Evolutionary Dynamics of Ant Colony Optimization

Haitham Bou Ammar, Karl Tuyls, and Michael Kaisers

Maastricht University, P.O. Box 616, 6200 MD Maastricht, The Netherlands

Abstract. Swarm intelligence has been successfully applied in various domains, e.g., path planning, resource allocation and data mining. Despite its wide use, a theoretical framework in which the behavior of swarm intelligence can be formally understood is still lacking. This article starts by formally deriving the evolutionary dynamics of ant colony optimization, an important swarm intelligence algorithm. We then continue to formally link these to reinforcement learning. Specifically, we show that the attained evolutionary dynamics are equivalent to the dynamics of Q-learning. Both algorithms are equivalent to a dynamical system known as the replicator dynamics in the domain of evolutionary game theory. In conclusion, the process of improvement described by the replicator dynamics appears to be a fundamental principle which drives processes in swarm intelligence, evolution, and learning.

1 Introduction

Artificial intelligence (AI) is a wide spread field. The applications of AI range from designing intelligent systems that are capable of dealing with un-anticipated changes in the environment to modeling interactions in biological and sociological populations. The success of learned behavior for any particular task greatly depends on the way the agent was designed and whether it suits that specific domain. In constructing artificial intelligence agents, designers have to choose from a wide range of available AI frameworks such as: swarm intelligence, reinforcement learning, and evolutionary game theory. The best choice depends on the application the agent is trying to tackle. For example, swarm intelligence is well suited for domains that require a robust behavior of the population of agents (i.e., if one agent fails the system still has to achieve its required goal). The merits of swarm intelligence have been exploited in applications like path planning, resource allocation and data mining [5, 13, 23, 27]. On the other hand, reinforcement learning algorithms best fit agents that individually need to maximize a possibly delayed feedback signal from the environment. As an illustration, reinforcement learning can best fit agents that need to master chess playing [19], play soccer robotics and control [14, 18] et cetera.

Both swarm intelligence and reinforcement learning have been explored empirically with considerable success, see for example [19, 16, 3, 10]. Reinforcement learning has been thoroughly discussed in single-agent environments, for which proofs of convergence have been given, e.g., Q-learning [26]. In multi-agent settings, theoretical analysis poses additional challenges such as a non Markovian environment from the agent's point of view. Theoretical guarantees have been difficult to achieve, but [1, 21] laid the foundation for a new approach, linking multi-agent reinforcement learning with evolutionary game theory. Recently, this link has been used to provide convergence guarantees for multi-agent Q-learning [11, 12]. In swarm intelligence, there has been much less progress in formalizing the dynamics of the domain or linking them to other learning techniques.

In this paper we provide a formal derivation of Ant Colony Optimization (ACO), a well known swarm intelligence algorithm. Interestingly, this mathematical derivation reveals that swarm intelligence and reinforcement learning are intrinsically equivalent, both following the replicator dynamics from evolutionary game theory. This is in line with the conceptual equivalence conjectured in [**?**]. More specifically, this related work has established conceptual and empirical evidence for the similarity of ACO and a network of Learning Automata. This network of learning automata is formally related to the replicator dynamics [1] that form the basis of the formal equivalence established in this article. Here, it becomes clear that whether swarm intelligence, or reinforcement learning was involved in a specific learning task, there seems to be a common process that drives the improvement of the agents behavior, namely the replicator dynamics.

The remainder of the paper is organized as follows: Section 2 introduces concepts from swarm intelligence, reinforcement learning and evolutionary game theory that are the basis of the upcoming analysis. We proceed to introduce our mathematical formalization of swarm intelligence in Section 3. Section 4, builds on the attained derivation to formally link swarm intelligence and reinforcement learning. Section 5 presents related work. Section 6 concludes and reflects upon various future research directions.

2 Background

This section introduces the main concepts from swarm intelligence, reinforcement learning, and evolutionary game theory that this article is based on. Specifically, ACO and the update rule are first discussed. Then Q-learning, a specific reinforcement learning algorithm, as well as the replicator dynamics are presented.

2.1 Ant Colony Optimization (ACO)

Ant Colony Optimization, which is widely used in swarm intelligence, is a class of algorithms that takes inspiration from the foraging behavior of certain ant species. These ants deposit pheromones to indicate favorable routes that should be followed by other ants in the colony. ACO exploits similar biologically inspired behaviors in order to solve optimization problems. Since the ACO's introduction, there has been several variations to the original optimization algorithm. Here we limit our focus to Ant Systems (AS), the original ACO algorithm [2, 4]. The main characteristic of AS is that all the pheromone levels are updated by all ants that have built a solution at the current iteration.

