Reaching Consensus in Ad-Hoc Diffusion Networks

Dariusz R. Kowalski[1,2] and Jarosław Mirek[1](✉)

[1] Department of Computer Science, University of Liverpool, Liverpool, UK
{D.Kowalski,J.Mirek}@liverpool.ac.uk
[2] SWPS University of Social Sciences and Humanities, Warsaw, Poland

Abstract. We consider an algorithmic model of diffusion networks, in which n nodes are distributed in a 2D Euclidean space and communicate by diffusing and sensing molecules. Such a model is interesting on its own right, although from the distributed computing point of view it may be seen as a generalization or even a framework for other wireless communication models, such as the SINR model, radio networks or the beeping model. Additionally, the diffusion networks model formalizes and generalizes recent case studies of simple processes in environment where nodes, often understood as biological cells, communicate by diffusing and sensing simple chemical molecules.

To demonstrate the algorithmic nature of our model, we consider a fundamental problem of reaching consensus by nodes: in the beginning each node has some initial value, e.g., the reading from its sensor, and the goal is that each node outputs the same value.

Our deterministic distributed algorithm runs at every node and outputs the consensus value equal to the sum of inputs divided by the sum of the channel coefficients of each cells. For a node v consensus is reached in $\mathcal{O}\left(\log_\rho\left(\frac{1}{2}\sqrt{\frac{d_{min}}{d_{max}}}\cdot\frac{d_v}{b\sum_i d_i}\right)\right)$ communication rounds, where d_v is the sum of molecule reachability ratios to node v in the medium, $d_{min} = \min_i d_i$, $d_{max} = \max_i d_i$, and b is the sum of the initial values. ρ represents the second largest eigenvalue of a matrix of normalised molecule reachability ratios, that we analyze together with an associated Markov Chain.

Keywords: Ad hoc networks · Cells · Diffusion networks · Molecules · Consensus

1 Introduction

The consensus problem in distributed systems, where multiple processes want to agree on a certain value, is considered to be fundamental for many fault-tolerant and sensors' systems [17]. We consider the consensus problem in an ad-hoc network of cells with a diffusion-based molecular communication model.

Supported by Polish National Science Center (NCN) grant UMO-2017/25/B/ST6/02553.

S. Gilbert et al. (Eds.): ALGOSENSORS 2018, LNCS 11410, pp. 180–192, 2019.
https://doi.org/10.1007/978-3-030-14094-6_12

1.1 Previous Work

The consensus problem has been widely considered in distributed systems in several papers; here we only sample some relevant work. The authors in [1] focused on solving the classic consensus problem in a network of n mobile nodes, where the communication in the environment is considered unpredictable, their main goal was to reach convergence on a common consensus value by all the n nodes. The authors assumed that each node has its own identifier, the communication is synchronous, and when a message is delivered the destination is not certain. The proposed protocols solved the consensus problem with crashes and Byzantine failures. The model in [18] studied systems of autonomous agents with biologically motivated interactions and the evolving of self-ordered motion in these systems. According to a local rule, the agents are updated from time to time. The authors in [18] show that despite the changing neighborhood of each agent and absence of centralized coordination, as the system gradually evolves, all agents eventually move in the same direction.

In [16] a profound study of the present consensus algorithms and convergence was presented accompanied by the performance analysis of these algorithms. It reveals collaboration between various fields of engineering and science like complex networks, distributed computing, graph theory, and Markov chains. The idea of cooperation among dynamic systems is explained through elaborated investigation of formation control for multi-vehicle systems. The connection between spectral and structural properties of complex networks and the speed of information propagation of consensus algorithms is demonstrated. The authors in [15] presented a method for solving the consensus problem in distributed systems. The consensus problem in this method is considered on two levels. The first level represent general methods that can help in solving multi-valued and multi-attributed conflicts. The second level relates to different applications of consensus methods in solving various conflicts that may occur in distributed systems.

