Geometric Computations by Broadcasting Automata on the Integer Grid

Russell Martin, Thomas Nickson, Igor Potapov

Department of Computer Science,
University of Liverpool, Ashton Building,
Ashton St, Liverpool, L69 3BX, U.K.

{Russell.Martin,T.Nickson,Potapov}@liverpool.ac.uk
+44 (0)151 795 4275

Abstract

In this paper we introduce and apply a novel approach for self-organiz- ation, partitioning and pattern formation on the non-oriented grid environment. The method is based on the generation of nodal patterns in the environment via sequences of discrete waves. The power of the primitives is illustrated by giving solutions to two geometric problems using the broadcast automata model arranged in an integer grid (a square lattice) formation. In this model automata cannot directly observe their neighbours' state and can only communicate with neighbouring automata through the non-oriented broadcast of messages from a finite alphabet. In particular we show linear time algorithms for: the problem of finding the centre of a digital disk starting from any point on the border of the disc and the problem of electing a set of automata that form the inscribed square of such a digital disk. Possible generalizations and applications of techniques based on nodal patterns and the construction of different discrete wave interference pictures are discussed in the conclusion.

Keywords: Broadcasting Automata, Discrete Waves, Distributed Algorithms, Geometric Computation, Pattern Formation

1 Introduction

In many cases it is deemed that large numbers of simple robots can achieve certain tasks with greater efficiency than a single complex robot. Without the reliance on a single automaton there is an increase in robustness, the ability of the system to function with partial failures or abnormal conditions; flexibility, the capability of the system to adapt to changes in the environment; scalability, being able to expand the size of the system without a significant impact on performance [8, 9]. This leads to the question of how to co-ordinate these robots in their task whilst retaining their simplicity of design, function and the robustness that is inherent to distributed systems. Such robots may only have local information or minimal sensor range, small finite memory, minimal communication and restricted common knowledge such as coordinates, orientation and synchronicity [11, 3]. A variety of models have been suggested for this task including those from areas such as Multi Agent Systems, Engineering, Cellular Automata (CA) and Distributed Algorithmics [12]. Problems currently being worked on are common subdivisions of more complicated and pragmatic tasks which are found to tackle a large set of problems simply through the coordinated use of such sub problems. These are often, but not restricted to, pattern formation[5], aggregation[2], chain formation[15], self assembly[16], coordinated movement[17], hole avoidance[13], foraging[18], path formation[14], etc.

This paper investigates the possibilities of geometric computations and constructions by a weak, static, swarm of robots which are arranged in a square lattice. The work extends previous work on algorithms for the generation of triangular, square and hexagonal lattices by a swarm of autonomous mobile robots with limited sensor ranges which were studied in [5]. Our results can also be interpreted for triangular and hexagonal lattices, both of which are alternative, realistic models for swarm topology. The problem of swarm (or a set of mobile robots) configuration into regular grids is well known for exploration tasks and environmental or habitat monitoring [5]. However, arranging robots into a regular grid structure has a number of other benefits in terms of self organization and efficient communication. In this paper we illustrate the possibility of employing the property of regular structure for complex geometric constructions via non-oriented broadcasting in the static cluster of robots (broadcasting automata).

We first define a model of swarms of robots arranged on the grid. Each robot is represented by a broadcasting automaton with finite memory that is unable to observe its neighbourhood, but can communicate through the non-oriented broadcasting of messages with its neighbours. As a model broadcasting automata has widely been used for designing communication protocols [1, 4] and provides a realistic abstraction for interactions via radio network communication. The model can be seen as a special sub-case of cellular automata and may have both practical and theoretical interests.

In the broadcasting automata model direct communication between automata is only possible via the broadcast of messages to all neighbouring automata within a certain communication range. In the case of the square lattice topology (as well as triangular and hexagonal lattices), non-oriented broadcasting can be used to efficiently solve a variety of geometric problems by utilizing the effects found in both real physical systems (i.e. waves and interference patterns) and computational systems (i.e. information processing by finite state automata).

The waves generated by activating processes in a digital environment can be used for designing a variety of wave algorithms. In fact, even very simple finite functions for the transformation and analysis of passing information provides more complex dynamics than classical wave effects. We generalize the notion of the standing wave which is a powerful tool for partitioning a cluster of robots on a non-oriented grid. In contrast to classical waves where interference patterns are generated by nodal lines (i.e. lines formed by points with constant values), an automata network can have more complex patterns which are generated by periodic sequences of states in time.

In this paper we introduce and apply a novel approach for self-organization, partitioning and pattern formation on the non-oriented grid environment based on the generation of nodal patterns in the environment via sequences of discrete waves. The power of the primitives are illustrated by giving solutions to two geometric problems: the problem of finding the centre of a digital disk starting from any point on the border of the disc and the problem of electing a set of automata that form the inscribed square of the same digital disk.