Typically in ACO the problem is defined by a graph $G = (V, E)$, with V being the set vertices, encoding states of the environment, and E representing the edges of the graph, representing state transitions. An example of such a graphical model is shown in Figure 1, whereby each of the edges admits a certain pheromone level $\tau_{i,j}$. The ants in such a scenario start at the initial vertex i and collectively find the shortest path to the goal vertex g. The learning behavior for such a system is better explained through

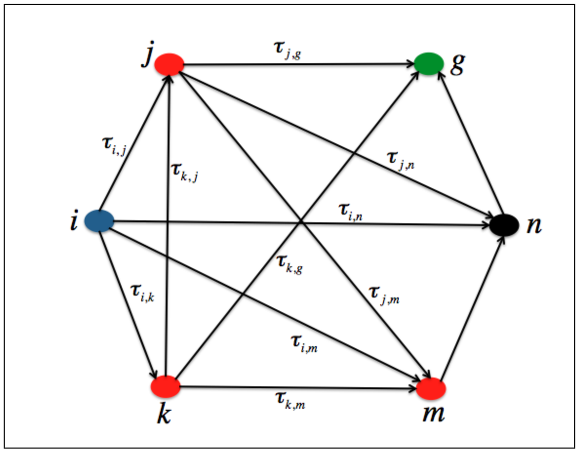

Fig. 1. Graph illustration of a domain in which Ant Colony Optimization can find the shortest path from i to g.

Algorithm 1. Learning starts by initializing the pheromone levels at each of the edges either randomly or uniformly. The ants transition from a certain node on the graph to another using a stochastic transition probability derived from the pheromones:

$$p_{i,j} = \frac{\tau_{i,j}^{\alpha}\eta_{i,j}^{\beta}}{\sum_c \tau_{i,c}^{\alpha}\eta_{i,c}^{\beta}} \tag{1}$$

where α and β are parameters that present the tradeoff between the pheromone levels $\tau_{i,j}$ and the heuristic information $\eta_{i,j}$[1]. Typically, $\eta_{i,j}$ is defined as inversely proportional to the distance between the nodes i and j.

Algorithm 1 Ant Colony Optimization Metaheuristic

```
Initialize the pheromone levels $\tau_{i,j} \leftarrow \tau_0 \; \forall (i,j) \in E$
Choose the starting vertex on the graph $v_{start} = i \in V$
for iteration $t = 1, 2 \ldots$ do
    for ant $m = 1, 2, \ldots$ do
        $v_{pos} = v_{start}$ and $u = \{\}$
        while $\exists$ feasible continuation $(v_{pos}, j)$ of $u$ do
            Select $j$ according to $p_{i,j}$, where
            $p_{i,j} = \begin{cases} 0 & \textbf{if (i,j) infeasible} \\ \frac{\tau_{i,j}^{\alpha}\eta_{i,j}^{\beta}}{\sum_c \tau_{i,c}^{\alpha}\eta_{i,c}^{\beta}} & \textbf{otherwise} \end{cases}$
            $u = u \oplus (v_{pos}, j)$ and $v_{pos} = j$
    Update the pheromone trails $\tau_{i,j}$ using Equation 2.
```

[1] It is worth noting that $\eta_{i,j}$ plays a role similar to reward shaping in reinforcement learning.

Assuming there exists a feasible path to the goal, or more formally assuming that the graph is an ergodic set if the goal state is removed, then all ants will be absorbed in that goal state given sufficient time. After reaching the goal state the pheromone trail of that path is updated according to the following Equation [3] :

$$\tau_{i,j}(t+1) = (1-\rho)\tau_{i,j}(t) + \sum_{m=1}^{M} \delta_{i,j}(t,m), \tag{2}$$

where $\tau_{i,j}$ represents the pheromone level at the edge (i,j), ρ denotes the pheromone evaporation rate, M is the number of ants and $\delta_{i,j}(t,m) = Q\frac{n_{i,j}}{L}$ with Q being a constant, $n_{i,j}$ being the number of times edge (i,j) has been visited[2] by the m^{th} ant and L being the total length of the m^{th} ant's trajectory[3]. It may be convenient to choose $Q = \frac{1}{M}$ to maintain a bounded sum for an arbitrary number of ants.

2.2 Q-learning

Q-learning [26] is a reinforcement learning algorithm that is designed to maximize a discounted sum of future rewards encountered by an agent interacting with an environment. Originally, Q-learning was used in a single-agent learning setting, where the learning process is markovian, i.e., the current state and action are considered to be sufficient statistics to determine the successor state and the reward.

