the lowest total cost: $1/0.8 + 1/0.3 < 1/0.4 + 1/0.1$.

However, the highest fitness can A obtain with the help of individual Y where the fitness score for A is: $((1 - 0.25) \times 20) - (1/0.4) = 12.5$. Cooperation with Z is less profitable for A. Individual Y gets in this case a fitness contribution of $(0.25 \times 20) - (1/0.5) = 3$. Keep in mind that this is the contribution score, this score is added to the current fitness score of the individual, which consists of the score from earlier performed assignments or cooperation tasks within the current generation.

When all individuals have performed the ten assignments the selection procedure starts. Hundred individuals are selected for the next generation, but before the new generation starts the mutation function is performed over all individuals.

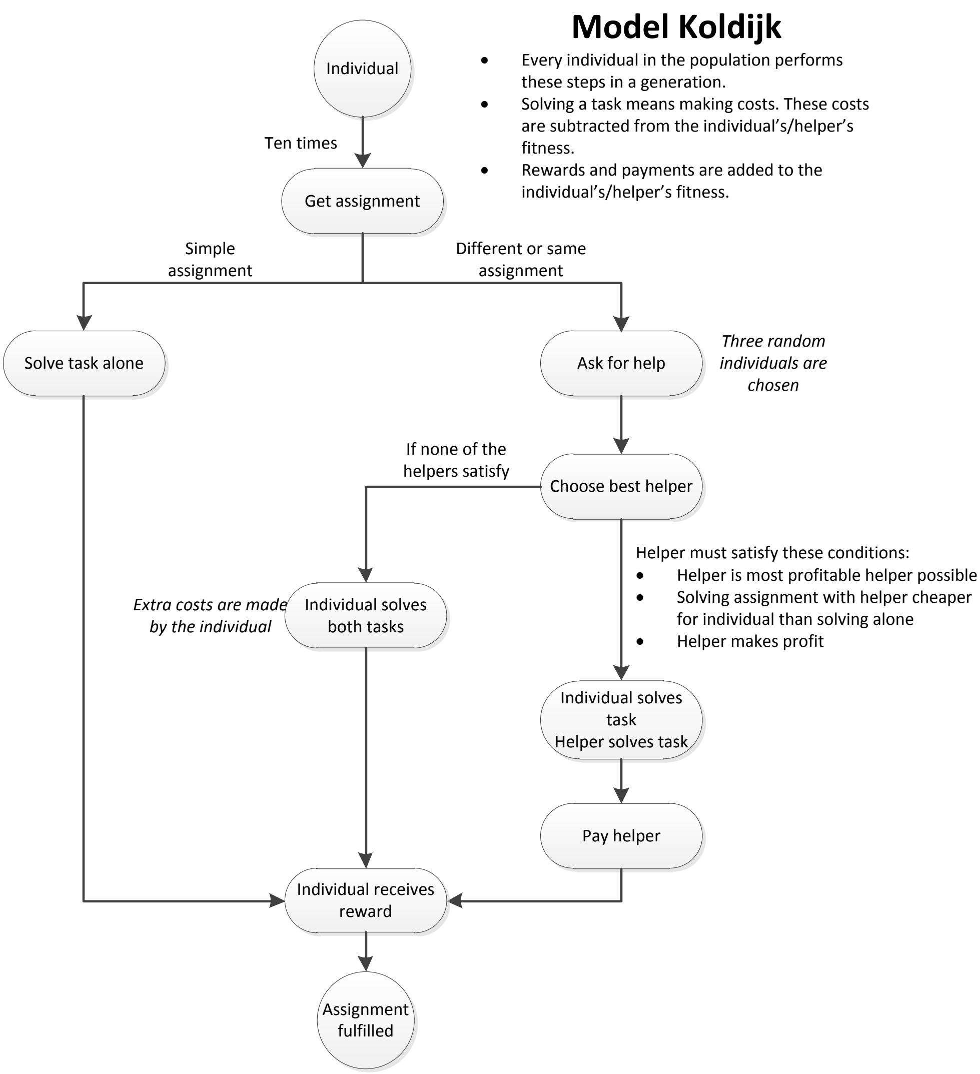


Figure 2.2: *Visual representation of the model from Koldijk [10].*

2.7 Implementation

Koldijk made her program code available to me for this further research. Therefore the exact same settings could be replicated in this experiment.

The model was programmed in Java and used the ECJ 18 framework [11], a freeware Java research system containing many methods to easily setup an evolutionary environment.

2.8 Experimental set-up

To see how and when cooperation evolves two variables are manipulated in the experiment:

- The proportion reward for the helper.

- The assignments: Simple assignments, different assignments or same assignments.

The proportion reward is manipulated since this is the so called cooperation mechanism and makes the cooperation possible. For each of the assignments a closer look is taken at specific pr values. The chosen pr values are: 0.15, .25 and 0.35[2]. The second variable is used to give a closer look at the different ways of cooperation that may occur.

2.9 Error in original model

The model made by Koldijk is a well designed model. As said, it unfortunately contained a programming mistake which was found by the author of this thesis. The error was found in the fitness function of the individuals. After each time cooperation took place, the individual was also given the reward of the helper. Meaning that the helper had worked for free. See appendix A for a more precise description of the error.

The results that are described in the next Chapter have this error fixed. These new results were significantly different compared to the original ones. The precise differences are shown in Section 3.3.

[2]Koldijk originally used the values 0, 0.25 and 0.5 to take a closer look at the results. But the new obtained results showed more interesting results at these values

Chapter 3

Results original model

In this Chapter the most important results of the model from Koldijk will be showed. Both errors were fixed and different results then originally reported by Koldijk were obtained.

In Section 3.1 the results of the simple assignments are shown. In Section 3.2 the results for the difficult assignments will be shown, where there will be looked at the fitness and cooperation (Subsection 3.2.1), the cooperation vs. proportion reward distribution (Subsection 3.2.2) and the gene distribution (Subsection 3.2.3). Next the differences between the original and new results are shown (Section 3.3). This Section closes with an interpretation of the results (Section 3.4) for both the simple as the different assignments.

3.1 Simple assignments

In this Section the results of the simple assignments are shown. As a reminder: these assignments contain only one task, therefore cooperation is not applicable.

The average fitness process over 100 runs for simple assignments is shown in Figure 3.1. One can see that different pr values have no effect, this makes sense since there is no cooperation. The fitness score ascends somewhat in the first five generations but then stays stable at a score of approximately 60.

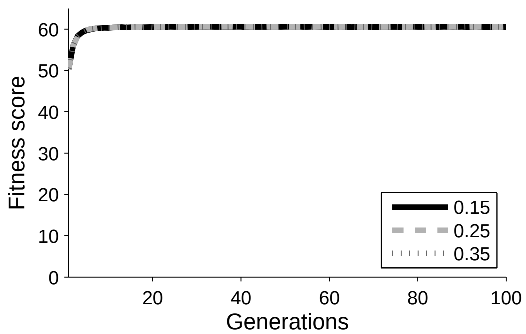


Figure 3.1: *Simple assignments: fitness score. Average over 100 runs*

To see what happens to the individuals a closer look at the genes is necessary. This is shown in Figure 3.2. What is striking is that all the individuals evolve their genes to around 0.33, they want to be equally good in every skill. A homogeneous population has evolved.

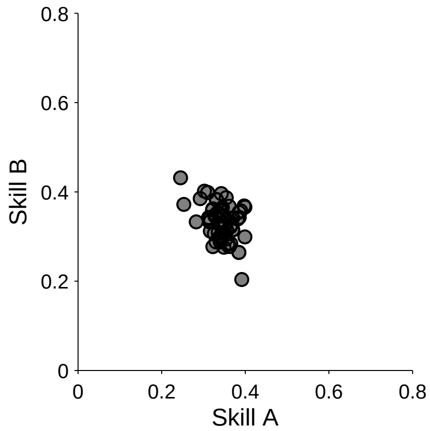


Figure 3.2: *Simple assignments: gene distribution last population*

3.2 Difficult assignments

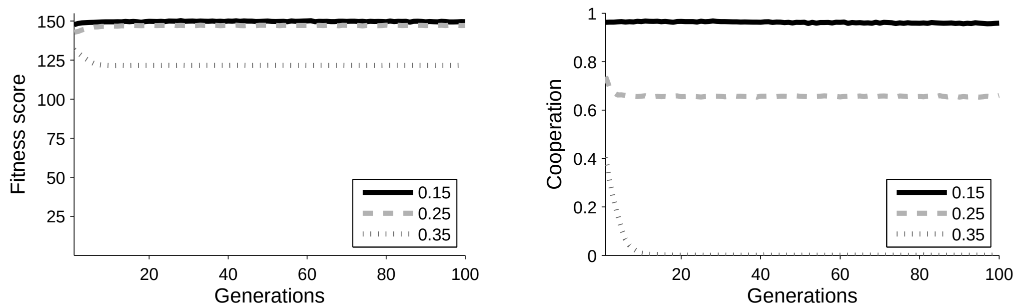
In this Section the result from the model when there are either different or same assignments given to the individuals are shown. Keep in mind that these assignments consists of two tasks, and therefore cooperation between the individuals is possible.