1.2 Related Work in Molecular Communication

Few research scopes [7–9] focused on studying the consensus problem in the molecular communication model. In the most relevant work [7], the authors explored consensus-type processes under simplified diffusion based molecular communication. They focused on studying the convergence of a random process of exchanging information about an event (measurement) through a diffusion based network. Through communication, all nodes try to obtain the best estimate. Additionally, they consider two simplified cases of diffusion environments, including distance unification and random deployment of cells.

1.3 Our Results

In this paper, we consider an ad-hoc network where nodes are deployed in a two-dimensional medium and communicate with each other by diffusing and sensing molecule concentrations. This model has been designed as a complete graph with weighted edges. We assume that the edge weight (that we called a channel coefficient up to this point) between two nodes depends on the distance between them, as signal strength in diffusion is inversely proportional to the cube of the distance [14], as well as other diffusion parameters. It is however important to highlight that the weights of the graph are not known a priori.

The most notable novelty of this paper is that the model is very basic and there are few assumptions that the system has to fulfill in order to execute the algorithm. This allows to think that our setup is actually a generalization or a framework for models such as the SINR [11,12]. Precisely, in the SINR model signals transmitted by processors weaken polynomially with respect to distance. Nevertheless the strongest signal contains a legible information and stations are distinguishable by having their unique identifiers. It is worth noticing that in the diffusion-based molecular model signals weaken in a similar manner. On the contrary, however, we do not distinguish nodes with identifiers and the information is actually just the amount of molecules - these similarities remind of the beeping model [5,6].

We assume that each node has its initial value, and these values are supposed to be an estimation of a parameter in the environment. The eventual goal is that nodes share their values between each other until they reach an agreement (consensus) about the average of these values. Nevertheless in biologically-motivated systems many processes are controlled by some overriding processes. Justification is to be found later in this paper. This leads to the notion of an eventual consensus that we introduce - even though the algorithm converges to a consensus, it is controlled by an overriding mechanism. Hence, each node does not know how long it should exchange information, yet will be notified when it is done.

In [7], the authors analyzed the processes of convergence of information-exchange between randomly distributed nodes in a diffusion medium. We generalize that work by proposing a model where nodes are placed arbitrarily. We propose a deterministic distributed algorithm which runs at every node and eventually computes the consensus value equal to the sum of inputs divided by the sum of channel coefficients. For a node v consensus is reached in $O\left(\log_\rho\left(\frac{1}{2}\sqrt{\frac{d_{min}}{d_{max}}}\cdot\frac{d_v}{b\sum_i d_i}\right)\right)$ rounds, where d_v is the sum of the *molecule reachability ratios* i.e. $\sum_i c_{iv}$, where a molecule reachability ratio c_{iv} between nodes i and v is the percentage of molecules sent by i that reaches v in a communication round, $d_{min} = \min_i d_i$, $d_{max} = \max_i d_i$, and b is the sum of the initial values. ρ represents the second largest eigenvalue of a special matrix of normalised reachability ratios that we used for our analysis (c.f. Sect. 3).

Finally, we also elaborate on how the length of a round influences the total time of consensus (defined as the number of rounds multiplied by the round duration) and propose study on local consensus for such systems.

1.4 Structure of the Paper

This paper is organized as follows. In Sect. 2 we give a detailed description of our model. Then we introduce the proposed consensus algorithm and its analysis in Sect. 3. Section 4 discusses time adjustments for optimizing round duration and how the local consensus in a molecular environment is implied by the eventual consensus. Finally, Sect. 5 combines conclusions and prospective future work.

2 Model

2.1 Network Environment

We consider an ad doc diffusion network consisting of n nodes, also called *cells*, deployed arbitrarily in a two dimensional Euclidean space. Nodes are identical and anonymous, in particular, they do not have distinguishing identifiers and have limited computational power. They also form an *ad-hoc* network structure, in the sense that they do not posses any a priori knowledge about network topology. In the remainder of the paper we use letters i, j, k to distinguish nodes for the purpose of notation and analysis only.