By definition, the Q-learner repeatedly interacts with its environment, performing an action i at time t, and receiving reward $r_i(t)$ in return. It maintains an estimation $Q_i(t)$ of the expected discounted reward for each action i. This estimation is iteratively updated according to the following equation, known as the Q-learning update rule, where α denotes the learning rate and γ is the discount factor:

$$Q_i(t+1) \leftarrow Q_i(t) + \alpha \left(r_i(t) + \gamma \max_j Q_j(t) - Q_i(t) \right) \tag{3}$$

2.3 Evolutionary Game Theory

Evolutionary game theory takes a rather descriptive perspective, replacing hyper-rationality from classical game theory by the concept of natural selection from biology [17]. It studies the population development of individuals belonging to one of several species. The two central concepts of evolutionary game theory are the replicator dynamics and evolutionary stable strategies [20]. The replicator dynamics presented in the next subsection describe the evolutionary change in the population. They are a set of differential equations that are derived from biological operators such as selection, mutation and cross-over. The evolutionary stable strategies describe the possible asymptotic behavior

[2] Note that in some related work a visited edge may only be updated once [3]. In contrast, our model corresponds to an ant that dispenses a fixed pheromone amount per step, and may thus update a link for every time it has been visited.

[3] Please note that the number of trajectories and the number of ants are the same under the presented assumptions.

of the population. However, their examination is beyond the scope of this article. For a detailed discussion, we refer the interested reader to [9].

Consider a population comprised of several species, where each species i makes up a fraction x_i of the population $x = (x_1, x_2, \ldots, x_n)$. The replicator dynamics represent a set of differential equations that formally describe the population's change over time. A population comprises a set of individuals, where the species that an individual can belong to relate to pure actions available to a learner. The Darwinian fitness f_i of each species i can be interpreted as the expectation of the utility function $r_i(t)$ that assigns a reward to the performed action. The distribution of the individuals on the different strategies can be described by a probability vector that is equivalent to a policy for one player, i.e., there is one population in every agents *mind*. The evolutionary pressure by natural selection can be modeled by the replicator equations. They assume this population to evolve such that successful strategies with higher payoffs than average grow while less successful ones decay. The general form relates the time derivative $\dot{x}$ to the relative fitness of species i.

$$\dot{x}_i = x_i \left[f_i(x) - \sum_k x_k f_k(x) \right], \text{where } f_i = E\left[r_i(t)|x\right] \tag{4}$$

In two-player asymmetric games, the fitness functions can be computed from the payoff bi-matrix (A, B), where e_i is the i^{th} unit vector:

$$\begin{aligned} \dot{x}_i &= x_i \left[e_i A y - x A y\right] \\ \dot{y}_j &= y_j \left[x B e_j - x B y\right] \end{aligned} \tag{5}$$

where x_i,y_i are the probabilities of a player picking action i and j respectively, x,y are the action probability vectors for each of the players and A,B are matrices representing the payoff, e.g., $f_i(x) = E\left[r_i(t)|y\right] = e_i A y$ for the first player. Equation 5 clearly quantifies the selection scheme of a strategy i as being the difference between the attained payoff of that same strategy, i.e., $(Ay)_i$ compared to the average payoff over all other strategies $x^T A y$.

These dynamics are formally connected to reinforcement learning [1, 21, 22]. For the ease of readability we leave this derivation for later sections. Namely, we present these equations in Section 4 once deriving the relation between ACO and reinforcement learning.

3 Mathematical Derivation

In this section we will provide the mathematical derivations involved in determining the theoretical link between swarm intelligence and evolutionary game theory.

3.1 Assumptions and General Framework

We examine the theoretical behavior of the Ant Colony Optimization (ACO) algorithm given an infinite number of ants. In order to bound the pheromone level, we assume that each ant dispenses an amount which is anti-proportional to the number of ants (i.e., $Q = \frac{1}{M}$). Note that the total trajectory length L equals $\sum_{b,c} n_{b,c}$. Under these mild assumptions, each pheromone update iteration can be described as follows:

$$\tau_{i,j}(t+1) \leftarrow (1-\rho)\tau_{i,j}(t) + \lim_{M\to\infty} \sum_{m=1}^{M} \frac{1}{M} \frac{n_{i,j}}{\sum_{b,c} n_{b,c}}, \tag{6}$$

where $n_{i,j}$ and $n_{b,c}$ present the number of visits to the edges (i,j) and (b,c) respectively. The limit is used to denote that we are interested in the behavior of such a system as the number of ants grows to infinity.

ACO Replicator Dynamics Here we will derive the replicator dynamics of ACO and reflect upon the technicalities involved.

