Applied Artificial Intelligence, 14:769–784, 2000

0883-9514/00 $12.00 + .00

BOOSTING COOPERATION BY EVOLVING TRUST

ANDREAS BIRK
Vrije Universiteit Brussel, Artificial Intelligence Laboratory, Brussels, Belgium

Instead of establishing trust through defining compliance-based standards like protocols augmented by cryptographic methods, it is shown that trust can emerge as a self-organizing phenomenon in a complex dynamical system. It is assumed that trust can be modeled on the basis of an intrinsic property called trustworthiness in every individual i. *Trustworthiness is an objective measure for other individuals, whether it is desirable to engage in an interaction with* i *or not. Trustworthiness cannot directly be perceived. Building trust, therefore, relates to estimating trustworthiness. Subjective criteria like outer appearance are important for building trust as they allow the handling of unknown agents for whom data from previous interactions do not exist. Here, trustworthiness is grounded in the strategies of agents who engage in an extended version of the iterated prisoner's dilemma. Trust is represented as a preference to be grouped together with agents with a certain label to play a game. It is shown that stable relations of trust can emerge and that the coevolution of trust boosts the evolution of cooperation.*

The investigation of the formation and application of trust is interesting from two different perspectives. First, it relates to basic research on fundamental principles of social interactions between living systems, especially humans. Second, constructive approaches investigating trust are important for applications, allowing autonomous interactions between artificial systems.

Almost all higher life forms interact with other individuals of their kind, leading to complex social behaviors. Man is no exception; on the contrary, social and cultural behaviors are among the most crucial aspects of the major pride of humans, namely cognition. The study of the social interactions of living systems in general and of humans in particular is, accordingly, an important issue of basic research, and it is pursued in many different fields like ecology, economics, and social science, to name just a few.

Andreas Birk is a research fellow (OZM-980252) of the Flemish Institution for Applied Research (IWT).

Address correspondence to Andreas Birk, Vrije Universiteit Brussel, Artificial Intelligence Laboratory, Pleinlaan 2, 106725, 1050 Brussels, Belgium. E-mail: birk@ieee.org

Today, there is an increasing amount of artificial autonomous systems, which control various devices without a continuous or explicit supervision by humans. To reach their full potential, artificial autonomous systems must be capable of interacting with each other. E-commerce can, for example, only be successful if a multitude of devices cooperates in an autonomous delivery of goods *into* the buyer's place. There must be, for example, a goods-port, which is capable of handling deliveries without the presence of a human supervisor.

The classic scientific fields investigating the basic principles of social interactions between living systems were forced to rely on *descriptive* approaches for their work, as the manipulation of living societies is simply infeasible or even immoral. The appearance of artificial autonomous systems now leads to the possibility as well as to the need to use *constructive* approaches. It is on the one hand a gift, as it allows one to "tinker" with artificial but nonetheless complex societies to investigate basic research questions. It poses, on the other hand, a serious challenge as further technological progress needs working solutions for concrete applications.

The field of multiagent systems (MAS) (Castelfranchi & Wemer, 1994; Demazeau & Müller, 1991; Garijo & Boman, 1999; Jennings & Wooldridge, 1998a, 1998b) focuses on compliance-based approaches for a constructive investigation or exploitation of artificial societies. This means this field tries to establish standards for agent languages and architectures (Müller et al., 1996; Müller et al., 1999; Singh et al., 1998; Wooldridge et al., 1996) within which interactions take place. In respect to trust, cryptographic methods are, for example, used (Harbison, 1998; Lehti & Nikander, 1998; Phillips, 1997), which establish a well-defined security at the cost of restricting interactions to systems that comply to the standard.

Here in contrast, trust is formed in a dynamical process. There is no absolute security as trusted systems can cheat. But the process is completely open and robust as trust is not predefined, but emerges from subtle interactions between the systems. The basic ideas of this process go back to two roots, namely, the field of artificial life or short Alife (Langton, 1989; Langton et al., 1990; Steels, 1994a) and the field of evolutionary game theory (Axelrod, 1984; Axelrod & Hamilton, 1981; Smith, 1984; Smith & Price, 1973). Before the process of the formation of trust can be described, it is necessary to first define the notion of trust itself as it is used here. The basis for trust is an intrinsic property of each individual in form of the so-called *trustworthiness*. The trustworthiness of an agent a_A is an *objective* measure for another agent a_B of the desirability of interactions with a_A. If the possibly continuous, trustworthiness of a_A is high, it is highly desirable for a_B to engage in trust-based interactions with a_A. Vice versa, if the trustworthiness of a_A is low, it is highly undesirable for A_B to engage in interactions with a_A.