First, the fitness and cooperation results(3.2.1) are shown, followed by the distribution of the proportion reward(3.2.2) and concluding the distribution of the genes(3.2.3) is shown.

3.2.1 Fitness and cooperation

In the results of the simple assignments a fitness score of 60 was observed. Since these assignments have two tasks there is a twice as big reward and a suspected fitness score of 120. That is without cooperation, a higher fitness score could even be derived if cooperation would evolve.

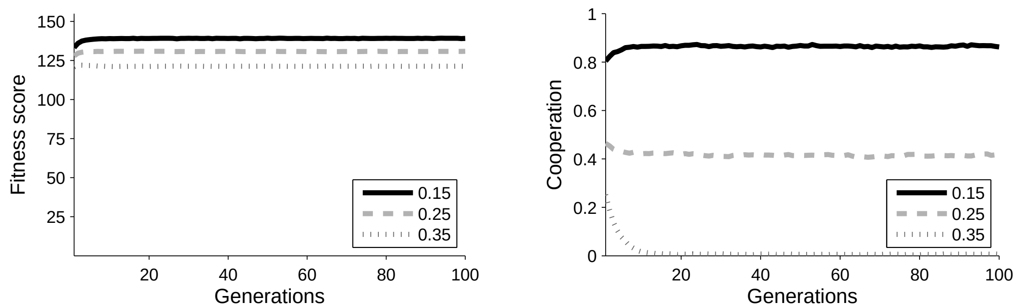
In Figure 3.3 the fitness score and the amount of cooperation are shown for the different assignments as an average over 100 runs. Both graphs show three different lines for the different proportion reward values used. One can see that with a proportion reward of 0.15 the cooperation is highest of the three plotted lines. What is interesting is that despite the random initialization the level of cooperation is from start already at its highest point. It seems that there is no evolution needed to learn cooperation here. For a proportion reward of 0.25 the average cooperation level is lower than for 0.15. Only in approximately 65% of the cases there is cooperation. For both the proportion reward of 0.25 and that of 0.35 the random population starts with a higher level cooperation then the evolution eventually converges to. In the case of the proportion reward of 0.35 one can clearly see that this disadvantages the fitness score.



(a) Fitness (b) Cooperation

Figure 3.3: *Different assignments: showing fitness(a) and cooperation(b) versus the generations. Average over 100 runs.*

In Figure 3.4 the fitness and amount of cooperation for the same assignments are shown. Again with the proportion reward of 0.15 the highest fitness and cooperation are obtained. However, a little lower, fitness now reaches up to about 140 and cooperation to about 85%. For the proportion reward levels of 0.25 and 0.35 the cooperation again drops at the start of the graph. This time the fitness score for the proportion reward of 0.35 is less influenced and stays a stable horizontal line. With a proportion reward of 0.25 the fitness score comes to a total 130 with a cooperation level of about 40%. For 0.35 the fitness comes to slightly above 120 and the evolution has terminated the cooperation completely.



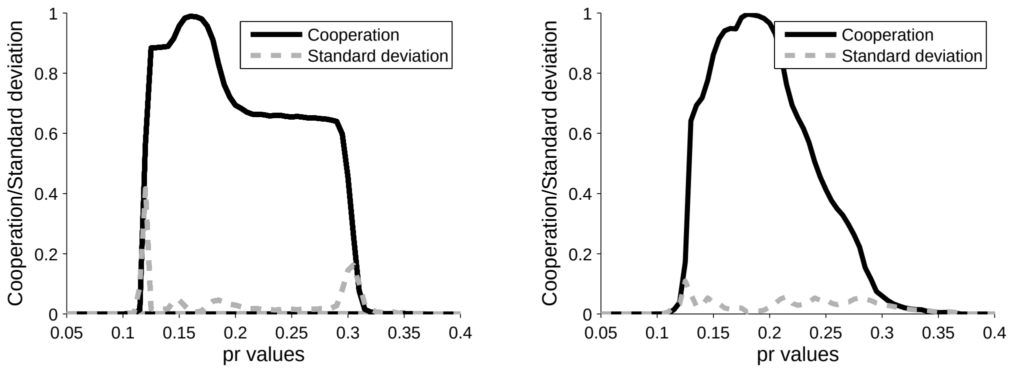
(a) Fitness (b) Cooperation

Figure 3.4: *Same assignments: showing fitness(a) and cooperation(b) versus the generations. Average over 100 runs.*

3.2.2 Proportion reward distribution

In Figure 3.5(a) one can see the distribution of cooperation versus different proportion rewards for the different assignments, and in Figure 3.5(b) for the same assignments. These measurements are from the last generation, averaged over 100 runs. The graphs also show the standard deviation over these 100 runs.

The first graph clearly shows that cooperation quickly initializes at a pr level of 0.12 and rises to almost its maximal point. From a pr of 0.31 the cooperation quickly drops to zero. What is especially interesting is that there seems to be a stable platform between the points 0.20 and 0.30. Between these points there is a continue cooperation going on in approximately 65% of the assignments.



(a) Different assignments

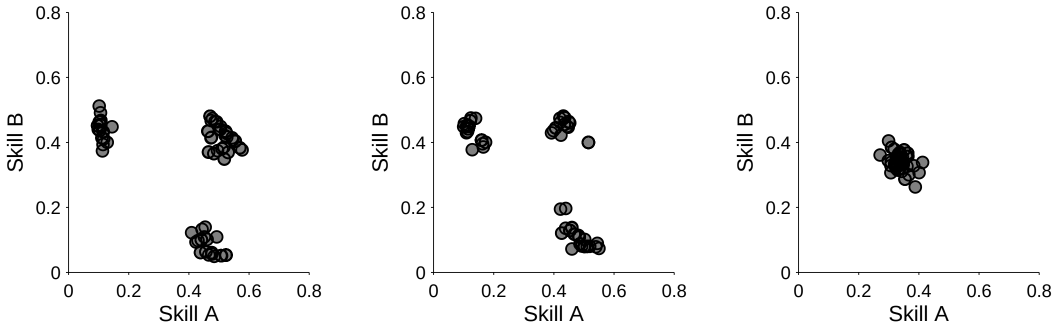
(b) Same assignments

Figure 3.5: *Amount of cooperation vs different proportion reward levels for the difficult assignments. Cooperation measured in generation 100, averaged over 100 runs.*

3.2.3 Gene distribution

To see what exactly happened with the individuals during the evolutionary process the distribution of the genes from the last generation are plotted. In Figure 3.6 the results from the different assignments are plotted and in Figure 3.7 the results for the same assignments. The figures each consists of three plots to show the results for the different pr values. These results are each from a single run.

The gene distribution for the different assignments show that for a proportion of 0.15 and of 0.25 three different groups have emerged. These groups find themselves around the points (0.1, 0.45), (0.45, 0.1) and (0.45, 0.45). This suggests that all the individuals have specialized in two different skills and forgot one skill. For a proportion reward of 0.35 the genes have again concentrated them self around the point (0.33, 0.33), which seems to be a result of the fact that there is no cooperation.



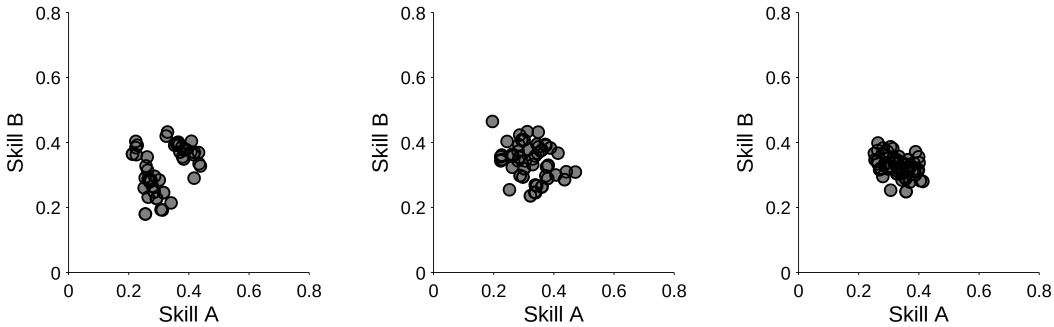
(a) pr = 0.15

(b) pr = 0.25

(c) pr = 0.35

Figure 3.6: *Gene distribution of the different assignments for different pr values. The genes are from the last generation.*

For the gene distribution of the same assignments with a proportion reward of 0.15 two different groups emerged. The individuals seem to be willing to specialize but are for some reason stopped before they can really release themselves from the middle point. With a proportion reward of 0.25 the groups have already disappeared, there is however a slightly wide spreading around the point of (0.33, 0.33). Again this seems to be an effort to specialize. For the proportion reward of 0.35 the groups have again all emerged around (0.33, 0.33), the result of the fact that no cooperation has evolved.



