3-5-2015

Reward Shaping in the Ant Colony System: Lessons for the Design of Collective Intelligence Systems

Alexander Kornrumpf

Ulrike Baumöl

Reward Shaping in the Ant Colony System: Lessons for the Design of Collective Intelligence Systems

Alexander Kornrumpf[1], Ulrike Baumöl[1]

[1] FernUniversität Hagen, Lehrstuhl für Informationsmanagement, 58084 Hagen
{alexander.kornrumpf, ulrike.baumoel}@fernuni-hagen.de

Abstract. The problem-oriented design of collective intelligence systems (CIS) is in itself an open problem. Previous research draws upon findings from biological swarm intelligence to derive guiding design principles but also highlights the importance of evaluating the system's state with respect to the given problem. We investigate this evaluation task on the individual and the global level within the framework inspired by reinforcement learning. We map different modes of evaluation to different schemes of rewarding agents, thereby illustrating that designer of CIS face the task of reward shaping. We simulate several reward schemes as variations of the well-known ant colony system (ACS). We show that rewards in the ACS, although they consist only of a single value, the metaphorical pheromone concentration, have complex semantics, and coordinate the distribution of information and allocation of work within the system. This makes the ACS a valuable source of inspiration for CIS with human agents.

Keywords: collective intelligence, swarm intelligence, reinforcement learning.

1 Introduction

The topic of *collective intelligence (CI)* is increasingly generating attention from researchers and practitioners alike [1, 2]. The last decade has produced new and successful forms of collaboration such as crowdsourcing [3], open innovation [4] and wikis [5]. Researchers and the general public have readily filed these successes as examples of CI or more colloquial the "wisdom of the crowds" [6, 7], the latter expression stemming from a book of the same name [8]. However, despite the aforementioned examples of success, a general, goal-oriented approach to the design and implementation of systems that enable CI – *collective intelligence systems (CIS)* – is still an open research problem [2, 9] and in fact has been since the 1990s [10].
To a great extent, CIS design has been inspired by the behavior of animal swarms, often called swarm intelligence (SI). A review of early work seeking to derive properties of CI from the behavior of insect swarms can be found in [11]. More recent examples of work on the application of SI principles – e. g. stigmergy, adaptivity, flexibility, robustness, and self-organization – to business and economics include [12–15]. There is a historic tendency to equate the SI of insects with the CI of humans. However, while SI principles provide valuable guidelines for CIS design, it is also obvious

12th International Conference on Wirtschaftsinformatik,
March 4-6 2015, Osnabrück, Germany

Kornrumpf, A.; Baumöl, U. (2015): Reward Shaping in the Ant Colony System: Lessons for the Design of Collective Intelligence Systems, in: Thomas. O.; Teuteberg, F. (Hrsg.): Proceedings der 12. Internationalen Tagung Wirtschaftsinformatik (WI 2015), Osnabrück, S. 842-856

that the analogy carries only so far. The differences between animal swarms and human collectives impose limitations on the degree to which SI is applicable to humans. Human abilities would be neglected in a CIS design guided solely by SI principles [16].

In this paper, we seek to investigate the transition between SI and CI more closely. We depart from three observations. First, CI often builds on SI only in a metaphorical sense. Vocabulary from SI is used to describe a working system, but the actual system is very different from the inspiring SI. A very well-known example of this is the *ant colony system (ACS)* metaheuristic [17, 18]. The ACS-algorithm draws on inspiration from SI to simulate a colony of artificial ants that can solve problems of combinatorial optimization. But the swarm metaphor is only one way of framing the algorithm. It can also be described within the framework of reinforcement learning [19]. Second, in previous work focusing on the systems aspect of CIS, the state of the CIS itself is assumed to have an intrinsic value with respect to a given problem [10, 20]. Assuming that this value is maximal for a solution state, the task of designing a problem-oriented CIS comes down to the alignment of the collective's evaluation of states with the evaluation of state with respect to the given problem [10] In other words, the system needs to be crafted so that its states align to a given utility function of the individuals. To achieve this, additional incentives beyond the intrinsic value of states might be needed [9]. Third, both, the SI-perspective and the systems-perspective share the notion that information is distributed within the collective, i. e. none of the individuals has complete knowledge or a complete representation of the problem to be solved [11, 21, 22].

Using ACS as an example and reinforcement learning as a formal frame of reference for the study of state-values, we conduct a simulation experiment to address the following questions:

1. Where does the ACS go beyond the inspiration of SI?
2. How does varying the reward which the individuals receive for their actions impact the system's performance?
3. How does varying the way in which individuals share information influence the system's performance?

We find that, despite its heavy reliance on the swarm metaphor, the behavior of agents in the ACS is very unlike that of real ants. On the contrary it is carefully crafted so that the structure of rewards incentivizes intelligent exploitation of shared information on the system-wide scale. These findings suggest that the structure of rewards and distribution of information are two aspects that need to be carefully considered when designing a CIS, even though they play almost no part in the swarm metaphor. We show that the performance of the whole system crucially depends on the rewarding scheme as selected by the designer. Individuals need not only to be incentivized to take actions directed towards solving the problem but also to disclose information gathered during the process, including information on failed attempts.