We commence by examining the variation in the transition probability for an ant being at a certain state i and moving to a state j. Taking the derivative of Equation 1 we get :

$$\begin{aligned}
\dot{p}_{i,j} &= \left(\frac{\tau_{i,j}^{\alpha} \eta^{\beta}}{\sum_c \tau_{i,c}^{\alpha} \eta_{i,c}^{\beta}} \right)' \\
&= \frac{\alpha \tau_{i,j}^{\alpha-1} \dot{\tau_{i,j}} \eta_{i,j}^{\beta}}{\sum_c \tau_{i,c}^{\alpha} \eta_{i,c}^{\beta}} - \frac{\tau_{i,j}^{\alpha} \eta_{i,j}^{\beta} \sum_c \alpha \tau_{i,c}^{\alpha-1} \dot{\tau_{i,c}} \eta_{i,c}^{\beta}}{\left(\sum_c \tau_{i,c}^{\alpha} \eta_{i,c}^{\beta} \right)^2} \\
&= \alpha p_{i,j} \frac{\dot{\tau}_{i,j}}{\tau_{i,j}} - \alpha p_{i,j} \sum_c \frac{\dot{\tau_{i,c}}}{\tau_{i,c}} p_{i,c} \\
&= \alpha p_{i,j} \left(\frac{\dot{\tau_{i,j}}}{\tau_{i,j}} - \sum_c \frac{\dot{\tau_{i,c}}}{\tau_{i,c}} p_{i,c} \right)
\end{aligned} \tag{7}$$

Equation 7 clearly resembles the general replicator dynamics given in Equation 4 when written in the following form, where $\Theta_{i,j} = \frac{\dot{\tau_{i,j}}}{\tau_{i,j}}$:

$$\dot{p}_{i,j} = \alpha p_{i,j} \left(\Theta_{i,j} - \sum_k p_{i,k} \Theta_{i,k} \right) \tag{8}$$

The attained model of ACO in Equation 8, quantifies the change in the probability of choosing an action j in a given state i and therefore, represents a generalization of the stateless replicator dynamics from Equation 4 to multiple-state games. This discussion is further detailed in Section 4. Equation 8 also clearly represents the change in the transition probabilities as a function of the level of the pheromone updates.

3.2 Rate of Change in the Pheromone Level

Next we will determine the rate of change in the pheromone level which is vital for solving Equation 8. Consider again the update rule given in Equation 6. Using the central limit theorem this is re-written in the following format :

$$
\begin{aligned}
\tau_{i,j}(t+1) &= (1-\rho)\tau_{i,j}(t) + \mathbb{E}\left(\frac{n_{i,j}}{\sum_{b,c} n_{b,c}}\right) \\
\mathit{\Delta}\tau_{i,j}(t) &= -\rho\tau_{i,j}(t) + \mathbb{E}\left(\frac{n_{i,j}}{\sum_{b,c} n_{b,c}}\right) \\
\frac{\mathit{\Delta}\tau_{i,j}(t)}{\tau_{i,j}(t)} &= -\rho + \frac{\mathbb{E}\left(\frac{n_{i,j}}{\sum_{b,c} n_{b,c}}\right)}{\tau_{i,j}(t)} \qquad (9)
\end{aligned}
$$

Considering infinitesimal time changes, Equation 9 can be written as :

$$
\frac{\dot{\tau_{i,j}}(t)}{\tau_{i,j}(t)} = -\rho + \frac{\mathbb{E}\left(\frac{n_{i,j}}{\sum_{b,c} n_{b,c}}\right)}{\tau_{i,j}(t)} \tag{10}
$$

Next we will provide a formal method to determine the expectation of Equation 10. This formalization is based on the Markov Chain theory that will be introduced.

Markov Chain Theory A Markov Chain (MC) is a discrete time random process that satisfies the Markov property. An MC typically involves a sequence of random variables that evolve through time according to a stochastic transition probability. In our current formalization we are more interested in an *Absorbing Markov Chain* (AMC). An AMC is a variation of the former to include an Absorbing state. In other words, the system has a state that once reach can never be left, therefore called absorbing.

In an AMC with t transient states and r absorbing state the probability matrix could be written in the canonical format of Equation 11,

$$
\mathbf{P} = \left[\begin{array}{c|c} \mathbf{Q} & \mathbf{R} \\ \hline \mathbf{0} & \mathbf{I} \end{array}\right], \tag{11}
$$

where $\mathbf{Q} \in \mathbb{R}^{t\times t}$ is the transient state probability matrix, $\mathbf{R} \in \mathbb{R}^{t\times r}$is the absorbing transition probability matrix, $\mathbf{0} \in \mathbb{R}^{r\times t}$ zero matrix, and $\mathbf{I} \in \mathbb{R}^{r\times r}$ identity matrix.