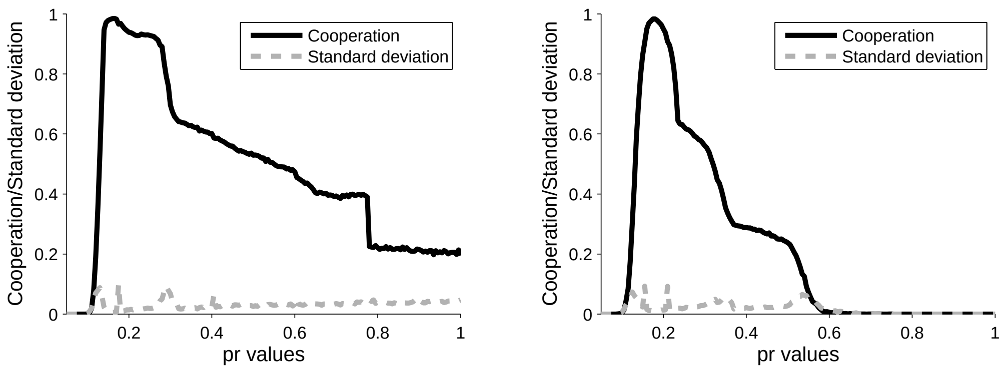
(a) pr = 0.15 (b) pr = 0.25 (c) pr = 0.35

Figure 3.7: *Gene distribution of the same assignments for different pr values. The genes are from the last generation.*

3.3 Differences

The newly obtained results differ at some points from the results originally reported by Koldijk. The main effects that can be found are however the same. There is stable level of cooperation established and the population shows specialization. The proportion reward distribution and the amount of specialization however differed.

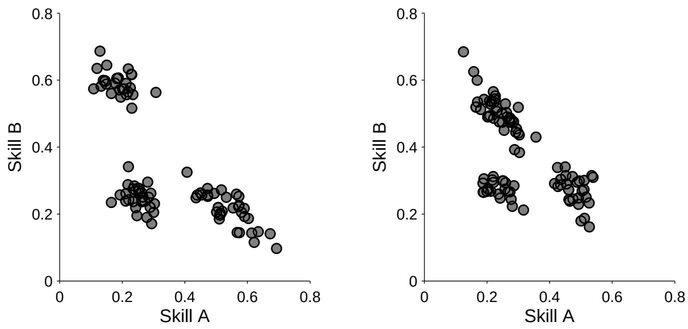
Figure 3.8 shows the amount of cooperation for different proportion reward values at generation 100 in the original model from Koldijk, averaged over 100 runs. The graph shows that there occurred cooperation for far higher proportion reward values than the new results show. There also occurred more cooperation at the different proportion levels. The different assignments even still show cooperation when the helper gets the full reward (pr = 1), something which seems not possible since an individual is only willing to cooperate when it is cheaper than working alone. However, the errors have resulted in only highly specialized individuals. When these individuals have to solve an assignment with a task in which they are not specialized they make very high costs, so high even that the fitness contribution score could be negative. As the results show, in approximately twenty percent of the time working together is actually less costly than working alone.



(a) Different assignments (b) Same assignments

Figure 3.8: *Pr distribution of the original model from Koldijk without the errors fixed. Cooperation measured in generation 100, averaged over 100 runs.*

Figure 3.9 shows the gene distribution for the different and same assignments at a pr level of 0.25 as originally reported by Koldijk. The distribution at the different assignments show an almost completely specialized population while the new results show far less specialization.



(a) Different assignments (b) Same assignments

Figure 3.9: *Gene distribution for difficult assignments with a proportion reward value of 0.25. From the original model from Koldijk without the errors fixed.*

Other differences were that with the original results more generations were required to come to a stable point in the fitness and cooperation. Also, the drawback of the lines that was visible in the new results is not visible in the original ones. Appendix A shows the fitness and cooperation process from the original results.

3.4 Interpretation of the results

3.4.1 Simple assignments

The simple assignments show a normal evolutionary situation when is looked at the fitness results. The individuals quickly learn that specialization is not the way to go and they evolved to a homogenous population. Each individual prepares herself for every possible task.

3.4.2 Difficult assignments

Fitness and cooperation:

First of all it is established that cooperation can evolve. However the cooperation mechanism, the proportion reward that was given to the helper, plays an important role in this. The proportion reward graphs showed this. When the proportion reward is too low, there will be no cooperation since the helper is not going to help if it cannot gain any fitness. When the proportion reward is too high, there will also be no cooperation since the employer will be better off solving the assignment alone since help is too expensive.

Second of all, it is possible to conclude that fitness and the cooperation do not have to be going hand in hand, however, cooperation does play a big role in the fitness score. This can easily be explained by the simple fact that solving an assignment together is more efficient. Less costs are made which directly results in a higher fitness.

Very remarkable was the drawback of the fitness and cooperation line that was observed for the different assignments with a proportion reward value of 0.35. This is the opposite of what is expected for an evolutionary algorithm. It can however be explained by the random initialization of the population. Calculations show that a balanced individual has a slightly better fitness gain then a specialized individual who cooperates. Therefore the specialized individuals will be extinced after a few generations, together with the cooperation. To understand why the total fitness of the population drops, one has to take a look at the cooperation procedure. When a specialized individual solves the task in which it is specialized it makes far less costs than a balanced individual. Costs made is a subtraction of the total fitness score which can never be returned. This means that the individuals choose for them self. Although specialization is better for the population they want to survive and choose to balance their skills.

Another remarkable thing was the platform that arised at the different assignments between a proportion reward of 0.20 and 0.30. This is a direct result of the choice of the individuals to specialize in two skills. The chance that an individual is getting an assignment with a task in which it did not specialize is 2/3 (the individual is not good in one of the three skills and the assignment consists of two different skills). Every time this happens it is cheaper to cooperate, however, when the individual will get an assignment with both skills in which it is specialized, it is cheaper off by working alone.

Gene distribution:

Between the different and same assignment there were some differences in the gene distribution. The individuals that performed the different assignments showed a slight specialization. Striking was that they always specialized in two tasks and never in one. This can be explained by the fact that when an individual specializes in one skill it has a high probability of getting an assignment with tasks in which it is not good, and solving a task with a low skill level is very expensive.

A second striking thing in the gene distribution of the different assignments are the different groups that arise. The individuals divide themselves around the three sorts of specialization. The same thing can be observed in the real world in the way of supply and demand.

The gene distribution for the same assignments showed very little specialization. This can be explained again by the fact that solving a task in which an individual is not good can be very expensive. And if an individual would specialize in one or two tasks, it still has a high chance of getting an assignment in which it is not good. Since the tasks will be the same, the cooperator cannot save the individual as was the case for the different assignments.

3.4.3 Other domains

Koldijk points also out how the effects relating this research after cooperation can affect other research domains, namely biology, society and robotics. Since this is beyond the scope of my research it is not discussed here, if you are however interested I would certainly recommend reading her thesis.

Chapter 4

Methods model with trust mechanism

To answer the research question, the previously used model from Koldijk needs to be extended with a trust mechanism. This Chapter starts with an explanation of the changes that are needed to the model (Section 4.1) followed by the actual implementation (Section 4.2). Section 4.3 gives a scheme of the new model and in Section 4.4 the differences between the new model and the prisoners dilemma is discussed. Concluding the new evolutionary settings (Section 4.5) and the experimental setup (Section 4.6) are explained.

4.1 Theory

In the model from Koldijk, without the trust mechanism, individuals could easily cooperate. There was no reason to distrust someone and there was no chance of betrayel. Cooperating in real life is not that easy. People always make an evaluation on the reliability of their possible cooperator, and if that cooperator is not found to be reliable enough, cooperation is not happening.

A simple way of how a trust mechanism can be added to the model is by giving each individual the choice to see if they can trust their helper. Of course, a helper should then also be able to betray the individual. This also works vice versa. The helper should decide if it can trust the individual, and the individual can betray the helper by not giving the helper its part of the reward.

The decision if an individual can be trusted should be based on relevant information. In daily life decisions to trust someone are made on numerous different things like the appearance of the possible helper, previous experiences, experiences from friends etc. These different things try to put an accurate number on the reliability of the other individual. In daily life we don't get betrayed, we are quite good at predicting someones reliability [7]. When we are betrayed we often knew that there was a chance. Therefore, we could use a abstraction in our model which says that we always know the exact reliability of an individual. This could be considered a fairly large abstraction, and one could argue if this would represent trust in its fullest. However, this research is not about how trust evolves, or when betrayal occurs. It is about if trust is even able to evolve in a population and, if so, how that affects the population. Therefore I believe this to be a valid abstraction.

When individuals cooperate, and the trust mechanism is added to the model, there are four situations that can occur between an employer and a helper:

- Both cooperate: Both decide to do the nice thing and cooperate. (CC)
- The employer betrays. (DC)
- The helper betrays. (CD)
- Both betray. (DD)

The first situation, where both individuals cooperate, is the situation that continuously occured in the model without the trust mechanism. Both individuals solve their part of the assignment and both get a part of the reward. In the second case the helper gets betrayed. This should impact the fitness of the helper negatively, and that of the employer positively. So this will be modeled as: the employer lets the helper perform the task but keeps the entire reward for itself. In the third case, where the helper betrays the employer, the helper walks away with its part of the reward but does not solve its task. Leaving the employer to also solve the second task.