2 Background

2.1 Collective Intelligence and the Swarm Metaphor

There are many definitions of CI. For this paper, we use a definition from [22], that "*CI is the ability of sufficiently large groups of individuals to create an emergent solution for a specific class of problems or tasks*". This is in line with the consensus of many researchers, as found in [1], that "*CI is greater than the sum of individual contributions*", which is a simplified way to describe emergence, and that "*CI is goal-oriented and focuses on specific problems*". To provide an understanding how this definition builds upon SI, we discuss the swarm metaphor for the ACS. The swarm metaphor is probably best introduced by the words of Dorigo and Gambardella:
"The natural metaphor on which ant algorithms are based is that of ant colonies. Real ants are capable of finding the shortest path from a food source to their nest without using visual cues by exploiting pheromone information. While walking, ants deposit pheromone on the ground, and follow, in probability, pheromone previously deposited by other ants."[17]
This conception draws on earlier work which proposes a simple pheromone model to explain two observations of the behavior displayed by colonies of real ants while foraging for food:

- Presented with a road bifurcation and later rejoin that creates two paths of equal length (a "diamond shaped bridge"), a colony of the ant *Iridomyrmex humilis* will arbitrate on one path and disregard the other one [23]
- Presented with a road bifurcation and later rejoin that creates two path of different length, a colony of the ant *Lasius niger* will arbitrate on the shorter path and disregard the other one [24].

The behavior of said species is *emergent* in the sense that the ants do not have any concept of efficient collective foraging on a global scale but follow only simple local rules. It is *self-organizing* in the sense that no designated leader decides what path should be taken and there is no explicit decision process. The colony arbitrating on one path is a result only of the accumulation of pheromones on that path [23, 24]. The properties of emergence and self-organization are the hallmark of CI as defined above.
The ACS-algorithm uses swarm principles to provide a heuristic solution for problems of combinatorial optimization such as the *travelling salesman problem (TSP)* [17]. In contrast, recent research applies the same swarm principles, in the form of CI, to tackle wicked problems, e. g. global climate change [2, 25]. While these contrasting applications give an impression of the power of the swarm metaphor, swarm principles alone provide little insight into the design of CIS or, as Introne et al. put it, *"there are many examples of such* [CI] *systems, but there is no clear recipe for their development"* [25]. Therefore, we are interested in a notion of what makes CIS work that can be operationalized in CIS design. We use the ACS as an example, because it consists of a very small set of rules and actions and it can be investigated in simulation.

2.2 The Intrinsic Value of States

To get a measure for the success of a CIS, we follow [21] and [20] by assuming that the state of the CIS, what we refer to as the global state, can be mapped to a solution candidate of the problem under consideration. A given solution candidate then can be evaluated with respect to the problem to get what we refer to as the *global* value of the underlying state. In principle, the state of spaces could simply be searched for the optimal solution, or, taking restrictions on transitions between states into account, a path to the optimal solution [26]. In practice, however, this is not possible, if the space of solutions, and thereby the space of states, is too large to be searched exhaustively or not clearly defined in the first place. It is no coincidence that these issues characterize the hardness of combinatorial optimization problems and wicked problems respectively. For a detailed discussion of wicked problems cf. [27]. The solution to this dilemma, as offered by the swarm metaphor, is that the transitions of the system's state are not guided by a global perception of value but by individual's *local* perceptions of the value of a given state. Individual behavior under such conditions can be modelled as reinforcement learning [10, 26, 28]. Emergence can be interpreted as the fact, that the criteria for global evaluation may be radically different from the individual evaluation and even include the translation of the state into a different ontology [22].

Reinforcement learning is agents learning what actions to take, on the basis of observing reward signals. This includes balancing observation and exploitation [29]. We focus specifically on the Q-Learning Algorithm [30]. Let $Q(s,a)$ be the expected value of performing an action a when in some state s from an agent's point of view. The agents can learn the values of Q from experience by repeatedly performing an action and observing the change of state and received rewards. An agent who in state s has performed action a, resulting in the new state s' and reward r may apply the following update rule:

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha(r + \gamma \max_{a'} Q(s',a')) \quad (1)$$

The intuition here is that the value of a state-action pair is given by the immediate reward r plus the future reward to be expected from exploiting the available information on the new state s' (hence the max) but discounted by some factor $0 \leq \gamma \leq 1$. In some scenarios, r is always zero, except for entering some goal state, so that the value of all other state-action pairs equals this reward discounted for the distance to the goal state. For typical reinforcement learning settings, the Q-learning update rule (1) converges to the true values of $Q(s,a)$ [30]. Once this fact is established, the optimal strategy for agents is obvious: In general choose a as to maximize $Q(s,a)$ for the current state s (exploitation). But with a small probability choose another action at random to learn about new state-action values (exploration).

For a functional CIS, global and local perception of a state's value must be in alignment [10]. In simulated SI, e. g. the ACS, this can be enforced by design. In CIS with human collectives the designer has to solve this problem by presenting the CIS in a way that aligns with the individual motivation and add incentives where this is not possible [9]. Translated to the language of reinforcement learning, *reward shaping* refers to the idea that at a state transition some additional reward $F(s,s')$ may be

provided by design, as to improve the speed of learning [31]. Earlier work has clearly emphasized the importance of motivation and incentives in CIS [32–34]. There are many known means of incentivisation for CIS [35, 36]. Successful examples include monetary incentives (e. g. the Netflix Prize [37]) and non-monetary rewards such as access to webpages (e. g. reCAPTCHA [38]) and gamification (e. g. the "ESP game" [39]).

