DiSIT, Computer Science Institute
Università del Piemonte Orientale "A. Avogadro"
Viale Teresa Michel 11, 15121 Alessandria
http://www.di.unipmn.it


**An Intelligent Swarm of Markovian Agents**

*A. Bobbio, B. Dario , C. Davide, M. Gribaudo, S. Marco*
*(andrea.bobbio@mfn.unipmn.it, dbruneo@unime.it, cerotti@elet.polimi.it, gribaudo@elet.polimi.it, mscarpa@unime.it)*

TECHNICAL REPORT TR-INF-2014-06-01-UNIPMN
*(June 2014)*

Research Technical Reports published by DiSIT, Computer Science Institute, Università del Piemonte Orientale are available via WWW at URL `http://www.di.unipmn.it/`. Plain-text abstracts organized by year are available in the directory

## Recent Titles from the TR-INF-UNIPMN Technical Report Series

2013-01 *Minimum pattern length for short spaced seeds based on linear rulers (revised)*, L. Egidi, G. Manzini, July 2013.

2012-04 *An intensional approach for periodic data in relational databases*, A. Bottrighi, A. Sattar, B. Stantic, P. Terenziani, December 2012.

2012-03 *Minimum pattern length for short spaced seeds based on linear rulers*, L. Egidi, G. Manzini, April 2012.

2012-02 *Exploiting VM Migration for the Automated Power and Performance Management of Green Cloud Computing Systems*, C. Anglano, M. Canonico, M. Guazzone, April 2012.

2012-01 *Trace retrieval and clustering for business process monitoring*, G. Leonardi, S. Montani, March 2012.

2011-04 *Achieving completeness in bounded model checking of action theories in ASP*, L. Giordano, A. Martelli, D. Theseider Dupré, December 2011.

2011-03 *SAN models of a benchmark on dynamic reliability*, D. Codetta Raiteri, December 2011.

2011-02 *A new symbolic approach for network reliability analysis* , M. Beccuti, S. Donatelli, G. Franceschinis, R. Terruggia, June 2011.

2011-01 *Spaced Seeds Design Using Perfect Rulers*, L. Egidi, G. Manzini, June 2011.

2010-04 *ARPHA: an FDIR architecture for Autonomous Spacecrafts based on Dynamic Probabilistic Graphical Models*, D. Codetta Raiteri, L. Portinale, December 2010.

2010-03 *ICCBR 2010 Workshop Proceedings*, C. Marling, June 2010.

2010-02 *Verifying Business Process Compliance by Reasoning about Actions*, D. D'Aprile, L. Giordano, V. Gliozzi, A. Martelli, G. Pozzato, D. Theseider Dupré, May 2010.

2010-01 *A Case-based Approach to Business Process Monitoring*, G. Leonardi, S. Montani, March 2010.

2009-09 *Supporting Human Interaction and Human Resources Coordination in Distributed Clinical Guidelines*, A. Bottrighi, G. Molino, S. Montani, P. Terenziani, M. Torchio, December 2009.

2009-08 *Simulating the communication of commands and signals in a distribution grid*, D. Codetta Raiteri, R. Nai, December 2009.

2009-07 *A temporal relational data model for proposals and evaluations of updates*, L. Anselma, A. Bottrighi, S. Montani, P. Terenziani, September 2009.

# An Intelligent Swarm of Markovian Agents

Andrea Bobbio [1], Dario Bruneo [3], Davide Cerotti [2],
Marco Gribaudo [2], Marco Scarpa [3]

[1] DiSIT, Università del Piemonte Orientale, Alessandria, Italy
Department of Computer Science, Indian Institute of Technology IITGN, Gandhinagar, India
[2] Dipartimento di Elettronica e Informazione, Politecnico di Milano, Milano, Italy
[3] Dipartimento di Matematica, Università degli Studi di Messina
andrea.bobbio@unipmn.it, {cerotti,gribaudo}@elet.polimi.it,
{dbruneo,mscarpa}@unime.it

## Abstract

We define a Markovian Agent Model (MAM) as an analytical model formed by a spatial collection of interacting Markovian Agents (MAs), whose properties and behavior can be evaluated by numerical techniques. MAMs have been introduced with the aim of providing a flexible and scalable framework for distributed systems of interacting objects, where both the local properties and the interactions may depend on the geographical position. MAMs can be proposed to model biological inspired systems since are suited to cope with the four common principles that govern swarm intelligence: positive feedback, negative feedback, randomness, multiple interactions. In the present work, we report some results of a MAM model for WSN routing protocol based on swarm intelligence, and some preliminary results in utilizing MAs for very basic ACO benchmarks.

## 1 Swarm Intelligence: a modeling perspective

Swarm intelligent (SI) algorithms are variously inspired from the way in which colonies of biological organisms self organize to produce a wide diversity of functions [1, 2]. Individuals of the colony have a limited knowledge of the overall behavior of the system and follow a small set of rules that may be influenced by the interaction with other individuals or by modifications produced in the environment. The collective behavior of large groups of relatively simple individuals, interacting only locally with few neighboring elements, produces global patterns. Even if many approaches have been proposed that differentiate in many respects, four basic common principles have been isolated that govern SI.

◇ *Positive feedback*
◇ *Negative feedback*
◇ *Randomness*
◇ *Multiple interactions*

The same four principles govern also a class of algorithms inspired by the expansion dynamics of slime molds in the search for food [3, 4], that have been utilized as the base for the generation of routing protocols in Wireless Sensor Networks (WSN).

Through the adoption of the above four principles, it is possible to design distributed, self-organizing, and fault tolerant algorithms able to self-adapt to the environmental changes, that present the following main properties [1]: *i)* Single individuals are assumed to be simple with low computational intelligence and communication capabilities; *ii)* individuals communicate indirectly, through modification of the environment (this property is known as *stigmergy* [2]); *iii)* The range of the interaction may be very short, nevertheless a robust global behavior emerges from the interaction of the nodes; *iv)* The global behavior adapts to topological and environmental changes.

The usual way to study such systems is through simulation, due to the large number of involved individuals that lead to the well-known state explosion problem. Analytical techniques are preferable if, starting from the peculiarities of SI systems, allow to realize effective and scalable models. Along this line, new stochastic entities, called Markovian Agents (MAs) [5, 6], have been introduced with the aim of providing a flexible, powerful, and scalable technique for modeling complex systems of distributed interacting objects, for which feasible analytical and numerical solution algorithms can be implemented. Each object has its own local behavior that can be modified by the mutual interdependencies with the other objects. MAs are scattered over a geographical area and retain their spatial position so that the local behavior and the mutual interdependencies may be related to their geographical positions and other features like the transmittance characteristics of the interposed medium. MAs are modeled by a discrete-state continuous-time finite CTMC whose infinitesimal generator is influenced by the interaction with other MAs. The interaction among agents is represented by a *message passing model* combined with a *perception function*. When resident in a state or during a transition, an MA is allowed to send messages that are perceived by the other MAs, according to a spatial dependent *perception function*, modi-

fying their behavior. Messages may model real physical messages (as in WSN) or simply the mutual influences of an MA over the other ones.

The flexibility of the MA representation, the spatial dependency, and the mutual interaction through message passing and perception function, make MA models suited to cope with various biological inspired mechanisms governed by the four aforementioned principles. In fact, the Markovian Agent Model (MAM), whose constituent elements are the MAs, was specifically studied to cope with the following needs [6]:

*i* Provide analytical models that can be solved by numerical techniques, thus avoiding the need of long and expensive simulation runs;
*ii* Provide a flexible and scalable modeling framework for distributed systems of interacting objects;
*iii* Provide a framework in which local properties can be coupled with global properties;
*iv* Local and global properties and interactions may depend on the position of the objects in the space (space-sensitive models);
*v* The solution algorithm self-adapts to variations in the system topology and in the interaction mechanisms.

Interactive Markovian Agents have been first introduced in [7, 5] for single class MAs and then extended to Multi-class Multi-message Markovian Agent Model in successive works [8–10]. In [9, 11, 12] MAs have been applied to routing algorithms in WSN, adopting SI principles [13].