In the last situation, where both individuals want to betray the other, two things could happen. One could imagine how the individual continues the assignment without the helper, as it would when it had chosen to work alone. Secondly, the assignment giver could lose confidence in such a situation and could revoke the assignment. In that case neither of the individuals would gain anything. Both situations are researched, they represent interesting situations which is explained in Section 4.4.

4.2 Implementation

The actual implementation in the model is done by adding four genes to every individual and stating some extra restrictions. In this Section these new genes and restrictions are explained. For arguments sake, the individual that has to solve the assignment is here notated as the employer.

4.2.1 Genes and restrictions

From the four new genes, two are to represent how reliable the individual is, and two are to represent how reliable the individual wants that their employer/helper is. These genes are all numbers between 0 and 1.

- Chance Payout (CP): The chance that this individual as an employer will pay its helper. (1-CP) is the chance that this individual will betray its helper.
- Chance Work (CW): The chance that this individual as a helper will work. (1-CW) is the chance that this individual will betray its employer by not working.
- Trustworthiness Payout(TP): This individual is only willing to work as a helper when the employer is at least this reliable.
- Trustworthiness Work(TW): This individual is, as an employer, only willing to work together with a helper when the helper is at least this reliable.

Together with the last two genes there are two resrictions:

- An employer will only work with a helper if it can trust the helper enough: $CW_{helper} >= TW_{emp}$.
- A helper will only help if it can trust the employer enough: $CPemp >= TP_{helper}$.

An individual now consists of seven genes. The three skillgenes, which represent how well the individual can solve a specific task, and the four trustgenes. From the trustgenes the first two genes ((CP) and (CW)) represent the reliability of the individual. The last two genes ((TP) and (TW)) represent how reliable the individual wants that its cooperator is.

The following restrictions are added to the original model:

- The chance that a helper or employer is betrayed is randomly determined. There is chance of $1 - Emp_{CP}$ that an employer will betray its helper. There is a chance of $1 - Hlp_{CW}$ that a helper will betray its employer.
- Betrayed still means paying. If an employer was willing to cooperate but it is betrayed by its helper than it still has to pay the helper. Because the assignment has to be solved the employer also needs to solve the task that was designated to the helper.

- Betrayed still means working. If a helper is betrayed it has to solve the task but gets no reward.
- If multiple possible helpers give the same profit, the one that is most reliable is chosen.

For the different kind of situations that can occur when both individuals defect there are different restrictions:

- Non-revoked assignments: the assignment still has to be solved. The employer solves the assignment alone, the same way as it would normally do. The helper gets no reward but makes also no costs.
- Revoked assignments: When the assignment giver revokes the assignment both the employer and the helper throw away their chance on a reward. They will get no reward but also make no costs.

4.3 Overview

In Figure 4.1 the new model is visualized in a scheme. It shows the steps taken by every individual every generation. The scheme clearly shows the differences with the original model.

4.4 Comparison between models

As mentioned in the introduction, the prisoners dilemma is an often used model to investigate trust. In this Section the differences between the prisoners dilemma, and the trust model are discussed.

The prisoners dilemma actually deals in penalties, because we want to compare the reward from the models we need to rewrite the penalty table to a reward table. Table 4.1 is an often used reward table for the prisoners dilemma. The table shows how defecting someone is very profitable, unless both prisoners defect, then almost no reward is obtained. When both prisoners cooperate the highest total reward is obtained. Striking at this dilemma is that for a prisoner defecting is always the best choice. Imagine that prisoner 1 has to make his decision. He thinks that when Prisoner 2 Cooperates I can better defect, a reward of five instead of three. And when prisoner 2 defects I can also better defect, a reward of one instead of zero. Hence, a prisoner who is unable to look at the bigger picture would always decide to defect. This is the clue of the prisoners dilemma in a nutshell.

		Prisoner 2 C	Prisoner 2 D
Prisoner 1	C	3 / 3	0 / 5
	D	5 / 0	1 / 1

Table 4.1: *The reward table of the prisoners dilemma.*

For the trust model one can also make a reward table, however, to create this it is it is necessary to define an example. Imagine the assignment {A, B} was assigned to an individual who has already chosen its helper. For a simple example imagine both the individual and the helper have a uniform distribution of skills: .33, .33, .33 and that the proportion reward is set at 0.2. In that case, the reward table for the non-revoked assignments would look like Table 4.2a. If the proportion reward would be equal to 0.3 then the table would look like Table 4.2b. Both are examples of a situation, and the values differ, however, the relative difference between the values remains. Note that when these table would have been plotted for the original model only the left corner of the tables would be filled.

(a) proportion reward = 0.2

		Helper C	Helper D
Individual	C	13 / 1	8.2 / 4
	D	17 / -3	12.2 / 0

(b) proportion reward = 0.3

		Helper C	Helper D
Individual	C	11 / 3	6.2 / 6
	D	17 / -3	12.2 / 0

Table 4.2: *The reward Table for different pr values of the trust model. The assignment {A, B} is given to two uniform individuals. The assignment will not be revoked.*

Both tables are similar to the prisoner dilemma's reward table. Defecting gives always a higher result for the individual, while when both cooperate the best global result is obtained. The Table shows that this situation could be described as a prisoner's dilemma [1].

The reward tables for the assignments who are revoked when both individuals defect is shown in Table 4.3. The Table shows that this time defecting for the individuals is not always the best solution. However, it could be still be a profitable experience.

(a) proportion reward = 0.2

		Helper C	Helper D
Individual	C	13 / 1	8.2 / 4
	D	17 / -3	0 / 0

(b) proportion reward = 0.3

		Helper C	Helper D
Individual	C	11 / 3	6.2 / 6
	D	17 / -3	0 / 0

Table 4.3: *The reward Table for different pr values of the trust model. The assignment {A, B} is given to two uniform individuals. The assignment will be revoked when both individuals defect.*

4.5 Evolutionary settings

Since the model has been changed, the evolutionary algorithm has to be examined to see if it is still well tuned. A major change has been the adding of genes, individuals now consists of seven genes instead of three. Therefore cross-over was chosen as an extra evolutionary method. Cross-over combines the genes of two individuals, this should make sure that learned skills are not so easily discarded. Using only mutation was sufficient when there were only three genes, however, with the adding of more genes the number of possibilities has exponentially grown. Using only mutation might be to random. The strength of cross-over is that it can combine the genes of two good individuals and making an even better combination[1].

The new evolutionary settings are:

[1]Of course, a worse combination is also possible, but that individual would soon enough be removed from the population by the selection mechanism.

- Cross-over type: uniform
- Genes to cross-over: 3
- Cross-over probability: 0.5
- Mutation probability per gene: 0.1
- Number of generations: 2000

The cross-over used is uniform cross-over. This means that there are no chunks of genes combined but random gene numbers are chosen and used to switch the genes between the individuals. The cross-over and mutation chance were chosen at these value because this gave the most stable results without destroying the variation in the population. A too high probability of cross-over might result in all individuals becoming the same. Destroying the variation in the population would in its place destroy the possibility of finding the best solution [8]. The number of generations was raised to 2000. Pilot experiments showed that with these settings the results were best visible.

The so-called 'evolutionary pipeline' will in the new model be as following:

1. Approximately 25 times two individuals are picked out of the old population by the selection mechanism. These individuals go through the cross-over procedure and these newly combined individuals are selected for the new population.
2. The selection mechanism selects the approximately 50 other individuals for the new population.
3. Exactly 100 individuals have been selected. For every individual, for every gene, with a chance of 0.1 mutation is performed on this gene.
4. The normalization procedure assures that the skill-genes sum up to exactly 1.

4.6 Experimental setup

In the experiment the parameters were manipulated as in the model without the cooperation mechanism: the type of assignments and the proportion reward. Furthermore there was also looked at the different results between the revoked and non-revoked assignments.