2.3 The Ant Colony System

We introduce the ACS as a heuristic solution to the TSP following [17]. First we define the TSP. Let $G = (V, E)$ be a weighted complete graph where the set of vertices $V = \{v_1, \dots, v_n\}$ is interpreted as a list of cities and the set of edges $E = \{(v_i, v_j) : v_i, v_j \in V\}$ contains connections between any two cities. The weight of an edge, $w(v_i, v_j)$ denotes the travelling distance between cities v_i and v_j. A Hamiltonian cycle in G is a closed path that visits each vertex exactly once. We interpret this as a salesman visiting each city in the list and call the Hamiltonian cycle a *tour*. Since G by definition is a complete graph there are $(n-1)!/2$ tours. The TSP is the problem of finding the tour for which the sum of weights, i. e. the total distance travelled, is minimal.

The ACS goes about the TSP as follows: Every edge gets assigned a level of pheromones $\tau(v_i, v_j)$ which is interpreted as a measure of how desirable using that edge in a tour is. τ is initialized at some τ_0 for each $(v_i, v_j) \in$ E. A number m of artificial agents, the so called ants, are placed on random vertices. Each ant l keeps track of its unvisited vertices U_l. The simulation proceeds in discrete timesteps. In each timestep, every ant l chooses the next city to visit according to the following rule:

Let $\eta(v_i, v_j)$ be $1/w(v_i, v_j)$ and $v_{i,l}$ be the current location of ant l. The new location $v_{j,l}$ is given by

$$v_{j,l} = \begin{cases} \arg\max_{v_j \in U_l} \left\{ \tau(v_{i,l}, v_j) \eta(v_{i,l}, v_j)^{\beta} \right\} & q \leq q_0 \\ S \quad otherwise \end{cases} \tag{2}$$

In the first case the ant *exploits* the pheromone information of how desirable an edge is, weighted against the length of this edge. β is a parameter that allows to manipulate the relative importance of the two factors. The exploiting move is chosen with probability q_0, i. e. q is a random number. The second case is the case where the ant *explores* other options. S is a city chosen at random according to the following distribution:

$$p_l(v_j) = \begin{cases} \frac{\tau(v_{i,l}, v_j) \eta(v_{i,l}, v_j)^{\beta}}{\sum_{v_k \in U_l} \tau(v_{i,l}, v_k) \eta(v_{i,l}, v_k)^{\beta}} & v_j \in U_l \\ 0 \quad otherwise \end{cases} \tag{3}$$

When an ant transitions from $v_{i,l}$ to $v_{j,l}$ the pheromone trail $\tau(v_{i,l}, v_{j,l})$ is updated according to the swarm metaphor with the following update rule:

$$\tau(v_{i,l}, v_{j,l}) \leftarrow (1-\alpha)\tau(v_{i,l}, v_{j,l}) + \alpha\tau_0 \quad (4)$$

This is called the *local* update. α is a parameter that models the decay of pheromones over time.

When all ants have finished their respective tours, the shortest tour observed so far is determined. Let L_g be the length of this tour and $T_g \subset E$ be the edges on it. A *global* update of pheromone concentration is performed so that:

$$\tau(v_i, v_j) \leftarrow (1-\alpha)\tau(v_i, v_j) + \alpha\tau_\Delta(v_i, v_j) \quad (5)$$

$$\tau_\Delta(v_i, v_j) = \begin{cases} 1/L_g & (v_i, v_j) \in T_g \\ 0 & otherwise \end{cases} \quad (6)$$

After the global update the whole procedure is repeated until an end condition, e. g. a fixed number of iterations has been reached [17].

This concludes the description of the ACS in terms of the swarm metaphor, using pheromones as the central concept. Alternatively it is possible to apply reinforcement learning to the ACS for the TSP. Agents map to ants, states map to current locations, choosing an action maps to choosing the next city to visit and Q generally maps to τ. There even is exploration and exploitation. However, the updating mechanism for τ in the ACS in eq. (4) differs from the Q-learning update rule in eq. (1). AntQ [19] is a hybrid algorithm that draws on both Q-learning and the swarm metaphor. AntQ equals the ACS but with (1) instead of (4) as the local update rule. In fact AntQ predates the ACS, and rule (4) was a later simplification [17].

3 Dissecting the ACS

3.1 Setting

In the following, we use the term *global state* of the ACS to refer to the $n \times n$ matrix that contains the values of τ for all edges. This is not to be confused with the *local state* an ant finds itself in, i. e. a current location and a list of unvisited locations. Ants estimate the local value of adjacent states using the distance to this state and the pheromone density on the way. One way to understand this, is to assume that an ant located at city $v_{i,l}$ can only *observe a subset* of the global state: the pheromones on edges adjacent to $v_{i,l}$. When moving to another location, the ant updates both its local state and the global state. The list of unvisited locations can be interpreted as the ant's working memory. It can never be observed by other ants. This is consistent with the model of CI introduced in [22]. The working memory is used only to guarantee a feasible solution, i. e. that no city is visited more than once. It is not used to make the choice which particular city should be visited next [17]. Note that this is a major limitation with respect to reinforcement learning and the TSP. In theory, the value of a (local) state-action pair represents the desirability of adding the respective edge to a tour. This value obviously depends on the position of the edge within the tour. Not being allowed to use its list of unvisited cities when estimating state-action values deprives the ant of potentially valuable information. In practice, there are $2^n - 1$ pos-

sibilities for the list of unvisited cities so making this list part of the local state rather than working memory as a separate entity would lead to a combinatorial explosion of the state space.

