

Article

## Distributed Consensus for Metamorphic Systems Using a Gossip Algorithm for $CAT(0)$ Metric Spaces <sup>†</sup>

Anass Bellachehab

Telecom SudParis, Institut Mines-Télécom, UMR CNRS 5157 SAMOVAR, 9 Rue Charles Fourier, 91000 Évry, France; E-Mail: anass.bellachehab@telecom-sudparis.eu ; Tel.: +33-67667424

<sup>†</sup> This paper is an extended version of my paper published in MaxEnt 2014, Amboise, France, 21–26 September 2014.

Received: 30 December 2014 / Accepted: 9 March 2015 / Published: 12 March 2015

---

**Abstract:** We present an application of distributed consensus algorithms to metamorphic systems. A metamorphic system is a set of identical units that can self-assemble to form a rigid structure. For instance, one can think of a robotic arm composed of multiple links connected by joints. The system can change its shape in order to adapt to different environments via reconfiguration of its constituting units. We assume in this work that several metamorphic systems form a network: two systems are connected whenever they are able to communicate with each other. The aim of this paper is to propose a distributed algorithm that synchronizes all of the systems in the network. Synchronizing means that all of the systems should end up having the same configuration. This aim is achieved in two steps: (i) we cast the problem as a consensus problem on a metric space; and (ii) we use a recent distributed consensus algorithm that only makes use of metrical notions.

**Keywords:** metamorphic systems

---

### 1. Introduction

Many problems in robotics, computer science and biology involve systems that can be described as reconfigurable or metamorphic. These systems change their state through a set of local rules, and in order to move from a given State A to another given State B, the system has to determine the sequence of local moves that it should perform. Examples of such systems are: metamorphic manufacturing systems [1], phylogenetic trees [2] and metamorphic robots [3].

The examples we will deal with in this paper will be about metamorphic robots, which consists of a collection of individual modules that can connect/disconnect from each other and form a rigid structure, referred to as a configuration or a state. Individual modules can change their position relative to their neighbors, as long as the whole system remains connected and according to a set of local rules. This allows the system to dynamically change its configuration and position. We assume here that we have many identical metamorphic robots that are able to communicate with each other and share information about their respective states. These robots form a communication network that we will assume to be decentralized (*i.e.*, without a central fusion node). We are interested in the problem of distributed consensus among metamorphic systems: each system is in its own initial state, and we would like all of them to end up in a common configuration.

Distributed consensus algorithms have been thoroughly studied, mainly in the cases of vector data [4–6] or ordered data (e.g., [7]). The algorithm in [5], for example, is a distributed algorithm that uses pairwise arithmetic averages of the data. However, without specific assumptions, configurations cannot be averaged. The algorithm presented in [7] relies on pairwise maximum computations. In the setting of metamorphic robots, however, configurations cannot be easily ordered, hence the need to find an appropriate framework.

In [8,9], the authors introduce a mathematical framework for analyzing metamorphic systems based on embedding the state space—the space of all possible states of a given metamorphic system—into a continuous space. This embedding, called the state complex is well suited to a recent consensus algorithm [10] that relies on pairwise midpoint computations. Indeed, the state complex can be equipped with an adequate metric, so as to yield a  $CAT(0)$  metric space. Hence, the main contribution of this paper is to propose a distributed consensus algorithm that provably converges and is well adapted to the state space of metamorphic systems.

The paper is organized as follows. The first section details the mathematical background underpinning the metamorphic systems state space embedding, as well as gives a formal description of the consensus problem. The second section exposes the random pairwise midpoint algorithm. Section 3 explains how to compute the midpoint of any two points in a cubical complex. Finally, Section 4 provides numerical simulations for the proposed consensus algorithm.

## 2. Framework

### 2.1. Metamorphic Systems: Definition and Examples

There are many examples of metamorphic/reconfigurable systems, such as reconfigurable manufacturing systems [1] and phylogenetic trees [2]. Another example, one in which we will be more interested in this paper, is that of metamorphic robots, which were described by Østergaard *et al.* in [11] as robotic systems:

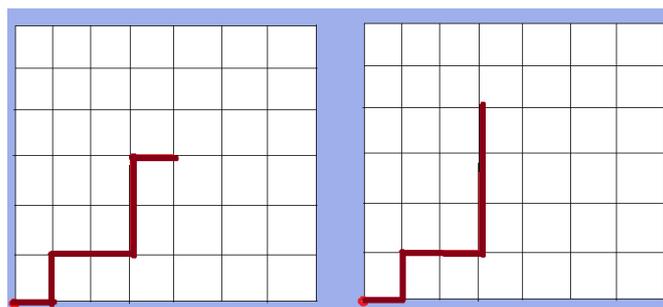
- (1) That consist of several identical and physically independent unit modules;
- (2) For which its modules can be connected to each other in many possible ways in order to form rigid structures;
- (3) For which its modules can disconnect and reconnect while the system is active;

(4) For which it can change the way its modules are connected, *i.e.*, it is fully automatic.

Some of these robots are lattice-based, meaning that the robotic modules occupy a discrete set of possible positions, this set of possible positions forming a lattice. The nature of the lattice depends on the geometry of the modules: it can be hexagonal, squared, dodecahedral, *etc.* These metamorphic systems can be mathematically described as a collection of states on a graph (see Appendix A).

Representing the various states of a metamorphic system by their lattice configuration will prove to be insufficient for finding a simple consensus protocol. Indeed, the lattice representation does not provide an ordering of states. Following [8], we represent a system configuration as a point in a cubical complex  $\mathcal{S}$ , called the state complex (see, also, Appendix B for a definition). The zero-dimensional skeleton of this complex is the set of states, and two vertices are linked by an edge if their corresponding states differ by a single action of a generator (see Appendix A). A  $k$ -cube of the complex represents  $k$  commutative movements, *i.e.*, movements that are non-overlapping whatever their order (see [8] for a rigorous definition).

An example of a metamorphic system is the robotic arm, which consists of attached links, inside a grid with one of its extremities attached to the base point  $(0, 0)$  of the grid (see Figure 1).



**Figure 1.** Example of a lattice-based reconfigurable system: the robotic arm. The edges in color indicate the presence of a unit module; a black edge indicates its absence. The arm is attached at its base point  $(0, 0)$ . Here, an elementary movement has been performed by the module at the end of the arm, which changes the overall form of the system.

We assume in this work that the metamorphic systems form a network; two systems are connected whenever they are able to communicate with each other. The aim is to synchronize all of the systems that compose the network, *i.e.*, all of the systems should have the same configuration, as shown in the example of Figure 2.