The trustworthiness of a_A can be dynamic, both in time as well as in

respect to the agent space, i.e., the trustworthiness of a_A for agent a_B can be different from the trustworthiness of a_A for agent a_C at the same moment in time. The problem of trustworthiness is that it is an internal state of agent a_A, which cannot be accessed by any other agent. It might even be based on hidden or so-called unconscious processes such that a_A himself cannot truly determine its own trustworthiness for others. In addition, trustworthiness is dynamic, i.e., even when correctly determining it once in respect to its meaning for one agent, this information can be useless shortly after time or in respect to another agent.

Any process that tries to establish an approximation of the trustworthiness of a_A is here denoted as building *trust*. Processes for building trust often include a nonrational component in the sense that decisions on how to deal with another individual are not only based on previous interactions with this individual, but also on other, presumingly *subjective* criteria. These criteria, for example, include outer appearance, recommendations from others, and so on. Subjective processes for building trust are extremely important as they allow decisions whether to interact or not with unknown individuals, i.e., individuals who have not been encountered in previous interactions.

There are more or less unlimited possibilities for the representation of trustworthiness and for the implementation of trust-building processes. In a purely descriptive approach to the matter, Bacharach and Gambetta (2000), for example, propose to use certain properties of pay-off matrices for representing trustworthiness and signaling theory to formalize the aspects of a subjective building of trust. In this article, trustworthiness has grounded itself in the strategies of agents who engage in an extended version of the iterated prisoner's dilemma. Trust is represented as a preference to be grouped together with agents with a certain label to play a game.

Furthermore, a constructive algorithm for establishing trust is presented. Starting with meaningless labels, agents develop preferences to interact with other agents with a certain type of label. The underlying process follows the principles of self-organization as investigated and used within Alife and evolutionary game theory. So there is no central control. Agents engage only in limited local interactions, and the information exchanged among agents is partial and unreliable. Nevertheless, a stable relation of trust emerges. In addition, trust boosts the evolution of cooperation in the underlying game.

The rest of this article is structured as follows. In the next section, the framework for the experiments is presented. A previously published, extended version of the prisoner's dilemma, allowing continuous degrees of cooperation and N-players as well as strategies for this game, are shortly introduced. The next section shows how trust can be embedded into this framework. A set of labels is used to mark agents. Labels as a kind of outer appearance of agents and the strategies of agents as a basis for trustwor-

thiness are not correlated in the beginning of each experiment. An algorithm is presented, which builds trust by evolving preferences for each agent to be grouped together with agents carrying a certain label. Results based on sets of experiments are presented in the section following. Experimental evidence is given that stable relations of trust can actually emerge from limited interactions among agents. In addition, it is shown that this trust boosts the evolution of cooperation, i.e., a higher cooperative level in the population is reached faster with the trust building mechanism then without it. The final section concludes the article.

THE FRAMEWORK FOR COOPERATION

A Continuous N-Player Prisoner's Dilemma

Roberts and Sherratt (1998) published results on the evolution of cooperation in an extension of the standard prisoner's dilemma (PD) to continuous degrees of cooperation. Partially inspired by this work and partially based on experiments with heterogeneous robots in an artificial ecosystem (Birk & Belpaeme, 1998; Steels, 1994b), following further extension to an N-player case was developed, leading to a continuous-cooperation N-player prisoner's dilemma (CN-PD).

In the artificial ecosystem, simple mobile robots, the so-called "moles," can autonomously recharge their batteries, thus staying operational over extended periods in time. As illustrated in Figure 1, a so-called "head" can track the mobile robots and it can perceive so-called pitfalls which are a kind of inverse charging station where the batteries of the moles are partially discharged via a resistor. When a mobile robot approaches a pitfall, which it cannot distinguish from a charging station, the head can warn the mobile

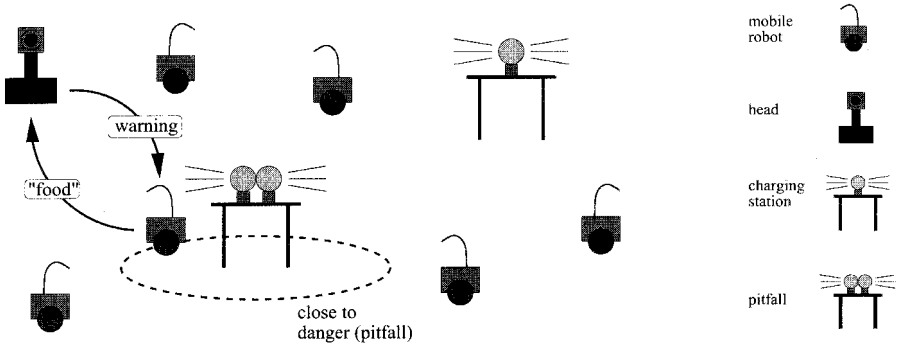

FIGURE 1. The extended artificial ecosystem of the VUB AI-Lab, including a head and several moles. So-called pitfalls in the form of inverse charging stations can suck energy out of a mole. Unlike moles, a head can distinguish pitfalls and charging stations, and it can warn a mole when being close to a pitfall. The mole in return feeds a part of its benefit in the form of energy to the head.

robot. The mobile robot, in exchange, can share the benefit of the saved energy with the head.