The power of the ACS stems from the fact that it produces a heuristic solution by considering a much smaller search space. This is the principle that makes the swarm metaphor attractive. Real ants, while solving the complex task of pathfinding, have no conception of the problem they solve. SI emerges from individuals of very limited intelligence following simple rules [12]. In the ACS setting, we use the term *emergence* to refer to the idea that individual ants evaluate the state, i. e. the pheromone trails, locally in terms of whether they should travel along a given edge. The local evaluation criterion is fundamentally different from the global value which we define as the length of a tour induced by the τ-matrix. Note that most local values refer to edges that are not part of the final tour and therefore have no direct connection to the global value. Since we are interested in emergence, we measure the ACS performance not in terms of how good it solves the TSP as an optimization problem, but in terms of the construction of an emergent solution. For global evaluation, we do not keep track of a best tour that might have been encountered by some "lucky" ant or a best global state the system might have inhibited at some point in time. Instead we measure the length of the tour induced by τ using the rule in eq. (2) with pure exploitation, i. e. $q_0 = 1$, after a fixed number of 500 iterations. The reasoning behind this measurement is first, that within the swarm metaphor an ant would not realize if it had hit a "lucky" tour and second, that, strictly speaking, emergence implies convergence to a stable state [40]. A system that deviates from an earlier superior state does not really exhibit emergence.

We generated ten random instances of the TSP. For each instance we choose 64 random points on a 256×256 grid and use the pairwise Euclidean distances as edge weights.

3.2 Baseline ACS

As a baseline, we implement the ACS from [17] with only a few minor changes. We use the parameter settings reported therein: $\beta = 2, q_0 = 0.9, \alpha = 0.1, \tau_0 = \frac{1}{nL_{nn}}$,where L_{nn} is the length of the tour generated by the nearest neighbor heuristic. We call the tour length emerging from the ACS, according to the measurement described above, L_{acs}. Note that, due to the way the ACS is initialized, the nearest neighbor heuristic is a special case of the ACS where the evaluation takes place after zero iterations, i. e. without running the ACS at all. In the following we will judge the performance of variations of the ACS relative to L_{nn} and L_{acs}. A variation performing worse than L_{nn} is one that actually evolved the system away from the solution state. We do not make comparisons with fundamentally different solution methods, e. g. branch-and-cut.

Earlier work reports simulating $m = 10$ ants over 2500 iterations [17]. However, we observe little to no improvement in the majority of iterations. Overall, we observe better results with $m = 50$ and 500 iterations. Note that the number of tours generated

is equal in both cases (25000), only the number of global updates is smaller. We discuss this in more detail later in this paper. One final change we have made, is that all the ants start at the same city and build their tours sequentially rather than concurrently. This is necessary to allow a fair comparison of all variations we implement. The resulting algorithm is given below:

```
Initialize τ to τ_0 and t to 0
while (t < 500)
  for each ant
    while |U_l| > 0
      select u_{j,l} according to eq. (2)
      update τ according to eq. (4)
      remove u_{j,l} from U_l
  globally update τ according to eq. (5)
  increment t
return length of tour induced by τ and eq. (2) with q_0 = 1
```

The ACS outperforms the nearest neighbor heuristic on all 10 instances of the TSP under consideration. On average, the relative tour length after 500 iterations of the ACS amounts to 82.4% of the tour length of the nearest neighbor heuristic.

3.3 Differences between Global and Local Updates

The ACS features two update mechanisms – cf. eq. (4) and (5) – which seem similar but actually work differently. The role of the global update is to keep the system's state near the best tour found so far. The role of the local update in contrast is to encourage ants to explore paths not considered by their peers [17]. The first point is easy to see. Disregarding local updates, given enough global updates the only edges with non-zero pheromone concentration are edges on the globally best tour. In that case, our measure of emergent performance becomes equal to measuring the length of the best individual tour. In terms of emergence, the global update uses information that is not available to individual ants. There is no meaningful way to attribute the global update to the actions of an individual ant, especially since the globally best tour could have been generated several iterations ago. Therefore, in terms of reinforcement learning, the global update should not be interpreted as an act of individual learning but rather as a change of the environment with the goal of guiding the system in the right direction, i. e. as reward shaping. Within the swarm metaphor there is no such conception. It is clear that natural environments cannot exhibit this kind of adaption. So at this point the ACS already goes beyond the framing of the swarm metaphor.
The role of local updates is less intuitive. In contrast to global updates, for local updates the "pheromone decay" only takes place on edges that are actually visited by an ant, cf. eq. (4). Note that the local reward signal τ_0 is smaller than the global reward signal $1/L_g$ by approximately a factor n. This leads to the phenomenon that most often the local update actually *decreases* the pheromone concentration on the respective edge. This ensures the diversity of the ant's respective tours and encourages ants