The present work describes the structure of MAMs and the numerical solution algorithms in Section 2. Then, applications derived from biological models are presented: a swarm intelligent algorithm for routing protocols in WSN (Section 3) and a simple Ant Colony Optimization (ACO) example (Section 4).

## 2 Markovian Agents Models

The structure of a single MA is represented in Figure 1. States $i, j, \ldots, k$ are the states of the CTMC representing the MA. The transitions among the states are of two possible types that are drawn differently:

- Solid lines (like the transition from $i$ to $j$ or the self-loops in $i$ or in $j$) indicate the fixed component of the infinitesimal generator and

represent the local or autonomous behavior of the object that is independent of the interaction with the other MAs (like, for instance, the time to failure distribution, or the reaction to an external stimulus). Note that we include in the representation also self-loop transitions that require a particular notation since are not visible in the infinitesimal generator of the CTMC [14].

- Dashed lines (like the transition from $i$ to $k$ or the transitions into $i$ or $j$) represent the transitions induced by the interaction with the other MAs. The way in which the rates of the induced transitions are computed is explained in the following section.

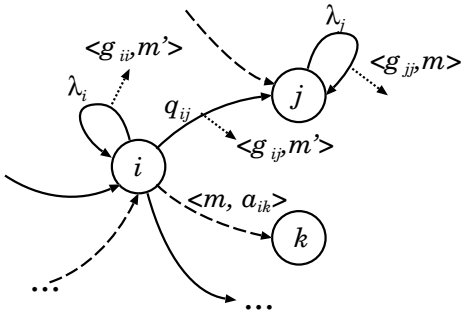



**Fig. 1.** Schematic structure of a Markovian Agent.

During a local transition (or a self-loop) an MA can emit a message of any type with an assigned probability, as represented by the dotted arrows in Figure 1 emerging from the solid transitions. The pair $< g_{ij}, m >$ denotes both the message generation probability and the message type. Messages generated by an MA may be perceived by other MAs with a given probability, according to a suitable perception function, and the interaction mechanism between emitted messages and perceived messages generates the induced transitions (dashed lines). The pair $< m, a_{ik} >$ denotes both the type of the perceived message and the corresponding acceptance probability.

An MAM is a collection of interacting MAs defined over a geographical space $\mathcal{V}$. Given a position $\mathbf{v}$ inside $\mathcal{V}$, we define $\rho(\mathbf{v})$ the density of MAs in $\mathbf{v}$. According to the definition of the density $\rho(\mathbf{v})$, we can classify a MAM with the following taxonomy:

- An MAM is *static* if $\rho(\mathbf{v})$ does not depend on time, and *dynamic* if it does depend on time;
- An MAM is *discrete* if the geographical area on which the MAs are deployed is discretized and $\rho(\mathbf{v})$ is a discrete function of the space or it is *continuous* if $\rho(\mathbf{v})$ is a continuous function of the space.

Further, MAs may belong to a single class or to different classes with different local behaviors and interaction capabilities, and messages may belong to different types where each type induces a different effect on the interaction mechanism. The perception function describes how a message of a given type emitted by an MA of a given class in a given position in the space is perceived by an MA of a given class in a different position.

**Mathematical formulation**

A *Multiple Agent Class, Multiple Message Type* MAM is defined by the tuple [12]:

$$MAM = \{\mathcal{C}, \mathcal{M}, \mathcal{V}, \mathcal{U}, \mathcal{R}\}, \tag{1}$$

where:

$\mathcal{C} = \{1 \dots C\}$ is the set of agent classes. We denote with $MA^c$ an agent of class $c \in \mathcal{C}$.

$\mathcal{M} = \{1 \dots M\}$ is the set of message types. Each agent (independently of its class) can send or receive messages of type $m \in \mathcal{M}$.

$\mathcal{V}$ is the finite space over which Markovian Agents are spread.

$\mathcal{U} = \{u^1(\cdot) \dots u^M(\cdot)\}$ is a set of $M$ perception functions (one for each message type).

$\mathcal{R} = \{\rho^1(\cdot) \dots \rho^C(\cdot)\}$ is a set of $C$ agent density functions (one for each agent class).

Each agent $MA^c$ of class $c$ is characterized by a state space with $n_c$ states, and it is defined by the tuple:

$$MA^c = \{\mathbf{Q^c}(\mathbf{v}), \mathbf{\Lambda}^c(\mathbf{v}), \mathbf{G}^c(\mathbf{v}, m), \mathbf{A}^c(\mathbf{v}, m), \boldsymbol{\pi}_0^c(\mathbf{v})\}. \tag{2}$$

where:

$\mathbf{Q^c}(\mathbf{v})$ is the local component of the infinitesimal generator;

$\mathbf{\Lambda}^c(\mathbf{v})$ is the vector of the self-jump transition rates;

$\mathbf{G}^c(\mathbf{v}, m)$ is the matrix containing the probabilities of generating a message of type $m$;

$\mathbf{A}^c(\mathbf{v}, m)$ is the matrix containing the probabilities of accepting a message of type $m$;

$\boldsymbol{\pi}_0^c(\mathbf{v})$ is the initial probability vector.

Note that even if the structure of the CTMC associated to each MA$^c$ of class $c$ is the same for all the objects, the values of the parameters may depend on position $\mathbf{v}$ and, therefore, may vary from MA$^c$ to MA$^c$.

An $MAM$ can be analyzed solving a set of coupled differential equations. Let us call $\rho_i^c(t, \mathbf{v})$ the density of agents of class $c$, in state $i$, located in position $\mathbf{v}$ at time $t$. In the following, we will focus on static MAMs thus assuming that the total density of agents in position $\mathbf{v}$ remains constant over the time; we have that:

$$\sum_{i=1}^{n_c} \rho_i^c(t, \mathbf{v}) = \rho^c(\mathbf{v}) \ , \quad \forall \mathbf{v}, \forall t \geq 0. \tag{3}$$

We collect the state densities into a vector $\boldsymbol{\rho}^c(t, \mathbf{v}) = [\rho_i^c(t, \mathbf{v})]$ and we are interested in computing the transient evolution of $\boldsymbol{\rho}^c(t, \mathbf{v})$.

From the above definitions, we can compute the total rate $\beta_j^c(\mathbf{v}, m)$ at which messages of type $m$ are generated by an agent of class $c$ in state $j$ in position $\mathbf{v}$:

$$\beta_j^c(\mathbf{v}, m) = \lambda_j^c(\mathbf{v})\, g_{jj}^c(\mathbf{v}, m) + \sum_{k \neq j} q_{jk}^c(\mathbf{v})\, g_{jk}^c(\mathbf{v}, m). \tag{4}$$

where the first term in the r.h.s is the contribution of the messages of type $m$ emitted during a self-loop from $j$ and the second term is the contribution of messages of type $m$ emitted during a transition from $j$ to any $k\,(\neq j)$.

The interdependencies among MA$^c$s are ruled by a set of perception functions whose general form is:

$$u^m(c, \mathbf{v}, i, c', \mathbf{v}', j, ) \tag{5}$$

The perception function $u^m(.)$ in (5) represents how an MA of class $c$ in position $\mathbf{v}$ in state $i$ perceives the messages of type $m$ emitted by an MA of class $c'$ in position $\mathbf{v}'$ in state $j$. The functional form of $u^m(.)$ identifies the perception mechanisms and must be specified for any given application since it determines how an MA is influenced by the messages emitted by the other MAs. The transition rates of the induced transitions are primarily determined by the structure of the perception function.

A pictorial and intuitive representation of how the perception function $u^m(c, \mathbf{v}, i, c', \mathbf{v}', j, )$ acts, is given in Figure 2. The MA in the top right portion of the figure in position $\mathbf{v}'$ broadcasts a message of type $m$ from state $j$ that propagates in the geographical area until reaches the MA in the bottom left portion of the figure in position $\mathbf{v}$ and in state $i$. Upon acceptance of the message according to the acceptance probability $a_{ik}(\mathbf{v}, m)$, a new induced transition from state $i$ to state $k$ (represented by a dashed line) is generated in the model.