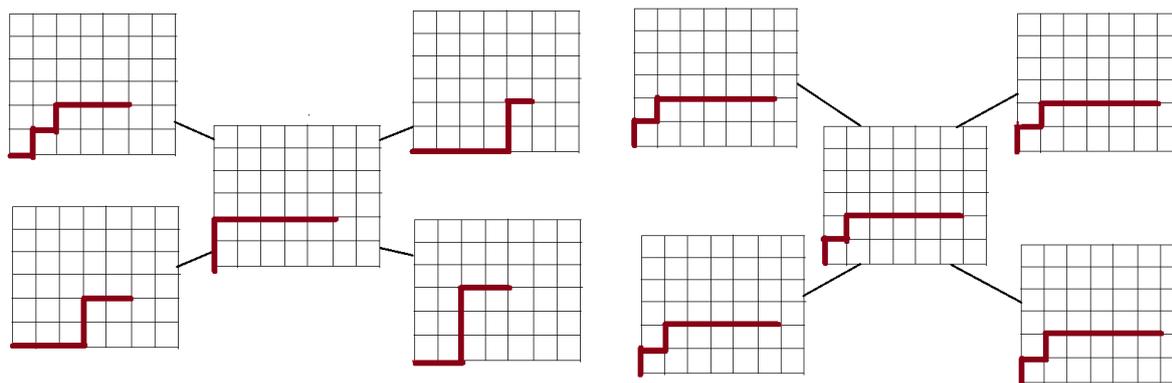
Having described the model chosen for metamorphic systems, we next review the mathematical framework of the consensus problem.

## 2.2. Framework of the Consensus Problem

### 2.2.1. Network

Following the approach of [5], we model the network of metamorphic systems by a connected graph  $G = (V, E)$  with vertices  $V$  representing the metamorphic systems (agents) and whose

edges  $E$  represent the communication links between these agents. We assume the graph to be undirected, which means that if an agent can communicate with another agent, then the converse is also true; this hypothesis is not too unrealistic if we suppose that all of the agents are identical and that the movement speed of the agents is very small compared to their communication speed. When a communication link exists between two agents, we say that the two agents are neighbors. We denote by  $\mathcal{N}(v)$  the set of all neighbors of the agent  $v \in V$ .



**Figure 2.** In this example, we are given five robotic arms. In the leftmost figure (describing the initial state), each arm has its own configuration. The rightmost figure represents a consensus state, in which all of the arms share a common configuration.

As in [5], we assume that the time model is asynchronous, *i.e.*, that each agent has its own Poisson clock that ticks with a common intensity  $\lambda$  (the clocks are identically made), and moreover, each clock is independent of the other clocks. When an agent clock ticks, the agent is able to perform some computations and wake up some neighboring agents. This time model has the same probability distribution as a global single clock ticking with intensity  $N\lambda$  and selecting uniformly randomly a single agent at each tick. This equivalence is described, e.g., in [5]. Notice also that link  $e = \{v, w\}$  is not necessarily used by agents  $v$  and  $w$  at a given time:  $v$  or  $w$  might not be awakened.

### 2.3. Communication

At a given time  $k$ , we denote by  $V_k$  the agent whose clock ticked and by  $W_k$  the neighbor that was in turn awoken. Therefore, at time  $k$ , the only communicating agents in the whole network are  $V_k$  and  $W_k$ . A single link is then active at each time; hence, at a given time, most links are not used. We assume that  $(V_k, W_k)$  are independent and identically distributed and that the distribution of  $V_k$  is uniform over the network, while the distribution of  $W_k$  is uniform in the neighborhood of  $V_k$ . More precisely, the probability distribution of  $(V_k, W_k)$  is given by:

$$\mathbb{P}[V_k = v, W_k = w] = \begin{cases} \frac{1}{N} \frac{1}{\text{deg}(v)} & \text{if } v \sim w \\ 0 & \text{otherwise} \end{cases}$$

Notice that this probability is not symmetric in  $(v, w)$ . It is going to turn out to be convenient to also consider directly the link  $\{V_k, W_k\}$ , forgetting which node was the first to wake up and which node was second. In this case,  $\mathbb{P}[\{V_k, W_k\} = \{v, w\}]$  is of course symmetric in  $(v, w)$ . One has:

$$\mathbb{P}[\{V_k, W_k\} = \{v, w\}] = \begin{cases} \frac{1}{N} \left( \frac{1}{\deg(v)} + \frac{1}{\deg(w)} \right) & \text{if } v \sim w \\ 0 & \text{otherwise} \end{cases}$$

The communication framework considered here is standard [5].

### 2.3.1. Data Space

Each node  $v \in V$  can store a value  $x_v \in \mathcal{S}$  that lies in a cubical complex  $\mathcal{S}$  (see Appendix B).

We assume that the cubical complex is equipped with the metric  $d$  induced from the Euclidean metric on each cube. The skeleton of the complex is a metric graph called the transition graph and is denoted.

Initially, each node  $v$  has an initial value  $x_v(0)$ , and  $X_0 = (x_1(0), \dots, x_N(0))$  is the tuple of initial values. A consensus state has the form  $X_\infty = (x_\infty, \dots, x_\infty)$  with:  $x_\infty \in \mathcal{S}$ . We denote by  $x_v(k)$  the value stored by the agent  $v \in V$  at time  $k$  and  $X_k = (x_1(k), \dots, x_N(k))$  the global state of the system at that instant.

We want to apply a distributed algorithm in order to achieve the consensus state.  $(\mathcal{S}, d)$  being a metric space, a candidate algorithm could be the average pairwise midpoint [10] algorithm. This algorithm requires  $(\mathcal{S}, d)$  to be a  $CAT(0)$  metric space.

**Definition 1.** Let  $(M, d)$  be a metric space. A geodesic curve  $c$  in  $X$  is a map  $c : [0, l] \rightarrow X$  from a closed interval  $I = [0, l]$  to  $X$ , such that, for all  $t, t' \in I$ :

$$d(c(t), c(t')) = |t - t'|$$

A metric space is said to be geodesic if and only if, for any two points  $x, y \in X$ , there exists a geodesic  $c : [0, l] \rightarrow X$ , such that  $c(0) = x$  and  $c(l) = y$ .