to explore new tours that are similar but different compared to the best currently known tour [17]. While this appears to be a plausible mechanism from an optimization point-of-view, it also breaks with the swarm metaphor. In the biological model, the net impact of laying pheromones is always positive in comparison to the decay of pheromones over time. This is because in the biological model, the latter effect is independent of the presence of an ant. Simulating the ACS, we observe a decrease of pheromone concentration at roughly 90% of all local updates. Coincidently, $m = 50$, a value selected empirically for good overall performance, is close to $n = 64$ so that, due to the factor n in τ_0, the sum of all local updates in an iteration has approximately the same impact as the global update. This is coherent with results from [17].

To further investigate the interaction of global and local rewards we consider three variations of the ACS. First we implement AntQ, i. e. we use eq. (1) instead of eq. (4) for the local update. Be T_l with length L_l the tour constructed by the current ant. We reinforce the final edge of T_l with the additional reward $1/L_l$ which becomes known only at the end of the tour. In theory, Q-learning should lead to this reward propagating back along tours thus shaping the local reward to follow the same principle as the global reward $1/L_g$. The reasoning behind this is simple: if the ants could learn to estimate the length of a tour at its outset, no longer would there be any need to discourage edges already visited by peers. On the contrary, every new tour would make an improvement towards estimating the true value of edges. In simulation, this variation of AntQ still outperforms the nearest neighbor heuristic on all instances but only at 88.2% of the average tour length. AntQ outperforms the ACS on one instance by a very small margin, but on average results in a tour length of 107% of the ACS.

The comparatively poor performance of AntQ suggests that information about the length of particular tours is of little value as a reward signal. Making local and global reward signals constant allows to estimate the value of this information. Therefore, as a second variation we test what we call the global-τ_0 ACS. We use τ_0 instead of $\frac{1}{L_g}$ in eq. (6), thus shaping the global reward to match the local reward. In a third variation, global-L_{nn} we use $\frac{1}{L_{nn}}$ instead, i. e. a constant value in the same magnitude as the original $\frac{1}{L_g}$. These variations allow us not only to investigate the impact of the relative magnitude of local and global rewards on the system's performance, but also the effect of shaping the global reward to match the local reward. In the simulation of four of the ten instances, global-τ_0 displays a performance inferior to the nearest neighbor heuristic. The average relative tour length is 96.7%, making global-τ_0 performance only slightly better than nearest neighbor and clearly worse than the ACS. In contrast, global-L_{nn} outperforms the ACS on two instances although, like AntQ, only by a small margin. On average, the performance of global-L_{nn} diverges from ACS performance by less than 1%.

3.4 Rewarding Global Progress

So far, we followed the swarm metaphor in viewing pheromone concentration as a measure of how desirable it is to follow a given edge from an ant's local perspective.

This needs to be understood as conceptually different from viewing system wide pheromone concentration as a basis for the global evaluation of the system's state. As we have pointed out, this dichotomy is a sign of the system's emergent properties. However, it is up to discussion, and ultimately the CIS designer's choice, what makes an edge desirable from the local perspective. In a general CIS with human agents, from the individual's perspective, there may be intrinsic factors determining the desirability of state-action pairs such as the perceived individual value of states and the individual cost of performing an action [26]. However, the CIS designer should seek to create incentives that let the individuals desire locally what lets the system make progress globally [9]. In the simulation of the ACS, this problem is greatly simplified, compared to a CIS with human agents, because we can shape the reward to any structure we want, and the individual ants are hardcoded to blindly adhere to any given measure of desirability. Nevertheless, the choice of a reward structure is not trivial. Our results from the previous section show, that rewarding ants locally with respect to the global ontology of tour lengths does not improve global performance. Empirically, it seems to be of very little value, to encode information on the tour length, as an indicator of progress, into the reward signals, on both, the global and the local level.

To further investigate this point, we consider the possibility of rewarding progress towards the global goal more directly. Wolpert and Tumer have shown that one way to implement successful CI is to define the local value of a state as the same state's global value minus a hypothetical value that would have occurred, had the individual not existed. They call this "wonderful life utility" (WLU) [10]. We adapt this conception to the ACS as follows: In every iteration, we calculate not only the global value of the pheromone matrix but also for each ant the value of the pheromone matrix without the pheromones added by that ant in the current iteration. The difference of the two values, is a measure of the ant's direct contribution, positive or negative, to the current global value. However, neither a strategy of shaping the AntQ-local reward to reflect this difference nor the strategy to reinforce a positive contribution in retrospective with a reward of $1/L_l$ display a better performance than the nearest neighbor heuristic.