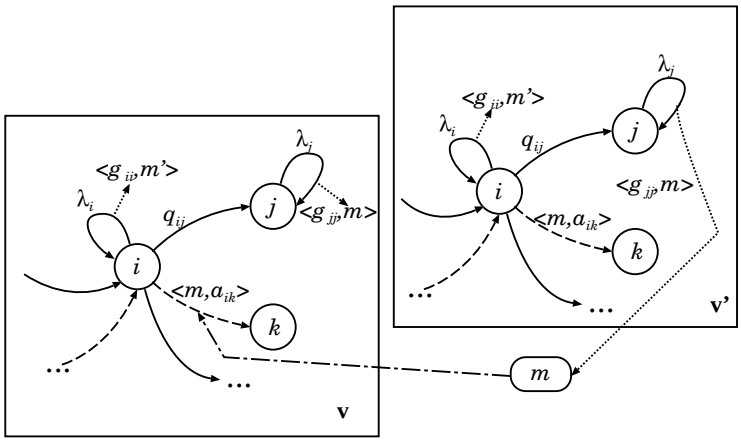



**Fig. 2.** Message passing mechanism ruled by a perception function.

With the above definitions we are now in the position to compute the components of the infinitesimal generator of an MA that depend on the interaction with the other MAs and that constitute the original and innovative part of the approach.

We define $\gamma_{ii}^c(t, \mathbf{v}, m)$ the total rate at which messages of type $m$ coming from the whole volume $\mathcal{V}$ are perceived by an MA in state $i$ in location $\mathbf{v}$.

$$\gamma_{ii}^c(t, \mathbf{v}, m) = \int_{\mathcal{V}} \sum_{c'=1}^{C} \sum_{j=1}^{n_{c'}} u^m(c, \mathbf{v}, i, c', \mathbf{v}', j, ) \beta_j^{c'}(m) \rho_j^{c'}(t, \mathbf{v}') d\mathbf{v}'. \tag{6}$$

$\gamma_{ii}^c(t, \mathbf{v}, m)$ in Equation (6) is computed by taking into account the total rate of messages of type $m$ emitted by all the MAs in state $j$ and in a given position $\mathbf{v}'$ (the term $\beta_j^c(\mathbf{v}, m)$) times the density of MAs in $\mathbf{v}'$ (the term $\rho_j(t, \mathbf{v}')$) times the perception function (the term $u^m(c, \mathbf{v}, i, c', \mathbf{v}', j, )$)

summed over all the possible states $j$ and class $c'$ of each MA and integrated over the whole space $\mathcal{V}$. From an MA in position $\mathbf{v}$ and in state $i$ an induced transition to state $k$ (drawn in dashed line) is generated with rate $\gamma_{ii}^c(t, \mathbf{v}, m)\, a_{ik}(\mathbf{v}, m)$ where $a_{ik}(\mathbf{v}, m)$ is the appropriate entry of the acceptance matrix $\mathbf{A}(\mathbf{v}, m)$.

We collect the rates (6) in a diagonal matrix $\mathbf{\Gamma}^c(t, \mathbf{v}, m) = \text{diag}(\gamma_{ii}^c(t, \mathbf{v}, m))$. This matrix can be used to compute $\mathbf{K}^c(t, \mathbf{v})$, the infinitesimal generator of a class $c$ agent at position $\mathbf{v}$ at time $t$:

$$\mathbf{K}^c(t, \mathbf{v}) = \mathbf{Q}^c + \sum_m \mathbf{\Gamma}^c(t, \mathbf{v}, m) \left[\mathbf{A}^c(m) - \mathbf{I}\right]. \tag{7}$$

The first term in the r.h.s. is the local transition rate matrix and the second term contains the rates induced by the interactions.

The evolution of the entire model can be studied by solving $\forall \mathbf{v}, c$ the following differential equations:

$$\boldsymbol{\rho}^c(0, \mathbf{v}) = \rho^c(\mathbf{v})\boldsymbol{\pi}_0^c \tag{8}$$

$$\frac{d\boldsymbol{\rho}^c(t, \mathbf{v})}{dt} = \boldsymbol{\rho}^c(t, \mathbf{v})\mathbf{K}^c(t, \mathbf{v}). \tag{9}$$

From the density of agents in each state, we can compute the probability of finding a class $c$ agent at time $t$ in position $\mathbf{v}$ in state $i$ as:

$$\pi_i^c(t, \mathbf{v}) = \frac{\rho_i^c(t, \mathbf{v})}{\rho^c(\mathbf{v})}. \tag{10}$$

We collect all the terms in a vector $\boldsymbol{\pi}^c(t, \mathbf{v}) = [\pi_i^c(t, \mathbf{v})]$. Note that the definition of Equation (10) together with Equation (3) ensures that $\sum_i \pi_i^c(t, \mathbf{v}) = 1, \forall t, \forall \mathbf{v}$.

Note that each equation in (9) has the dimension $n_c$ of the CTMC of a single MA of class $c$. In this way, a problem defined over the product state space of all the MAs is decomposed into several subproblems, one for each MA, having decoupled the interaction by means of Equation (6). Equations (9) provide the basic time-dependent measures to evaluate more complex performance indices associated to the system. Equations (9) are discretized both in time and space and are solved by resorting to standard numerical techniques for differential equations.

## 3 A consolidated example: routing in WSN

In this section, we present our first attempt to model swarm intelligence inspired mechanisms through the MAM formalism. This application describes a MAM model to the analysis of a swarm intelligence routing protocol in WSN and was first proposed in [9] and then enriched in [12]. In the present work, we show new experiments to illustrate the self-adaptability of the MAM model to the changing of environmental conditions.

WSN are large networks of tiny sensor nodes that are usually randomly distributed over a geographical region. The network topology may vary in time in an unpredictable manner due to many different causes. For example, in order to reduce power consumption, battery operated sensors undergo cycles of sleeping - active periods; additionally, sensors may be located in hostile environments increasing their likelihood of failure; furthermore, data might also be collected from different sources at different times and directed to different sinks. For this reason, multi-hop routing algorithms used to route messages from a sensor node to a sink should be rapidly adaptable to the changing topology. Swarm intelligence has been successfully used to face these problems thanks to its ability in converging to a single global behavior starting from the interaction of many simple local agents.

### 3.1 A swarm intelligence based routing

In [15] a new routing algorithm, inspired to the biological process of pheromone emission, has been proposed. The routing table in each node stores the "*pheromone level*" owned by each neighbor, coded as a natural integer quantity [15]; when a data packet has to be sent it is forwarded to the neighbor with the highest pheromone level. This approach correctly works only if a sequence of increasing values of pheromone levels towards the sinks exists; in other words, the sinks must have the maximum pheromone level in the WSN and a decreasing pheromone gradient must be established around the sinks covering all the net.

To build the pheromone gradient, the initial setting of the WSN is as follow: the sinks are set to a fixed maximum pheromone level, whereas the sensor nodes' pheromone levels are set to $0$. When the WSN is operating, each node periodically sends a signaling packet with its pheromone level and updates its value based on the level of its neighbors.

More specifically, the algorithm for establishing the pheromone gradient is based on two types of nodes in the WSN, called *sinks* and *sensors* respectively, and the pheromone is assumed discretized into $P$ different levels, ranging from $0$ to $P-1$. In this way, routing paths toward the sink are established through the exchange of pheromone packets containing the pheromone level $p$ $(0 \leq p \leq P-1)$ of each node.

Sink nodes, once activated, set their internal pheromone level to the highest value $p = P-1$. Then, they, at fixed time interval, broadcast a pheromone message to their neighbors with the value $p$. We assume $T1$ is the time interval incurring between two consecutive sending of pheromone message.

Instead, the pheromone level of a sensor node is initially set to $0$ and then it is periodically updated according to two distinct actions: *Excitation action (the positive feedback)* and *Evaporation action (the negative feedback)*.

*Excitation action*: sensor nodes periodically broadcast to the neighbors a pheromone message containing their internal pheromone level $p$. Like the sink node, sensor nodes perform the sending at regular time interval $T1$. When a sensor node receives a pheromone level $p_n$ sent by a neighbor it compares $p_n$ with its own level $p$ and updates the latter if $p_n > p$. The new value is computed as a function of the current and the received pheromone level $update(p, p_n)$. In this context, we use the mean value as the new updating value, thus the function is assumed to be $update(p, p_n) = round((p + p_n)/2)$.