The expected number of visits to a transient state is determined using, $\mathbf{N} = \sum_{k=0}^{\infty} \mathbf{Q}^k = (\mathbf{I} - \mathbf{Q})^{-1}$, while the expected transient probabilities of visiting all the transient states is given by $\mathbf{H} = (\mathbf{N} - \mathbf{I})\mathbf{D}^{-1}$, where $\mathbf{D} \in \mathbb{R}^{t\times t}$ is a diagonal matrix having the same diagonal entries as $\mathbf{N}$. The probability of an absorbing transition to occur is given by $\mathbf{B} = \mathbf{NR}$ with $\mathbf{B} \in \mathbb{R}^{t\times r}$.

Equation 12 represents the expectation using the AMC theory. We have divided a trajectory into three essential parts: (1) a transition from an initial state to a certain state i, (2) a transition between two transient states (i, j) and (3) a transition probability from a transient state to the goal or end state. These three parts present the possible combinations of an ant to start at a certain initial state and reach the required destination.

Using the AMC chain theory and the trajectory division idea, we can re-write in expectation. Many terms appear with powers related to a, b, α, and β, with a being the number of visits to a certain link (i,j), b represents the length of a trajectory, $(\mathbf{Q}^b\mathbf{R})_{end}$ signifies the probability of a transient state to be absorbed, $\mathbf{Q}^{\alpha}_{1,i}$ is the expected probability to transient from an initial to an i^{th} state, $\mathbf{Q}^{\beta_1}_{j,i}\dots\mathbf{Q}^{\beta_n}_{j,i}$ are all the expected probabilities of looping between states (i,j), and $\psi = b - a - \alpha - \sum_{i<j}\beta_i$, representing the remaining division to get absorbed.

$$\mathbb{E}\left(\frac{n_{i,j}}{\sum_{b,c} n_{b,c}}\right) = \sum_{b=1}^{\infty}\sum_{a=0}^{b}\frac{a}{b}$$
$$\left[(\mathbf{Q}^b\mathbf{R})_{end}\sum_{\alpha=0}^{b-a}\sum_{\beta_j\in\{1,\dots,n\}}^{\psi}\mathbf{Q}^{\alpha}_{1,i}\mathbf{Q}^{a}_{i,j}\mathbf{Q}^{\beta_1}_{j,i}\dots\mathbf{Q}^{\beta_n}_{j,i}\mathbf{Q}^{b-a-\alpha-\sum_i\beta_i}_{j,end}\right] \tag{12}$$

The attained results could now be substituted back in Equation 8, with $\Theta_{i,j}$ being:

$$\Theta_{i,j} = -\rho + \frac{\sum_{b=1}^{\infty}\sum_{a=0}^{b}\frac{a}{b}\left[\boldsymbol{\chi}\sum_{\alpha=0}^{b-a}\sum_{\beta_j\in\{1,\dots,n\}}^{\psi}\boldsymbol{\Gamma}\boldsymbol{\Xi}\boldsymbol{\Lambda}\right]}{\tau_{i,j}} \tag{13}$$

with $\boldsymbol{\chi} = (\mathbf{Q}^b R)_{end}$, $\boldsymbol{\Xi} = \mathbf{Q}^{\beta_1}_{j,i}\dots\mathbf{Q}^{\beta_n}_{j,i}$, $\boldsymbol{\Gamma} = \mathbf{Q}^{\alpha}_{1,i}\mathbf{Q}^{a}_{i,j}$, and $\boldsymbol{\Lambda} = \mathbf{Q}^{b-a-\alpha-\sum_i\beta_i}_{j,end}$. Substituting these in Equation 8 we get the full ACO model as :

$$\frac{\dot{p}_{i,j}}{p_{i,j}} = -\alpha\rho + \alpha\frac{\sum_{b=1}^{\infty}\sum_{a=0}^{b}\frac{a}{b}\left[\boldsymbol{\chi}\sum_{\alpha=0}^{b-a}\sum_{\beta_j\in\{1,\dots,n\}}^{\psi}\boldsymbol{\Gamma}\boldsymbol{\Xi}\boldsymbol{\Lambda}\right]}{\tau_{i,j}} \tag{14}$$
$$-\alpha\sum_{l}p_{i,l}\left(\rho + \frac{\sum_{b=1}^{\infty}\sum_{a=0}^{b}\frac{a}{b}\left[\boldsymbol{\chi_l}\sum_{\alpha=0}^{b-a}\sum_{\beta_j\in\{1,\dots,n\}}^{\psi}\boldsymbol{\Gamma_l}\boldsymbol{\Xi_l}\boldsymbol{\Lambda_l}\right]}{\tau_{i,l}}\right)$$

where the subscript l denotes each of the matrices depending on the required edge.

4 Relation to reinforcement learning

In this section we will draw the connection between the attained model of ACO and reinforcement learning. Namely, we start by presenting the dynamics of Q-learning and then linking them to the attained ACO equations.