Note that WLU, in the implementation described above, does not measure indirect contributions of a given ant to the global value. Other ants, having access to the pheromone traces laid by the given ant, change their behavior accordingly but this happens after the performance of the given ant has been measured and rewarded. While this is most likely the reason for the poor performance of an otherwise theoretically sound measurement of state-value, the "real" WLU is impractical to measure, because indirect effects extend indefinitely into the future. Every newly generated tour has the effect of branching the system into two possible futures, one with this tour and one without it. At every point, both branches would need to be tracked to measure indirect effects, creating an exponential number of branches. According to [10], calculating WLU is a "fictional counter-factual operation" that ignores the system's dynamics. In case of the ACS, this is done by cancelling out the effect of a given ant on the state's value. However, this is of little use, if most of the effect arises precisely from the system's dynamics and is therefore not independently measurable. In addition to this conceptual problem, designing a CIS with human agents on the basis of WLU, is to

ignore any intrinsic state-action values and superimpose an external system of values on the individuals. This is likely to be an inefficient and costly approach.

3.5 The Value of Information

Given the problems with the WLU approach and the poor performance of seemingly sophisticated rewarding schemes in simulation, we explore a different approach. Both reward schemes, AntQ and WLU, seek to include information on the problem into rewards. Accepting this to be a reasonable idea, we reinvestigate the simpler ACS reward scheme with respect to the information conveyed. From the discussion in Section 3.3, it follows that the informational content of global reward essentially is "this edge is part of the best currently known tour". The content of local reward is "this edge is part of one of your peer's current tour". The way in which ants balance this information and act upon it is hardcoded into the ACS. In terms of WLU, if indirect contributions are to be taken into account, the impact of a single ant on the system is not as much it's construction of a new tour but rather broadcasting this tour to the other ants. It follows that WLU in the ACS amounts to a measure of the global value of the information that ant l took tour T_l in the current iteration.

In absence of a way to approximate this value directly, we approach the problem by asking, under what circumstances an agent individually would be inclined to share this information. Note that the ACS assumes that the ants fully cooperate, i. e. share every tour with the other ants. Deactivating this mechanism completely lets the ACS become non-functional [17]. However there is another way of modelling non-cooperative ants: let the ants keep track of pheromone concentrations but only for tours they constructed themselves, i. e. every ant has a private pheromone matrix. The global reward also is awarded individually, i. e. for the globally best tour that is known to a particular ant. For fair comparison with the ACS, global reward is scaled by $1/n$ so that a single ant's local reward may still counterweight the global reward. The tour lengths produced by the non-cooperative ACS register between ACS tour length and nearest neighbor tour length for all instances. On average the tour length is 105.5% of ACS tour length and 86.9% of nearest neighbor tour length thus indicating that the global value of cooperation is rather small in the given scenario but the global value of keeping track of previous tours is vital to performance.

We investigate two alternative modes of cooperation. In the first mode we allow the agents, as the term ants no longer seems appropriate, to form permanent coalitions. If two agents or two groups of agents join forces, they gain access to a common pheromone matrix, a common best tour and all tours generated in the future by any member of the coalition. The results so far indicate that getting access to future tour information is always a benefit for both sides. Economically it can be argued that having more agents exploiting the best tour creates additional welfare that can be split among the coalition in a way that creates incentives to join for both sides, e. g. using Shapley-value. However, it directly follows from the reasoning above that there is no meaningful way for agents or groups of agents to compare the value of their private pheromone matrices. Therefore we allow ants to join forces only if their respective pheromone matrices currently induce the same tour, i. e. both parties can be sure that

they do not lose an advantage to the other party. Simulation shows that these preconditions for cooperation occur frequently and the system quickly converges to the grand coalition, with a performance almost equal to the ACS.
As a second mode of cooperation we consider non-permanent cooperation and apply a slightly different reasoning. An agent may estimate the (local) value of a newly generated tour by comparing its length to that of the previously generated tour. If the new tour is longer than the old one the agent has made a discovery that is of no future value to itself, e. g. in terms of a competitive advantage, but has some value to other agents because they can avoid making the same mistake. Therefore, from an individual perspective the agent can reasonably choose to "sell" the newly generated tour. This has the effect of allowing the agents to make local updates with selected external tours. The simulation results in a performance level between the non-cooperative variation and the variation with coalitions at an average of 102.2% of the ACS tour length.

4 Discussion and Conclusion

Our findings show, that the ACS conceptually goes beyond the swarm metaphor. Where real ants follow a simple mechanism of laying and following pheromone tracks, we have shown, that the pheromones in the ACS are used to convey different types of information, thus making for more complex semantics of rewards. However, the results also show that not all information is equally valuable in terms of the resulting global performance. Another aspect where the ACS goes beyond the swarm metaphor is the global update in itself. We have shown that the global update can be viewed as the doing of what we might call a "benevolent" environment, actively guiding the individuals towards global success.

Table 1. Overview of simulation results

	Average tour length	
Variation	*relative to L_{nn} [%]*	*relative to L_{ACS} [%]*
ACS	82.4	100.0
AntQ	88.2	107.0
global-τ_0	96.7	117.4
global-L_{nn}	82.9	100.6
WLU	>100%	-
non-cooperative	86.9	105.5
coalitions	82.7	100.4
information-selling	84.2	102.2

We made changes to the ACS, not with the goal to improve its overall performance but seeking to investigate what factors are responsible for performance. Therefore, it comes to no surprise that none of the variations tested, was able to outperform the original ACS on average. An overview of all simulation results reported is given in Table 1. Variations, which leave the original relationship between global and local

rewards intact, tend to display performance similar to the original, whereas reward schemes that change this relationship, no matter how well-motivated in theory, lead to inferior performance. We view this as evidence that the balance of global and local rewards as proposed in [17] and confirmed by our own findings, is a central feature for the success of the ACS.