Let there be N moles and one head. Each mole m_i ($1 \leq i \leq N$) has a gain G_i based on the avoidance of pitfalls due to warnings of the head. This gain only depends on the so-called headsight $hs \in [0,1]$, i.e., the percentage with which the head perceives dangerous situations. Concretely, the gain is the headsight times one energy unit (EU):

$$G_i = hs \cdot 1.0EU.$$

Furthermore, in the beginning of each time step t, each mole m_i invests up to 0.75 energy units to feed the head. The investment I_i is proportional to the continuous cooperation level $co_i \in [0,1]$ of m_i:

$$I_i = co_i \cdot 0.75EU.$$

The headsight hs depends on the amount of food the head receives from the moles, i.e., the head is completely fed when it receives the 0.75 energy units from every mole. Concretely, the headsight hs is defined as the average sum of cooperation levels in time step t:

$$hs = \sum_{1 \leq i \leq N} co_i/N.$$

The pay-off po_i for a mole m_i is the difference between gain and investment:

$$po_i = G_i - I_i = \sum_{1 \leq j \leq N} co_j/N \cdot 1.0EU - co_i \cdot 0.75EU. \tag{1}$$

So, a dilemma for the moles arises. On the one hand, it is in the interest of each mole that the head is well fed. On the other hand, there is the temptation to leave the task of actual feeding to others, as the head does not react to the behavior of a single mole, i.e., it does not punish a mole when it does not donate energy.

Note that the pay-off for a mole depends on its own cooperation level and on the cooperation levels of all other moles. Let $\bar{co}$ denote the average cooperation level of the group, i.e.,

$$\bar{co} = \sum_{1 \leq i \leq N} co_i/N.$$

The pay-off for a mole m_i can directly be computed for co_i and $\bar{co}$. Namely the pay-off function f_p: $[0,1] \times [0,1] \rightarrow IR$ is

$$f_p(co_i, \bar{co}) = co_i \cdot -0.75EU + \bar{co} \cdot 1.0EU. \tag{2}$$

Based on this, we can extend the terminology for pay-off values in the standard prisoner's dilemma with pay-off types for cooperation (C), punishment (P), temptation (T), and sucking (S), as follows:

- Full cooperation as all fully invest: $C_{all} = f_p(1.0,1.0) = 0.25$;
- All punished as nobody invests: $P_{all} = f_p(0.0,0.0) = 0.0$;
- Maximum temptation: $T_{max} = f_p\left(0.0, \frac{N-1}{N}\right) \geq 0.5$;
- Maximum sucking: $S_{max} = f_p\left(0.0, \frac{1}{N}\right) \leq -0.25$.

For co, $\bar{co} \neq 0.0$, 1.0, one gets the following additional types of pay-offs, the so-called partial temptation, the weak cooperation, the single punishment, and the partial sucking. They are not constants (for a fixed N) like the previous ones, but actual functions in $(co, \bar{co})$. Concretely, they are subfunctions of f_p $(co, \bar{co})$, operating on subspaces defined by relations of co in respect to $\bar{co}$ (Table 1).

Note that for a fixed average cooperation level $\bar{co}$ and two individual cooperation levels $co' > co''$, it always holds that $f_p(co', \bar{co}) < f_p(co'', \bar{co})$. Therefore, it holds for an individual player in a single game that

- The partial temptation always pays better than weak cooperation.
- The partial temptation increases with decreasing individual cooperation.
- The absolute value of partial sucking increases with increasing individual cooperation.

This can also be stated as

$$\begin{aligned} T_{max} > T_{partial}(.) > C_{all} > C_{weak}(.) > 0.0 \\ P_{single}(.) = P_{all} = 0.0 \\ S_{max} < S_{partial}(.) < 0.0. \end{aligned} \tag{3}$$

Equation 3 illustrates the motivation for the names of the different types of pay-off. The attribute *max* for temptation T and sucking S indicates that these are the maximum absolute values. The *partial* accordingly indicates

TABLE 1 Additional Pay-Off Types in the CN-PD

$co < \bar{co}$	$\bar{co} \leq co < 4/3 \cdot \bar{co}$	$co = 4/3 \cdot \bar{co}$	$co > 4/3 \cdot \bar{co}$
$T_{partial}(co, \bar{co})$ $\in]0, 1.0[$ partial temptation	$C_{weak}(co, \bar{co})$ $\in]0, 0.25[$ weak cooperation	$P_{single}(co, \bar{co})$ $= 0$ single punish	$S_{partial}(co, \bar{co})$ $\in]-1.0, 0[$ partial sucking

that these values are only partially reached through the related T or S functions. The attribute *weak* for the cooperation function C relates to the fact that though the player receives a positive pay-off, it is less than in the maximum cooperation case where *all* players fully cooperate. When no player invests, *all* are punished with a zero pay-off. Whereas in the *single* case, at least the individual player we are looking at gets punished with a zero pay-off; other players can receive all types of pay-off.

