

Theoretical Distributed Computing meets Biology: A Review

Ofer Feinerman^{1*} and Amos Korman^{2**}

¹ The Weizmann Institute of Science, Israel

² CNRS and University Paris Diderot, France.

Abstract. In recent years, several works have demonstrated how the study of biology can benefit from an algorithmic perspective. Since biological systems are often distributed in nature, this approach may be particularly useful in the context of distributed computing. As the study of algorithms is traditionally motivated by an engineering and technological point of view, the adaptation of ideas from theoretical distributed computing to biological systems is highly non-trivial and requires a delicate and careful treatment. In this review, we discuss some of the recent research within this framework and suggest several challenging future directions.

* The Weizmann Institute of Science, Rehovot, Israel. E-mail: feinermanofer@gmail.com. Supported by the Israel Science Foundation (grant 1694/10).

** CNRS and University Paris Diderot, France. E-mail: amos.korman@liafa.univ-paris-diderot.fr. Supported by the ANR projects DISPLEXITY and PROSE, and by the INRIA project GANG.

1 Introduction

1.1 Background and motivation

Nature serves as inspiration for scientists in all disciplines and computer science is, certainly, no exception. The reverse direction, that of applying studies in computer science to improve our understanding of biological organisms, is currently dominated by the field of bioinformatics. A natural question to be asked is thus the following: how can we apply our knowledge in other aspects of computer science to enhance the study of biology? This direction of research may become particularly fruitful in the context of distributed computing, since indeed, biological systems are distributed in nature (e.g., cells are composed of proteins, organs of cells, populations of organisms and so on).

It is important to note that distributed computing is traditionally studied from an engineering and technological point of view where the focus is on the design of efficient algorithms to solve well defined problems. Analyzing algorithms as used in the biological world requires a different point of view, since the setting is usually unknown, as are the details of the algorithm and even the problem that it aims at solving. Hence, the adaptation of ideas from theoretical distributed computing to biological systems is highly non-trivial and requires a delicate and careful treatment. In many cases, such a study would require a collaboration between biologists that empirically investigate the phenomena, and computer science theoreticians who analyze it. The hope is that despite their enormous complexity, some aspects of biological systems can still be captured by relatively simple abstract models which can be analyzed using distributed computing techniques. In this case, the potential benefit of this direction of research would be huge, not only in terms of understanding large biological systems but also in enriching the scope of theoretical distributed computing.

The synergy between distributed computing and experimental biology is being tightened by methodological advances on both sides. On the distributed computing side, the last twenty years have been very fruitful in terms of advancing our fundamental understanding on topics such as dynamic networks, mobile agents, population protocols, and network computing in general. These advances may indicate that the field of distributed computing has reached the maturity level of being useful also for the context of understanding large biological systems. Since distributed computing addresses the relations between the single entity and the group from the theoretical perspective, experimentally, there is a need to simultaneously probe these two scales. Indeed, from the biological side, the main experimental challenge lies in being able to follow large numbers of identified individuals within behaving populations. The sheer size of these ensembles has, for many years, made this a formidable task. However several technological advances, and above all the huge increase in the availability of computing power, have brought this goal to within our reach.

1.2 Recent technological advances in experimental biology

Examples for the observation of large population are numerous and span various methodologies and biological systems. Fluorescence tagging methods provide detailed information of the internal state of cells in terms of both protein levels and fast phosphorylation dynamics. Coupled with microscopy or FACS measurements systems, fluorescence can be used to simultaneously and quantitatively measure ten or more such different internal variables over populations of millions of cells [27]. A second example comes from the field of neuroscience where there has been a growing emphasis on recording from large neuron populations of behaving animals. The relevant techniques include light-weight tetrodes and multi-electrode arrays [53] as well as the use of activity sensitive fluorescent proteins [10]. Cooperating biological individuals often engage in collective motions. The availability of high-resolution cameras and strong computers together with image processing analysis make these movements tractable. Here, too, examples are many ranging from immune cells moving within the body [29], cells within growing plants [45], birds in flocks [7], fish in schools, and ants within their colonies. Recent tagging technology has been applied to allow for long-term individual tracking of all members within a group (see Figure 1 and [42]). Another interesting example lies in the ability to measure and quantify population heterogeneity and specifically map the spread of behavioral thresholds of individuals to different stimuli [42, 55].

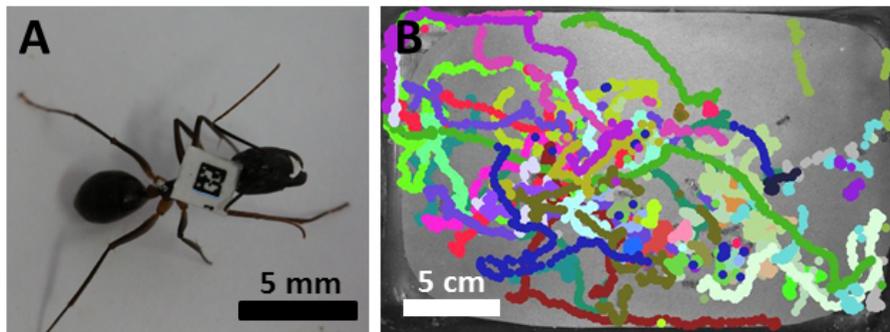


Fig. 1. A. a photo of a tagged carpenter ant worker. B. walking trajectories obtained from automatically tracking tagged ants.