**Definition 2** ( $CAT(0)$  inequality). Assume  $(X, d)$  is a geodesic metric space (a metric space in which any two points can be related by a geodesic) and  $\Delta = (c_0, c_1, c_2)$  is a geodesic triangle with vertices  $p = c_0(0)$ ,  $q = c_1(0)$  and  $r = c_2(0)$ . Let  $\bar{\Delta} = (\bar{p}, \bar{q}, \bar{r})$  denote a comparison triangle (a triangle with the same edge lengths as  $\Delta$ ) in the Euclidean space  $\mathbb{R}^2$ .  $\Delta$  is said to satisfy the  $CAT(0)$  inequality if, for any  $x = c_0(t)$  and  $y = c_2(t')$ , one has:

$$d(x, y) \leq \bar{d}(\bar{x}, \bar{y})$$

where  $\bar{x}$  is the unique point of  $[\bar{p}, \bar{q}]$ , such that  $d(p, x) = \bar{d}(\bar{p}, \bar{x})$ , and  $\bar{y}$  on  $[\bar{p}, \bar{r}]$ , such that  $d(p, y) = \bar{d}(\bar{p}, \bar{y})$ .

A geodesic metric space is said to be locally  $CAT(0)$  if any geodesic triangle of sufficiently small perimeter verifies the  $CAT(0)$  inequality. It is said to be globally  $CAT(0)$  if any geodesic triangle verifies the  $CAT(0)$  inequality.

For a thorough introduction to the subject, see [12,13].

Any state complex can be shown to be a locally  $CAT(0)$  space [14]. The global  $CAT(0)$  propriety requires an additional constraint on the state complex. In [14], a combinatorial criterion based on the notion of posets with inconsistent pairs is provided to verify whether a state complex is globally  $CAT(0)$ .

**Assumption 1.** We shall make the fundamental assumption that the state complex of any metamorphic system involved in this paper is globally  $CAT(0)$ . This assumption, while restrictive, still covers many interesting examples, like the robotic arm [3], phylogenetic trees [2], the hexagonal system [8], etc.

The following proposition links the existence and uniqueness of geodesics and midpoints with the  $CAT(0)$  property:

**Proposition 1.** If  $x$  and  $y$  are two points in a globally  $CAT(0)$  space  $X$ , then there is but one geodesic  $\gamma : [0, 1] \rightarrow X$ , such that  $\gamma(0) = x$  and  $\gamma(1) = y$ , which we will denote from now on as  $[x, y]$ . The midpoint of  $x$  and  $y$  is defined as  $\langle \frac{x+y}{2} \rangle := [x, y](\frac{1}{2})$  and is always well defined and unique.

Now that the working framework is set (connected and undirected communications graph, Poisson clocks, data space, etc.), we will propose in the next section a consensus algorithm based on the gossip protocol [5] adapted to  $CAT(0)$  metric spaces.

### 3. Algorithm

#### 3.1. Description

In this section, we expose a distributed algorithm that relies on distributed midpoint computation in  $\mathcal{S}$  to drive a system of identical metamorphic systems into a consensus configuration—the random pairwise midpoint algorithm [10]—which works as follows: At each count of the virtual global clock, one node  $v$  is selected uniformly randomly from the set of agents  $V$ . The node  $v$  then randomly selects a node  $w$  from the neighbors of  $v$  in the communications graph. Both node  $v$  and  $w$  then compute and update their value to  $\langle \frac{x_v + x_w}{2} \rangle$ .

This algorithm is well defined, since for any couple of points  $x, y \in \mathcal{S}$ , their midpoint exists and is unique. This is due to the fact that  $\mathcal{S}$  is a globally  $CAT(0)$  space.

---

#### Algorithm 1 Random Pairwise Midpoint.

---

**Input:** a graph  $G = (V, E)$  and the initial nodes configuration  $X_v(0), v \in V$

**for all**  $k > 0$  **do**

At instant  $k$ , uniformly randomly choose a node  $V_k$  from  $V$  and a node  $W_k$  uniformly randomly from  $\mathcal{N}(V_k)$ .

Update:

$$X_{V_k}(k) = \left\langle \frac{X_{V_k}(k-1) + X_{W_k}(k-1)}{2} \right\rangle$$

$$X_{W_k}(k) = \left\langle \frac{X_{V_k}(k-1) + X_{W_k}(k-1)}{2} \right\rangle$$

$$X_v(k) = X_v(k-1) \text{ for } v \notin \{V_k, W_k\}$$

**end for**

---

This algorithm belongs to the class of consensus protocols, which has been shown to achieve convergence towards a consensus state in [15]; the convergence rate is established as linear in [10]. This assumes, however, that it is possible to compute the midpoint of any two points with reasonable complexity. While there is no closed expression for the midpoint of any two points in  $\mathcal{S}$ , there is a procedure described in the next section that permits the computation of the midpoint.

### 4. Computing the Midpoint

In [16], an algorithm is given for computing means and medians in general  $CAT(0)$  metric spaces; we are here interested in finding the midpoint  $\langle \frac{x+y}{2} \rangle$  of any two given points  $x$  and  $y$  of a  $CAT(0)$  cubical complex  $\mathcal{S}$ , so we first determine the geodesic  $[x, y]$  between  $x$  and  $y$ .

Let  $x = (x_1 \dots x_N)$  and  $y = (y_1 \dots y_N)$  be the coordinates of  $x$  and  $y$  in the current standard embedding of  $\mathcal{S}$  in  $\mathbb{Z}^N$  (see Appendix C for an exposition of the standard embedding).

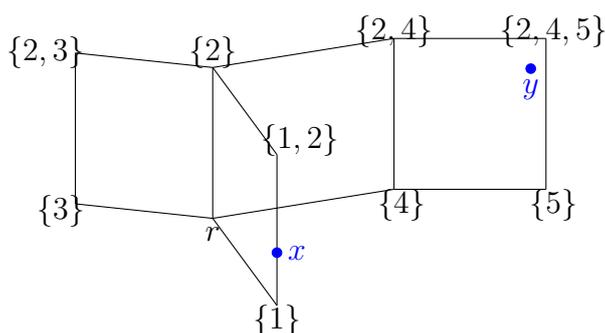
Then, we define  $v = (v_1, \dots, v_N)$  and  $w = (w_1, \dots, w_N)$ , such that for  $i \in \{1, \dots, N\}$ :

$$\begin{cases} v_i = 0 & \text{if } x_i \leq 0.5 \\ v_i = 1 & \text{if } x_i > 0.5 \end{cases}$$

Additionally:

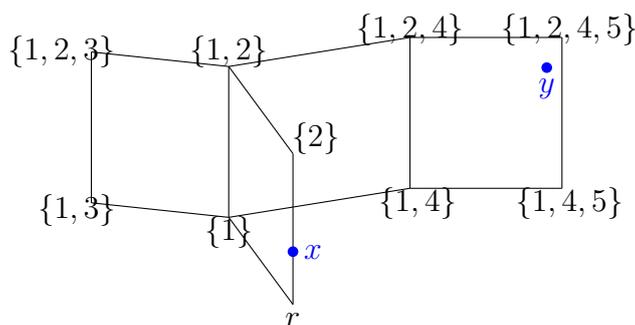
$$\begin{cases} w_i = 0 & \text{if } y_i \leq 0.5 \\ w_i = 1 & \text{if } y_i > 0.5 \end{cases}$$

$v$  and  $w$  are, respectively, the closest vertices of  $\mathcal{S}$  to  $x$  and  $y$  (see Figure 3).