*Evaporation action*: it is triggered at regular time interval $T2$ and it simply decreases the current value of $p$ by one unit assuring it maintains a value greater or equal to $0$.

We note that, despite all nodes perform their excitation action with same fixed time interval $T1$, no synchronization activity is required among the nodes; all of them act asynchronously in accordance with the principles of biological systems where each entity acts autonomously with respect to the others.

The excitation-evaporation process, like in biological systems, assures the stability of the system and the adaptability to possible changes in the environment or in some nodes. In fact any change in the network condition is captured by an update of the pheromone level of the involved nodes that modifies the pheromone gradient automatically driving the routing decisions toward the new optimal solution. In this way, the net-

work can self-organize its topology and adapt to environmental changes. Moreover, when link failures occur, the network reorganization task is accomplished by those nodes near the broken links. This results in a robust and self-organized architecture.

The major drawback of this algorithm is the difficulty in appropriately setting the parameter $T1$ and $T2$; in fact, as shown in [15, 12], the stability of the system and the *quality* of the produced pheromone gradient is strictly dependent on the parameters ratio. When $T1$ decreases and $T2$ is fixed, pheromone messages are exchanged more rapidly among the nodes and their pheromone level tends to the maximum level because the sink node always sends the same maximum level thus, without an appropriate balancing action, the pheromone level saturates all the nodes of the WSN. At the opposite, let us suppose $T1$ is fixed and $T2$ decreases; in this case the pheromone level in each sensor node decreases more quickly than its updating according to the value of the neighbors, as a result all the levels will be close to zero. From this behavior, we note that: 1) both timers are necessary to ensure that the algorithm could properly work, and 2) a smart setting of both timers is necessary in order to have the best gradient shape all over the network.

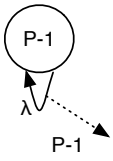
The $MAM$ model we are going to describe in the next section helps in easily setting the best parameter values.

### 3.2 The $MAM$ model

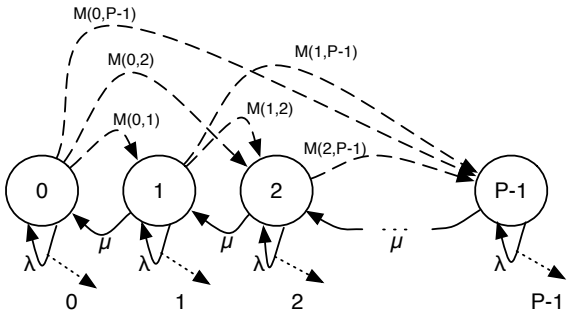
The $MAM$ model used to study the gradient formation is based on two agent classes: the class *sink node* denoted by a superscript $s$ and the class *sensor node* denoted by a superscript $n$. The pheromone intensity is discretized it into $P$ integer levels (ranging from $0$ to $P-1$) in accordance with the algorithm described earlier. The message exchange is modeled by using $M$ different message types. As we will explain later, since each message is used to send a pheromone level, we set $M = P$.

**Geographical space** The geographical space $\mathcal{V}$ where the $N$ agents are located is modeled as a $n_h \times n_w$ rectangular grid, and each cell has a square shape with side $d_s$. Sensors can only be located in the center of each cell and we allow at most one node per cell: i.e., some cell might be empty, and $N \leq n_h \times n_w$. Moreover, sink nodes are very few with respect to the number of sensor nodes.

**Agent's structure and behavior** Irrespective of the MA class considered, we model the pheromone level of a node with a state and this choice determines two different MA structures.




(a) Agent class = sink.




(b) Agent class = sensor.

**Fig. 3.** Markovian agent models.

The *sink* class (Fig. 3(a)) is very simple and is characterized by a single state labeled $P-1$ with a self-loop of rate $\lambda = \frac{1}{T1}$. In fact the sink has always the same maximum pheromone level, and emits a single message of type $P-1$ with rate $\lambda$.

Instead, the *sensor* class (Fig. 3(b)) has $P$ states identifying the range of all the possible pheromone levels. Each state is labeled with the pheromone intensity $i$ $(i = 0, \ldots, P-1)$ in the corresponding node and has a self-loop of rate $\lambda = \frac{1}{T1}$ that represents the firing of timer at regular intervals equal to $T1$. This event causes the sending of a message (see Section 3.2). The evaporation phenomenon is modeled by the solid arcs (local transitions) connecting state $i$ with state $i-1$ $(0 < i \leq P-1)$. The evaporation rate is set to $\mu = \frac{1}{T2}$; in such a way we represent the firing of timer $T2$.

**Message types** The types of messages in the model correspond to the different levels of pheromone a node can store, thus we define $\mathcal{M} = \{0, 1, \ldots, P-1\}$. Any self loop transition in state $i$ emits a message of the corresponding type $i$ at a constant rate $\lambda$, either in sink and in sensor nodes. The sink message is always of type $P-1$, representing the maximum pheromone intensity, whereas the messages emitted by a sensor node corresponds to the state where it actually is.

When a message of type $m$ is emitted neighboring nodes are able to receive it changing their state accordingly. This behavior is implemented through the dashed arcs (whose labels are defined through eq. (11)) that model the transitions induced by the reception of a message. In particular, when a node in state $i$ receives a message of type $m$, it immediately jumps to state $j$ if $m \in M(i,j)$, with:

$$M(i,j) = \{m \in [0 \cdots P-1] : round((m+i)/2) = j\} \quad \forall i,j \in [0 \cdots P-1] : j > i. \tag{11}$$

In other words, an MA in state $i$ jumps to the state $j$ that represents the pheromone level equal to the mean between the current level $i$ and the level $m$ encoded in the perceived message.

**Perception function** Messages of any type sent by a node are characterized by the same transmission range $t_r$ that defines the radius of the area in which an MA can perceive a message produced by another MA. This property is reflected in the perception function $u^m(\cdot)$ that, $\forall m \in [1 \cdots M]$, is defined as:

$$u^m(\mathbf{v}, c, i, \mathbf{v}', c', i') = \begin{cases} 0 \text{ dist}(\mathbf{v}, \mathbf{v}') > t_r \\ 1 \text{ dist}(\mathbf{v}, \mathbf{v}') \leq t_r, \end{cases} \tag{12}$$

where dist$(\mathbf{v}, \mathbf{v}')$ represents the distance between two nodes in position $\mathbf{v}$ and $\mathbf{v}'$.

As can be observed, the perception function in eq. (12) is defined irrespective of the message type, because in this kind of application the reception of a message of any type $i$ depends only on the distance between the emitting and the perceiving node. The transmission range $t_r$ depends on the properties of the sensor and it influences the number $\eta$ of neighbors perceiving the message. In the numerical experimentation, we consider $d_s \leq t_{r4} < \sqrt{2}\, d_s$ corresponding to $\eta = 4$.

**Generation and acceptance probabilities** In this application, messages are only generated during self-loop transitions with probability 1, so that $\forall i, j$, $g^c_{ii}(m) = 1$ and $g^c_{ij}(m) = 0$, $(i \neq j)$. Similarly, we assume either $a^c_{ij}(m) = 0$ or $a^c_{ij}(m) = 1$, that is incoming messages are always accepted or always ignored.

### 3.3 Numerical results

In order to analyze the behavior of the WSN model, the main measure of interest is the evolution of $\pi^n_i(t, \mathbf{v})$ i.e., the distribution of the pheromone intensity of a sensor node over the entire area $\mathcal{V}$ as a function of the time. The value of $\pi^n_i(t, \mathbf{v})$ can be computed from (10) and allows us to obtain several performance indices like the average pheromone intensity $\phi(t, \mathbf{v})$ at time $t$ for each cell $\mathbf{v} \in \mathcal{V}$:

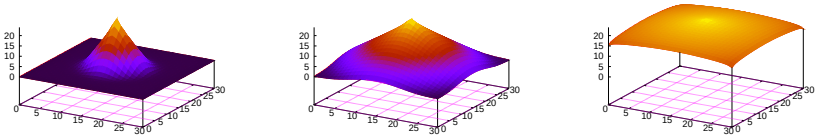
$$\phi(t, \mathbf{v}) = \sum_{i=0}^{P-1} i \cdot \pi^n_i(t, \mathbf{v}). \tag{13}$$

The distribution of the pheromone intensity over the entire area $\mathcal{V}$ depends both on the pheromone emission rate $\lambda$ and on the pheromone evaporation rate $\mu$; furthermore, the excitation-evaporation process depends on the transmission range $t_r$ that determines the number of neighboring cells $\eta$ perceived by an MA in a given position. To take into account this physical mechanism, we define the following quantity:

$$r = \frac{\lambda \cdot \eta}{\mu}, \tag{14}$$

that regulates the balance between the pheromone emission and evaporation in the SI routing algorithm. For a complete discussion about the performance indices that can be derived and analyzed using the described MAM, refer to [12].