Tracking all individuals within a group leads to the accumulation of huge data sets. On the one hand, this detailed provides us with the rich and detailed information that is required to test theoretical hypothesis. On the other hand, enormous datasets are difficult to manage towards the extraction of relevant information. This challenge has sparked the development of high-throughput automated ethnomics [8].

Experimental manipulations offer a great tool towards deciphering the data we collect and deciding between alternative hypotheses. Traditionally, such manipulations were administered at the level of the group. More recently, several methodologies for manipulating specific individuals within the group have been described. The most striking example is probably the ability to excite neurons by directed illumination which has opened up the field of opto-genetics [10]. Automated manipulation of the trajectories of specific individuals within groups have been demonstrated both in the context of fish and ant colonies [43].

1.3 Bringing the two disciplines closer

Many fields have contributed to biology but we believe distributed computing can bring a new and fresh perspective. To elaborate more on that, we first discuss an interesting difference in the way the notion of a *model* is typically perceived by natural scientists and by computer scientists. In a sense, when it comes to settings that involve multiple individual entities (e.g., network processors, mobile agents, sensors, robots, etc.), computer scientists distinguish the *model* (sometimes called also *setting*), which includes a description of the environment and the restricted capabilities of the individuals, from the *algorithm*, which includes the course of actions which individuals follow. In a way, this corresponds to the distinction between *hardware* and *software*. For example, the topological space in which entities operate as well as their memory capacity are considered to be part of the model, and the particular way in which the entities utilize their memory is part of their algorithm. In contrast, researchers in the natural sciences typically do not make this distinction and treat all these components together as part of the model. We believe that the point of view which makes a distinction between the model and the algorithm can benefit the study of natural phenomena for several reasons.

Although life is highly plastic and evolvable one can still make distinctions between factors that constrain a living system and courses of actions that may be employed within such constraints. A simple example involves physical constraints such as quantum shot noise that defines a lower bound for any light perception be it biological [6] or artificial. Other examples are more subtle and include the distinction between the different time scales inherent to an evolutionary process. One can clearly classify slow evolving features as constraints within which fast evolving can be tuned [30]. Similarly, evolution is not a reversible process so that historical evolutionary decisions may constrain organisms for prolonged periods of times, this may even lead to evolutionary traps [44] which may be perilous for species survival. These internal “hardware” constraints together with environmental constraints are analogous to the computer scientific “model”. Faster evolving parameters as well as actual cognitive, behavioral decisions are analogous to an “algorithm”. We therefore believe that the distinction between these two terms promises a novel and relevant perspectives which can benefit the biological sciences. Theoretical computer science, and theoretical distributed computing in particular, can contribute in this direction.

Recently, several works have utilized methods from distributed computing to improve our understanding of biological systems . It is our belief, however, that all current achievements are very preliminary, and that this direction of research is still making its first steps. Indeed, one important issue to note in this respect, is that, currently, all corresponding works suffer from an (arguably inherent) *gap* between the analyzed setting and the “real setting”, that is, the one apparent in nature. In each particular case, bridging this gap (or even just slightly reducing it), is a challenging task that must be a combined effort of both field biologists and theoreticians.

In this review, we describe four paradigms that aim to incorporate distributed computing with biological phenomena. In the following sections, we shall discuss the paradigms in detail, list current results, explain their potential impact on biology, and suggest future research directions. The four paradigms are divided according to the extent in which the model or the algorithm are assumed to be “known” (or “given”). That is, even though the biological setting is never, actually, known; assumptions regarding the model or algorithm can actually facilitate different lines of investigation.

1.4 The four paradigms

Before dwelling into the details, let us first list the paradigms and discuss them briefly. The paradigms are segregated using the distinction between model and algorithm as described above. At this point we would like to stress that our classification of papers into the four paradigms below is, to some extent, (as many other classifications) a matter of opinion, or interpretation.

1. **“Unknown” algorithm and “unknown” model.** Here, both the model and the algorithm are, mostly, absent, and two approaches have been considered to handle this complex situation.
 - (a) **Surmising.** This classical approach aims at understanding some particular phenomena that has been empirically observed. The approach involves coming up with (somewhat informed) guesses regarding both the model and the algorithm, together with their analysis. The assumed model and algorithm typically do not claim to represent reality accurately, but rather to reflect some aspects of the real setting. The distributed computing experience would come handy for allowing the guessing of more involved and efficient algorithms and for enhancing their rigorous analysis. This type of study was conducted recently by Afek et al. [2] concerning the fly’s brain, where the phenomena observed was that a variant of the Minimum Independent Set (MIS) problem is solved during the nervous system development of the fly.
 - (b) **Finding dependencies between parameters.** This new paradigm aims at obtaining knowledge regarding the model by connecting it to the output of the algorithm, which is typically more accessible experimentally. The paradigm is composed of three stages. The first stage

consists of finding an abstract setting that can be realizable in an experiment, parameterized by an unknown parameter a (e.g., the parameter can be the number of possible states that entities can possibly possess). The second stage involves analyzing the model and obtaining theoretical tradeoffs between the parameter a and the performance efficiency of the algorithm; the tradeoffs are obtained using techniques that are typically associated with distributed computing. Finally, the third stage consists of conducting suitable experiments and measuring the actual efficiency of the biological system. The idea is, that using the tradeoffs and the experimental results on the efficiency of the algorithm, one would be able to deduce information (e.g., bounds) regarding the parameter a . A first step to demonstrate this paradigm was recently made by the authors of this review [20], in the context of foraging strategies of ants.