Obviously, the result from [17] that modelling non-cooperative settings by making ants blind to pheromones ruins performance still holds. In addition to that, we have explored new ways of interpreting non-cooperation and limited cooperation within the ACS with respect to the agents' willingness to share information based on a prior estimate of its value. We have shown that rational reasons to share information can be found. Agents acting according to these reasons, while not completely matching the performance of an ACS with total information, still perform relatively well.

Earlier work, as discussed in the first part of this paper, basically has two guidelines to offer to the CIS designer. Follow the principles of the swarm metaphor [12] and incentivize desired behavior [36]. However, in spite of many successful CIS, it is often unclear how to apply these guidelines to a given problem. There are many examples for sophisticated ways to get collectives to fulfil the desired task, but the task itself, although often presented in a novel context, remains otherwise unchanged. This is only of limited use for wicked problems. For such problems, it is unclear, which course of actions would best fit the task at hand and therefore unclear which behavior should be incentivized. To mitigate this problem, CIS designers should not only tailor problem-specific reward schemes taking into account both global and local utility but also design the system to present the problem in a form that is accessible and interesting for the collective, even if this leads to a complete reframing [9]

Our investigation of the ACS adds empirical evidence to these conceptions, suggesting that systems built on swarm principles are similar whether they solve optimization problems or wicked problems. But there is more. The rewarding scheme of the ACS is shown to have a dual function of encouraging stability and diversity. Rewards are not given on the basis of a global value as WLU might suggest, but as a direct consequence of the cumulated earlier actions of the collective. It shows that a simple reward mechanism that adapts as the system progresses as to guide individual action towards the global solution allows for a truly emergent solution that not only takes into account the limited processing capacity of individuals – which seems to be more of a problem for ants than for humans – but also the fundamental difficulty to provide meaningful local feedback in the same ontology in which global value is measured. Rewarding schemes with such a fundamentally dynamical component are yet to be tested on human CI.

In addition, the pheromone mechanism, or more generally the incentive structure of the ACS makes it easy and attractive for agents to share relevant information thus allowing for cooperation and minimization of redundant work, even if this is not enforced by design. In terms of individual interests, the modified ACS creates an environment in which the gain of sharing information massively outweighs the loss. Not only have we discussed rewarding principles of SI, but also have done this while considering agents' individual interests, thereby making a first step towards what we might call "humanizing the swarm".

References

1 Schoder, D., Gloor, P.A., Metaxas, P.T.: Social Media and Collective Intelligence—Ongoing and Future Research Streams. Künstl. Intell. 27, 9–15 (2013)
2 Schoder, D., Putzke, J., Metaxas, P.T., Gloor, P.A., Fischbach, K.: Informationssysteme für „Wicked Problems". Forschung an der Schnittstelle von Social Media und Collective Intelligence. Wirtschaftsinf 56, 3–11 (2014)
3 Howe, J.: The Rise of Crowdsourcing, http://www.wired.com/wired/archive/14.06/crowds.html
4 Chesbrough, H.W.: Open innovation. The new imperative for creating and profiting from technology. Harvard Business School Press, Boston, Mass (2003)
5 Tapscott, D., Williams, A.D.: Wikinomics. How mass collaboration changes everything. Portfolio, New York (2006)
6 Bonabeau, E.: Decisions 2.0. The Power of Collective Intelligence. MIT Sloan Manage. Rev. 50, 45-52 (2009)
7 Buecheler, T., Sieg, J.H., Füchslin, R.M., Pfeifer, R.: Crowdsourcing, Open Innovation and Collective Intelligence in the Scientific Method - A Research Agenda and Operational Framework. In: Fellermann, H., Dörr, M., Hanczy, M.M., Laursen, L.L., Maurer, S., Merkle, D., Monnard, P.-A., Støy, K., Rasmussen, S. (eds.) Artificial Life XII, pp. 679–686. MIT Press, Cambridge, MA (2010)
8 Surowiecki, J.: The wisdom of crowds. Why the many are smarter than the few and how collective wisdom shapes business, economies, societies, and nations. Doubleday, New York (2005)
9 Kornrumpf, A., Baumöl, U.: A Design Science Approach to Collective Intelligence Systems. 2014 47th Hawaii International Conference on System Sciences, 361–370 (2014)
10 Wolpert, D.H., Tumer, K.: An Introduction to Collective Intelligence (1999)
11 Sulis, W.: Fundamental Concepts of Collective Intelligence. Nonlinear Dynamics, Psychology, and Life Sciences 1, 35-53 (1997)
12 Bonabeau, E., Meyer, C.: Swarm Intelligence. A whole new way to think about business. Harvard Bus. Rev. 79, 106-114 (2001)
13 Gloor, P.A.: Swarm creativity. Competitive advantage through collaborative innovation networks. Oxford University Press, Oxford, New York (2006)
14 Gloor, P., Cooper, S.: The new principles of a swarm business. MIT Sloan Manage. Rev. 48, 81-84 (2007)
15 Kazadi, S., Lee, J.: Swarm Economics. In: Ao, S.-I., Rieger, B., Chen, S.-S. (eds.) Advances in Computational Algorithms and Data Analysis, 14, pp. 249–278. Springer (2009)
16 Krause, J., Ruxton, G.D., Krause, S.: Swarm intelligence in animals and humans. Trends. Ecol. Evol. 25, 28–34 (2010)
17 Dorigo, M., Gambardella, L.M.: Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. Trans. Evol. Comp 1, 53–66 (1997)
18 Dorigo, M., Stützle, T.: The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances. In: Glover, F., Kochenberger, G.A. (eds.) Handbook of Metaheuristics, 57, pp. 250–285. New York (2003)
19 Gambardella, L.M., Dorigo, M.: Ant-Q: A Reinforcement Learning approach to the traveling salesman problem. In: Proceedings of Twelfth International Conference on Machine Learning, pp. 252–260. Morgan Kaufmann (1995)
20 Lykourentzou, I., Vergados, D.J., Loumos, V.: Collective Intelligence System Engineering. In: Proceedings of the International Conference on Management of Emergent Digital EcoSystems, pp. 134-140. ACM, New York, NY (2009)