2.2 Communication Model

Communication between nodes is by diffusion (also called transmission) and sensing of molecules. More precisely, each node i could decide to diffuse an amount Q of molecules at time τ, and any other node j of distance d from node i can sense altogether

$$c(Q, d, T) = \int_{\tau}^{\tau+T} Q \cdot \frac{1}{(4\pi D t)^{\frac{2}{3}}} \cdot \exp(\frac{-d^2}{4Dt})\ \delta t \tag{1}$$

of these molecules within time interval $[\tau, \tau + T]$, where D is the diffusion coefficient, resulting from the communication medium. In case when more than one node diffuses molecules, a receiver node j accumulates the sensed molecules through the summation of the values $c(Q, d, T)$ over diffusing nodes i, i.e., in the interval $[\tau, \tau + T]$ node j senses

$$\sum_i c(Q_i, \mathrm{dist}(i, j), T_i) - c(Q_i, \mathrm{dist}(i, j), [T_i - T]_+)$$

molecules in total, where $\mathrm{dist}(i, j)$ is the Euclidean distance between i and j, T_i is the time that passed from diffusion of node i up to time $\tau + T$ (i.e., node i diffused at time $\tau + T - T_i$), Q_i is the amount of molecules diffused by i at that time, and $[T_i - T]_+$ equals to $\max\{T_i - T, 0\}$. In other words, the receiver senses the total amount of molecules that have been in its nearest proximity in the time interval $[\tau, \tau + T]$ without being able to distinguish which molecules come from which transmitter. In this sense, molecules are indistinguishable.

2.3 Control Variables

In the bloodstream there is a mechanism, called chemotaxis which gives cells the ability to migrate in a specific direction. During chemotaxis, cells move in response to an external signal, most frequently a small molecule, known as a chemoattractant. When cells sense the concentration of the chemical, they move in the direction where the concentration of that chemical is higher. Such a mechanism of directional movement may be observed while wound healing [3]. Consequently, cells' migration stops when they no longer sense the external signal of the chemoattractant chemical.

Another example of cell movement are the lymphocyte responses to [Ca2+] signals, ranging from short-term cytoskeletal modifications to long-term changes in gene expression [10]. These are only few examples of how biological activities end.

Such biological processes motivated us to introduce *control variables* that allow some *oracle* to provide additional information to the algorithm during the execution. In this work we use it in a very basic form to recognize (e.g., based on some external system behavior) whether the task of consensus has been actually done in the system or not.

2.4 Rounds and Synchronization

We assume that nodes are synchronized and can work in rounds of some predefined length T. We assume that T is a system parameter and discuss its length in Sect. 4. However this parameter strongly implies the topology of the network.

Nodes which want to diffuse molecules do it in the beginning of a round, and all nodes sense the aggregated level of molecules in their close proximities during the whole round. At the end of the round, we assume that the amount of molecules drops to zero; in practice, in biological systems this could be guaranteed by the ability of cells to absorb the molecules remaining at the end of the round in negligible time, or releasing other type of molecules that chemically transform the remaining communication molecules into some other type of molecules without any implications for communication.

Consequently, inspired by a model from [13] we assume that nodes can utilize molecules that were used for communication through the use of a mechanism known as *Destroyer molecules*, which help controlling the communication channel by eliminating remaining molecules from the environment. The authors in [13] assume that destroyer molecules are immobile in the environment and their size is bigger compared to information molecules. Besides, when an information molecule gets close to a destroyer molecule, it binds to it and is removed from the environment, due to the chemical attraction between them.

The conclusion is that treating the system as synchronous is possible because cells perform a repeating sequence of: diffusing molecules, sensing the concentration of molecules and cleaning the environment. Parameter T is therefore strongly connected with sensing concentrations because the longer the round the further broadcasts may be heard. The propagation of molecules across the medium in diffusion based communication is due to spontaneous diffusion, which is a stochastic process and occurs at a much lower speed, i.e. the order of a few millimeters per second. Finally, after the sensing phase, the environment is cleaned and a new round begins.

2.5 (Eventual) Consensus Problem

In the consensus problem, in the beginning each node has its initial value. The goal is that all nodes output the same value. We assume that initial values b_i, for a node i, are in $[0, 1]$, which covers a wide spectrum of potential input types (i.e., many other input ranges could be easily transformed 1–1 into values in $[0, 1]$). Let $I = (b_1, b_2, \ldots, b_n)$ denote the vector of initial values.