2. **“Known” algorithm and “unknown” model.** This paradigm corresponds to the situation in which the algorithm is already fairly understood, and the challenge is to find a simple and abstract setting that, on the one hand, somehow captures some essence of the “real” setting, and, on the other hand, complies well with the given algorithm to explain some empirically observed phenomena. We are not aware of any work in the framework of biology where this paradigm was employed. However, the work of Kleinberg [31] on *small world* phenomena can serve as an example for this paradigm within the context of sociology.
3. **“Unknown” algorithm and “known” model.** In the context of large biological ensembles, this paradigm fixes an abstract model consisting of multiple processors (either mobile, passively mobile, or stationary) operating in a given setting. Two approaches are considered.
 - (a) **Complexity analysis.** Here, the complexity of the abstract model is analyzed, aiming at bounding the power of computation of the processors as a group. In all cases studied so far, the model seems to represent a very high level abstraction of reality, and the resulting computational power is typically very strong. Furthermore, known complexity results are obtained on models that seem too far from reality to be realized in experiments. Hence, this line of research currently not only does not involve experimental work, but also does not seem to be related to such experiments in the near future. As the focus of this review is on connections between theoretical work in distributed computing and experimental work in biology, we decided to only briefly discuss this line of research in this review.
 - (b) **Guessing an algorithm.** Here, again, the aim is to provide an explanation to some observed phenomena. The model is given to the researchers, and the goal is to come up with a simple algorithm whose analysis complies with the phenomena. For example, Bruckstein [9] aims at explaining the phenomena in which ant trails seem to be relatively straight. The paper relies on a fairly reasonable model, in which ants can see their nearest neighbors, all ants walk in one direction and in the same speed.

Then the paper contributes by providing an algorithm whose outcome is straight lines.

4. **“Known” algorithm and “known” model.** This paradigm assumes a model and algorithm that have been substantiated to some extent by empirical findings. The efficiency of the algorithm is then theoretically analyzed, as is, with the goal of obtaining further insight into system function. The corresponding study is currently very limited: to the best of our knowledge, its merely contains works analyzing bird flocking algorithms.

1.5 Other related work

Computer science and biology have enjoyed a long and productive relationship for several decades. One aspect of this relationship concerns the field of *bioinformatics*, which utilizes computer science paradigms to retrieve and analyze biological data, such as nucleic acid and protein sequences. The field of *natural computing* is another aspect of this relationship; this field brings together nature and computing to encompass three classes of methods: (1) simulating natural patterns and behaviors in computers, (2) potentially designing novel types of computers, and (3) developing new problem-solving techniques. (For detailed reviews, see *e.g.*, [11, 33, 40].)

The motivation for the latter class of methods is to provide alternative solutions to problems that could not be (satisfactorily) resolved by other, more traditional, techniques. This direction of research is termed *bio-inspired computing* or *biologically motivated computing* [12, ?], or *computing with biological metaphors* [50]. *Swarm intelligence*, for instance, refers to the design of algorithms or distributed problem-solving devices inspired by the collective behavior of social insects and other animal societies. This sort of work is inspired by biology but not bound by biology, that is, it doesn't have to remain true in the biology context. Examples of this approach can be found, *e.g.*, in the book by Dorigo and Stutzle [16] which describes computer algorithms inspired by ant behavior, and particularly, the ability to find what computer scientists would call shortest paths. Another example of this approach comes in the context of applications to robotics. *Swarm robotics*, for example, refers to the coordination of multi-robot systems consisting of large numbers of mostly simple physical robots [17].

The reverse direction, that of applying ideas from theoretical computer science to improve our understanding of biological phenomena, has received much less attention in the literature, but has started to emerge in recent years, from different perspectives. An example for such an attempt is the work of Valiant [51] that introduced a computational model of evolution and suggested that Darwinian evolution be studied in the framework of computational learning theory. Several other works (which are discussed in the following sections) took this direction by applying algorithmic ideas from the field of distributed computing to the context of large and complex biological ensembles.

At this point, we would like to note that in several works both directions of research co-exist. However, since the topic of this review focuses on implying

ideas from distributed computing to biology, then, when discussing particular works, we shall focus our attention on this direction of research and typically ignore the bio-inspired one.

2 “Unknown” model and “unknown” algorithm

This paradigm corresponds to the situation in which both model and algorithm are, to some extent, absent. Roughly speaking, two approaches have been considered to handle this complex situation. The first approach is classical in the context of science. It concerns the understanding of a particular phenomena that has been observed experimentally, and consists in guessing both a model and an algorithm to fit the given phenomena. The second approach aims at improving our understanding of the connections between model and/or algorithmic parameters, and to reduce the parameter space by finding dependencies between parameters. Furthermore, when coupled with suitable experiments, this approach can be used to obtain bounds on parameters; which may be very difficult to obtain otherwise. Let us first describe the more classical approach.

2.1 Surmising

This approach concerns a particular phenomena that has been observed empirically. After some preprocessing stage, consisting of high level observations and some data-analysis, the main goal is to come up with (informed) guesses for both a model and an algorithm to fit the given phenomena. In other words, the model and algorithm are tailored to the particular phenomena. Note that this guessing approach is hardly new in biology, and in fact, it is one of the more common ones. However, it is far less common to obtain such guesses using the types of reasoning that are typically associated with distributed computing. Indeed, the distributed computing experience would come handy here for two main reasons: (1) for allowing the guessing of more involved efficient algorithms and (2) for enhancing their rigorous analysis. This was the case in the recent study by Afek et al. [2] concerning the fly’s brain, where the phenomena observed was that a variant of the Minimum Independent Set (MIS) problem is solved during the development of the nervous system of the fly.