2 Broadcasting Automata

General Model of Broadcasting Automata

In general, a model of broadcasting automata is defined as a **network of finite automata** which is represented by a triple (G, Λ, V_0) where: $G = (V, E)$ is a graph, with set of vertices V and set of edges E. Λ is a deterministic I/O automaton which is at each vertex, v, of the graph G. The set V_0 is the set of initial transmitters which are the set of automata that are active at the start of the computation.

The communication between automata is organized by message passing, where **messages** are from the alphabet Σ and are generated as the output symbols of the automata. The set of vertices connected to the automaton $a \in \Lambda$ at some vertex $v_a \in V$ is given by $\Gamma(a)$ and is the **set of neighbours** for that automaton, a. Messages generated by an automaton, $a \in \Lambda$, are passed to the automaton's **transmission neighbourhood**, which is a subset of its set of neighbours, $\Gamma_T(a) \subseteq \Gamma(a)$. Messages are generated and passed instantaneously at discrete time steps, resulting in synchronous steps. Those automata that have received a message are said to be **activated** along with those automata that started the computation in the set V_0 We will assume that if several messages are transmitted to a particular automaton a, it will receive only a set of unique messages, i.e. for any multiset of transmitting messages received by a, over some number of rounds, the information about quantity of each type, within a single round, will be lost. For any multiset of transmitting messages received by a in a single round, the information about the quantity of each type will be lost.

Let us consider two variants of broadcasting automata: **synchronous** and **asynchronous** (or reactive) models. In the case of the synchronous model, every automaton from the moment of activation, which occurs upon the receipt of a signal, follows discrete time steps and reacts on a set of received events (including an empty set, which corresponds to a 'no event' situation). In the case of the asynchronous (reactive) model, every automaton follows discrete time steps but can only react to a non-empty set of events. Although such automata have no shared notion of time, a form of implied synchronicity may be derived from the constant amount of time taken for each round of communication. Both models are computationally equivalent, but the algorithms designed in each model may require different amount of resources such as the size of the messaging alphabet, number of broadcasts and the overall execution time.

Proposition 1 *The synchronous and asynchronous models are computationally equivalent. Any algorithm for broadcasting automata in the reactive model with a message alphabet of size $|\Sigma|$ can be simulated by the synchronous model with a singular alphabet and $|\Sigma|$ slowdown. Any algorithm for broadcasting automata in the synchronous model with a message alphabet of size $|\Sigma|$ can be simulated by the reactive model with an alphabet $|\Sigma'|$ where $|\Sigma'| = |\Sigma| + 1$.*

Proof: Broadcasting a message u_i in the reactive model with a message alphabet of size $|\Sigma|$ can be simulated in synchronous model by introducing rounds of size $|\Sigma|$. In each round with $|\Sigma|$ discrete time steps the broadcasting of a message u_i will be simulated by broadcasting at time i within the current round. The simulation of the synchronous mode can be modelled by introducing a new symbol u_0 corresponding to a 'no event' situation. Every automaton will transmit either a message u_i from the alphabet Σ or the message u_0. $\square$

Broadcasting automata on $\mathbb{Z}^2$

In this paper we consider a model of broadcasting automata on the non-oriented square lattice (integer grid). We note, however, a similar model can be defined on any other lattice or graph structure. On the square lattice it is possible to vary the transmission neighbourhood, Γ_T, by varying the transmission range. In particular, with a transmission radius equal to 1 or 1.5 (which covers those nodes at an exact distance of 1 and $\sqrt{2}$), the so-called **Von Neumann neighbourhood** and **Moore neighbourhoods** are generated, respectively.

In Figure 2 the source of the transmission is shown as the circle at the centre of the surrounding automata, those that are black are within the transmission range and thus in the transmission neighbourhood. The large outer circles represent the transmission range which can be changed to alter the automata that are included in the transmission neighbourhood. If the transmission radius is equal to 1, as in Figure 2 diagram i) then only four of the eight automata can be reached. If the radius is made slightly larger and is equal to 1.5, it can encompass all eight automata in its neighbourhood. As we will show later, iterative broadcasting within Von Neumann and Moore neighbourhoods can distribute messages in the form of a diamond wave and a square wave [1] .

Whilst these are the only two radii that will be addressed here, in general any radius may be chosen where each radius forms a distinct convex polygon.

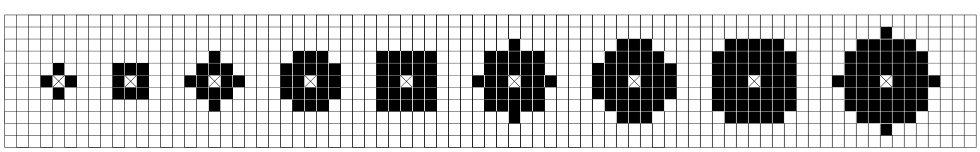

Figure 1: A variety of transmission radii are shown (l-r) squared radii $r^2 = \{1, 2, 4, 5, 8, 9, 10, 13, 16\}$. Crosses represent the centre of the respective discrete disc.

3 Computational Primitives for Broadcasting Automata

In this section we discuss a number of computational primitives that can be applied to non-oriented broadcasting automata on the grid in order to generate geometrical constructions.