The definition of consensus is actually meaningless because it has been already shown in classical distributed systems that if the nodes do not have knowledge about the network topology, then it may be impossible to compute any non-constant function even on a ring, see e.g., [2] for details. Consequently, in such a weak model as the diffusion-based networks some computations may not be exact and are of a different nature. This brings us to the concept of *eventual consensus*, considered before in weak distributed systems: there is a round starting from which candidate values stored at nodes are the same. Note that in some executions nodes may not know this stabilization round, even though it exists.

2.6 Performance Measure

We consider time as the performance measure, defined in two ways. First, as the number of rounds in which all nodes reach consensus, regardless of the initial setting. Second, also called the *total time*, as the number of rounds multiplied by the round duration; we will show that the total time of an algorithm may depend on the actual duration of a round.

3 Concellsus - The Consensus Algorithm in Diffusion Networks

In this section we introduce a deterministic algorithm Concellsus that allows cells to arrive at a consensus in a diffusion-based ad hoc network.

3.1 Technical Preliminaries

The nature of a diffusion-based model allows us to think that nodes/cells deployed in a medium form a distributed system, which can be represented as a complete graph with weighted edges and self loops added to each node. More formally, we denote by $\mathcal{N} = (V, E)$ the graph representing a given ad-hoc network, where $V = \{1, 2, \ldots, n\}$ and $E = V \times V$. Notice that we allow self-loops in the network graph, as when a cell diffuses a certain amount of molecules, it may also sense it from the medium. For an edge $(i, j) \in E$, we associate a weight c_{ij} equal to $c(1, \mathrm{dist}(i, j), T)$, where T is the length of a round. We call this weight the *molecule reachability ratio* between cell i and cell j. Furthermore, it is worth noticing that for any i, j we have that $c_{ij} = c_{ji}$.

Let $C = \{c_{ij}\}$ be the matrix of edges' weights in network $\mathcal{N}$. We denote the sum of weights of edges connected to node i by $d_i = \sum_{j \in V} c_{ji}$, and call it the *impact* of cell i. We associate the stochastic matrix $P = \{p_{ij}\}$, where $p_{ij} = \frac{c_{ij}}{d_i}$, with network $\mathcal{N}$. We call it the *matrix of normalised molecule reachability ratios*.

3.2 The Algorithm

Algorithm 1. Concellsus, code for node i and input b_i; control variable `process`

```
Input: $b_i$, process
Output: $w_i$
diffuse a unit of molecules to all neighbors of $i$;
receive value $d_i$;
initialize $value_i$ to $\frac{b_i}{d_i}$;
while process do
    diffuse $value_i$ (to all neighbors of $i$);
    receive $received_i$ (accumulated from all neighbors $j$ of $i$ or $i$ itself);
    update $value_i := \frac{received_i}{d_i}$;
end
$w_i := [\frac{value_i}{d_i}]$
```

Nodes start with some initial local measurements resulting from the environment, which could be arbitrary. Recall that we denote the initial sequence of values as $I = (b_1, \ldots, b_n)$.

Having those initial measurements, nodes start exchanging values, stored as a local variable $value_i$ at some node i, in order to arrive at a consensus. They proceed in rounds, starting from round 0, which is slightly different from the other rounds. In round 0, each node diffuses a *unit* of molecules, and during the round, senses/receives some accumulated value (of molecules) from all its neighbors. The node stores it as d_i. We may think that cell i receives a form of a weighted average, which in case of a diffusion of units of molecules corresponds to value d_i. We will call this the *impact* of node i.

In the main part of the algorithm each node diffuses its initial measurement divided by its impact to all the neighbors. During the round, each node i senses the medium to receive some accumulated value (of molecules) from all its neighbors and then updates $value_i$ with $\frac{received_i}{d_i}$. Intuitively, cells that exist farther from cell i have a smaller impact on the actual $value_i$ (in a single round) than cells living closer. Thus, if some cell j with a huge measurement is very far from cell i then this value will more likely "reach" i via multiple other cells (which will be sensing and accumulating a large portion of this value) rather than directly.