The numerical results have been obtained with the following experimental setting. The geographical space is a square grid of sizes $n_h = n_w$ = 31, where $N$ = 961 sensors are uniformly distributed with a spatial density equal to 1 (one sensor per cell). Further, we set $\lambda = 4.0$, $P = 20$, and $\eta = 4$. The first experiment aims at investigating the formation of the pheromone gradient around the sink as a function of the model parameters. To this end, a single sink node is placed in the center of the area



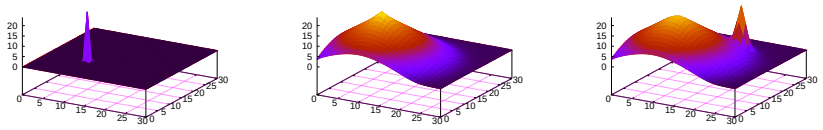
(a) r=1.2. (b) r=1.8. (c) r=2.4.

**Fig. 4.** Distribution of the pheromone intensity varying $r$.

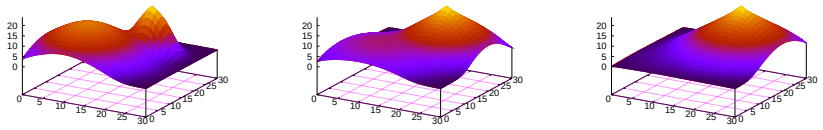
and the pheromone intensity distribution is evaluated as a function of the parameter $r$, by varying $\mu$ being $\lambda$ and $\eta$ fixed.

Figure 4 shows the distribution of the pheromone intensity $\phi(t, \mathbf{v})$ measured in the stable state for three different values of $r$. If the value of $r$ is small ($r = 1.2$) or high ($r = 2.4$), the quality of the gradient is poor. This is due to the prevalence of one of the two feedbacks: negative (with $r = 1.2$ evaporation prevails) or positive ( with $r = 2.4$ excitation prevails and all sensors saturate). On the contrary, intermediate values ($r = 1.8$) generate well-formed pheromone gradients able to cover the whole area, thanks to the correct balance between such two feedbacks. Then, an opportune evaluation of the value of $r$ has to be carried out in order to generate a pheromone gradient that fits with the topological specification of the WSN under exam.

In order to understand the dynamic behavior of the SI algorithm, we carried out a transient analysis able to highlight the different phases of the gradient construction process when the position of the sink changes in time. In particular, in the following experiment (see Figure 5) we analyzed how the algorithm self adapts to topological modifications by recalculating the pheromone gradient when two different sinks are present in the network and they are alternately activated. Figures 5(a)-5(b) show how the pheromone signal is spread on the space $\mathcal{V}$ until the stable state is reached. At this point ($t = 17.5sec$), we deactivated the old sink and we activated a new one in a different position (Figure 5(c)). Figures 5(d)-5(e) describe the evolution of the gradient modification. It is possible to observe that, thanks to the properties of the SI algorithm, the WSN is able to rapidly discover the new sink and to change the pheromone gradient by forgetting the old information until a new stable state is reached (Figure 5(f)).



(a) $t = 0sec$ (b) $t = 17sec$ (c) $t = 17.5sec$



(d) $t = 19sec$ (e) $t = 24sec$ (f) $t = 29sec$

**Fig. 5.** Distribution of the pheromone intensity with respect to $t$ when two sinks are alternately activated. The change is applied at time $t = 17.5sec$.

Finally, in order to test the scalability of the MAM in more complex scenarios, we have assumed a rectangular grid with $n_h = n_w = 100$ hence with $N = 100 \times 100 = 10,000$ sensors, and we have randomly scattered 50 sinks in the grid. The grid is represented in Figure 6, where the sinks are drawn as black spots. Since each sensor is represented by an MA with $P = 20$ states (see Figure 3(b)), the product state space of the overall system has $N = 20^{10,000}$ states!

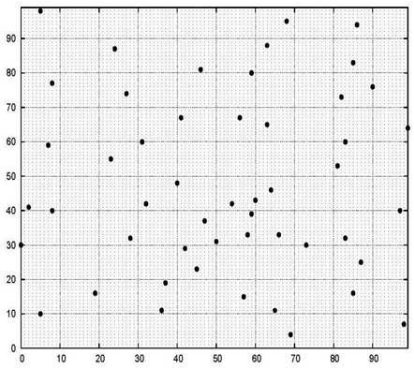


**Fig. 6.** The $100 \times 100$ grid with $10,000$ cells and 50 randomly scattered sinks

The steady pheromone intensity distribution for the geographical space represented in Figure 6 is reported in Figure 7. Through this experiment, we can assess that the pheromone gradient is reached also when no symmetries are present in the network and that the proposed model is able to capture the behavior of the protocol in generating a correct pheromone gradient also in presence of different maximums. In fact, using the same protocol configurations found for a simple scenario, the SI algorithm is able to create a well formed pheromone gradient also in a completely different situation, making such routing technique suitable in non predictable scenarios. Such scenario also demonstrates the scalability of the proposed analytical technique that can be easily adopted in the analysis of very large networks.

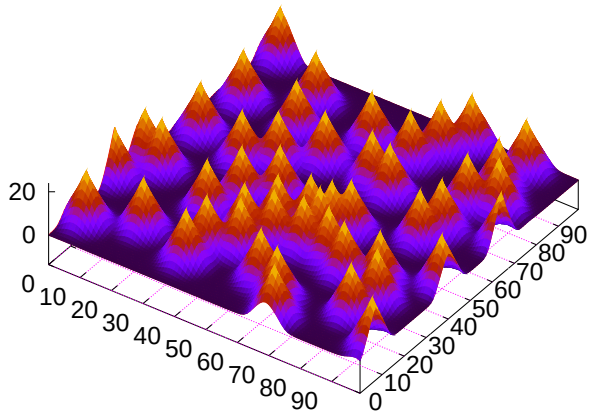


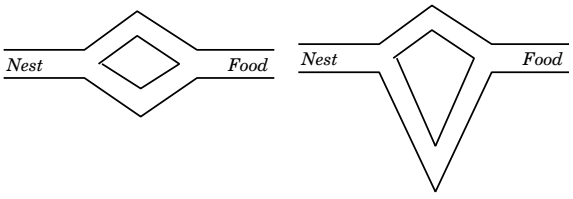

**Fig. 7.** Distribution of the pheromone intensity when the network is composed by a grid of 10,000 sensor nodes with 50 sinks.

## 4 Ant Colony Optimization

The aim of this section is to show how MAMs can be adopted to represent one of the more classical swarm intelligence problem known as Ant Colony Optimization (ACO) [2], that was inspired by the foraging behavior of ant colonies which, during food search, exhibit the ability to

solve simple shortest path problem. To this end, in the present work, we simply show how to build a MAM that solve the famous *Double Bridge Experiment* which was first proposed by Deneubourg and colleagues in the early 90's [16, **?**], and that has been proposed as an entry benchmark for ACO models.