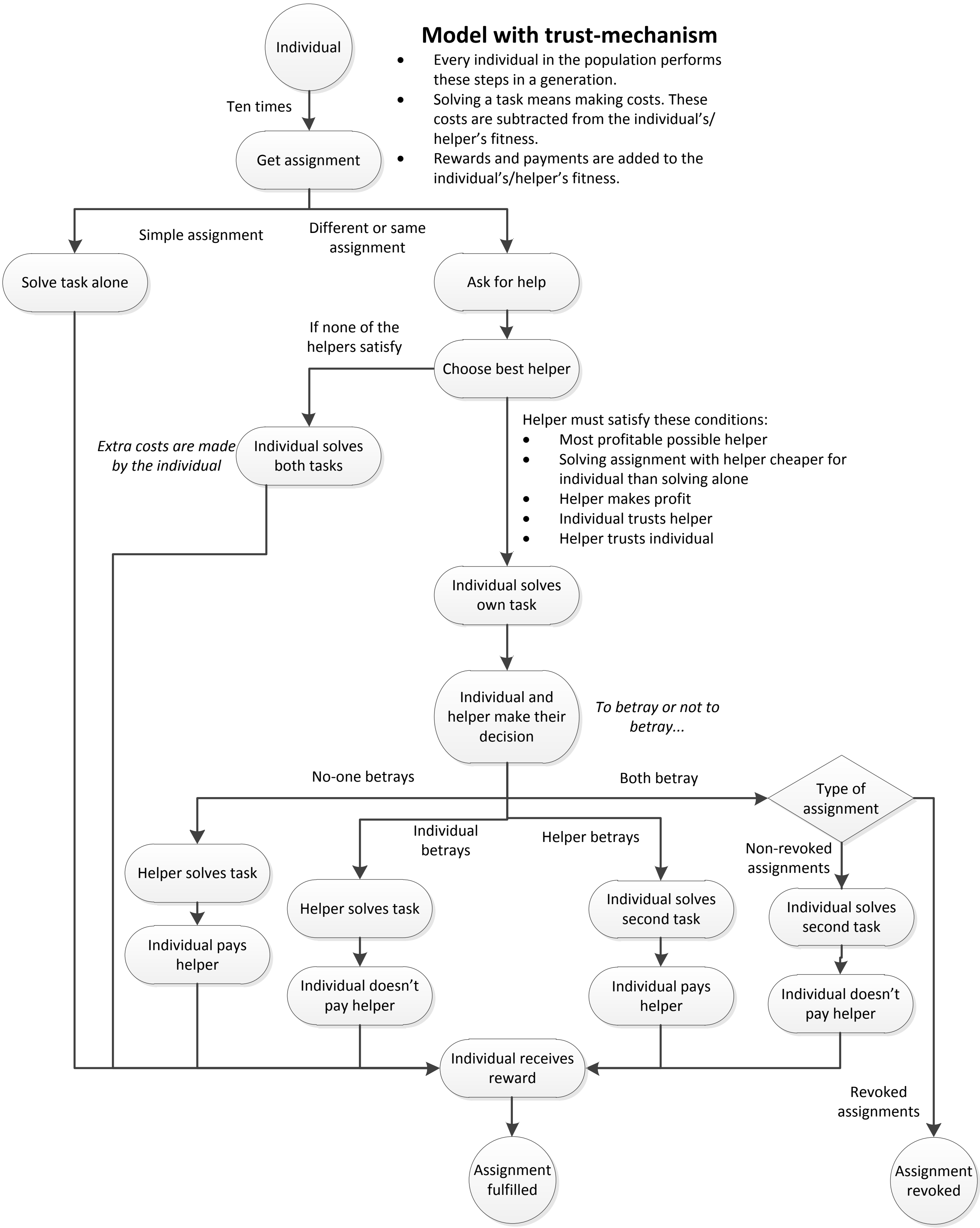


Figure 4.1: *Scheme of the trust model.*

Chapter 5

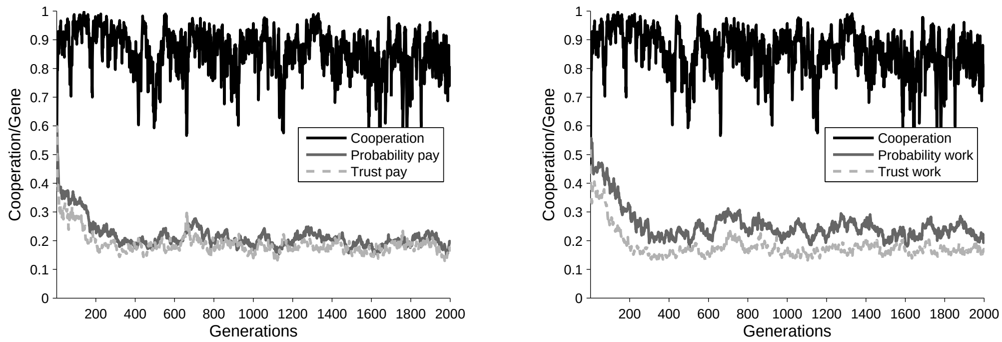
Results model with the trust-mechanism

In this Chapter the results of the extended model are shown. This time, the simple assignments are not discussed. For a simple assignment there can be no cooperation and therefore trust has no influence. The results would be the same as in the original model.

This Chapter starts with the results from the non-revoked assignments (Section 5.1), followed by the results with the revoked assignments (Section 5.2). The Chapter closes with an interpretation of the results (Section 5.3).

5.1 Non-revoked assignments

Especially interesting in this new model are of course the trust genes, the newly added genes. The values of these genes determine the amount of trust and therefore the amount of cooperation that will evolve. The average trust genes for a single run are plotted for the pay-genes in Figure 5.1(a) and for the work-genes in Figure 5.1(b), both graphs show the amount of cooperation in the population for clarity reasons. The run is with different assignments and a proportion reward of 0.2. The plots show that cooperation again evolves, however, the cooperation is jumping up and down between approximately the levels 0.5 and 1. All the trust genes have after 300 generations evolved to approximately 0.2, which suggest that there is very little trust. That cooperation still exists is because all the individuals evolve their trust-work and trust-pay levels even lower, they do not seem to care that they are being betrayed.



(a) Pay genes

(b) Work genes

Figure 5.1: *The amount of cooperation plotted with the different trust genes averaged from the individuals. Single run for a different non-revoked assignment with a proportion reward of 0.2. The cooperation is plotted twice for clarity.*

In Figure 5.2 the total experiences of all the individuals are plotted per generation. The graph belongs to the same run as the above example, and one can see that it also shows corresponding data. Most of the time the individuals betray each other, only in a very low amount of the time they both work together. Working alone shows high fluctuations, sometimes very popular, sometimes not.

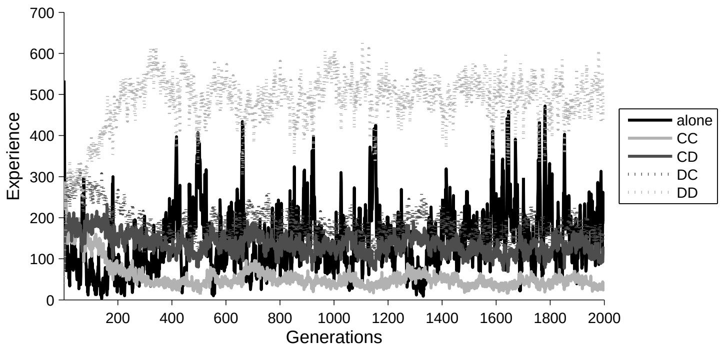
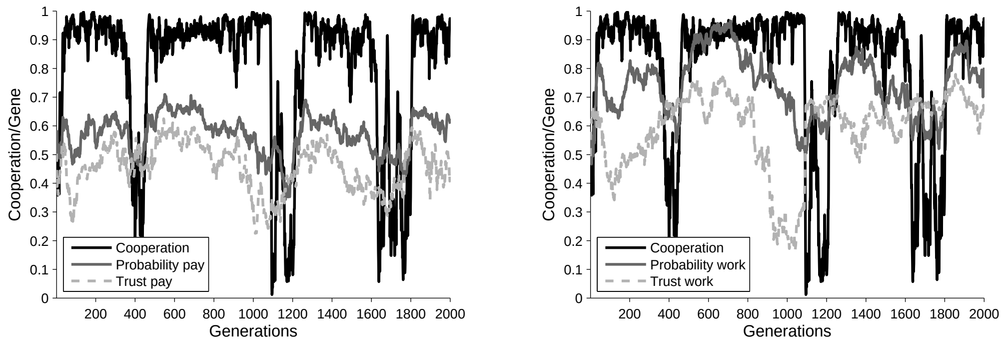


Figure 5.2: *The experiences from the individuals vs. the generations. Results with non-revoked assignments. The different lines plotted mean; alone: individual works alone, CC: both cooperate, CD: helper defects, DC: individual defects, DD: both defect.*

The results that were obtained with different proportion reward values and for the same assignments differ very little from the above presented results. It was only the amount of cooperation that varied, at the same way however that it did as without cooperation. The trust genes behaved the same way as in the above results. See appendix B for some of these graphs.

5.2 Revoked assignments

When the model was adjusted for revoked assignments significantly different results were obtained. Figure 5.3 shows the results from the trust-genes together with the amount of cooperation. One can clearly see different states in the population, states with much cooperation and states where there is no trust in the population and almost no cooperation occurs. One can see that the times that there is no cooperation goes together with the times that one of the trust genes goes above its corresponding probability gene.



(a) Pay genes

(b) Work genes

Figure 5.3: *The amount of cooperation plotted with the different trust genes averaged from the individuals. Single run for a different revoked assignment with a proportion reward of 0.2. The cooperation is plotted twice for clarity.*

In Figure 5.4 the experiences of the individuals have been plotted for the same run as the above example. The graph clearly shows the different states in the population. Three high peaks show the times when almost no cooperation occurred and the individual had to work alone. In the other moments the state where both individuals cooperate has the upper hand.