MIS on the fly: Informally, the classical Maximal Independent Set (MIS) problem aims at electing a set of leaders in a graph such that all other nodes in the graph are connected to a member of the MIS and no two MIS members are connected to each other. This problem has been studied extensively for more than twenty years in the distributed computing community. Very recently, Afek et al. [2] observed that a variant of the distributed MIS problem is solved during the development of the fly’s nervous system. More specifically, during this process, some cells in the pre-neural clusters become Sensory Organ Precursor (SOP) cells. The outcome of this process guarantees that each cell is either an SOP or directly connected to an SOP and no two SOPs are connected. This is similar

to the requirements of MIS. However, the solution used by the fly appears to be quite different from previous algorithms suggested for this task. This may be due to the limited computational power of cells as compared to what is assumed for processors in traditional computer science solutions. In particular, in the "fly's solution", the cells could not rely on long and complex messages or on information regarding the number of neighbors they have.

Afek et al. suggested an abstract (relatively restrictive) model of computation that captures some essence of the setting in which flies solve the SOP selection problem. For this abstract model, the authors were able to develop a new MIS algorithm that does not use any knowledge about the number of neighbors a node has. Instead, with probability that increases exponentially over time, each node that has not already been connected to an MIS node proposes itself as an MIS node. While the original algorithm of Afek et al [2] requires that nodes know an upper bound on the total number of nodes in the network, a new version of this algorithm [1] removes this requirement.

The algorithms in [1, 2] are motivated by applications to computer networks, and hence primarily follow the bio-inspired approach. Nevertheless, it is interesting to note that some aspects of these algorithms are consistent with empirical observations. Indeed, in [2] the authors used microscopy experiments to follow SOP selection in developing flies, and discovered that a stochastic feedback process, in which selection probability increases as a function of time, provides a good match to the experimental results; such a stochastic feedback process is also evident in the corresponding algorithms. Hence, these works also follow the direction of research which is the topic of this review.

2.2 Finding dependencies between parameters

In contrast to the previous approach, this approach does not focus on understanding a particular phenomena, but instead aims at understanding the underlying connections between the model and/or algorithm ingredients. Indeed, a common problem, when studying a biological system is the complexity of the system and the huge number of parameters involved. Finding ways of reducing the parameter space is thus of great importance. One approach is to divide the parameter space into critical and non-critical directions where changes in non-critical parameters do not affect overall system behavior [22, 26]. Another approach, which is typically utilized in physics, would be to define theoretical bounds on system performance and use them to find dependencies between different parameters. This approach may be particularly interesting in the case where tradeoffs are found between parameters that are relatively easy to measure experimentally and others that are not. Indeed, in this case, using such tradeoffs, relatively easy measurements of the "simple parameters" would allow us to obtain non-trivial bounds on the "difficult parameters". Note that such theoretical tradeoffs are expected to depend highly on the setting, which is by itself difficult to understand.

Very recently, a first step in the direction of applying this approach has been established by the authors of this review, based on ideas from theoretical

distributed computing [20]. That work considers the context of central place foraging, such as performed by ants around their nest. The particular theoretical setting, involving mobile probabilistic agents (e.g., ants) that search for food items about a source node (e.g., the nest) was introduced in [21] and has two important advantages. On the one hand, this setting (or, perhaps, a similar one) is natural enough to be experimentally captured, and on the other hand, it is sufficiently simple to be analyzed theoretically. Indeed, in [20] we establish tradeoffs between the time to perform the task and the amount of memory (or, alternatively, the number of internal states) used by agents to perform the task. Whereas the time to perform the task is relatively easy to measure, the number of internal states of ants (assuming they act similarly to robots) is very difficult to empirically evaluate directly.

As mentioned in [20], the natural candidates to test this paradigm on would be desert ants of the genus *Cataglyphys* and the honeybees *Apis mellifera*. These species seem to possess many of the individual skills required for the behavioral patterns that are utilized in the corresponding upper bounds in [20, ?], and hence are expected to be time efficient.

It is important to note that it is not claimed that the setting proposed in [20] precisely captures the framework in which these species perform search (although it does constitute a good first approximation to it). Indeed, it is not unreasonable to assume that a careful inspection of these species in nature would reveal a somewhat different framework and would require the formulation of similar suitable theoretical memory bounds. Finding the “correct” framework and corresponding tradeoffs is left for future work. Once these are established, combining the memory lower bounds with experimental measurements of search speed with varying numbers of searchers would then provide quantitative evidence regarding the number of memory bits (or, alternatively, the number of states) used by ants. Furthermore, these memory bits must mainly be used by ants to assess their own group size prior to the search. Hence, such a result would provide insight regarding the ants’ quorum sensing process inside the nest.

3 “Known” algorithm and “unknown” model

This paradigm corresponds to the situation in which the algorithm is already fairly understood, and the challenge is to find a simple and abstract model that, on the one hand, somehow captures some essence of the “real” setting, and, on the other hand, complies well with the given algorithm to explain some empirically observed phenomena. Although this paradigm could be applied within the framework of biology we are not aware of any such work. Nevertheless, to clarify the usefulness of this paradigm, we describe works on *small world* phenomena and their applicability to sociology.