The Evolution of Cooperation in the Iterated CN-PD

Much like in the standard prisoner's dilemma, rational agents are also trapped in the global punishment as nobody will feed the head in a single game of the CN-PD. But when iterating the game over several time-steps t, strategies taking previous cooperation or noncooperation of others into account can lead to the agents into cooperation. In Birk & Wiernik (2000), where the continuous N-player prisoner's dilemma is also described in more detail, a novel strategy, the so-called justified-snobism (JS), is presented.

Justified-snobism cooperates slightly more than the average cooperation level of the group of N players if a nonnegative pay-off was achieved in the previous iteration, and it cooperates exactly at the previous average cooperation level of group otherwise. So JS tries to be slightly more cooperative than the average. This leads to the name for this strategy as the snobbish belief to be "better" (in terms of altruism) than the average of the group is somehow justified for players which use this strategy. It can be shown that JS is a successful strategy for the CN-PD and especially that JS is evolutionary stable. In the experiments reported here, JS has to compete with following strategies in iterated CN-PDs:

Follow-the-masses (FTM): match the average cooperation level from the previous iteration, i.e., $co_i[t] = \bar{co}[t-1]$.

Hide-in-the-masses (HIM): subtract a small constant c from the average cooperation level, i.e., $co_i[t] = \bar{co}[t-1] - c$.

Occasional-short-changed-JS ($OSC\text{-}JS$): a slight variation of JS, where occasionally the small constant c is subtracted from the JS-investment.

Occasional-cheating-JS ($OC\text{-}JS$): another slight variation of JS, where occasionally nothing is invested.

Challenge-the-masses (CTM): Zero cooperation when the previous average cooperation is below one's one cooperation level, a constant cooperation level c' otherwise, i.e.,

- $co_i[t-1] \geq \bar{co} : co_i[t] = c'$
- $co_i[t-1] < \bar{co} : co_i[t] = 0$.

Nonaltruism (NA): always completely defect, i.e., $co_i[t] = 0$.

Anything-will-do (AWD): always cooperate at a fixed level, i.e., $co_i[t] = c'$.

The strategies compete in an evolutionary tournament proceeding in time-steps t. In the beginning, a population of 1000 agents is randomly created, i.e., each agent gets a randomly selected strategy following an even distribution. In the iterations $t \rightarrow t + 1$, the population is divided in a random manner into groups with 20 agents each. Each group plays a CN-PD. The so-called fitness $fit(a_i)$ of agent a_i in time-step t is determined by the running average of its pay-offs, i.e.:

$$fit(a_i)[t] = (1 - q) \cdot po_i[t] + q \cdot fit(a_i)[t - 1]$$

with

$$q \in]0.0, 1.0[.$$

Reproduction of agents is proportional to their fitness as roulette-wheel selection keeps the population size fixed to 1000. In addition, new agents are randomly created in each time-step with a small likelihood $p_{new} = 0.05$. When running the evolutionary tournament without trust, JS multiplies and starts to take over the population (Figure 5).

THE EVOLUTION OF TRUST

The Outer Appearance of Agents

As mentioned is the introduction, trust is seen here as a kind of subjective criterion guiding the interaction with others. More concretely, a strategy is based on the objective measures on the performance of other agents, namely, their cooperation level in previous iterations. Furthermore, the given strategy $strat_A$ of an agent a_A establishes its trustworthiness in an objective manner. If another agent a_B would explicitly known $strat_A$, then a_B could rationally decide whether it is desirable to play a game with a_A or not.

Trust, in contrast, is based on secondary, derived measures, here the "outer appearance" of an agent in form of a marker. The main idea is as follows: Agents are randomly marked with labels from a finite set $S_L = \{l_1, \ldots, l_k\}$. The function L maps labels to agents. Whenever a new agent a is created, a label l is randomly selected and assigned to a, i.e., $L(a) = l$. Note that this assignment is completely independent of the strategy of the agent, even during the course of the evolution. The set $S_{color} =$ `{red,green,blue}` will serve as an extremely simple example of labels in the remainder of this article.