4.1 Q-learning Dynamics

In [21], the authors extend the work of [1] to derive the dynamics of Q-learning. Consider the policy of two players to be represented as probability vectors $x = (x_1, \dots, x_k)$ and $y = (y_1, \dots, y_z)$, where x_i indicates the probability for player one to choose an action i, k presents the k^{th} allowed action for player one, y_j manifests the probability of the second player to pick action j, and z is the z^{th} allowed action for player two.

Based on these definitions the dynamics of Q-learning in a two-player stateless matrix game are derived as,

$$\begin{aligned}\dot{x}_i &= x_i\alpha\left(\tau^{-1}\dot{Q}_i - \sum_k \dot{Q}_k x_k\right)\\ \dot{y}_j &= y_j\alpha\left(\tau^{-1}\dot{Q}_j - \sum_z \dot{Q}_z x_z\right)\end{aligned} \tag{15}$$

Taking the update rule of Equation 3 into account, the dynamics of Q-learning could be written as,

$$\begin{aligned}\dot{x}_i &= x_i\alpha\left(\tau^{-1}[e_iAy - xAy] - \log x_i + \sum_k x_k \log x_k\right) \qquad (16)\\ \dot{y}_j &= y_j\alpha\left(\tau^{-1}[xBe_j - xBy] - \log y_j + \sum_z y_z \log z_z\right)\end{aligned}$$

with x,y the policies of each player, α the learning rate, τ temperature parameter, A, B the payoff matrices, and e_i the i^{th} unit vector. The most interesting part of this results is that the previous equations contain a selection part that is equivalent to the replicator dynamics and a mutation part. For more details interested readers are referred to [21].

With these equations the dynamics of Q-learning in two player stateless games could be better understood and analyzed. As we will discuss later in the document, our formulation of ACO and its relation to reinforcement learning requires an extension of the above equations to multi-state games. Some work on the generalization of the replicator dynamics to multiple-states in [8] had been achieved. But for a better understanding of the replicator dynamics of ACO we require a more general framework that explains the dynamics of more coupled states. This has not been achieved yet, but we hope that our ACO formulation could give a better insight on the analysis of complex forms multi-state games and present a rigorous starting point for more complex tasks.

4.2 Ant Colonies as Reinforcement Learning Agents

In analyzing this relation it is important to note that here we look at the dynamical model of ACO from a slightly different perspective where $p_{i,j}$ now denotes the probability of an agent to choose an action j at a certain state i. This view is intuitively equivalent to the previous where $p_{i,j}$ was manifesting the probability for an ant to visit an edge (i, j).

The dynamical model in Equation 8 clearly resembles close connections to the replicator equations of Q-learning in Equation 15. Both models have similar time derivatives representing the change in the probability of choosing a certain action by a player. On one hand, in reinforcement learning this probability, x_i in Equation 15, depends on the difference between the Q values for choosing the action i compared to that of picking all the others, i.e., $\dot{Q}_i - \sum_k \dot{Q}_k x_k$. On the other hand, in ACO the probability of choosing an action j in a state i, $p_{i,j}$ in Equation 8, depends on both the change in the pheromone level for $\dot{\tau}_{i,j}$ and on the original pheromone level $\tau_{i,j}$, i.e., $\frac{\dot{\tau_{i,j}}}{\tau_{i,j}} - \sum_c \frac{\dot{\tau_{i,c}}}{\tau_{i,c}} p_{i,c}$. The above observation reflects that the solution to the probability of a certain action in ACO lies in a higher dimensional space than that of the pheromone space and resembles additional complexities to Q-learning.

Another important observation is to been seen from a *state-locality* point of view. If each state was viewed as an agent by its own, then at this locality each agent, equivalently a state, admits its own replicator dynamics that are described in Equation 8. In such a setting the relation between swarm intelligence and reinforcement learning becomes apparent and clear. Using this *state-locality* view, Equation 8 could be re-written in the following format for each state-agent (i, j):

$$\dot{p}_i = \alpha p_i \left(\Theta_i - \sum_l p_l \Theta_l \right) \tag{17}$$

Here the dynamics of ACO clearly resemble those of Q-learning whereby Θ_i plays the same role as that of $\dot{Q}_i$ in Equation 15. At this stage, it is worth noting two important characteristics of the attained model: (1) The pheromone levels in ACO play the same role to the Q values, i.e., the expected total payoff in a reinforcement learning agent and (2) The variation of the action selection probabilities in ACO resemble more complexities compared to normal form Q-learning.[4] It is important to mention, that although the swarm intelligence and reinforcement learning seem to admit similar local behaviors the analysis of the dynamics of ACO is more complex from the global point of view as these states-agents are dynamically coupled.