Small world phenomena: It is long known that most people in social networks are linked by short chains of acquaintances. The famous “six degrees of separation” experiment by Milgram [38] implied not only that there are short chains between

individuals, but also that people are good at finding those chains. Indeed, individuals operating with only local information are expected to find these chains by using the simple *greedy* algorithm. Kleinberg [31] investigated this *small world* phenomena from a distributed computing point of view, and came up with an extremely simple abstract model that, on the one hand, somehow captures some essence of the “real” setting, and, on the other hand, complies well with a greedy algorithm to explain the small world phenomena.

The model consists of a two-dimensional grid topology augmented with long-range connections, where the probability $\Pr(x, y)$ of a connecting node x with a node y is some (inverse proportional) function of their lattice distance $d(x, y)$, that is, $\Pr(x, y) \approx 1/d(x, y)^\alpha$, for some parameter α . This abstract model does seem to represent some essence of social networks, where it is reasonable to assume that each person has several immediate acquaintances and fewer “long range acquaintance”, and that it is less likely to have a long range acquaintance if this acquaintance is “far away” (in some sense). Kleinberg [31] then studied distributed greedy algorithms that resemble the one used by Milgram: for transmitting a message, at each step, the holder of the message must pass it across one of its (either short or long range) connections, leading to one who minimizes the distance to the destination. Crucially, this current holder does not know the long range connections of other nodes. The algorithm is evaluated by its expected delivery time, which represents the expected number of steps needed to forward a message between a random source and target in a network generated according to the model. It turns out that when the long-range connections follow an inverse-square distribution, i.e., the case $\alpha = 2$, the expected time to deliver a message is small: polylogarithmic in the number of nodes. The setting was further analyzed proving that the exponent $\alpha = 2$ for the long range distribution is the only exponent at which any distributed algorithm on the grid can achieve a polylogarithmic time delivery.

Following [31], several other works investigated extensions of this model, in what has become a whole area of research (see [23] for a survey). In particular, Fraigniaud and Giakkoupis proved [24] that all networks are smallworldizable, in the sense that, for any network G , there is a natural way to augment G with long-range links, so that (a minor variant of) greedy routing performs in $2\sqrt{\log n}$ steps. Before, [25] proved that this bound is essentially the best that you can expect in arbitrary networks. In addition, Chaintreau et al. [14] studied how the Kleinberg’s harmonic distribution of the long-range links could emerge naturally from a decentralized process. It appears that if individuals move at random and tend to forget their contact along with time, then we end up with connections between individuals that are distributed harmonically, as in Kleinberg’s paper.

4 “Unknown” algorithm and “known” model

In the context of large biological ensembles, this paradigm fixes an abstract model consisting of multiple processors (either mobile, passively mobile, or stationary). Two approaches are considered. The first approach analyzes the com-

putational power of such models, and the second suggests simple algorithms that can potentially operate within model constraints and explain known phenomena.

4.1 Computational aspects

The fundamental question of what can be computed by biological systems is fascinating. One of the aspects of this question concerns, for example, the computational power of ants. It is quite evident that the computational abilities of the human brain are much more impressive than those of a typical ant colony, at least in some respects. A basic philosophical question is whether this computational gap is a consequence of the different physical settings (e.g., in the case of ants, this includes the physical organization of ants and their individual limitations), or because it was simply not developed by evolution as it wasn't necessary for survival. To put it more simply: is the reason that an ant colony is not as smart as a human brain because it *cannot* be or because it doesn't *need* to be?

While we are very far away from answering such a question, some very initial steps have been taken in this direction. Various abstract models which loosely represent certain settings in nature are suggested and their computational power is analyzed. Broadly speaking, in all such previous works, the analyzed model appears to be very strong, and is often compared to a Turing machine. It is important to note, however, that, as a generalization, the motivations for some of the suggested models in the literature come only partially from biology, and are affected equally by sensor networks and robotics applications. Indeed, in all cases studied so far, the model seems to represent a very high level abstraction of reality, and in particular, seems too far from reality to be realized in experiments. Hence, this line of research currently not only does not involve experimental work, but also does not seem to be related to such experiments in the near future. As the focus of this review is on connections between theoretical work in distributed computing and experimental work in biology, we decided to only briefly discuss corresponding computational results.

Population protocols: The abstract model of population protocols, introduced by Angluin et al. [3], was originally intending to capture abstract features of computations made by tiny processes such as sensors, but it was observed also that it may be useful for modeling the propagation of diseases and rumors in human populations as well as stochastically interacting molecules. The question of what can be computed by population protocols has been studied quite thoroughly. Specifically, perhaps the main result in the setting of population protocols is that the set of computable predicates under a “fair” adversary is either exactly equal to or closely related to the set of semilinear predicates [4]. The model was shown to be much more powerful under a (uniform) random scheduler, as it can simulate a register machine. For a good survey of population protocols, refer to [5].

A related model was studied by Lachmann and Sella [34], which is inspired by task switching of ants in a colony. The model consists of a system composed

of identical agents, where each agent has a finite number of internal states. The agent's internal state may change either by interaction with the environment or by interaction with another agent. Analyzing the model's dynamics, the authors prove it to be computationally complete. A result in a similar flavor was obtained by Sole and Delgado [49].

Ant robotics: Ant robotics is a special case of swarm robotics, in which the robots can communicate via markings [54]. This model is inspired by some species of ants that lay and follow pheromone trails. Recently, Shiloni et al. showed that a single ant robot (modeled as finite state machine) can simulate the execution of any arbitrary Turing machine [46]. This proved that a single ant robot, using pheromones, can execute arbitrarily complex single-robot algorithms. However, the result does not hold for N robots.