The coevolution of strategies and trust proceeds in time-steps t, much like the evolution of strategies on its own. Figure 2 illustrates the overall program in pseudo-code. The crucial change to the mere evolution of strategies is that groups are not randomly formed anymore, but based on prefer-

```
co-evolve strategies and trust {
/* random initialization */
    t = 1
    pop[1] = ∅
    while #pop[1] < N {
        random create agent a
        pop[1] = pop[1] ∪ {a}
    }
/* evolutionary step (t → t + 1) */
    while(True) {
        ∀1 ≤ i ≤ N_G : form group G_i[t]
        ∀1 ≤ i ≤ N_G : play k CN-PD on G_i
        pop[t] = ⋃_{1≤i≤N_G} G_i[t]
        evolve strategies on pop[t]
        t = t + 1
    }
}

random create agent a {
    l = random select(S_L)
    L(a) = l
    strat = random select(S_strat)
    Strat(a) = strat
    }
}
```

with

- N_G is the (fixed) number of groups in the population and N is the (fixed) total number of agents in the population
- `random select` (set S) returns an element from S following an even distribution of probabilities

FIGURE 2. The pseudo-code of the overall coevolution of trust and strategies.

ences on labels. The exact mechanism is described in detail in the next section.

Trust as Preferences in the Group Formation

Trust as estimation of trustworthiness is established through preferences in the group formation, i.e., agents prefer to play games with agents carrying a certain marker. Note that this trust cannot be justified by rational means

alone, at least in the beginning of the iterated games, as there is no correlation between a certain label and a certain strategy.

Concretely, the so-called trust function $trust_i: L \rightarrow [0.0,1.0]$ of an agent a_i maps a weight w to each possible label l_j, such that $trust_i(l_j) = w$. The weight w represents a_i's preference to interact with an agent with label l_j. If w is high, i.e., close to 1.0, a_i prefers to interact with agents with label l_j, or it simply trusts them. If w is low, *i.e.*, close to 0.0, a_i prefers not to interact with agents with label l_j, or it simply does not trust them.

The pseudo-code program in Figure 3 shows how a trust functions are concretely used to form a new group G. The simple example of color labels S_{color} is now used to illustrate how a new group is formed. Given the trust functions of several agents at time-step $t-1$ as listed in Table 2, a new group G of size $N_A = 6$ is formed as follows. First, an agent is selected from $pop[t-1]$ with the roulette-wheel principle based on the fitness of all agents. Let us assume agent a_1 is selected. After the first agent a_1 has been added to the group G, further agents are added in iterations of the lines 8 to 10. In the first iteration, the trust-function of a_1 is used to initialize the summed weight sw for each possible label l_i (line 8). Table 3 shows the results of this first iteration and for the further iterations. In general, the

```
1  form group G {
2  /* initialize the group G with one agent based on fitness */
3      G = ∅
4      a = roulette-wheel selection (pop[t − 1], fit())
5      G = G ∪ {a}
6  /* add agents to G based on the trust of the agents already in G */
7      while #G < N_A
8          ∀l_i ∈ S_L : sw(l_i) = Σ_{a_j ∈ G} trust_j(l_i)
9          a' = roulette-wheel selection (pop[t], sw())
10         G = G ∪ {a'}
11     }
12 }
```

with

- N_A is the (fixed) number of agents per group
- `roulette-wheel selection` (set S, function $f : S \rightarrow \mathbb{R}$) returns an agent a from set S with a likelihood $prob$ proportional to $f(a)$, i.e., $prob(a) = f(a)/\sum_{a' \in S} f(a')$

FIGURE 3. The pseudo-code of a group formation based on the trust functions.

TABLE 2 An Example of Trust Functions for the Agents a_1 to a_6 in Time-Step $t - 1$

Agent a_i	$Trust_i$(red)	$Trust_i$(green)	$Trust_i$(blue)
a_1	0.241	0.987	0.328
a_2	0.793	0.846	0.201
a_3	0.086	0.392	0.003
a_4	0.393	0.586	0.245
a_5	0.230	0.567	0.045
a_6	0.187	0.793	0.627

function $sw(\,)$ is used to bias the selection of the next agent which is included in the group (line 9) for each possible label. Table 4 shows the according probabilities for this first and the further iterations with the color set example.

As the number of labels is usually much smaller than the size of the population, the roulette-wheel selection of an agent biased with label preferences is done for efficiency reasons as follows. First, a label l is chosen with roulette-wheel selection using the bias $sw(\,)$. Then a sequential search is used starting from a random position in the population. The first agent with label l which is encountered is added to the group G.

Back to the color set example, let us assume that agent a_2 is selected and added to the group at the end of the first iteration. In the next second iteration of lines 8 to 10, the summed preferences of agent a_1 and a_2 are used to select the next agent a_3, and so on.