**Figure 3.** Example of a cubical complex with an initial root vertex  $r$ . We want the geodesic between the points  $x = (1, 0.25, 0, 0, 0)$  and  $y = (0, 0.75, 0, 1, 0.75)$ .

Next, we re-root the cubical complex at  $v$  and change the labeling of its vertices consequently, as well as the coordinates of  $x$  and  $y$ . To obtain the new coordinates of any point  $a = (a_1, \dots, a_N)$  in the new standard embedding, we update:  $a_{i,new} = 1 - a_{i,old}$  if  $v_i = 1$  and  $a_{i,new} = a_{i,old}$  if  $v_i = 0$  (see Figures 3 and 4).



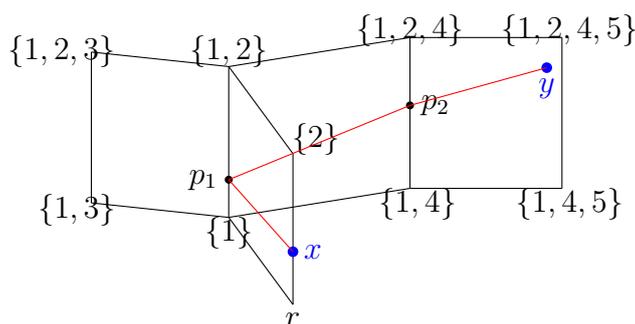
**Figure 4.** We re-root the cubical complex at  $v$ , and the new coordinates are:  $x = (0, 0.25, 0, 0, 0)$  and  $y = (1, 0.75, 0, 1, 0.75)$ . The coordinates of  $w$  are  $(0, 1, 0, 1, 1)$ .

We denote by  $I_w = \{i \in \{1, \dots, N\} | w_i = 1\}$ , the order ideal associated with the vertex  $w$ . Now that the complex is re-rooted, we need to find the cubical sequence that contains the geodesic between  $x$  and  $y$ . According to [14], a valid sequence  $(C_k)_{1 \leq k \leq n} = (I_k, M_k)_{1 \leq k \leq n}$  of cubes containing the geodesic, consists of a sequence of order ideals  $I_1 \subset I_2 \subset \dots \subset I_n = I_w$  and maximal antichains  $M_k \subset I_k, 1 \leq k \leq n$  (see Appendix B for the representation of individual cubes of a complex). To find this sequence, we take the subset of minimal elements of  $I_w$  and  $I_1$ , and set  $M_1 = I_1$ . Then, in order to form  $C_k = (I_k, M_k)$  from  $C_{k-1} = (I_{k-1}, M_{k-1})$ , we take the subset of minimal elements  $m_k$  of  $Q - I_{k-1}$ , put  $I_k = I_{k-1} \cup m_k$  and let  $M_k$  be the maximal antichain of  $I_k$ . Using this procedure until  $I_k = P$ , we obtain a valid cube sequence, which contains the geodesic. In the example of Figures 3 and 4, we have:  $I_1 = \{2, 3\}, M_1 = \{2, 3\}$  and  $I_2 = \{2, 3, 4\}, M_2 = \{2\}$ .

After the cube sequence has been determined, we then have to find the ‘breakpoints’ from which the geodesic passes. The points  $\{p_1, \dots, p_{n-1}\}$  are such that:  $\|x - p_1\| + \|p_1 - p_2\| + \dots + \|p_{n-1} - p_n\| + \|p_n - y\|$  is minimal and that for each  $k \in \{1, \dots, n - 1\}$   $p_k \in F_k$  with  $F_k$  being the common frontier of  $(I_k, M_k)$  and  $(I_{k+1}, M_{k+1})$ , which is the cubical cell  $(I_k, M_k \cap M_{k+1})$ . This problem can be cast as a touring problem with  $n$  polyhedral regions and  $2n$  facets [14].

$$\begin{aligned} &\min t_0 + t_1 + \dots + t_n \\ &\forall k \in \{0 \dots n\} : t_k \geq \|p_k - p_{k+1}\| \\ &p_k \in F_k, \quad k = 1 \dots n \\ &p_0 = x, p_{n+1} = y \end{aligned}$$

The touring problem is a second order cone optimization problem, for which numerical solvers exist. We solve the touring problem and obtain a series of points  $p_1 \dots p_n$  and the distances  $t_1 \dots t_{n-1}$  between  $p_k$  and  $p_{k+1}$  (see Figure 5).



**Figure 5.** In this example, the cube complex and the points  $x$  and  $y$  are such that we have a non-co-linear sequence of points  $\{p_1, p_2\}$  that the geodesic between  $x$  and  $y$  (drawn in red) passes through. The cube sequence  $\{C_1, C_2, C_3\}$  contains this geodesic.

Knowing the sequence of points  $(p_i)_{1 \leq i \leq n}$  and distances  $(t_i)_{0 \leq i \leq n}$ , we want to determine the midpoint of  $x$  and  $y$ ; we have two cases:

Case 1:  $n > 0$ .

First, we determine the cubical cell of  $(C_k)_{1 \leq k \leq n+1}$  that contains the midpoint. Let  $i_0 = \min \left\{ 1 \leq i \leq n \mid d(x, p_i) \geq \frac{d(x,y)}{2} \right\}$ ; then, we have:  $\langle \frac{x+y}{2} \rangle \in C_{i_0}$  and  $\langle \frac{x+y}{2} \rangle \in [p_{i_0}, p_{i_0+1}]$ . Using the convention  $p_0 = x$  and  $p_{n+1} = y$ , the analytic expression of  $\langle \frac{x+y}{2} \rangle$  is:

$$\left\langle \frac{x+y}{2} \right\rangle = p_{i_0} + \frac{D}{t_{i_0}}(p_{i_0+1} - p_{i_0})$$

with:

- $D = \frac{1}{2} \left| \sum_{k=0}^{i_0-1} t_k - \sum_{k=i_0+1}^{n+1} t_k - t_{i_0} \right|$ : If  $2 \leq i_0 \leq n - 1$ .
- $D = \frac{1}{2} t_{i_0}$ : If  $i_0 = 1$ .
- $D = \frac{1}{2} \left| t_{i_0} - \sum_{k=0}^{i_0-1} t_k \right|$ : If  $i_0 = n$ .