In the experiment a nest of Argentine ants is connected to a food source using a double bridge as shown in Figure 8. Two scenarios are considered: in the first one the bridges have equal length (Figure 8 (a)), in the second one the lengths of the bridges are different (Figure 8 (b)). The collective behavior can be explained by the way in which ants communicate indirectly among them (*stigmergy*). During the journey from the nest to the food source and vice versa, ants release on the ground a chemical substance called *pheromone*, moreover ants can perceive pheromone and they choose with greater probability a path marked by a stronger concentration of pheromone. As a results, ants releasing pheromone on a branch, increase the probability that other ants choose it. This phenomenon is the realization of the positive feedback process described in Section 1 and it is the reason for the convergence of ants to the same branch in the equal length bridge case. When lengths are different, the ants choosing the shorter path reach the food source quicker then those choosing the longer path. Therefore, the pheromone trail grows faster on the shorter bridge and more ants choose it to reach food. As a result, eventually all ants converge to follow the shortest path.

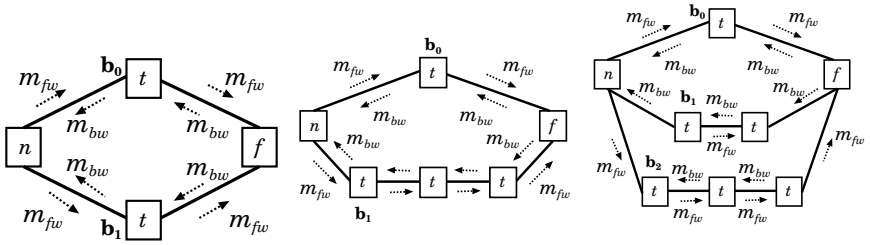



(a) Equal branches. (b) Different branches.

**Fig. 8.** Experiment scenarios. Modified from Goss et al [17].

### 4.1 The $MAM$ model

We represent the double bridge experiment through a Multiple Agent Class and Multiple Message Type MAM. We model ants by messages,




(a) Equal branches. (b) Two different branches. (c) Three different branches.

**Fig. 9.** Graph used to model the experiment scenarios.

and locations that ants traverse by MAs. Three different MA classes are introduced: the class *Nest* denoted by superscript $n$, the class *Terrain* denoted by superscript $t$, and the class *Food* denoted by superscript $f$. Two types of messages are used: ants walking from the nest to the food source correspond to messages of type $fw$ $(forward)$, whereas ants coming back to the nest correspond to messages of type $bw$ $(backward)$.

**Geographical space** Agents (either nest, terrain, or food source) are deployed on a discrete geographical space $\mathcal{V}$ represented as an undirected graph $G = (V, E)$, where the elements in the set $V$ are the vertices and the elements in the set $E$ are the edges of the graph. In Figures 9(a) and 9(b) we show the locations of agents for the equal and the different length bridge scenarios, respectively. The squares are the vertices of the graph and the labels inside them indicate the class of the agent residing on the vertex. In this model we assume that only a single agent resides on each vertex. Message passing from a node to another is depicted as little arrows labeled by the message type. As shown in Figure 9, the different lengths of the branches are represented by a different number of hops needed by a message to reach the food source starting from the nest. Figure 9(c) represents a three branches bridge with branches of different length.

**Agent's structure and behavior** The structure of the three MA classes is described in the following.

**MA Nest -** The nest is represented by a single MA of class $n$, shown in Figure 10(a). The nest MA$^n$ is composed by a single state that emits

messages of type $fw$ at a constant rate $\lambda$, modeling ants leaving the nest in search for food.

**MA Terrain -** An MA in class $t$ (Figure 10(c)) represents a portion of terrain on which an ant walks and encodes in its state space the concentration of the pheromone trail on that portion of the ground. We assume that the intensity of the pheromone trail is discretized in $P$ levels numbered 0, 1, ... $P-1$.

With reference to Figure 10(c), the meaning of the states is the following:

$t_0$ - denotes no pheromone on the ground and no ant walking on it;
$t_i$ - denotes a concentration of pheromone of level $i$ and no ant on the ground;
$t_{if}$ - denotes an ant of forward type residing on the terrain while the pheromone concentration is at level $i$;
$t_{ib}$ - denotes an ant of backward type residing on the terrain while the pheromone concentration is at level $i$.

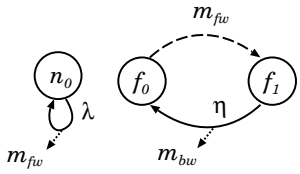
The behavior of the MA$^t$ agent at the reception of the messages is the following:

$fw$ - forward ant; a message of type $fw$ perceived by an MA$^t$ in states $t_i$, induces a transition to state $t_{(i+1)f}$ meaning that the arrival of a forward ant increases the pheromone concentration of one level *(positive feedback)*;
$bw$ - backward ant; a message of type $bw$ perceived by an MA$^t$ in states $t_i$, induces a transition to state $t_{(i+1)b}$ meaning that the arrival of a backward ant increases the pheromone concentration of one level *(positive feedback)*;

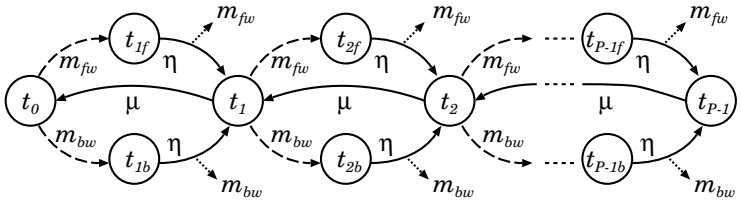
Ants sojourn on a single terrain portion for a mean time of $1/\eta$ sec., then they leaves towards another destination. The local transitions from states $t_{if}$ to states $t_i$ and the generation of message $fw$ model such behavior for forward ants. An analogous behavior is represented for backward ants by local transitions from states $t_{ib}$ to states $t_i$. The local transitions at constant rate $\mu$ from states $t_i$ to states $t_{i-1}$ indicate the decreasing of one unit of the concentration of pheromone due to evaporation *(negative feedback)*.

**MA Food source -** An MA of class $f$ represents the food source (Figure 10(b)). The reception of a message of type $fw$ in state $f_0$ indicates that a forward ant has reached the food source. After a mean time of $1/\eta$ sec., such an ant leaves the food and starts the way back to the nest becoming a backward ant (emission of message of type $bw$).

We assume that no more than one ant at the time can reside on a portion of terrain or in the food source.




(a) MA$^n$: Agent of class nest. (b) MA$^f$: Agent of class food.




(c) MA$^t$: Agent of class terrain.

**Fig. 10.** Markovian agent models for the ACO experiment.

**Perception function** The perception function rules the interactions among agents and, in this particular example, defines the probability that a message (ant) follows a specific path both on the forward and backward direction. The definition of the perception function takes inspiration on the stochastic model proposed in [16, **?**] to describe the dynamic of the ant colony. In such a model the probability of choosing the shorter branch is given by:

$$p_{is}(\tau) = \frac{(k + \varphi_{is}(\tau))^\alpha}{(k + \varphi_{is}(\tau))^\alpha + (k + \varphi_{il}(\tau))^\alpha} \tag{15}$$

where $p_{is}(\tau)$ ($p_{il}(\tau)$) is the probability of choosing the shorter (longer) branch, $\varphi_{is}(\tau)$ ($\varphi_{il}(\tau)$) is the total amount of pheromone on the shorter

(longer) branch at a time $\tau$. The parameter $k$ is the degree of attraction attributed to an unmarked branch. It is needed to provide a non-null probability of choosing a path not yet marked by pheromone. The exponent $\alpha$ provides a non-linear behavior.