TABLE 3 The Weights for the Roulette-Wheel Selection of Additional Agents for Group G

Iteration	Group G	Sw(red)	Sw(green)	Sw(blue)
1	$G = \{a_1\}$	0.241	0.987	0.328
2	$G = \{a_1, a_2\}$	1.034	1.833	0.529
3	$G = \{a_1, \ldots, a_3\}$	1.120	2.225	0.532
4	$G = \{a_1, \ldots, a_4\}$	1.513	2.811	0.777
5	$G = \{a_1, \ldots, a_5\}$	1.743	3.378	0.822
6	$G = \{a_1, \ldots, a_6\}$	1.930	4.171	1.449

TABLE 4 The Development of the Probabilities of the Color of the Agent that Is Added to the Group

Iteration	Group G	$Prob$(red)	$Prob$(green)	$Prob$(blue)
1	$G = \{a_1\}$	0.155	0.634	0.210
2	$G = \{a_1, a_2\}$	0.304	0.540	0.156
3	$G = \{a_1, \ldots, a_3\}$	0.289	0.574	0.137
4	$G = \{a_1, \ldots, a_4\}$	0.297	0.551	0.152
5	$G = \{a_1, \ldots, a_5\}$	0.293	0.568	0.138
6	$G = \{a_1, \ldots, a_6\}$	0.255	0.552	0.191

The Update of the Trust Functions

The trust function of an agent a_i is updated in each time-step t based on the (very limited) experiences with other agents with a certain label. Concretely, the weight of trusting agents with label l_j is updated in each game proportionally to the pay-off and the number of agents with that label in the group. This means that when many agents with label l_j are in the group and the pay-off is high, then the agent a_i increases its trust in agents with that label l_j. A running average is used to sum the updates over consecutive time-steps:

$$trust_i(l_j)[t] = (1 - q) \cdot trust_i(l_j)[t-1] + q \cdot po_i[t-1] \cdot \#\{a_k \in G \text{ with } L(a_k) = l_j\}/N_A$$

with $q \in]0.0, 1.0[$.

The constant q is set to 0.1 in all experiments reported here.

Again, let us return to the example with the set S_{color} of color labels. Assume that the group G = $\{a_1 \ldots, a_6\}$ has played a CN-PD game in time-step $t - 1$. The pay-offs for each agent in this game and the (fixed) labels of each agent are shown in Table 5. Agent a_4, for example, has received a rather high pay-off. As there are rather many blue agents in the group, a_4 increases its trust in this color as

$$\begin{aligned} trust_4(\texttt{blue})[t] &= (1 - 0.1) \cdot trust_4(\texttt{blue})[t-1] \\ &\quad + 0.1 \cdot po_4[t-1] \cdot \#\{a_k \in G \text{ with } L(a_k) = \texttt{blue}\}/N_A \\ &= 0.9 \cdot 0.245 \\ &\quad + 0.1 \cdot 0.404 \cdot 4/6 \\ &= 0.247. \end{aligned}$$

Note that each of the blue agents can have a different strategy. Especially in the beginning of the evolution, where labels and strategies are independently

TABLE 5 The (Fixed) Labels of the Agents a_1 to a_6 and Their Pay-Offs in Time-Step t

Agent a_i	Label $L(a_i)$	Pay-off $po_i[t-1]$
a_1	blue	0.257
a_2	green	−0.035
a_3	blue	0.392
a_4	red	0.404
a_5	blue	0.157
a_6	blue	0.289

distributed among randomly created agents, this is very likely. Note also that the relatively high pay-off for agent a_4 can be due to an exploitation of the green agent a_2 and rather independent from the presence of the four blue agents.

RESULTS

Trust Becomes Stable

There is neither a meaningful form of trust nor a (obvious) basis for it in the beginning of each experiment. Or more concretely, neither the labels nor the trust functions contain any information or meaning in the beginning of each experiment:

- The labels are randomly assigned to the agents. Therefore, there is no meaningful relation between an agents's label and its strategy.
- The trust functions of the agents are randomly initialized. Therefore, there is no a priori, global preference of agents to be grouped together.

Nevertheless, a stable relation of trust emerges. This means, the agents evolve fixed preferences for interacting with agents with a certain label. Figure 4 shows the percentage of agents that cannot "decide" which type of agent they should trust. More precisely, the graph shows the percentage of agents where the highest preference of a particular agent for a certain label

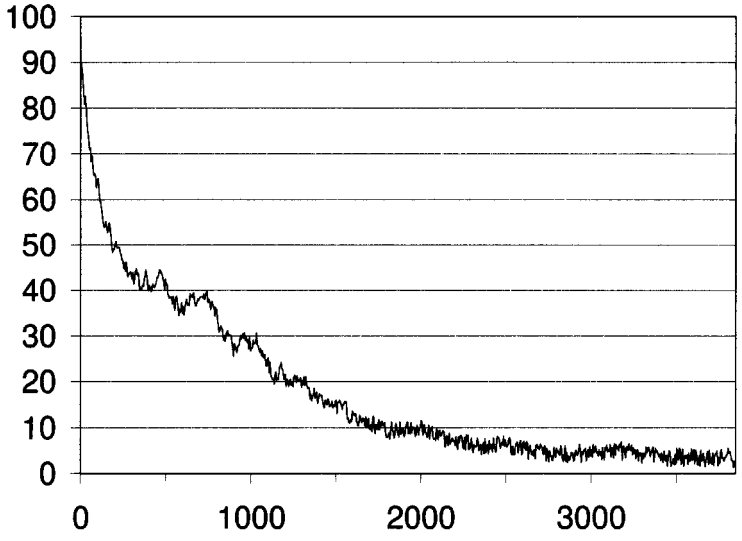

FIGURE 4. The percentage of agents in the population which cannot "decide" which types of agents they should trust. In the beginning of the run, this percentage is high as most agents change their preference in every time-step, more or less randomly guessing. After a while, fixed preferences evolve.

in the current step is different from its highest preference in the previous step. In the beginning of the run, the percentage of "undecided" agents is very high, i.e., the agents are more or less randomly guessing in each step which type of agents they should trust. After a while, this indecision is dropping to almost zero, i.e., the agents evolve fixed preferences for certain labels.