A high-level pictorial view of this observation is shown in Figure 2. Under the state-agent equivalence view each state resembles its own replicator equations. The switching dynamics between each of the state-agent in ACO are beyond the scope of this paper and are left for further studying in the future work. A good starting point here is the work of [25], whereby the authors provided the switching dynamics for piece-wise linear systems. It is worth noting though, that this work still needs to be extended to suit the ACO model, as piece-wise linear behavioral models are not a suitable assumption in this case.

5 Related Work

Ant Colony Optimization has been shown to work empirically in various domains. Theoretically there have been various successes in providing convergence proofs for the ACO algorithms, see [3, 6] for a thorough discussion. In the realm of modeling the dynamics of ACO less research has been done so far. For example [15] model the dynamical properties of the ACO algorithm using a deterministic model that assumes an average expected behavior of the ants. The model is best suited for a specific class of permutation problems. Another attempt to formalize the dynamics of ACO was provided in [7]. Gutjahr provides an analytical analysis of the finite-time dynamics of ACO

[4] Although this is not investigated in this paper, we speculate that ACO relates to Q-learning with eligibility traces which will be studied in deeper details in our future work.

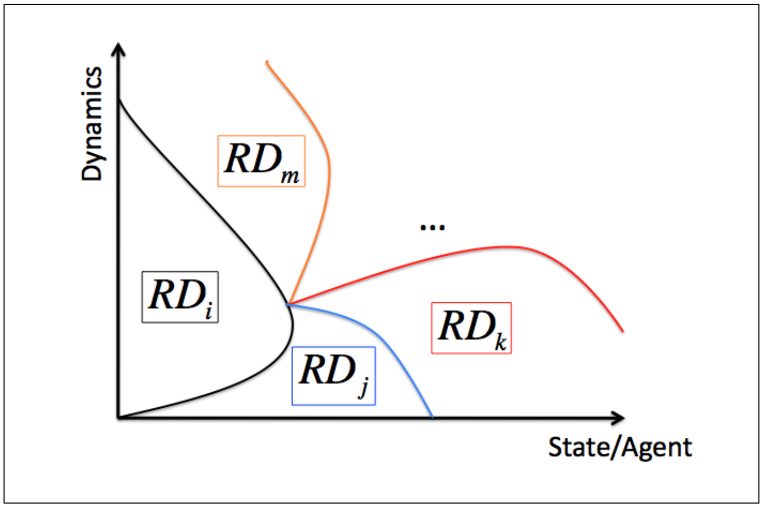

Fig. 2. A high level pictorial illustration of the dynamics of ACO at a local state level where RD_i denoted the replicator dynamics for a certain state-agent i. From the state-agent equivalent view, each state, is described by its own replicator differential equations dynamics.

based on a fitness-proportional pheromone update rule on arbitrary construction graphs. In [24], the authors attempt to formalize the dynamics of ACO based on an average reward Markov Decision Process model. It uses a very simple pheromone update rule ignoring the heuristics used in normal ACO.

The presented paper differs from these works by deriving a more general framework that describes the dynamical behavior of ACO which is applicable to a wider range of problems utilizing the full pheromone update rule. We further present, for the first time to the best of our knowledge, a formal link between swarm intelligence, evolutionary game theory and reinforcement learning.

6 Conclusions and Future Work

The contributions of this paper are twofold: (1) We presented a formal model of the dynamics of one of the most famous swarm intelligence algorithms, namely the replicator equations of ant colony optimization. (2) We provide a formal link between three well known learning schemes: swarm intelligence, evolutionary game theory and reinforcement learning. Although the update rules for the behavioral improvement of swarm intelligence and reinforcement learning appear to be very different, formally there is a common underlying process for improvement which is driven by the replicator dynamics.

The contributions of this paper range beyond the scope of linking the three different learning domains together and might serve as a rigorous starting point to the extension of the replicator dynamics beyond two-player stateless games to multi-players multi-state games which is considered to be one of the hardest problems in evolutionary game theory.

The established link provides several opportunities for future work. First, the ant colony optimization replicator dynamics (ACO-RD) provide a model that can be used to analyze the convergence behavior of the stochastic algorithm with tools from dynamical systems. In addition, they can be extended to obtain error bounds on the performance of the stochastic algorithm. Second, these ACO-RD are inherently multi-state, and we expect similarities to the dynamics of learning algorithms such as Q-learning with eligibility traces and bee-inspired swarm intelligence algorithms, to which this theoretical foundation may extend.