Let us take a closer look at the main loop of Concellsus, c.f., Algorithm 1. Motivated by biological processes and the way how they finish, we assume having a control variable `process` that indicates whether cells should continue exchanging information or not. For algorithmic reasons, we treat `process` as a boolean variable, however we may think that this is an overriding mechanism that controls the protocol, as it was motivated in Sect. 2.3.

The main loop in Concellsus represents the molecule exchange between cells in consecutive rounds. Each iteration consists of three steps. (1) Initially a cell i diffuses its own actual value to the medium and then (2) receives the values, modified by the formula of the communication medium, from all the neighbors. The value represents the sum $\sum\{\frac{value_j}{d_j}c_{ij} : j$ is a neighbor of i or $j = i\}$, where value c_{ij} is the weight factor between the cells (the farther the cells, the smaller the value received). Finally, (3) cell i updates and rescales its own value accordingly with its sum of weights d_i that it learned in the initial round.

Cells, or devices of nanometer order size are considered as very simple and of weak computational capabilities. Consequently, the computations may not necessarily need to be of a significant precision. One may imagine that a group of cells needs to decide which direction they should move. It is somehow more natural that the cells will decide on a general direction and repeat the procedure after some movement - especially if the topology of the medium is not known. As a result, we round the consensus value to a natural number when the estimation of each cell is sufficiently close to that number.

Eventually, in the described process, each cell will compute a sort of an average from all the initial measurements, hence make a consensus. The main loop terminates when $value_i$ is sufficiently close to $b\pi_i$, where $b = \sum_{i\in V} b_i$ and π_i is the ith component of the stationary distribution vector. Thus, the value that the algorithm finally computes is $b\frac{d_i}{\sum_j d_j}\frac{1}{d_i}$ that actually is $\frac{value_i}{d_i}$. The most important question, from our perspective is, however, what is the convergence rate of such a process; we will analyze it in the following subsection.

3.3 Analysis of Concellsus

Each cell has an initial measurement as an input, but does not know the measurements of other cells. Cells exchange their measurements, which are implicitly weighted by the medium. This process begins with vector $I = (b_1, \ldots, b_n)$ denoting cells' initial measurements. It changes with time as cells exchange and update their values round by round. Hence the vector of current values I_x in step x corresponds to $I_x = IP^x$, where P is a stochastic matrix defined in Sect. 3.1.

We show first that the Markov chain defined by matrix P converges to a stationary distribution $\pi = (\pi_1, \ldots, \pi_n)$, where $\pi_i = \frac{d_i}{\sum_j d_j}$ for every node i.

Lemma 1. *Consider a Markov chain with transition matrix P defined in Sect. 3.1. Then its stationary distribution is $\pi = (\pi_1, \ldots, \pi_n)$ where $\pi_i = \frac{d_i}{\sum_j d_j}$ for every node i.*

Proof. Firstly, we show that $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ is a stationary distribution for the process described by P, where $\pi_i = \frac{d_i}{\sum_i d_i}$ for every node i:
As $\pi_i = \frac{d_i}{\sum_j d_j}$, then $\sum_i \pi_i = \sum_i \frac{d_i}{\sum_j d_j} = 1$, and π is a proper stationary distribution.

Let $N(i)$ be the neighborhood of i. The condition $\pi = \pi P$ together with the fact that $p_{ij} = \frac{c_{ij}}{d_i}$ means that

$$\pi_i = \sum_{k \in N(i)} \left(\frac{d_i}{\left(\sum_j d_j\right)} \frac{c_{ik}}{d_i} \right) = \frac{d_i}{\sum_j d_j},$$

what ends the proof. □

Now we need to show that the process of exchanging and updating values converges and that eventually all the nodes end with the same value. Besides we would like to know what is the rate of convergence. Let us assume that control variable *process*, driven by an external system oracle, does not switch off, or switches off at the same round after (eventual) consensus has been reached in the system.