Note that the basis for this evolving trust as fixed preferences is really subjective in some sense. First, it is grounded on very limited data, i.e., there are many agents with label l_j in the population, but an agent a_i builds up some belief by interacting with just a few of them. Second, within the group to which agent a_i belongs to at a time-step t, there are (most probably) many different agents in respect to labels. The update of trust does not distinguish between those labels, though different agents, and accordingly labels, do (most probably) contribute very differently to the pay-off that a_i receives.

Evolution of Trust Boosts the Evolution of Cooperation

Despite the lack of meaning for the labels in the beginning, the evolution of trust boosts the evolution of cooperation in these experiments. Figure 5 shows the development of the general cooperation level for both cases, namely, respectively 50 averaged runs with and without a coevolution of trust. When the coevolution of trust is activated, a higher general level of cooperation is reached much faster than without an evolution of trust.

In each of the 50 runs, the population evolved into a set of agents, which in their majority had the following properties:

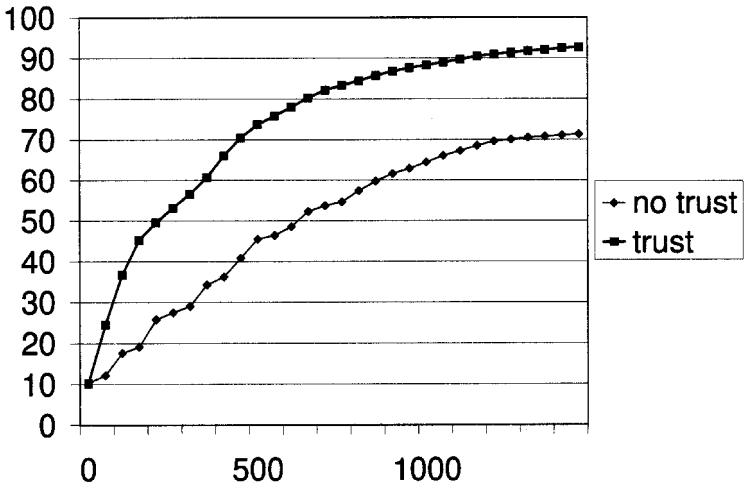

FIGURE 5. The general cooperation levels, averaged from, respectively, 50 runs with and without a coevolution of trust. When trust is activated, a higher general cooperation level is reached much faster than without trust.

- they all follows the cooperative strategy JS
- they all are marked with the same label l
- they all have a high trust in the label l
- they all have a low trust in other labels.

This result indicates a possible explanation for the boosting of cooperation in the reported experiments. In the beginning, labels are evenly distributed over agents and thus strategies. Also, preferences are evenly distributed. Assume random fluctuations cause a slightly above-average likelihood that trustworthy agents, which means here agents with a cooperative strategy, have a certain label. If there is, in addition, a slightly above-average likelihood that trustworthy agents trust this label, then there is the possibility that this subtle effect reinforces itself. As a result, trustworthy agents can so-to-say recognize each other and actively group together.

CONCLUSION

Trust is here modeled as emergent property in a complex dynamical system. Its basis is trustworthiness, which is defined as an intrinsic property of an individual i_A in respect to another individual i_B. It is an objective criterion in the sense that it gives i_B a measure allowing a rational choice of whether to interact with i_A or not. Unfortunately, the trustworthiness i_A is not perceivable by i_B in the general case. It is even questionable if i_A can access its very own trustworthiness for i_B, as it is an internal state that can be derived from well-hidden or so-called unconscious processes.

The building of trust deals with the approximation of trustworthiness. When an individual meets another one for the first time, there is no objective data from previous interactions allowing a rational choice of whether to interact or not. Subjective criteria like outer appearance must be used in those situations. Here, (in the beginning) meaningless labels are used for this purpose. Trust as approximation of trustworthiness is established through the preferences of agents to be grouped together with other agents carrying a certain marker. Groups play a game based on an extended version of the prisoner's dilemma. Strategies of the agents in the iterated game establish their trustworthiness.