Hopfield model: The celebrated Hopfield model [28] uses intuition from the field of statistical mechanics and neuroscience to provide intuition for associative memory in the brain. The model includes a learning stage at which a large number of distributed memories are imprinted onto a neuronal network. This is achieved by a very simple learning algorithm that fine tunes the connections (synapses) between the neurons. Once the memories are set, they can be retrieved by employing simple dynamics which is inspired by actual neuronal dynamics. Namely, neurons are modeled to be in one of two states as inspired by the all-or-none nature of action potentials; Second, neuronal dynamics are governed by local threshold computations as inspired by the actual physiological membrane thresholds. Formed memories are associative in the sense that partial memories are enough to reconstruct the full ones. The model was shown to be highly robust to failures such as connection, or neuronal deletions. Further, it has been proven, that this model for neuronal networks is computationally complete [47].

The Hopfield model is not only biologically inspired but also inspires biologists. It would be fair to say, that much of our intuition for distributed information storage and associative memories in the brain derives from this model. On the other hand, the Hopfield model is very far from being biologically accurate and as such it is not as useful as one might expect in the modeling of actual micro-circuits in the brain.

Distributed computing on fixed networks with simple processors: Recently, Emek et al. [18] introduced a new relaxation of the Beeping model from [1, 2], where the computational power of each processor is extremely weak and is based on thresholds. Despite the weak restrictions of the model, the authors show that some of the classical distributed computing problems (e.g., MIS, coloring, maximal matching) can still be solved somewhat efficiently. This shows that the power computation of such a model is high, at least by judging it from the point of view of tasks typically associated with computer networks.

4.2 Guessing an algorithm

Here, again, the goal is to provide an explanation to some observed phenomena. The model is given to the researchers, and the goal is to come up with a simple algorithm whose analysis complies with the phenomena. For example, Bruckstein [9] aims at explaining the phenomena in which ant trails seem to be relatively straight. The paper relies on a fairly reasonable model, in which ants walking from their nest towards a food source initially follow a random, convoluted path laid by the first ant to find the food. The model further includes the reasonable assumption that an ant on the trail can see other ants in its proximity. The paper goes on to suggest a simple algorithm in which each ant continuously orients her direction toward the ant walking directly in front of her. It is shown that this algorithm results in a “corner-cutting” process by which the ant trail quickly converges to the shortest distance line connecting the nest and the food source.

5 “Known” algorithm and “known” model

This paradigm concerns the analysis of specific algorithms operating in particular given models. The corresponding study currently includes works on *bird flocking* algorithms. In these cases, the specific algorithms and models studied are supported by some empirical evidence, although this evidence is, unfortunately, quite limited. The main objective of this paradigm is to have a better, more analytical, understanding of the algorithms used in nature. Nevertheless, potentially, this paradigm can also be used to give some feedback on the validity of the proposed algorithm. Indeed, on the one hand, some positive evidence is obtained if the (typically non-trivial) analysis finds the algorithm feasible and/or explaining a certain phenomena. On the other hand, proving its unfeasibility, e.g., that it requires unreasonable time to be effective, serves as a negative feedback, which may lead to disqualifying its candidature.

Bird flocking algorithms: The global behavior formed when a group of birds are foraging or in flight is called *flocking*. This behavior bears similarities with the swarming behavior of insects, the shoaling behavior of fish, and herd behavior of land animals. It is commonly assumed that flocking arises from simple rules that are followed by individuals and does not involve any central coordination. Such simple rules are, for example: (a) alignment - steer towards average heading of neighbors, (b) separation - avoid crowding neighbors, and (c) cohesion - steer towards average position of neighbors (long range attraction). With these three simple rules, as initially proposed by Reynolds [41], computer simulations show that the flock moves in a “realistic” way, creating complex motion and interaction. The basic flocking model has been extended in several different ways since [41]. Measurements of bird flocking have been established [19] using high-speed cameras, and a computer analysis has been made to test the simple rules of flocking mentioned above. Evidence was found that the rules generally hold true and that the long range attraction rule (cohesion) applies to the nearest

5-10 neighbors of the flocking bird and is independent of the distance of these neighbors from the bird.

Bird flocking has received considerable attention in the scientific and engineering literature, and was typically viewed through the lens of control theory and physics. Computer simulations support the intuitive belief that, by repeated averaging, each bird eventually converges to a fixed speed and heading. This has been proven theoretically, but how long it takes for the system to converge has remained an open question until the work of Chazelle [13]. Using tools typically associated with theoretical computer science, Chazelle analyzed two classical models that are highly representative of the many variants considered in the literature, namely: (a) the kinematic model, which is a variant of the classical Vicsek model [52], and (b) the dynamic model [15]. Chazelle proved an upper bound on the time to reach steady state, which is extremely high: a tower-of-twos of height linear in the number of birds. Furthermore, it turns out that this upper bound is in fact tight. That is, Chazelle proved that with a particular initial settings, the expected time to reach steady state is a tower-of-twos of height.

That lower bound is, of course, huge, and no (reasonably large) group of real birds can afford itself so much time. At first glance, it may seem as if this result already implies that real birds do not perform the considered algorithms. However, for such an argument to be more convincing, several assumptions need to be modified to suit better the setting of real birds. The first issue concerns the notion of convergence. As defined in [13], reaching a steady state, means that the system no longer changes. Of course, birds are not expected to actually achieve this goal. It would be interesting to come up and investigate some relaxation to this notion of convergence, that would correspond better to reality. Second, it would be interesting to prove lower bounds assuming the “average initial setting”, rather than the worst case one.