Theorem 1. *Algorithm* Concellsus *performs an eventual consensus in* $\mathcal{O}\left(\log_\rho \left(\frac{1}{2}\sqrt{\frac{d_{min}}{d_{max}}} \cdot \frac{d_v}{b \sum_i d_i}\right)\right)$ *iterations.*

Proof. We are interested in bounding the number of iterations r after which IP^r will be sufficiently close to the vector of the consensus value. To do this, we will analyze some property of the Markov chain described by P.

Recall that matrix $P = \{p_{ij}\}$ is a stochastic matrix of a primitive, reversible Markov chain, that describes a random walk on $\mathcal{N}$. According to Lemma 1 its stationary distribution is $(\pi_1, \pi_2, \ldots, \pi_N)$. We know that the eigenvalues of P are in the following relation: $1 = \lambda_1 > \lambda_2 \geq \cdots \geq \lambda_k$. Let ρ denote the second largest eigenvalue of P.
From Lemma 2 in [4] we know that

$$p_{ij}^{(r)} = \pi_j + \mathcal{O}\left(\sqrt{\frac{\pi_j}{\pi_i}} \cdot \rho^r\right),$$

where p_{ij} corresponds to the appropriate entry of the rth power of matrix P. Taking $M_P = \max_{i,j}\{\sqrt{\pi_j/\pi_i}\}$, the matrix form of the equation above is

$$P^r = P^\infty + \mathcal{O}(M_P \cdot \rho^r \cdot E),$$

where E is the matrix with all the entries equal 1. The limit $P^\infty = \lim_{r\to\infty} P^r$ of the chain is an $N \times N$ matrix where all the entries in the ith column are equal π_i. In our case we have that $M_p = \sqrt{d_{max}/d_{min}}$. In what follows

$$P^r = P^\infty + \mathcal{O}\left(\sqrt{\frac{d_{max}}{d_{min}}} \cdot \rho^r \cdot E\right).$$

For any vector of the initial measurements of the cells I, we have that $L = IP^\infty$ is the eigenvector of P. The ith entry of L equals $b\pi_i$, where b is the sum of the input vector I (the initial measurements vector). Consequently

$$IP^r = L + \mathcal{O}\left(\sqrt{\frac{d_{max}}{d_{min}}} \cdot \rho^r \cdot b \cdot e\right),$$

where e is the row vector with all its entries equal 1.

In order to be sure that all cells compute the correct value we need to be sure that for every cell v, $c\sqrt{\frac{d_{max}}{d_{min}}} \cdot \rho^r \cdot b < \frac{1}{2}\pi_v$, for some $c > 0$ resulting from the $\mathcal{O}$ notation. The reason being is that the final step of the algorithm computes a value rounded to the closest integer. Hence, as $\pi_v = \frac{d_v}{\sum_i d_i}$ we have that

$$\rho^r < \frac{1}{2}\sqrt{\frac{d_{min}}{d_{max}}} \cdot \frac{d_v}{cb\sum_i d_i},$$

and finally

$$r < \log_\rho\left(\frac{1}{2}\sqrt{\frac{d_{min}}{d_{max}}} \cdot \frac{d_v}{cb\sum_i d_i}\right).$$

We conclude that the whole network of cells will reach a consensus in r iterations that is bounded from above by $\log_\rho\left(\frac{1}{2}\sqrt{\frac{d_{min}}{d_{max}}} \cdot \frac{d_v}{cb\sum_i d_i}\right)$. □

A final remark is that the result of the theorem concentrates on some values specific for each node. However, one may think that the overall number of iterations (for the whole system to terminate) is bounded by extreme values. Furthermore the number of rounds is defined by the base of the logarithm - value ρ, the second largest eigenvalue of the transition matrix. This leads to an obvious conclusion that the number of iterations of Concellsus differs according to the topology of the network.

4 Extensions of Results

Biologically motivated systems have many subtleties, yet to be discovered. Consequently there are numerous problems that one may encounter, even when establishing the model. In this section we present notions and ideas resulting from our investigations. On one hand they justify, to some extent, our model, but on the other they may be inspirational in the sense of follow-up work.