In our MA model the perception function $u^m(\cdot)$ is defined, $\forall m \in \{fw, bw\}$, as:

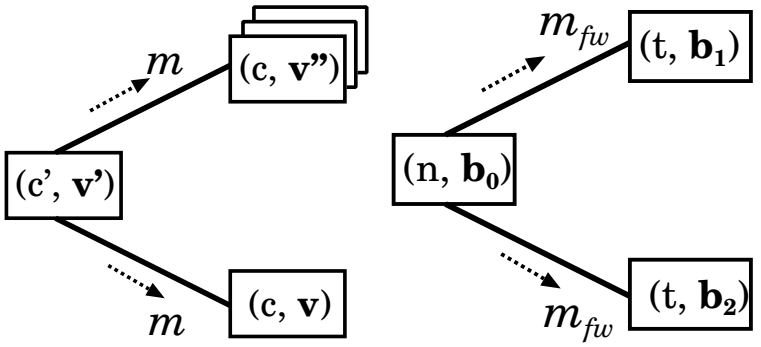
$$u^m(\mathbf{v}, c, i, \mathbf{v}', c', j, \tau) = \frac{(k + E[\boldsymbol{\pi}^c(\tau, \mathbf{v})])^\alpha}{\sum_{(c'',v'')\in Next^m(\mathbf{v}',c')}(k + E[\boldsymbol{\pi}^{c''}(\tau, \mathbf{v}'')])^\alpha} \tag{16}$$

where $k$ and $\alpha$ have the same meaning of Equation (15), $E[\boldsymbol{\pi}^c(\tau, \mathbf{v})]$ gives the mean value of the concentration of pheromone at a time $\tau$ in position $\mathbf{v}$ on the ground, and corresponds to $\varphi(\tau)$. The function $Next^m(\mathbf{v}', c')$ gives the set of pairs $\{(c'', \mathbf{v}'')\}$ such that the agent of class $c''$ in position $\mathbf{v}''$ perceives a message of type $m$ emitted by the agent of class $c'$ in position $\mathbf{v}'$. Figure 11(a) helps to interpret Equation (16). The multiple box stands for all the agents receiving a message $m$ sent by the agent of class $c'$ in position $\mathbf{v}'$. The value of $u^m(\mathbf{v}, c, i, \mathbf{v}', c', j, \tau)$ is proportional to the mean pheromone concentration of the agent in class $c$ at position $\mathbf{v}$ with respect to the sum of the mean concentrations of all the agents that receive message $m$ by the agent in class $c'$ and position $\mathbf{v}'$. For instance, we consider the scenario depicted in Figure 11(b), where a class $n$ agent in position $\mathbf{b_0}$ sends messages of type $fw$ to two other class $t$ agents at position $\mathbf{b_1}$ and $\mathbf{b_2}$, and we compute $u^{fw}(\mathbf{b_2}, t, i, \mathbf{b_0}, n, j, \tau)$. In such case, the evaluation of function $Next^{fw}(\mathbf{b_0}, n)$ gives the set of pair $\{(t, \mathbf{b_1}), (t, \mathbf{b_2})\}$ and the value of the function is:

$$u^{fw}(\mathbf{b_2}, t, i, \mathbf{b_0}, n, j, \tau) = \frac{(k + E[\boldsymbol{\pi}^t(\tau, \mathbf{b_2})])^\alpha}{(k + E[\boldsymbol{\pi}^t(\tau, \mathbf{b_1})])^\alpha + (k + E[\boldsymbol{\pi}^t(\tau, \mathbf{b_2})])^\alpha} \tag{17}$$

As a final remark, we highlight that $u^m(\cdot)$ does not depend on the state variables $i$ and $j$ of the sender and receiver agents even if these variables appear in the definition of $u^m(\cdot)$ (Equation 16). Instead, $u^m(\cdot)$ depends on the whole probability distribution $\boldsymbol{\pi}^c(\tau, \mathbf{v})$ needed to compute the mean value $E[\boldsymbol{\pi}^c(\tau, \mathbf{v})]$.

**Generation and acceptance probabilities** As in Section 3, also in this ACO-MAM model we allow only $g^c_{i,j}(m) = 0$ or $g^c_{i,j}(m) = 1$ and




(a) General case. (b) Example of scenario in Figure 9(b)

**Fig. 11.** Perception function description.

$a^c_{i,j}(m) = 0$ or $a^c_{i,j}(m) = 1\ \forall c, m$. In particular, for the terrain agent $\text{MA}^t$, messages of type $fw$ are sent with probability $g^t_{if,i}(fw) = 1$, and are accepted with probability $a^t_{i,(i+1)f}(fw) = 1$ only in a $t_i$ state inducing a transition to a $t_{(i+1)f}$ state. An analogous behavior is followed during emission and reception of messages of type $bw$.

### 4.2 Numerical results for ACO Double Bridge Experiment

We have performed several experiments on the ACO model. In particular we study *the mean value of the concentration of pheromone* at a time $\tau$ in position $\mathbf{v}$ for a class $c$ agent, $E[\boldsymbol{\pi}^c(\tau, \mathbf{v})]$, defined as:

$$E[\boldsymbol{\pi}^c(\tau, \mathbf{v})] = \sum_{s \in S^c} \pi_s(\mathbf{v}, c) I(s) \tag{18}$$

where $S^c$ denotes the state space of a class $c$ agent, $I(s)$ represents the pheromone level in state $s$, and it corresponds to:

$$I(s) = i \quad \forall\ s \in \{t_i\} \cup \{t_{if}\} \cup \{t_{ib}\} \tag{19}$$

This value is used in Equation (16) to compute $u^m(\cdot)$ which, as previously said, rules the ant's probability to follow a specific path, therefore such performance index provides useful insight of the modeled ant's behavior.

We consider the three scenarios depicted in Figure 9, the labels $\mathbf{b_i}$ denote the positions where we compute the mean value of the concentration of pheromone. In all the experiments the intensity of the pheromone trail is discretized in $P = 8$ levels.

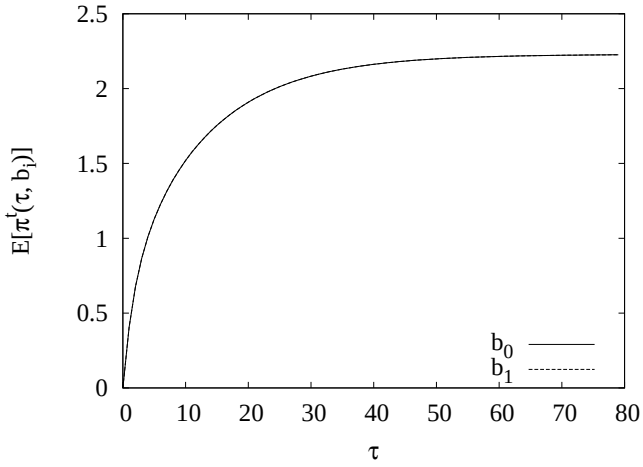



**Fig. 12.** Mean pheromone concentration with $\lambda = 1.0$, $\mu = 1$ and $\eta = 1$ for the equal branches experiment.

In Figure 12, the mean pheromone concentration $E[\boldsymbol{\pi}^c(\tau, \mathbf{b_i})]$ over the time for the equal branches experiment is plotted. As it can be seen, both mean pheromone concentrations have exactly the same evolution proving that ants do not prefer one of the routes.
The case with two different branches is considered in Figure 13. Speed of the ants (i.e., parameter $\eta$) is considered in the column (the left column corresponds to $\eta = 1.0$ and the right column to $\eta = 10$), while the evaporation of the pheromone is taken into account on the rows (respectively with $\mu = 0$, $\mu = 0.5$, and $\mu = 2$). When no evaporation is considered (Figure 13(a) and 13(b)), both paths are equally chosen due to the finite amount of the maximum pheromone level considered in this work. However the shorter path reaches its maximum level earlier than the longer route. In all the other cases, it can be seen that the longer path is abandoned after a while in favor of the shorter one. The evaporation of the pheromone and the speed of the ants both play a role in the time required to drop the longer path. Increasing either of the two, reduces the

time to discover the shorter route.

Finally, Figure 14 considers a case with three branches of different length and different evaporation levels ($\eta = 1$ and $\eta = 10$). Also in this case the model is able to predict that ants will choose the shortest route. It also shows that longer paths are dropped in an order proportional to their length: the longest route is dropped first, and the intermediate route is discarded second. Also in this case, the evaporation rates determine the speed at which paths are chosen and discarded.

## 5 Conclusions

In this work, we have presented how the Markovian Agents performance evaluation formalism can be used to study swarm intelligent algorithms. Although the formalism was developed to study largely distributed systems like sensor networks, or physical propagation phenomena like fire or earthquakes, it has been proven to be very efficient in capturing the main features of swarm intelligence.

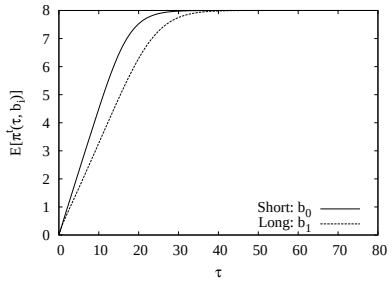
Beside the two cases presented in this chapter, routing in WSN and Ant Colony Optimization, the formalism is capable of considering other cases like Slime Mold models.