Case 2:  $n = 0$ .

In this case,  $x$  and  $y$  belong to the same cubical cell, and we have:  $\langle \frac{x+y}{2} \rangle = \frac{x+y}{2}$ .

### 5. Numerical Results

In this section, we apply the previously described algorithm to two examples of a lattice-based metamorphic systems, the robotic arm, and the hexagonal lattice robot.

To understand why the robotic arm is  $CAT(0)$ , one must construct a poset with inconsistent pairs associated with the state complex of the arm (see Appendix B.2).

**Definition 3.** Define for any robotic arm with  $n$  articulations  $R_n$  the set  $P_n = \{(x, y) \mid y \geq 0, y \leq x, x \leq n - 1\}$ , and define the partial order relation  $\leq$ , such that  $(x_1, y_1) \leq (x_2, y_2)$  if and only if  $x_1 \leq x_2$  and  $y_1 \leq y_2$ .

Using the partially ordered set  $P_n$ , we can show that the state complex of  $R_n$  is  $CAT(0)$  through the following proposition:

**Proposition 2.** [3] Let  $\mathcal{S}_n$  be the cubical complex of  $R_n$  rooted at the state where the arm is completely horizontal.

Then, there is a bijection between the possible states of  $R_n$  and the order ideals of  $P_n$ .

We analyze the convergence of the pairwise midpoint algorithm using as a criterion the variance function:

**Definition 4.** Given a configuration  $x = (x_1, \dots, x_N) \in \mathcal{S}^N$ , the variance function is defined as:

$$\sigma^2(x) = \frac{1}{N} \sum_{\{v,w\} \in \mathcal{P}_2(V)} d^2(x_v, x_w)$$

where  $d$  is the distance between two points in  $\mathcal{S}$

If  $\sigma^2(x) = 0$ , then  $x_1 = x_2 = \dots = x_N$ , and we have achieved a consensus state. In [10], it is proven that for a sequence of points  $X(k) = (x_1(k), \dots, x_N(k))$  generated according to the pairwise midpoint algorithm, the function  $\sigma^2$  converges to zero at a linear rate.

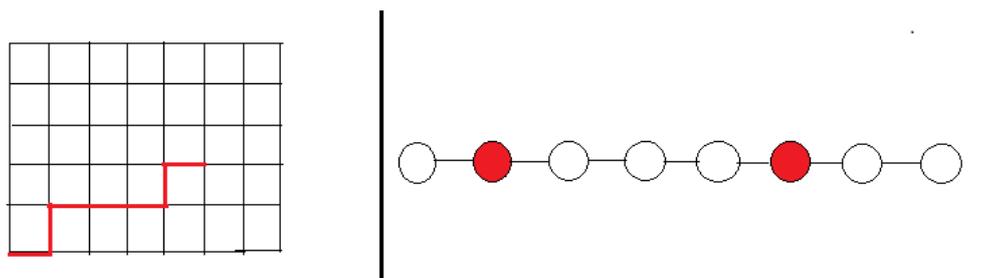
**Theorem 1.** Let  $X_k = (x_1(k), \dots, x_N(k))$  denote the sequence of random variables generated by Algorithm 1; then, there exists  $L < 0$ , such that,

$$\limsup_{k \rightarrow \infty} \frac{\log \mathbb{E}[\sigma(X_k)]}{k} \leq L$$

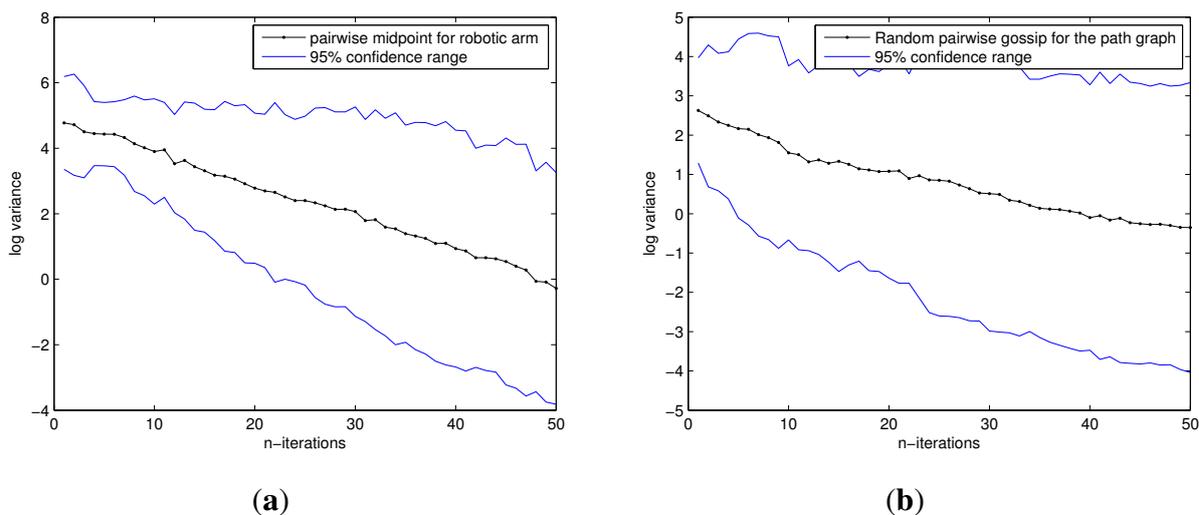
In order to randomly sample a set of initial coordinates, we begin by assigning to each metamorphic system a vector  $(x_1, \dots, x_N)$ , where each  $x_i$  is sampled randomly and uniformly from the interval  $[0, 1]$ . Then, we do the following: we check if there exists  $i, j$ , such that  $i \prec j$  and  $x_j \neq 0$ ; if so, then set  $x_i = 1$  (the  $\prec$  relation being the partial order of the poset associated with the cubical complex  $\mathcal{S}$  in its initial rooting).

5.1. Results for the Robotic Arm

We plot the log-variance  $\log \sigma^2$  as a function of the number of iterations  $k$ , for a complete graph of 10 robotic arms of  $n = 7$  joints (see Figure 6), then for a path graph of 10 robotic arms of the same type. In both cases, we observe in Figure 7 a linear curve with a negative slope in accordance with the results of [10]. In the case of the path graph, however, the slope is smaller than for the complete graph.