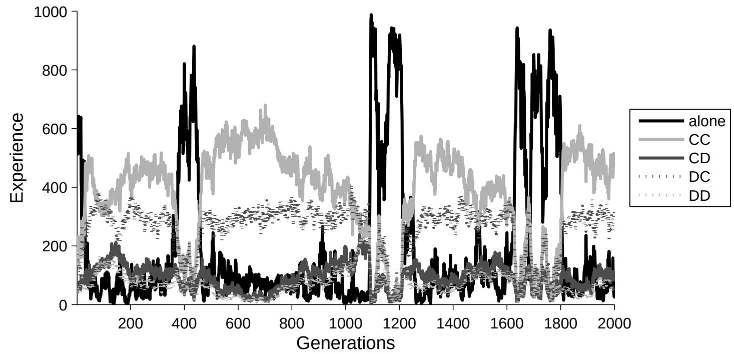
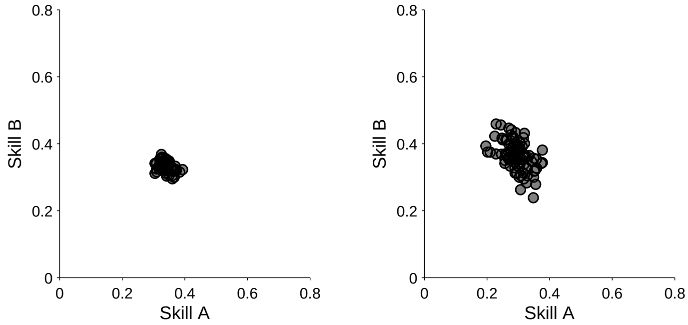


Figure 5.4: *The experiences from the individuals vs. the generations. Results with revoked assignments. The different lines plotted mean; alone: individual works alone, CC: both cooperate, CD: helper defects, DC: individual defects, DD: both defect.*

Figure 5.5(a) shows the skill-gene distribution of the population in generation 1200, at that time the population is at a low trust level for some generations. The distribution show that an homogeneous population has evolved where no individual is specialized. In Figure 5.5(b) the skill-gene distribution has been plotted of the population in generation 800, a generation preceded by a relative stable amount of cooperation. Where the same settings in the original result would lead to three different groups of specialized individuals this time there seems to be almost no specialization.



(a) Plot of the skill-genes distribution in generation 1200.

(b) Plot of the skill-gene distribution in generation 800.

Figure 5.5: *Plot of skill-gene distribution at two different generations in the trust model with revoked-assignments*

Again, almost the same results were obtained for different parameter settings. See appendix B for more graphs with different parameter settings.

5.3 Interpretation of the results

The results that were obtained in the adjusted model showed interesting results. First, the non-revoked assignments showed that when defecting can be near as profitable as cooperating the individuals quickly chose to defect. This corresponds with the difficult situation of the prisoners dilemma. Evolution seems not to be able overcome this. Striking is that even though both the individual as the helper have a high chance of defecting they still cooperate. The individuals probably see a chance of betraying the other and gaining more than when one would work alone. This can be explained by the fact that if this works, only for one time in a generation, a higher fitness score is obtained which increases the chance of survival.

The results of the revoked assignments showed far more trust than for the non-revoked assignments. Most noticeable are the different states that evolved. These states can be explained by individuals that at a certain point take advantage of the gullible evolved individuals. The population than flips to a population where almost no-one is anymore trusted. These states can also be observed in real populations where different states of calm and peacefulness are changed for state of savage and warlike states. It also shows how quickly an environment can flip from a trusted population to a population that trusts nobody.

The adding of trust has also been an adding of an unstable environment to the individuals. Cooperation showed, even in the peaceful states, more variations than in the original model. These variations have limited the amount of specialization that occurred with the individuals. In a unstable environment specialization can be a risky business, if the population goes in to another direction or if the population flips extinction is very near.

This unstable environment was not expected, but it is not new. The unstable states that reflect a ‘peaceful’ or ‘warlike’ state have been earlier observed [18] [13].

Chapter 6

Further research

The model from Koldijk has proven to be a model worth researching. Many simplifications, or different aspects could be further researched. In my opinion it is a great model for further interesting extensions and experiments. This Section provides some things that are certainly worth further research, first named for the original model and then for the trust model.

6.1 Koldijk's model

Some interesting things worth researching for the original model are:

- Currently all the individuals are equally good, they all have the same total amount of skills. In real life people are not equally talented.
- Cooperation does in real life not take place with random people. It happens with people that live near you, or with people that you know in some way.

Furthermore this model could also be very well used to gain more insight in to the necessary aspects of specialization. The model currently contains some simplifications that may restrict the evolution of specialization:

- Cost of help is a global constant instead of the cost being calculated per individual so that a more concurrent mechanism would evolve.
- When helpers are equally expensive the first one is automatically picked instead of the one that can solve the task as efficient as possible.
- Assignments are obligatory. Since an individual is required to solve every assignment it is possible that it has to solve a task in which it totally lacks any skill and would have very high expenses. This might be a reason why individuals would not specialize.

If specialization would be researched the effects of an unstable environment might also be an interesting aspect. In nature there is also an unstable environment. The trust model also showed that an unstable environment does effect the specialization.

6.2 Trust model

For the trust model there can also be done some interesting further research. The things named for Koldijk's model could of course also be applied in combination with this model. Other things worth researching might be:

- Currently the way individuals decide how well they can trust another individual is transparent. In real life it is of course not this easy. One might research how other aspects are important for the evolution of trust.

- The possibility for individuals to deceive other individuals. Pretending that they are better than they are actually are.

Chapter 7

Conclusion

This paper addressed the question how trust would affect the cooperation in a population during evolution. Results were obtained for a model without a trust mechanism and one with a trust mechanism.

Conclusion 1

How does cooperation evolve without a trust mechanism?

The results from the model without a trust mechanism clearly showed that cooperation can evolve and are in line with Koldijk's conclusion: "... the availability of (innate) cooperation mechanisms in a give population can indeed, in some situations, lead to the evolution of a population of cooperating individuals". However, the effect she reported, how a lot of cooperation between individuals can lead to specialization, is not this strong. Results showed that cooperation can indeed lead to specialization but only for a small amount. Other aspects like supply and demand seem to play a far bigger role.

Conclusion 2

How does cooperation evolve with a trust mechanism?

The trust model consisted of two situations. First of all the prisoner dilemma like assignments which showed that individuals are too selfish and a total rebellish population arises where everyone defects. This shows that trust cannot be that easily evolved and that in an environment where defecting is rewarded too much, a rebellish population will evolve. The second situation where defecting was less profitable showed that individuals are willing to trust each other to a certain amount. Some individuals will be selfish and stand in the way of a total trustworthy society.

Also, the second situation showed different states of high and low cooperation, which was also found in similar research from Bravo and Tamburino [5]. These results could represent 'peace' and 'war' times in a society.

Conclusion 3

How does trust affect the evolution of cooperation?

The results showed that a different population evolved with the trust mechanism than without trust the trust mechanism. Approximately the same amount of cooperation was obtained, however, the individuals had a far less diverse population. Furthermore, the trust did affect the stability of the cooperation. In the first place just locally, where some individuals are willing to cooperate with each other and others are not, resulting in small variations in the average cooperation. In

the second place it showed that trust can explain different global states where almost every one trusts each other or no-one trusts anyone.
There is one last remark to be made. Computer simulations are a great way to gain insight into large phenomenas in nature and society. However, as shown here, a small program error can give plausible looking, but wrong, results. We need to be careful and always validate every aspect of our data...

Chapter 8

Acknowledgements

I want to especially thank Saskia Koldijk. Her model and her previous research have been a guide through my own research. Also I thank her for sharing her model with me, without that, this research would not have been possible.

Second of all I would like to thank my supervisor Franc Grootjen. I am grateful for all his assistance and guidance during the development of my thesis.

Finally, I would also like to thank Ida Sprinkhuizen-Kuyper for the thoroughly reading of my thesis and her insightful help and tips.

Bibliography

[1] R. Axelrod. Effective choice in the prisoner's dilemma. *Journal of Conflict Resolution*, 24(1):3, 1980.

[2] R. Axelrod. The emergence of cooperation among egoists. *The American Political Science Review*, 75(2):306–318, 1981.

[3] R. Axelrod and W.D. Hamilton. The evolution of cooperation. *Science*, 211(4489):1390, 1981.

[4] T. Balch. Behavioral diversity as multiagent cooperation. In *Proc. of SPIE*, volume 99. Citeseer, 1999.

[5] G. Bravo and L. Tamburino. The evolution of trust in non-simultaneous exchange situations. *Rationality and Society*, 20(1):85, 2008.

[6] K.S. Cook, R. Hardin, and M. Levi. *Cooperation without trust?* Not Avail, 2007.

[7] L. Cosmides. The logic of social exchange: Has natural selection shaped how humans reason? Studies with the Wason selection task* 1. *Cognition*, 31(3):187–276, 1989.

[8] A.E. Eiben and J.E. Smith. *Introduction to evolutionary computing.* Springer Verlag, 2003.

[9] F. Fukuyama. Social capital and the global economy: A redrawn map of the world. *Foreign Affairs*, 75(4), 1995.

[10] S. Koldijk. Helping one another: How cooperation and diversity evolve. Website, July 2009. `http://www.ru.nl/publish/pages/529310/koldijks.bathesis09.pdf`.