Future research lines will try to emphasize the relations between Markovian Agents and Swarm Intelligence, trying to integrate both approaches: using Markovian Agents to formally study new swarm intelligent algorithms, and use swarm intelligent techniques to study complex Markovian Agents models in order to find optimal operation points and best connection strategies.

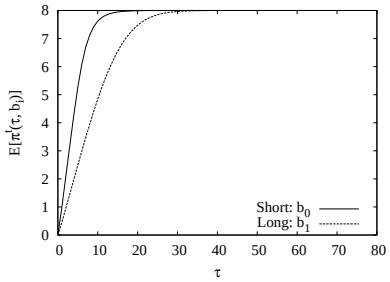
## References


1. M.G. Hinchey, R. Sterritt, C. Rouff: Swarms and swarm intelligence, IEEE Computer, 111 – 113 (April 2007)
2. M. Dorigo, T. Stützle: *Ant Colony Optimization* (MIT Press, 2004)
3. K. Li, K. Thomas, L. F. Rossi, Chien-Chung Shen: Slime mold inspired protocols for wireless sensor networks, 2nd IEEE International Conference on Self-Adaptive and Self-Organizing Systems 2008 (IEEE, Venice Italy 2008) 319 – 328
4. K. Li, C. E. Torres, K. Thomas, L. F. Rossi, Chien-Chung Shen: Slime mold inspired routing protocols for wireless sensor networks, Swarm Intelligence **5**(3-4), 183 – 223 (2011)
5. D. Cerotti, M. Gribaudo, A. Bobbio: Analysis of on-off policies in sensor network using Markovian agents, 4th Int. Workshop PerSens 2008 (IEEE, Hong Kong 2008) 300 – 305

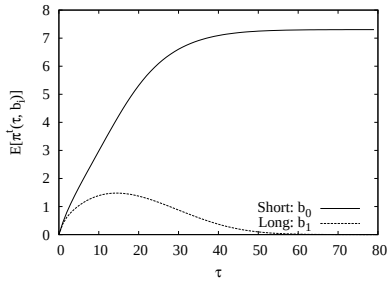
6. A. Bobbio, D. Bruneo, D. Cerotti, M. Gribaudo: Markovian Agents: A New Quantitative Analytical Framework for Large-Scale Distributed Interacting Systems, International Conference on Design and Modeling in Science, Education and Technology - DeMset2011, Orlando (Fl) 2011) 327 – 332
7. M. Gribaudo, A. Bobbio: Performability Analysis of a Sensor Network by Interacting Markovian Agents, Proceedings 8-th International Workshop on Performability Modeling of Computer and Communication Systems, Scotland, UK 2007)
8. A. Bobbio, D. Cerotti, M. Gribaudo: Presenting Dynamic Markovian Agents with a Road Tunnel Application, MASCOTS-09, London, UK 2009)
9. D. Bruneo, M. Scarpa, A. Bobbio, D. Cerotti, M. Gribaudo: Analytical modeling of swarm intelligence in Wireless Sensor Networks, Fourth International Conference on Performance Evaluation Methodologies and Tools (Valuetools 2009), Pisa, Italy 2009)
10. D. Cerotti, M. Gribaudo, A. Bobbio, C.T. Calafate, P.Manzoni: A Markovian agent model for fire Propagation in outdoor environments, Computer Performance Engineering (EPEW2010) (Springer Verlag - LNCS, Vol 6342, Bertinoro, Italy 2010) 131 – 146
11. D. Bruneo, M. Scarpa, A. Bobbio, D. Cerotti, M. Gribaudo: Adaptive Swarm Intelligence Routing Algorithms for WSN in a Changing Environment, IEEE Sensors 2010 Conference, Waikoloa, HI, USA 2010) 1813 – 1818
12. D. Bruneo, M. Scarpa, A. Bobbio, D. Cerotti, M. Gribaudo: Markovian agent modeling swarm intelligence algorithms in wireless sensor networks, Performance Evaluation **In Press, Corrected Proof**, – (2011)
13. M. Saleem, G.A. Di Caro, M. Farooq: Swarm intelligence based routing protocol for wireless sensor networks: survey and future directions, Information Sciences **181**, 4597 – 4624 (2011)
14. K. Trivedi: *Probability & Statistics with Reliability, Queueing & Computer Science applications* (Wiley, II Edition, 2002)
15. M. Paone, L. Paladina, D. Bruneo, A. Puliafito: A Swarm-based Routing Protocol for Wireless Sensor Networks,, 265 – 268 (2007)
16. Deneubourg, S. Aron, S. Goss, J. M. Pasteels: The self-organizing exploratory pattern of the argentine ant, Journal of Insect Behavior **3**(2), 159 – 168 (March 1990)
17. S. Goss, S. Aron, J. Deneubourg, J. Pasteels: Self-organized shortcuts in the Argentine ant, Naturwissenschaften **76**(12), 579 – 581 (December 1989)



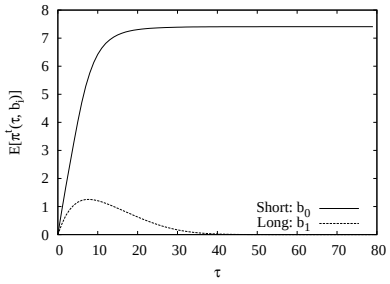

(a) Mean pheromone concentration $\lambda = 1.0$, $\mu = 0$ and $\eta = 1$.



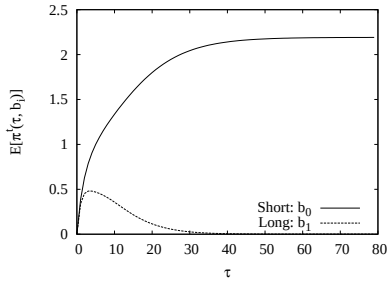

(b) Mean pheromone concentration $\lambda = 1.0$, $\mu = 0$ and $\eta = 10$.



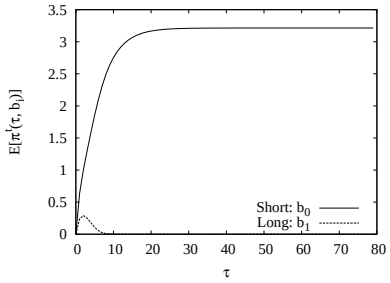

(c) Mean pheromone concentration $\lambda = 1.0$, $\mu = 0.5$ and $\eta = 1$.




(d) Mean pheromone concentration $\lambda = 1.0$, $\mu = 0.5$ and $\eta = 10$.

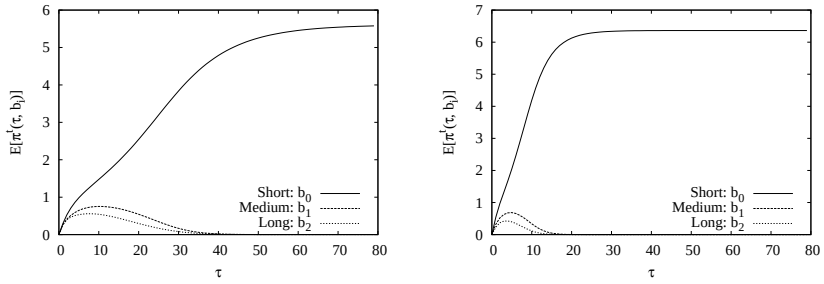



(e) Mean pheromone concentration $\lambda = 1.0$, $\mu = 2$ and $\eta = 1$.




(f) Mean pheromone concentration $\lambda = 1.0$, $\mu = 2$ and $\eta = 10$.

**Fig. 13.** Mean pheromone concentration for the case with two different branches.




(a) Mean pheromone concentration $\lambda = 1.0$, $\mu = 1$ and $\eta = 1$.

(b) Mean pheromone concentration $\lambda = 1.0$, $\mu = 1$ and $\eta = 10$.

**Fig. 14.** Mean pheromone concentration for the case with three different branches.