**Figure 6.** On the right, a robotic arm with  $n = 7$  articulations and the associated state graph.



**Figure 7.** Plot of  $n \rightarrow \log \sigma^2$  for a network of  $k = 10$  robotic arms with  $n = 7$  articulations. (a) The underlying communications graph is the complete graph, while on (b), it is the path graph. Because of the stochastic nature of the algorithm, the procedure is averaged over 30 runs. The resulting curve is a line of a negative slope of bigger magnitude for the complete graph than for the path graph.

5.2. Results for the Planar Hexagonal Lattice

The same analysis is applied to the hexagonal lattice system [8], which is a connected aggregate of hexagonally-shaped modules that occupy a planar lattice. Its graph representation and associated cubical complex can be seen in Figure 8.

We observe similar patterns as for the robotic arms experiment; in Figure 9 the function  $\log \sigma^2$  decreases at a linear rate for both the complete graph and the path graph cases.

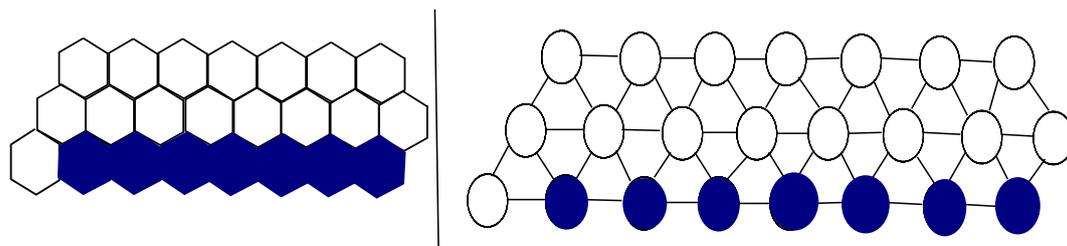


Figure 8. On the right, a hexagonal lattice system and its associated state graph.

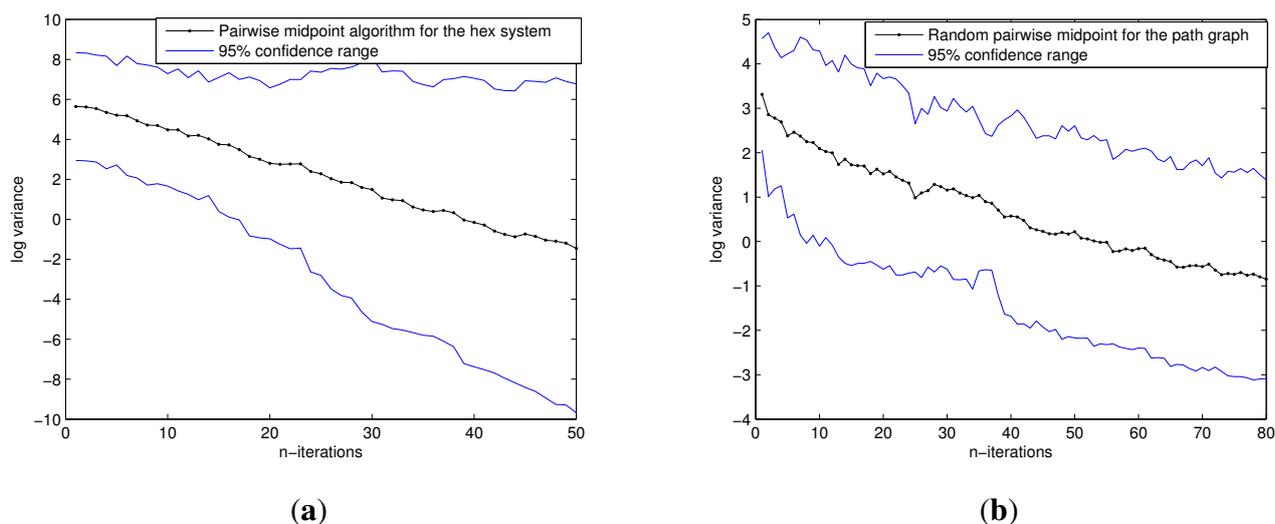


Figure 9. Plot of  $n \rightarrow \log \sigma^2$  for a network of  $k = 10$  hexagonal systems. (a) The underlying communications graph is the complete graph, while on (b), it is the path graph. Because of the stochastic nature of the algorithm, the procedure is averaged over 30 runs. The resulting curve is a line of a negative slope of bigger magnitude for the complete graph than for the path graph.

6. Conclusions

The random pairwise midpoint algorithm can be successfully applied to discrete combinatorial systems, like the case of metamorphic systems. This is a non-trivial example of a purely metric application of this algorithm, whose exponential convergence towards a consensus state has been confirmed through numerical experiments on two examples of metamorphic systems. The same method could be applied to the space of phylogenetic trees. Another application could be that of

distributed optimization, where  $N$  identical metamorphic systems, each with its own utility function  $f_i : \mathcal{S} \rightarrow \mathbb{R} : i \in \{1, \dots, N\}$ , try to minimize a collective objective function  $\frac{1}{N} \sum_{i=1}^N f_i$ ; the underlying communication network has no fusion node.

## Appendix

### A. Modeling Metamorphic Systems

For a rigorous definition of metamorphic systems, we follow the approach of [8]. We represent the lattice by its dual graph  $\mathcal{G} = (\mathcal{V}(\mathcal{G}), \mathcal{E}(\mathcal{G}))$ , whose vertices represent the individual cells of the lattice; two vertices are joined by an edge if and only if their corresponding cells are adjacent in the lattice. We associate with this graph a set of labels  $\mathcal{A}$  on the set of vertices to indicate whether the corresponding cell is occupied or not and, in the former case, whether it is occupied by an obstacle or by a module and of which type. A state of the system is any map  $U : \mathcal{G} \rightarrow \mathcal{A}$ . The metamorphic system dynamically changes its state through a set of elementary movements that satisfy the following rules: (i) units cannot overlap during reconfiguration; and (ii) overall connectivity should always be maintained. This is done via generators, which are defined below:

**Definition 5.** [9] Let  $\mathcal{G} = (\mathcal{V}(\mathcal{G}), \mathcal{E}(\mathcal{G}))$  be a graph and  $\mathcal{A}$  a set of labels. A generator  $\phi$  is a collection of three objects:

- A support  $\text{SUP}(\phi) \subset \mathcal{G}$ , which is a subgraph of  $\mathcal{G}$ ;
- A trace,  $\text{TR}(\phi) \subset \text{SUP}(\phi)$ , which is a subgraph of  $\text{SUP}(\phi)$ ;
- An non-ordered pair of states  $u_0, u_1 : \mathcal{V}(\text{SUP}(\phi)) \rightarrow \mathcal{A}$ , verifying:

$$u_0|_{\mathcal{V}(\text{SUP}(\phi)) - \mathcal{V}(\text{TR}(\phi))} = u_1|_{\mathcal{V}(\text{SUP}(\phi)) - \mathcal{V}(\text{TR}(\phi))}$$

where  $\mathcal{V}(\text{SUP}(\phi))$  are the vertices of  $\text{SUP}(\phi)$ .