21 Heylighen, F.: Collective Intelligence and its Implementation on the Web: Algorithms to Develop a Collective Mental Map. Comput. Math. Organ. Theory 5, 253-280 (1999)
22 Kornrumpf, A., Baumöl, U.: From Collective Intelligence to Collective Intelligence Sytems: Definitions and a Semi-Structured Model. Int. J. Coop. Info. Syst. 22 (2013)
23 Deneubourg, J.-L., Aron, S., Goss, S., Pasteels, J.M.: The Self-Organizing Exploratory Pattern of the Argentine Ant. Journal of Insect Behaviour 3, 159–168 (1990)
24 Beckers, R., Deneubourg, J.-L., Goss, S.: Trails and U-turns in the Selection of a Path by the Ant Lasius niger. J. Theor. Biol., 397–415 (1992)
25 Introne, J., Laubacher, R., Olson, G., Malone, T.: Solving Wicked Social Problems with Socio-computational Systems. Künstl Intell 27, 45-52 (2013)
26 Kornrumpf, A., Baumöl, U.: Towards a Model for Collective Intelligence, Emergence and Individual Motivation in the Web 2.0. In: Mattfeld, D.C., Robra-Bissantz S. (eds.) Multikonferenz Wirtschaftsinformatik 2012, pp. 1809–1820. Berlin (2012)
27 Rittel, H.W.J., Webber, M.M.: Dilemmas in a general theory of planning. Policy Sci 4, 155-169 (1973)
28 Kornrumpf, A., Baumöl, U.: On the State of the Art in Simulation of Collective Intelligence – A Literature Review. In: Kundisch, D., Suhl, L., Beckmann, L. (eds.) Multikonferenz Wirtschaftsinformatik 2014, pp. 1746–1760. Paderborn (2014)
29 Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction. The MIT Press, Cambridge, MA (1998)
30 Watkins, C.J., Dayan, P.: Q-learning. Machine Learning 8, 279–292 (1992)
31 Devlin, S., Kudenko, D.: Theoretical Considerations of Potential-Based Reward Shaping for Multi-Agent Systems. In: Proceedings of The Tenth AAMAS, pp. 225–232 (2011)
32 Leimeister, J.M., Huber, M., Bretschneider, U., Krcmar, H.: Leveraging Crowdsourcing: Activation-Supporting Components for IT-Based Ideas Competition. J. Manage. Inf. Syst 26, 197-224 (2009)
33 Malone, T.W., Laubacher, R., Dellarocas, C.N.: The Collective Intelligence Genome. MIT Sloan Manage. Rev. 51, 21-30 (2010)
34 Quinn, A.J., Bederson, B.B.: Human computation: a survey and taxonomy of a growing field. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1403–1412. ACM, Vancouver, BC, Canada (2011)
35 Chamberlain, J., Kruschwitz, U., Poesio, M.: Motivations for Participation in Socially Networked Collective Intelligence Systems. In: Malone, T.W., Ahn, L. von (eds.) Proceedings CI 2012, vol. abs/1204.4071vol. (2012)
36 Scekic, O., Truong, H.-L., Dustdar, S.: Incentives and rewarding in social computing. Commun. ACM 56, 72–82 (2013)
37 Villarroel, J.A., Taylor, J.E., Tucci, C.L.: Innovation and learning performance implications of free revealing and knowledge brokering in competing communities: insights from the Netflix Prize challenge. Comput. Math. Organ. Theory 19, 42–77 (2013)
38 Ahn, L. von, Maurer, B., McMillen, C., Abraham, D., Blum, M.: reCAPTCHA: Human-Based Character Recognition via Web Security Measures. Science 321, 1465–1468 (2008)
39 Ahn, L. von, Dabbish, L.: Labeling images with a computer game. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 319–326. ACM, Vienna, Austria (2004)
40 Müller, J.-P.: Emergence of Collective Behaviour and Problem Solving. In: Omicini, A., Petta, P., Pitt, J. (eds.) Engineering Societies in the Agents World IV, 3071, pp. 1–20. Springer, Berlin Heidelberg (2004)