4.1 Optimizing Round Duration

We stated the notion of rounds in the section regarding the model, but up to this point we treated it as a priori given. One may ask what is the optimal time duration of a round.

It is worth emphasizing that the duration of round a influences the weights in the graph of the network. Clearly, if the round is longer then each transmission of a certain cell reaches further distances, and consequently higher amount of molecules is sensed by nodes during rounds; hence the corresponding weights are also greater. However, the absolute time of the algorithm execution, defined as the number of rounds multiplied by the duration of a round, may be very long. On the other hand, too short rounds may cause that the graph becomes practically "disconnected" in the sense that weights will be too small to guarantee propagation across the network in a reasonable time. Therefore, our goal is to find round duration that provides a trade-off between the above mentioned two extremes.

From the point of view of the analysis from the previous section, we know that ConCellSus requires $r = O\left(\log_{\rho}\left(\frac{1}{2}\sqrt{\frac{d_{min}}{d_{max}}} \cdot \frac{d_v}{cb\sum_i d_i}\right)\right)$ rounds to reach consensus. Hence, the absolute time needed for the algorithm to terminate is bounded by rT there T is the duration of a single round (round a together with round b). The question is what is the optimal value of T, in order to minimize expression rT.

Considering T as a factor that influences the topology of the graph leads us to a conclusion that values d_{min}, d_{max}, d_v and $\sum_i d_i$ from the bound for r, are functions of T.

Thus in order to minimize the function $u(T) = rT$, we solve $\frac{\delta u}{\delta T} = 0$ to find the argument T^* for which $u(T^*)$ is minimal. However, we need to take into consideration, that the found value T^* has to satisfy $T^* \geq T_0$, where T_0 is the minimal duration of a round required to assure connectivity of the network graph. Otherwise reaching consensus would be impossible.

4.2 Local Consensus

The classical consensus problem considered in computer science concerns the matter of achieving a consistent opinion among all nodes in a network. In the world of cells and diffusion-based communication many processes are flexible. This means that even if there are numerous cells in the medium, only a subset of them may be employed for a specified task. If one considers the circulatory system then certainly not every platelet takes part in covering a particular wound. This brought us to a notion of a local consensus.

When considering the consensus problem, cells communicate with each other in order to reach an eventual consensus, where every cell finishes with the same value. What is more, we showed the rate of convergence of such a process for the protocol proposed in the previous section. Nevertheless to the best of our knowledge processes taking place in the human body are often controlled by some overriding mechanisms or organs. This leads to the notion of a local consensus.

If a certain task is considered as performed, by an overriding mechanism, then it is likely to happen that a process will be stopped and some cells may end without reaching the eventual value. Yet such an occurrence could suffice for the considered task, when the consensus has been achieved by a subset of cells. We say that there has been a local consensus.

The local consensus is implied by the eventual consensus. Precisely, if a value is becoming closer to be agreed among the whole network of cells with divergence α, then the divergence between a subset of cells is equal α' and $\alpha' \leq \alpha$.

5 Conclusions and Open Problems

In this work we demonstrated that the presented bio-inspired model of diffusion networks is algorithmically challenging - even on the level of establishing a simple and coherent model - by the design and analysis of a consensus algorithm.

In the consensus problem, it is interesting to see to what extent the external adversary could distract such a simple ad-hoc system from convergence.

Further study includes classical distributed computing and communication problems, mobility, fault tolerance, security, or even robotics.

Acknowledgements. We would like to thank Athraa Juhi Jani for fruitful talks and insight into the literature on the subject.