The support of the generator corresponds to the region of the graph that contains the necessary information to verify whether the movement is feasible (*i.e.*, the absence of obstacles, no modules overlapping and the connectivity of the system is maintained). The trace of the generator is the region of the graph where the movement actually takes place.

**Definition 6.** [9] A generator is said to be admissible at a state  $U$  if:  $U|_{\mathcal{V}(\text{SUP}(\phi))} = \hat{u}_0$ . The action of  $\phi$ , denoted  $\Phi$ , maps a state  $u : \mathcal{V}(\mathcal{G}) \rightarrow \mathcal{A}$  to a new one  $\Phi[u] : \mathcal{V}(\mathcal{G}) \rightarrow \mathcal{A}$ , given by:

$$\Phi[U] := \begin{cases} U & : \text{on } \mathcal{V}(\mathcal{G}) - \mathcal{V}(\text{TR}(\phi)) \\ \hat{u}_1 & : \text{on } \mathcal{V}(\text{TR}(\phi)) \end{cases}$$

Thus, a generator  $\phi$  acts on a state  $U$  by modifying its restriction on the vertices of  $\text{TR}(\phi)$  from  $u_0$  to  $u_1$ . This corresponds to an elementary movement of unit modules. However, a metamorphic system should be able to perform many such elementary movements simultaneously; in order for that to be feasible, the movements have to be compatible, *i.e.*, their simultaneous execution does not lead to module overlap or loss of system connectivity. For that, we introduce the following definition:

**Definition 7.** [9] In a metamorphic system, a collection of generators  $\{\phi_i\}$  is said to be commutative if:

$$i \neq j \Rightarrow \text{TR}(\phi_i) \cap \text{SUP}(\phi_j) = \emptyset,$$

## B. The State Complex

### B.1. Definition

A cubical complex can be seen as a collection of cubes glued together using isometries.

**Definition 8.** [17] (p.112) Let  $\Gamma \subset \mathbb{N}$  and  $(C_i)_{i \in \Gamma}$  be a collection of Euclidean unit cubes of various dimensions. Additionally, let  $X = \coprod_{i \in \Gamma} C_i$  be a disjoint union of these cubes. A cubical complex  $\mathcal{S}$  is the quotient of  $X$  by an equivalence relation  $\sim$ , such that if  $p : X \rightarrow \mathcal{S}$  is the natural projection, then:

- For every  $i \in \Gamma$ , the restriction  $p_i$  of  $p$  to the cube  $C_i$  is injective.
- If  $p_i(C_i) \cap p_j(C_j) \neq \emptyset$ , then there is an isometry  $h_{i,j}$  from a face  $T_i \subset C_i$  onto a face  $T_j \subset C_j$ , such that  $p_i(x) = p_j(x')$  if and only if  $x' = h_{i,j}(x)$ .

**Definition 9.** [9] The state complex  $\mathcal{S}$  of a metamorphic system is a cubical complex. Each  $k$ -cube  $e^{(k)}$  of  $\mathcal{S}$  is an equivalence class  $[u, (\phi_i)_{i=1}^k]$ , where:

- $(\phi_i)_{i=1}^k$  is a  $k$ -uple of commuting generators.
- $u$  is a state for which the generators  $(\phi_i)_{i=1}^k$  are admissible.
- $[u_0, (\phi_i)_{i=1}^k] = [u_1; (\phi'_i)_{i=1}^k]$  if and only if  $\exists \sigma \in \mathfrak{t}_k$ , such that:  $\forall i \in \{1, \dots, k\}$ ; we have:  $\phi_i = \phi'_{\sigma(i)}$ ; and  $u_0 = u_1$  on the subgraph:  $\mathcal{G} - \bigcup \text{TR}(\phi_{\alpha_i})$ .

The boundary of each  $k$ -cube is a collection of  $2k$  faces:

$$\partial[u; (\phi_{\alpha_i})_{i=1}^k] = \bigcup_{i=1}^k ([u; (\phi_{\alpha_j})_{j \neq i}] \cup [\phi_{\alpha_i}[u]; (\phi_{\alpha_j})_{j \neq i}])$$

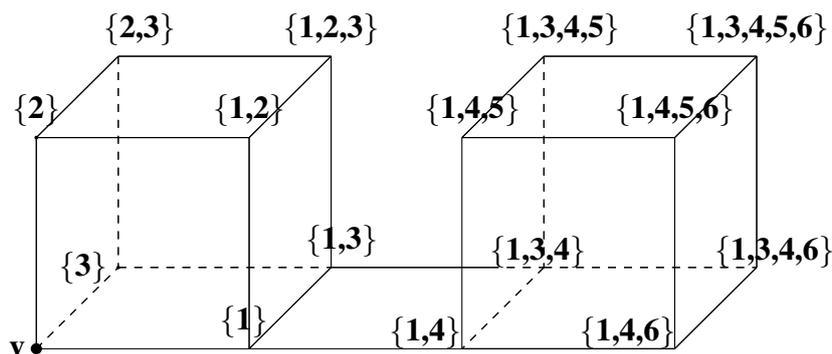
One advantage of using the cubical complex representation over that of the transition graph is that it shows which elementary movements can be performed simultaneously; and, thus, contains more information than the transition graph. It is not generally computationally feasible to construct the cubical complex associated with a given reconfigurable system, but there are some interesting cases where it is possible to do so, like the case of the robotic arm [3].