6 More future directions

Generally speaking, the direction of applying ideas from theoretical distributed computing to biology contexts is currently making its first baby steps. Hence, this direction is open to a large variety of research attempts. We have described several research paradigms to proceed in. Particular suggestions for future work using these paradigms can be found by inspecting some of the examples mentioned in the survey by Navlakha and Bar-Joseph [40] on bio-inspired computing. In particular, as mentioned in [40], two important examples of problems common to both biological and computational systems are distributed *consensus* [36] and *synchronization* [35, 48]. In biology, consensus has an important role in coordinating populations. Fish, for instance, must be able to quickly react to the environment and make collective decisions while constantly being under the threat of predation. Synchronization is apparent in fireflies that simultaneously flash [39, 56] and pacemaker cells in the heart [32]. We believe that the vast literature and advancements concerning synchronization and consensus in the

context of theoretical distributed computing may also be used to enhance the understanding of corresponding biological phenomena.

References

1. Y. Afek, N. Alon, Z. Bar-Joseph, A. Cornejo, B. Haeupler, and F. Kuhn. Beeping a Maximal Independent Set. In Proc. 25th International Symposium on Distributed Computing (DISC), 32–50, 2011.
2. Y. Afek, N. Alon, O. Barad, E. Hornstein, N. Barkai, and Z. Bar-Joseph. A biological solution to a fundamental distributed computing problem. *Science*, 331(6014):183–185, 2011.
3. D. Angluin, J. Aspnes, Z. Diamadi, M. J. Fischer, and R. Peralta. Computation in networks of passively mobile finite-state sensors. *Distributed Computing*, 18(4), 235–253, 2006.
4. D. Angluin, J. Aspnes, D. Eisenstat, E. Ruppert. The computational power of population protocols. *Distributed Computing* 20(4), 279–304, 2007.
5. J. Aspnes and E. Ruppert. An introduction to population protocols. *Bulletin of the European Association for Theoretical Computer Science, Distributed Computing Column*, 93:98–117, October 2007. An updated and extended version appears in *Middleware for Network Eccentric and Mobile Applications*, Benoit Garbinato, Hugo Miranda, and Luis Rodrigues, eds., Springer-Verlag, pp. 97–120, 2009.
6. W. Bialek. Physical limits to sensation and perception. *Annual Review of Biophysics and Biophysical Chemistry*, 16:455–478, 1987.
7. W. Bialek, A. Cavagnab, I. Giardinab, T. Morad, E. Silvestrib, M. Vialeb, and A. M. Walczake. Statistical mechanics for natural flocks of birds. *PNAS*, 109(13), 4786–4791, 2012.
8. K. Branson, A. A. Robie, J. Bender, P. Perona, and M. H. Dickinson. High-throughput ethomics in large groups of *Drosophila*. *Nature Methods* 6, 451–457, 2009.
9. A.M. Bruckstein. Why the ant trails look so straight and nice. *The Mathematical Intelligencer*, Vol. 15/2, pp. 58–62, 1993.
10. J.A. Cardin, M. Carlén, K. Meletis, U. Knoblich, F. Zhang, K. Deisseroth, L.H. Tsai, and C.I. Moore. Targeted optogenetic stimulation and recording of neurons in vivo using cell-type-specific expression of Channelrhodopsin-2. *Nature Protocols*, 5, 247–254, 2010.
11. L. N. de Castro. *Fundamentals of Natural Computing: Basic Concepts, Algorithms, and Applications*. CRC Press, 2006.
12. L. N. de Castro and F.J. Von Zuben. *Recent developments in biologically inspired computing*. Idea Group Publishing; 2004.
13. B. Chazelle. Natural algorithms. *Proc. 19th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 422–431, 2009.
14. A. Chaintreau, P. Fraigniaud, and E. Lebar. Networks Become Navigable as Nodes Move and Forget. *Proc. 35th International Colloquium on Automata, Languages and Programming (ICALP)*, 133–144, 2008.
15. F. Cucker, S. Smale. Emergent behavior in flocks. *IEEE Trans. Automatic Control* 52, 852–862, 2007.
16. M. Dorigo and T. Stutzle. *Ant colony optimization*. MIT Press, 2004.