[11] Sean Luke, Liviu Panait, Gabriel Balan, Sean Paus, Zbigniew Skolicki, Elena Popovici, Keith Sullivan, Joseph Harrison, Jeff Bassett, Robert Hubley, Alexander Chircop, Jack Compton, William Haddon, Stephen Donnelly, Beenish Jamil, and James O'Beirne. A Java-based Evolutionary Computation Research System. Website, 2010. `http://cs.gmu.edu/~eclab/projects/ecj/`.

[12] M.W. Macy and J. Skvoretz. The evolution of trust and cooperation between strangers: A computational model. *American Sociological Review*, 63(5):638–660, 1998.

[13] Y. Nakai and M. Muto. Emergence and collapse of peace with friend selection strategies. *Journal of Artificial Societies and Social Simulation*, 11(3):6, 2008.

[14] M.A. Nowak. Five rules for the evolution of cooperation. *Science*, 314(5805):1560, 2006.

[15] R.D. Putnam. Tuning in, tuning out: the strange disappearance of social capital in America. *PS: Political Science and Politics*, 28(4):664–683, 1995.

[16] M. Quinn, L. Smith, G. Mayley, and P. Husbands. Evolving teamwork and role allocation with real robots. In *Proceedings of the 8th International Conference on Artificial Life*, volume 302. Citeseer, 2002.

[17] J.L. Sachs, U.G. Mueller, T.P. Wilcox, and J.J. Bull. The evolution of cooperation. *The Quarterly Review of Biology*, 79(2), 2004.

[18] E. van Dijk. Grouping by encounter probability in an FSS society. Website, August 2009. `http://www.nici.ru.nl/~idak/teaching/batheses/BaScriptie_ETvanDijk.pdf`.

Appendix A

Error

The fitness of the helper was wrongfully set in the original programming code from Koldijk. The following code shows the error made. The function *evaluate* takes, among other variables, two individuals as a parameter, the individual (*ind*) and the helper (*helper*). Koldijk uses the letter A in her variable names to state that these variables are for the individual, and the letter B for the helper.

In the last two lines of the function the error is made. First of all the new total fitness score of the helper is calculated with the fitness score of the individual (*ind.fitness.fitness()*) instead of the score of the helper. Secondly, in the functions last line, the calculated new fitness score for the helper is awarded to the individual instead of to the helper.

The combinations of these two lines resulted in no additional fitness contribution for the helper (it worked for free) and the individual obtained the entire reward with free help from the helper. This resulted in more cooperation and more specialisation of the individuals.

```
public void evaluate(EvolutionState state, Individual ind, int pop,
  int threadnum, Task task, Individual helper)
{
  if( !( ind instanceof DoubleVectorIndividual ) )
    state.output.fatal(
  "The individuals for this problem should be DoubleVectorIndividuals." );

  //Reward berekenen
  double reward = task.difficulty() * 10;
  double rewardA = (1 - task.PROP_REWARD) * reward;
  double rewardB = task.PROP_REWARD * reward;

  //Taak verdelen
  Task[] devidedTask = task.devideTask((DoubleVectorIndividual)ind,
    (DoubleVectorIndividual)helper);

  //Kosten berekenen
  double[] skillA = ((DoubleVectorIndividual)ind).genome;
  int[] subtasksA = devidedTask[0].t;
  double[] skillB = ((DoubleVectorIndividual)helper).genome;
  int[] subtasksB = devidedTask[1].t;
  double costA = 0;
  double costB = 0;

  for(int i = 0; i < subtasksA.length; i++)
  {
```

```
    costA += subtasksA[i] / skillA[i];
    costB += subtasksB[i] / skillB[i];
  }

  //Fitness berekenen
  double gainedFitnessA = rewardA - costA;
  double totalFitnessA = ind.fitness.fitness() + gainedFitnessA;
  ((SimpleFitness)(ind.fitness)).setFitness( state,
    (float)totalFitnessA, false );

  double gainedFitnessB = rewardB - costB;
  double totalFitnessB = ind.fitness.fitness() + gainedFitnessB;
  ((SimpleFitness)(ind.fitness)).setFitness( state,
    (float)totalFitnessB, false );
}
```

Appendix B

Original results Koldijk

The following graphs show the results that were generated from the model of Koldijk, where the error remained.

In Figure B.1 the results of the simple assignment are shown. One can see that there is no difference compared to the previously shown results. This is because both programming errors were in the cooperation part of the code.

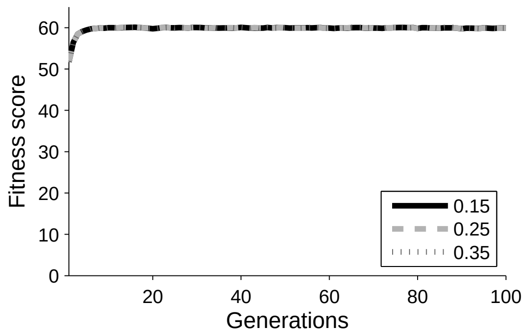
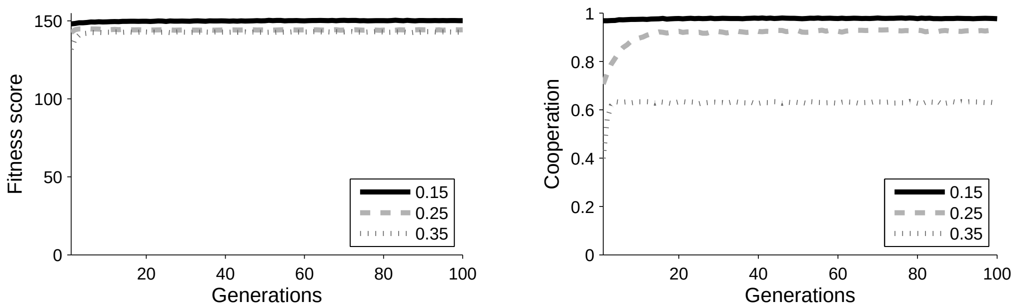


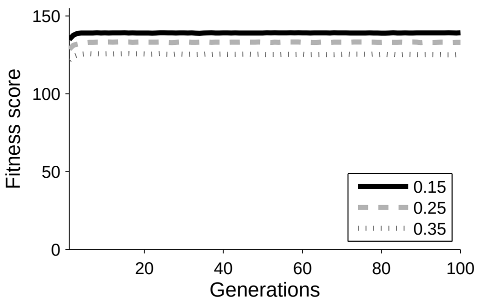
Figure B.1: *Simple assignments: fitness score. Average over 100 runs.*

In Figure B.2 and B.3 the fitness and cooperation results for both the different as the same assignments are shown. The results clearly show that the errors stimulate cooperation.

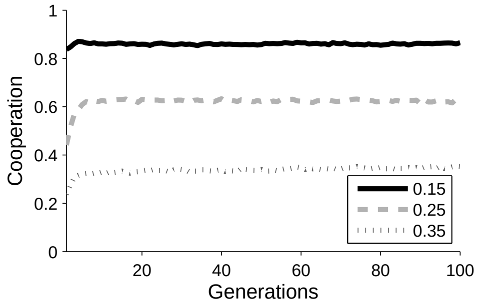


(a) Fitness (b) Cooperation

Figure B.2: *Different assignments: showing fitness(a) and cooperation(b) versus the generations. Average over 100 runs.*



(a) Fitness



(b) Cooperation

Figure B.3: *Same assignments: showing fitness(a) and cooperation(b) versus the generations. Average over 100 runs.*

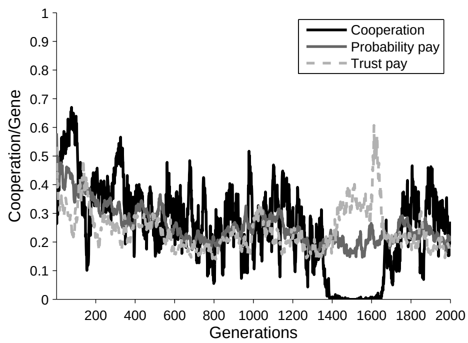
Appendix C

Results model with the trust mechanism

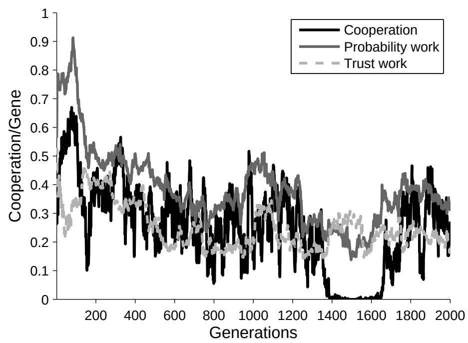
This chapter shows more results from the model that was appended with the trust mechanism. Section C.2 shows this for the non-revoked assignments, and Section C.2 for the revoked assignments.