References

1. Tilman Börgers and Rajiv Sarin. Learning through reinforcement and replicator dynamics. *Journal of Economic Theory*, 77(1):1–14, 1997.
2. M. Dorigo, A. Colorni, and V. Maniezzo. Positive feedback as a search strategy. Technical Report 91-016, Dipartimento di Elettronica, Politecnico di Milano, Milan, Italy, 1991.
3. Marco Dorigo, Mauro Birattari, and Thomas Sttzle. Ant colony optimization – artificial ants as a computational intelligence technique. *IEEE COMPUT. INTELL. MAG*, 1:28–39, 2006.
4. Marco Dorigo, Vittorio Maniezzo, and Alberto Colorni. The ant system: Optimization by a colony of cooperating agents. *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS-PART B*, 26(1):29–41, 1996.
5. Crina Grosan, Ajith Abraham, and Monica Chis. Swarm intelligence in data mining. In *Swarm Intelligence in Data Mining*, pages 1–20. 2006.
6. Walter J. Gutjahr. A graph-based ant system and its convergence. *Future Gener. Comput. Syst.*, 16:873–888, June 2000.
7. Walter J Gutjahr. On the finite-time dynamics of ant colony optimization. *Methodology and Computing in Applied Probability*, 8(1):105–133, 2006.
8. Daniel Hennes, Karl Tuyls, and Matthias Rauterberg. State-coupled replicator dynamics. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '09, pages 789–796, Richland, SC, 2009. International Foundation for Autonomous Agents and Multiagent Systems.
9. J. Hofbauer. *Evolutionary Games and Population Dynamics*. Cambridge University Press, Cambridge, 1998.
10. Michael Kaisers and Karl Tuyls. *Frequency adjusted multi-agent Q-learning*, page 309316. International Foundation for Autonomous Agents and Multiagent Systems, 2010.
11. Michael Kaisers and Karl Tuyls. FAQ-Learning in Matrix Games: Demonstrating Convergence near Nash Equilibria, and Bifurcation of Attractors in the Battle of Sexes. In *Workshop on Interactive Decision Theory and Game Theory (IDTGT 2011)*. Assoc. for the Advancement of Artif. Intel. (AAAI), 2011.
12. Ardeshir Kianercy and Aram Galstyan. Dynamics of softmax q-learning in two-player two-action games. *CoRR*, abs/1109.1528, 2011.
13. Defang Liu and Bochu Wang. Biological swarm intelligence based opportunistic resource allocation for wireless ad hoc networks. *Wireless Personal Communications*, 2011.
14. S. Mahadevan, J. Connell, C. Sammut, R. Sutton, and Temporal Phd. Automatic programming of behavior-based robots using reinforcement learning, 1991.
15. Daniel Merkle and Martin Middendorf. Modeling the dynamics of ant colony optimization. *Evol. Comput.*, 10:235–262, September 2002.

16. Jan Peters and Stefan Schaal. Reinforcement learning of motor skills with policy gradients, 2008.
17. John Maynard Smith. *Evolution and the Theory of Games*. Cambridge University Press, Cambridge, UK, 1982.
18. Peter Stone, Tucker Balch, and Gerhard Kraetzschmar, editors. *RoboCup-2000: Robot Soccer World Cup IV*, volume 2019 of *Lecture Notes in Artificial Intelligence*. Springer Verlag, Berlin, 2001.
19. Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.
20. Peter D Taylor and Leo B Jonker. Evolutionary stable strategies and game dynamics. *Mathematical Biosciences*, 40(1-2):145–156, 1978.
21. Karl Tuyls, Pieter Jant Hoen, and Bram Vanschoenwinkel. An evolutionary dynamical analysis of multi-agent learning in iterated games. *The Journal of Autonomous Agents and Multi-Agent Systems, JAAMAS*, 12(1):115 – 153, 2006.
22. Karl Tuyls and Simon Parsons. What evolutionary game theory tells us about multiagent learning. *Artificial Intelligence*, 171(7):406–416, 2007.
23. Christopher M. Vigorito. Distributed path planning for mobile robots using a swarm of interacting reinforcement learners. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, AAMAS '07, pages 120:1–120:8, New York, NY, USA, 2007. ACM.
24. P. Vrancx, K. Verbeeck, and A. Nowé. Networks of learning automata and limiting games. *Lecture Notes in Artificial Intelligence, ALAMAS III*, 4865:224–238, 2008.
25. Peter Vrancx, Karl Tuyls, and Ronald L. Westra. Switching dynamics of multi-agent learning. In *AAMAS (1)*, pages 307–313, 2008.
26. Christopher J. C. H. Watkins and Peter Dayan. Technical note q-learning. *Machine Learning*, 8:279–292, 1992.
27. Jing Zhou, Guanzhong Dai, De-Quan He, Jun Ma, and Xiao-Yan Cai. Swarm intelligence: Ant-based robot path planning. In *Proceedings of the Fifth International Conference on Information Assurance and Security, IAS 2009, Xi An, China, 18-20 August 2009*, pages 459–463. IEEE Computer Society, 2009.