### B.2. Encoding as a Partially-Ordered Set

Given a cubical complex  $\mathcal{S}$  and a vertex  $v$  (called the root vertex) of  $\mathcal{S}$ , one can introduce a partial order relation in the set of vertices of  $\mathcal{S}$  by stating that a  $u \prec w$  if and only if there is an edge path geodesic (i.e., a geodesic on the metric graph associated with  $\mathcal{S}$ ) from the root  $v$  to  $w$  that passes through  $u$ . The complex  $\mathcal{S}$  can thus be seen as a partially-ordered set. In [14], it is shown that if one can choose the special node  $v$ , such that the set of vertices of  $\mathcal{S}$  equipped with the partial order relation  $\prec$  is a poset with inconsistent pairs, then  $\mathcal{S}$  is a globally  $CAT(0)$  cubical complex. This last propriety guarantees the existence and uniqueness of the midpoint of any given two points and allows us to apply the random

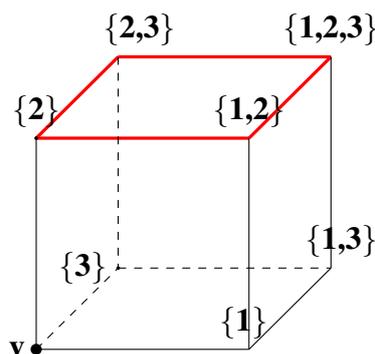
midpoint algorithm on a network of agents whose data are encoded in a cubical complex. We assume in this paper that all of the studied complexes are  $CAT(0)$ .

Reciprocally, given a poset with inconsistent pairs  $(P, \prec)$ , one can build a lattice whose summits are the order ideals of  $(P, \prec)$  that do not contain any inconsistent pairs (such order ideals are said to be consistent). An edge is drawn between two summits if their corresponding ideals differ by exactly one element. The vertices are ordered according to their corresponding ideals,  $u < w$  if and only if the order ideal  $I_u$  corresponding to the vertex  $u$  is a subset of  $I_w$ , the order ideal associated with the vertex  $w$  ( $u < w \Leftrightarrow I_u \subset I_w$ ), as shown in Figure B1.



**Figure B1.** The cubical complex associated with the poset  $(P = \{1, 2, 3, 4, 5, 6\}, \prec)$ , such that:  $1 \prec 4, 1 \prec 5$  and  $4 \prec 6$ . Each vertex is labeled by a consistent order ideal of  $P$ .

Individual cubes are represented by a pair  $(I, M)$ , with  $I$  a consistent order ideal and  $M$  a subset of the maximal element of  $I$ ; such a cube is of dimension  $|M|$ , and its vertices are found by removing from  $I$  all of the possible subsets of  $M$  (Figure B2 gives an example).



**Figure B2.** In this example, the 2-dimensional cube highlighted in color is represented by the pair  $(I, M)$  with  $I = \{1, 2, 3\}$  and  $M = \{1, 3\}$ .

### C. The Standard Embedding

One way to embed the cubical complex associated with a poset with inconsistent pairs of cardinal  $N$  is the so-called standard embedding [14]. An element  $u \in \mathcal{S}$  is represented by an  $N$ -tuple  $u = (u_1, \dots, u_N)$ , where  $\forall i u_i \in [0, 1]$ , and if for some  $(i, j) \in \{1, \dots, N\}^2$ , we have  $u_j \neq 0$  and  $u_j > u_i$ , then we must have  $u_i = 1$ . Furthermore, if  $x_i$  and  $x_j$  are inconsistent, then:  $u_i u_j = 0$ .

Figure 3 shows an example of a cubical complex where the standard embedding coordinates of the points  $x$  and  $y$  are:  $x = (1; 0.25; 0.25; 0)$  and  $y = (1; 0.75; 1; 0.75)$ .

### Conflicts of Interest

The authors declare no conflict of interest.

### References

1. Koren, Y.; Heisel, U.; Jovane, F.; Moriwaki, T.; Pritschow, G.; Ulsoy, G.; van Brussel, H. Reconfigurable Manufacturing Systems. In *Manufacturing Technologies for Machines of the Future*; Dashchenko, A., Ed.; Springer: Berlin/Heidelberg, Germany, 2003; Part IV, Chapter 19, pp. 627–665.
2. Billera, L.J.; Holmes, S.P.; Vogtmann, K. Geometry of the space of phylogenetic trees. *Adv. Appl. Math.* **2001**, *27*, 733–767.
3. Ardila, F.; Baker, T.; Yatchak, R. Moving Robots Efficiently Using the Combinatorics of CAT(0) Cubical Complexes. **2012**, arXiv preprint arXiv:1211.1442.
4. Tsitsiklis, J. Problems in Decentralized Decision Making and Computation. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 1984.
5. Boyd, S.; Ghosh, A.; Prabhakar, B.; Shah, D. Randomized gossip algorithms. *IEEE Trans. Inf. Theory* **2006**, *52*, 2508–2530.
6. DeGroot, M.H. Reaching a consensus. *J. Am. Stat. Assoc.* **1974**, *69*, 118–121.
7. Nejad, B.M.; Attia, S.A.; Raisch, J. Max-Consensus in a Max-Plus Algebraic Setting: The Case of Fixed Communication Topologies. In Proceedings of the IEEE XXII International Symposium on Information, Communication and Automation Technologies (ICAT 2009), Sarajevo, Bosnia and Herzegovina, 29–31 October 2009; pp. 1–7.
8. Abrams, A.; Ghrist, R. State complexes for metamorphic robots. *Int. J. Robot. Res.* **2004**, *23*, 811–826.
9. Ghrist, R.; Peterson, V. The geometry and topology of reconfiguration. *Adv. Appl. Math.* **2007**, *38*, 302–323.
10. Bellachehab, A.; Jakubowicz, J. Random Pairwise Gossip on  $CAT(\kappa)$  Metric Spaces. **2014**, arXiv preprint arXiv:1405.4190.
11. Østergaard, E.H.; Kassow, K.; Beck, R.; Lund, H.H. Design of the ATRON lattice-based self-reconfigurable robot. *Auton. Robots* **2006**, *21*, 165–183.
12. Bacak, M. *Convex Analysis and Optimization in Hadamard Spaces*; De Gruyter Series in Nonlinear Analysis and Applications; De Gruyter: Genthiner Strasse 13, Berlin, Germany, 2014; Volume 22.
13. Jost, J. *Nonpositive Curvature: Geometric and Analytic Aspects*; Lectures in Mathematics. ETH Zurich; Springer: Basel, Switzerland, 1997.
14. Ardila, F.; Owen, M.; Sullivant, S. Geodesics in CAT (0) cubical complexes. *Adv. Appl. Math.* **2012**, *48*, 142–163.
15. Grohs, P. *Wolfowitz's Theorem and Convergence of Consensus Algorithms in Hadamard Spaces*; Technical Report SAM Report 2012-27; ETH: Zurich, Switzerland, 2012.

16. Bacak, M. Computing medians and means in Hadamard spaces. *SIAM J. Optim.* **2014**, *24*, 1542–1566.
17. Bridson, M.; Haefliger, A. *Metric Spaces of Non-Positive Curvature*; Springer: Berlin/Heidelberg, Germany, 1999.

© 2015 by the author; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>).