17. M. Dorigo, E. Sahin. Swarm Robotics - Special Issue. *Autonomous Robots*, vol. 17, pp. 111-113, 2004.
18. Y. Emek, J. Smula and R. Wattenhofer. Stone Age Distributed Computing. Arxiv 2012.
19. T. Feder. Statistical physics is for the birds. *Physics today* 60 (10): 2830, 2007.
20. O. Feinerman and A. Korman. Memory Lower Bounds for Randomized Collaborative Search and Implications to Biology. In Proc. 26th International Symposium on Distributed Computing (DISC), 61-75, 2012.
21. O. Feinerman, A. Korman, Z. Lotker and J.S Sereni. Collaborative Search on the Plane without Communication. In Proc. 31st Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing (PODC), 77-86, 2012.
22. O. Feinerman, J. Veiga, J.R. Dorfman, R.N. Germain and G. Altan-Bonnet. Variability and robustness in T Cell activation from regulated heterogeneity in protein levels. *Science* 321(5892): 1081-1084, 2008.
23. P. Fraigniaud. Small Worlds as Navigable Augmented Networks. Model, Analysis, and Validation. Proc. 15th Annual European Symposium on Algorithms (ESA), 2-11, 2007.
24. P. Fraigniaud and G. Giakkoupis. On the searchability of small-world networks with arbitrary underlying structure. Proc. 42th ACM Symposium on Theory of Computing (STOC), 389-398, 2010.
25. P. Fraigniaud, E. Lebar, and Z. Lotker. A Lower Bound for Network Navigability. *SIAM J. Discrete Math.* 24(1): 72-81, 2010.
26. R.N. Gutenkunst, J.J. Waterfall, F.P. Casey, K.S. Brown, C.R. Myers and J.P. Sethna. Universally sloppy parameter sensitivities in systems biology models. *PLOS Computational Biology* 3(10): e189, 2007.
27. L.A. Herzenberg, and S.C. De Rosa. Monoclonal antibodies and the FACS: complementary tools for immunobiology and medicine. *Immunology Today* 21(8), 383-390, 2000.
28. J.J. Hopfield Neural networks and physical systems with emergent collective computational abilities. *PNAS*, 79(8), 2554-2558, 1982.
29. T. Ishii and M. Ishii Intravital two-photon imaging: a versatile tool for dissecting the immune system. *Ann Rheum Dis* 70, 113-115, 2011.
30. E. Jablonka and M.J. Lamb. Evolution in four dimensions: Genetic, epigenetic, behavioral, and symbolic variation in the history of life. MIT Press, 2005.
31. J. M. Kleinberg. Navigation in a small world. *Nature* 406 (6798): 845, 2000.
32. T. Kuramoto and H. Yamagishi. Physiological anatomy, burst formation, and burst frequency of the cardiac ganglion of crustaceans. *Physiol Zool* 63: 102-116, 1990.
33. S. Olariu and A.Y. Zomaya. Handbook of Bioinspired Algorithms and Applications. Chapman & Hall/CRC, 2005.
34. M. Lachmann and G. Sella. The Computationally Complete Ant Colony: Global Coordination in a System with No Hierarchy. *ECAL*, 784-800, 1995.
35. C. Lenzen, T. Locher, and R. Wattenhofer. Tight bounds for clock synchronization. *J. ACM* 57(2), 2010.
36. N.A. Lynch. Distributed Algorithms. San Francisco, CA: Morgan Kaufmann Publishers Inc. 1996.
37. D. Mange and M. Tomassini. Bio-inspired computing machines: towards novel computational architecture. Presses Polytechniques et Universitaires Romandes; 1998.
38. S. Milgram. The small-world problem. *Psychol. Today* 1, 61-67, 1967.
39. R.E. Mirollo and S.H. Strogatz. Synchronization of pulse-coupled biological oscillators. *SIAM J. Applied Math* 50: 1645-1662, 1990.

40. S. Navlakha and Z. Bar-Joseph. Algorithms in nature: the convergence of systems biology and computational thinking. *Nature-EMBO Molecular Systems Biology*, 7:546, 2011.
41. C. Reynolds. Flocks, herds and schools: A distributed behavioral model. *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, 25–34, 1987.
42. E. J. H. Robinson, T. O. Richardson, A. B. Sendova-Franks, O. Feinerman, and N. R. Franks. Radio tagging reveals the roles of corpulence, experience and social information in ant decision making. *Behavioral Ecology and Sociobiology*, 63(5), 627–636, 2009.
43. E.J.H Robinson, O. Feinerman, and N.R. Franks Experience, corpulence and decision making in ant foraging. *Journal of Experimental Biology*, 215, 2653–2659, 2012.
44. M.A. Schlaepfer, M.C. Runge, and P.W. Sherman. Ecological and evolutionary traps. *Trends in Ecology and Evolution*, 17(10), 474–480, 2002.
45. G. Sena, Z. Frentz, K. D. Birnbaum, and S. Leibler. Quantitation of Cellular Dynamics in Growing Arabidopsis Roots with Light Sheet Microscopy. *PLOS1*, 2011.
46. A. Shiloni, N. Agmon, and G. A. Kaminka. Of Robot Ants and Elephants: A Computational Comparison. In *Theoretical Computer Science*, 412, 5771-5788, 2011.
47. H.T. Siegelmann and E.D. Sontag. On the Computational Power of Neural Nets. *Journal of Computer and System Sciences* 50, 132–150, 1995.
48. O. Simeone, U. Spagnolini, Y. Bar-Ness, and S. Strogatz. Distributed synchronization in wireless networks. *IEEE Signal Process Mag* 25: 81–97, 2008.
49. R. V. Sole and J. Delgado. Universal Computation in Fluid Neural Networks. *Complexity* 2(2), 49-56, 1996.
50. R. Paton. *Computing with biological metaphors*. Chapman & Hall; 1994.
51. L. G. Valiant. Evolvability. *J. ACM* 56(1), 2009.
52. T. Vicsek, A. Czirok, E. Ben-Jacob, I. Cohen, and O. Shochet. Novel type of phase transition in a system of self-driven particles. *Physical Review Letters* 75, 1226–1229, 1995.
53. J. Viventi et al. Flexible, foldable, actively multiplexed, high-density electrode array for mapping brain activity in vivo. *Nature Neuroscience* 14, 1599–1605, 2011.
54. I. Wagner and A. Bruckstein. Special Issue on Ant Robotics. *Annals of Mathematics and Artificial Intelligence*, 31(1-4), 2001.
55. C. Westhus, C.J. Kleineidam, F. Roces and A. Weidenmuller. Behavioural plasticity in the fanning response of bumblebee workers: impact of experience and rate of temperature change. *Animal Behavior*, in press, 2012.
56. A.T. Winfree. Biological rhythms and the behavior of populations of coupled oscillators. *J. Theor. Biololgy*. 16: 15–42, 1967.