References

1. Angluin, D., Fischer, M.J., Jiang, H.: Stabilizing consensus in mobile networks. In: Gibbons, P.B., Abdelzaher, T., Aspnes, J., Rao, R. (eds.) DCOSS 2006. LNCS, vol. 4026, pp. 37–50. Springer, Heidelberg (2006). https://doi.org/10.1007/11776178_3
2. Attiya, H., Snir, M., Warmuth, M.K.: Computing on an anonymous ring. J. ACM **35**(4), 845–875 (1988). https://doi.org/10.1145/48014.48247
3. Bray, D.: Cell Movements: From Molecules to Motility. Garland Science (2001)
4. Broder, A., Karlin, A.: Bounds on the cover time. J. Theoret. Probab. **2**(1), 101–120 (1989). https://doi.org/10.1007/BF01048273
5. Cornejo, A., Kuhn, F.: Deploying wireless networks with beeps. In: Lynch, N.A., Shvartsman, A.A. (eds.) DISC 2010. LNCS, vol. 6343, pp. 148–162. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15763-9_15. http://dl.acm.org/citation.cfm?id=1888781.1888802
6. Du, D.-Z., Hwang, F.K.: Combinatorial Group Testing and Its Applications, 2nd edn. World Scientific (1999). https://doi.org/10.1142/4252. https://www.worldscientific.com/doi/abs/10.1142/4252
7. Einolghozati, A., Sardari, M., Beirami, A., Fekri, F.: Consensus problem under diffusion-based molecular communication. In: 2011 45th Annual Conference on Information Sciences and Systems, pp. 1–6, March 2011. https://doi.org/10.1109/CISS.2011.5766149
8. Einolghozati, A., Sardari, M., Beirami, A., Fekri, F.: Data gathering in networks of bacteria colonies: collective sensing and relaying using molecular communication. In: 2012 Proceedings IEEE INFOCOM Workshops, pp. 256–261, March 2012. https://doi.org/10.1109/INFCOMW.2012.6193501

9. Einolghozati, A., Sardari, M., Fekri, F.: Networks of bacteria colonies: a new framework for reliable molecular communication networking. Nano Commun. Netw. **7**, 17–26 (2016). https://doi.org/10.1016/j.nancom.2015.01.003, http://www.sciencedirect.com/science/article/pii/S1878778915000046
10. Gallo, E.M., Canté-Barrett, K., Crabtree, G.R.: Lymphocyte calcium signaling from membrane to nucleus. Nat. Immunol. **7**(1), 25–32 (2006)
11. Gupta, P., Kumar, P.R.: The capacity of wireless networks. IEEE Trans. Inf. Theoret. **46**(2), 388–404 (2006). https://doi.org/10.1109/18.825799
12. Jurdziński, T., Kowalski, D.R.: Distributed randomized broadcasting in wireless networks under the SINR model. In: Kao, M.Y. (ed.) Encyclopedia of Algorithms, pp. 577–580. Springer, New York (2016). https://doi.org/10.1007/978-1-4939-2864-4_604
13. Kuran, M., Yilmaz, H.B., Tugcu, T.: A tunnel-based approach for signal shaping in molecular communication. In: 2013 IEEE International Conference on Communications Workshops (ICC), pp. 776–781, June 2013. https://doi.org/10.1109/ICCW.2013.6649338
14. Meng, L.S., Yeh, P.C., Chen, K.C., Akyildiz, I.F.: Mimo communications based on molecular diffusion. In: 2012 IEEE Global Communications Conference (GLOBECOM), pp. 5380–5385. IEEE (2012)
15. Nguyen, N.T.: Consensus system for solving conflicts in distributed systems. Inf. Sci. Inform. Comput. Sci. **147**(1–4), 91–122 (2002). https://doi.org/10.1016/S0020-0255(02)00260-8
16. Olfati-Saber, R., Fax, J.A., Murray, R.M.: Consensus and cooperation in networked multi-agent systems. Proc. IEEE **95**(1), 215–233 (2007). https://doi.org/10.1109/JPROC.2006.887293
17. Turek, J., Shasha, D.: The many faces of consensus in distributed systems. Computer **25**(6), 8–17 (1992). https://doi.org/10.1109/2.153253
18. Vicsek, T., Czirók, A., Ben-Jacob, E., Cohen, I., Shochet, O.: Novel type of phase transition in a system of self-driven particles. Phys. Rev. Lett. **75**(6), 1226 (1995)