C.1 Non-revoked assignments

C.1.1 Different assignment with pr=0.3



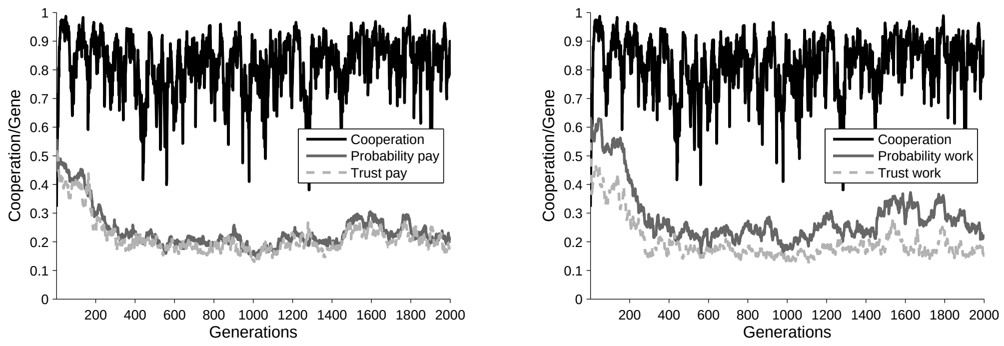
(a) Pay genes



(b) Work genes

Figure C.1: *The amount of cooperation plotted with the different trust genes averaged from the individuals. Single run for a different revoked assignment with a proportion reward of 0.3. The cooperation is plotted twice for clarity.*

C.1.2 Same assignment with pr=0.2



(a) Pay genes (b) Work genes

Figure C.2: *The amount of cooperation plotted with the different trust genes averaged from the individuals. Single run for a different revoked assignment with a proportion reward of 0.3. The cooperation is plotted twice for clarity.*

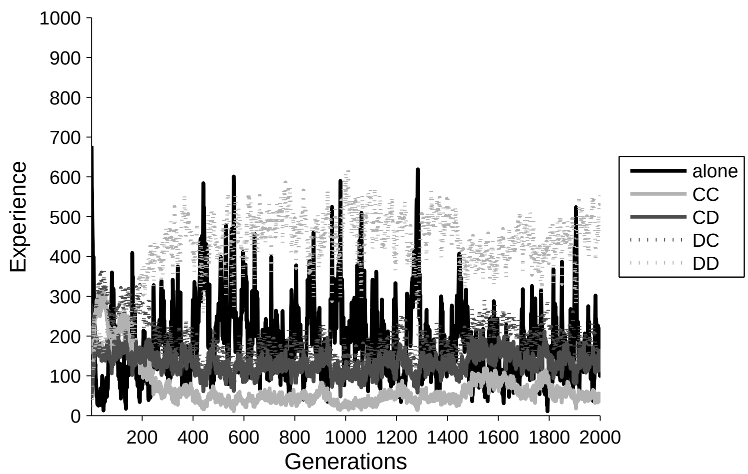
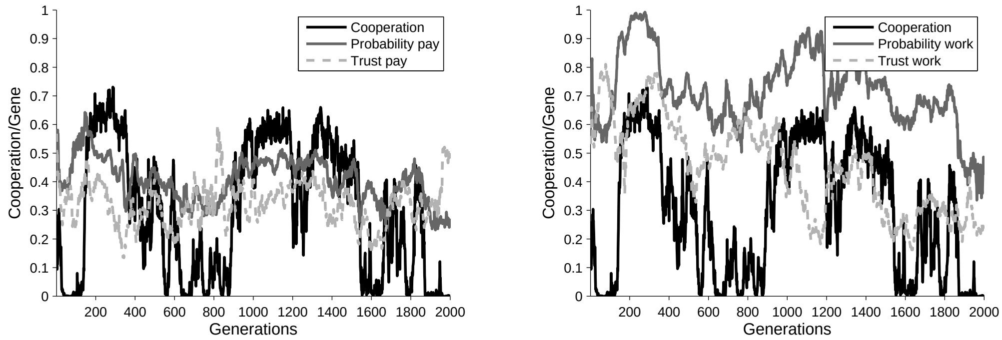


Figure C.3: *The experiences from the individuals vs. the generations. Results with revoked assignments. The different lines plotted mean; alone: individual works alone, CC: both cooperate, CD: helper defects, DC: individual defects, DD: both defect.*

C.2 Revoked assignments

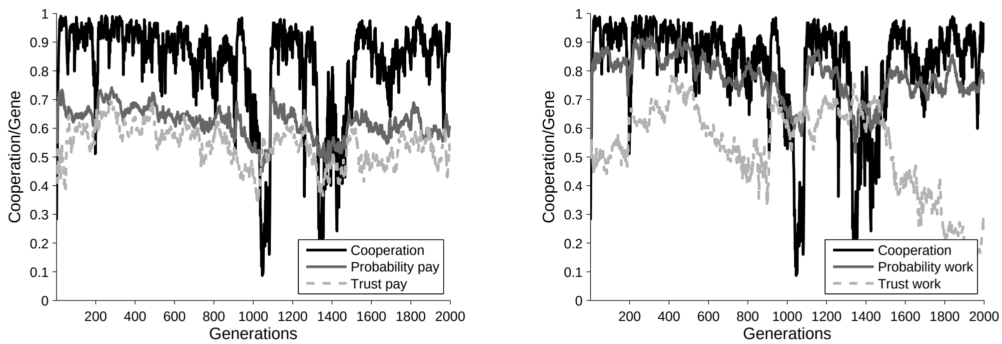
C.2.1 Different assignment with pr=0.3



(a) Pay genes (b) Work genes

Figure C.4: *The amount of cooperation plotted with the different trust genes averaged from the individuals. Single run for a different revoked assignment with a proportion reward of 0.3. The cooperation is plotted twice for clarity.*

C.2.2 Same assignment with pr=0.2



(a) Pay genes (b) Work genes

Figure C.5: *The amount of cooperation plotted with the different trust genes averaged from the individuals. Single run for a different revoked assignment with a proportion reward of 0.3. The cooperation is plotted twice for clarity.*

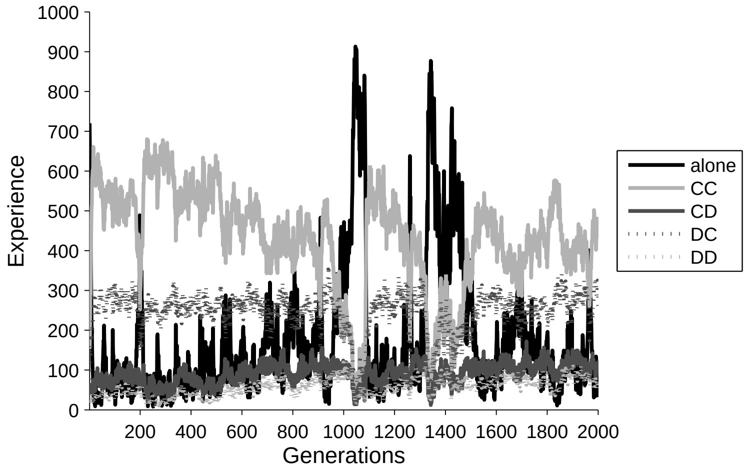


Figure C.6: *The experiences from the individuals vs. the generations. Results with revoked assignments. The different lines plotted mean; alone: individual works alone, CC: both cooperate, CD: helper defects, DC: individual defects, DD: both defect.*

Appendix D

Parameter settings

This appendix shows the different parameter settings used in both models. In Section D.1 the parameters for the model without the trust mechanism is given, and in Section D.2 the additional or overwritten parameters for the model with the trust mechanism is shown.

D.1 Model without trust mechanism

Table D.1 shows the fixed parameter settings as original defined by Koldijk. The same settings were used in this thesis, except for the number of generations which was reduced to 100. Table D.2 shows the manipulated parameters for the model without the cooperation mechanism.

From how many individuals a helper can be chosen	NR HELPERS	3
When expected pay-off threshold a helper helps	HELPER THRESHOLD	0
How much working alone costs extra	EXTRA COSTS	1.3
Populationsize	pop.subpop.0.size	100
Number of generations	generations	200
Length of genome	genome-size	3
Minimum value of genes	min-gene	0.1
Maximum value of genes	max-gene	1
Mutation probability	mutation-prob	0.33
Mutation type	mutation-type	gauss
Standard deviation of Gaussian	mutation-stdev	0.05
Selection method	pipe.source.0	Tournament Selection
Tournament size	select.tournament.size	3

Table D.1: *The fixed parameters settings (with the variable names in the programming code) as used by Koldijk. The same settings were used in this research, except for the number of generations which was in this research set to 100. Table from Koldijk[10].*

Assignments	Simple, different, same
Proportion of the reward the helper gets	0 0.15 0.25

Table D.2: *The manipulated parameters settings used in the model without the cooperation mechanism.*

D.2 Model with trust mechanism

In the model with the trust mechanism af few parameters were added or changed. Table D.3 shows the fixed parameters that were replaced or added in the model with the trust mechanism. Table D.4 shows the addition of the manipulated parameter.

Number of generations	2000	changed
Length of genome	7	changed
Minimum value of genes	skill-genes 0.1, trust-genes 0	changed
Mutation probability	0.1	changed
Cross-over type	uniform	added
Cross-over probability	0.5	added
Genes to cross-over	3	added

Table D.3: *The changed and added fixed parameters settings used in the model with the cooperation mechanism compared to the model without the cooperation mechanism.*

Action when both betray	revoked assignments, non-revoked assignments	added

Table D.4: *The manipulated parameters settings used in the model without the cooperation mechanism.*