A constructive way to update trust, based on limited interactions with other agents, is presented. In the experiments reported here, there is neither a correlation between labels and strategies, nor between preferences and strategies in the beginning of each experiment. Nevertheless, stable relations of trust emerge. Furthermore, the coevolution of trust can significantly boost the evolution of cooperation. This means that in the underlying evolutionary game, a higher cooperative level in the population is reached faster with the trust building than without it.

REFERENCES

Axelrod, R., and W. D. Hamilton. 1981. The evolution of cooperation. *Science* 211:1390–1396.

Axelrod, R. 1984. *The evolution of cooperation*. New York: Basic Books.

Bacharach, M., and D. Gambetta. 2000. Trust in signs. In *Trust and Social Structure*, ed. K. Cook. New York: Russell Sage Foundation.

Birk, A., and T. Belpaeme. 1998. A multi-agent-system based on heterogeneous robots. In *Collective Robotics Workshop 98. Lecture Notes in Artificial Intelligence*. Berlin: Springer.

Birk, A., and J. Wiernik. 2000. An N-player prisoner's dilemma in a robotic ecosystem. *Proceedings of the 8th International Symposium on Intelligent Robots*.

Castelfranchi, C., and E. Wemer, eds. 1994. Artificial social systems—Selected papers from the Fourth European Workshop on Modelling Autonomous Agents in a Multi-Agent World, MAAMAW-92, *Lecture Notes in Artificial Intelligence 830*, Heidelberg, Germany: Springer-Verlag.

Demazeau, Y., and J.-P. Müller, eds. 1991. *Decentralized AI 2—Proceedings of the Second European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-90)*. Amsterdam, The Netherlands: Elsevier Science Publishers B.V.

Garijo, F. J., and M. Boman, eds. Proceedings of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World: Multi-Agent System Engineering (MAAMAW-99), *LNAI 1647*, June 30–July 2, 1999. Berlin: Springer.

Harbison, W. S. 1998. Delegating trust. In IWSP: International Workshop on Security Protocols, *LNCS*.

Jennings, N. R., and M. R. Wooldridge, eds. 1998. *Agent technology; Foundations, applications, and markets*. New York: Springer.

Jennings, N. R., and M. R. Wooldridge. 1998. Applications of intelligent agents. In *Agent technology; Foundations, applications, and markets*, eds. N. R. Jennings and M. R. Wooldridge, New York: Springer.

Langton, C. C. Sept. 1989. Artificial life. In *Proceedings of the Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems (ALIFE '87)*, Santa Fe Institute Studies in the Sciences of Complexity, Volume 6, 1–48, Redwood City, CA. Reading, MA: Addison–Wesley.

Langton, C. G., C. E. Taylor, J. D. Farmer, and S. Rasmussen, eds. 1990. *Artificial life II*. Reading, MA: Addison–Wesley.

Lehti, I., and P. Nikander. 1998. Certifying trust. In PKC: International Workshop on Practice and Theory in Public Key Cryptography. *LNCS*.

Müller, J. P., N. R. Jennings, and M. R. Wooldridge, eds. 1996. *Intelligent Agents III; Proc. of the ECAI'96 Workshop on Agent Theories, Architectures, and Languages*. New York: Springer.

Müller, J., M. P. Singh, and A. S. Rao, eds. 1999. Proceedings of the 5th International Workshop on Intelligent Agents V: Agent Theories, Architectures and Languages (ATAL-98), *LNAI 1555*, July 4–7 1999. Berlin: Springer.

Phillips, D. J. 1997. Cryptography, secrets, and the structuring of trust. In *Technology and privacy: The new landscape*, eds. P. E. Agre and M. Rotenberg, Cambridge, MA: The MIT Press.

Roberts, G., and T. N. Sherratt. 1998. Development of cooperative relationships through increasing investment. *Nature* 394 (July):175–179.

Singh, M. P., A. Rao, and M. J. Wooldridge, eds. 1998. Proceedings of the 4th International Workshop on Agent Theories. Architectures and Languages (ATAL-97), *LNAI 1365*, July 24–26. Berlin: Springer.

Smith, J. M. 1984. The evolution of animal intelligence. In *Minds, machines and evolution*, ed. C. Hookway, London: Cambridge University Press.

Smith, J. M., and G. R. Price. 1973. The logic of animal conflict. *Nature* 246:441–443.

Steels, L. 1994. The artificial life roots of artificial intelligence. *Artificial Life Journal* 1(1).

Steels, L. 1994. A case study in the behavior-oriented design of autonomous agents. 1994. In *From Animals to Animats 3. Proceedings of the Third International Conference on Simulation of Adaptive Behavior*, eds. D. Cliff, P. Husbands, J. A. Meyer, and S. W. Wilson, Cambridge: The MIT Press/ Bradford Books.

Wooldridge, M., J. P. Muller, and M. Tambe, eds. 1996. *Intelligent Agents Volume II—Proceedings of the 1995 Workshop on Agent Theories, Architectures, and Languages (ATAL-95)*. Berlin: Springer-Verlag.