In the **synchronous** model waves are passed from automaton to automaton according to the following rules:

1. An automaton a receives a message from an activating source at a time t;
2. At a time $t+1$, a sends a message to all automata within its transmission neighbourhood $\Gamma_T(a)$;
3. At a time $t+2$, a ignores all incoming messages for this round.

In the **asynchronous model** simulation we require at least 3 distinct symbols $\{u_0, u_1, u_2\}$:

1. An automaton a receives a message u_i from an activating source;
2. The automaton a broadcasts a message $u_{(i+1) \mod 3}$ to all automata within its transmission neighbourhood $\Gamma_T(a)$;
3. Ignore next incoming message $u_{(i+2) \mod 3}$.

In each model Step 3 prevents a node from receiving back the wave that it just passed to its neighbours by ignoring all transmissions received the round after transmission and ensuring that the front of the wave is always carried away from the source of transmission. Waves are passed using two different transmission neighbourhoods referred to as square and diamond waves, which are equivalent to Moore and Von Neumann neighbourhoods respectively, see Figure 2. If two waves are broadcast towards each other at the same time and meet then both waves halt due to the newly elected automata are surrounded by automata that are in the ignore state. Figure 2

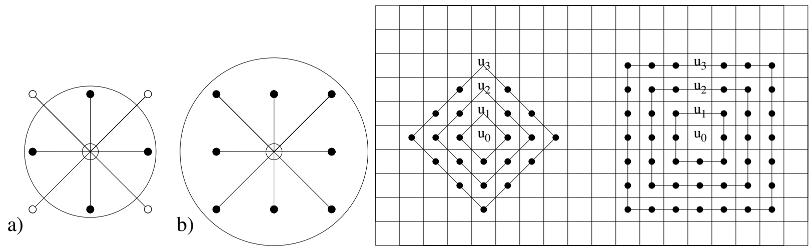

Figure 2: Diagram $a)$ represents the propagation pattern for a diamond wave (Von Neumann neighbourhood) and diagram $b)$ shows the propagation pattern for a square wave (Moore neighbourhood). Also wave propagation with the asynchronous model is shown on the square grid: Moore neighbourhood (left) and Von Neumann neighbourhood (right).

The Wave Propagation Algorithms defined above can also be modified to simulate the transmission of any periodic word $(u_0, ..., u_k)^*$, including a 'discrete wave'. In the case of the **synchronous** model with a singular alphabet the initial source transmits a message and all automata reached will use the above defined algorithm to pass the activation wave. At the same time and since the system is synchronous, all activated automata can go through a cycle of states corresponding to $u_0, ..., u_k$. Alternatively for the **asynchronous** model the same can be simulated with an alphabet of size k. When an automaton receives a symbol, u_i, it will broadcast the message $u_{i+1 \mod k}$ once.

Whilst within the model an automaton is unable to directly access the state of its neighbours it is still possible to design many primitives in a similar style as designed for the cellular automata model [6], including synchronization procedures, finding the edge elements on a line (Lemma 1), shortest branch of a tree (Lemma 2), etc. These basic tools, based on ideas from cellular automata, will be utilized to illustrate more sophisticated algorithmic methods in Section 4.

Lemma 1 *Given a network of finite automata, (G, Λ, V_0) with a graph which form a single line and is of size n and a single source v, such that $v \in V_0$, it is possible to* *elect the automata with no successor in $(n-k)+3$ time steps, where k is the number of automata from the initiating automaton, v, to its closest side, and with an alphabet, Σ, of size $|\Sigma| = 3$, in the asynchronous model.*

Proof: The proof is given as a constructive argument such that the following algorithm describes a method for any automaton to establish itself as an automaton that is, or is not, on the edge and correctness of the algorithm is given. The algorithm, presented below, assumes that there is some $v \in V_0$ that begins the initial transmission and that the standard form of transmission for broadcast automata is utilised. After use of the following procedure, which is utilised by all broadcast automata, then any such automata which have no successor should have elected themselves as such.

The following algorithm is followed by every automata and provides a method for locating the edge automata in any line of size n with an alphabet of size 3 and is visualised in Figure 3.

1. Automaton v_i receives symbol $s_i \in \Sigma$ activating it at time t
2. For the following three time steps, $t + \{1, 2, 3\}$, it emits symbol $s_{i+1 \bmod 3}$
3. If at time $t + 2$ automaton v_i receives two signals of differing types, s_i and $s_{i+2 \bmod 3}$ then it is not on the edge. Should the v_i at time $t = 3$ receive only signal s_i then the automaton has no successor.

The above algorithm elucidates the observation that those automata on the edge of the graph have no successor and as such it is not possible for them to receive two signals of differing types. Let us assume for contradiction that there is such an element v_i that has no successor but which will not be elected by the above algorithm. It must be that v_i, to have successfully been elected by the algorithm, has received two differing symbols at the same time step. As each automaton is only able to transmit a single symbol during one time step this means that the automaton v_i must have at least two automata in its neighbourhood. As the automaton lies on a line it must be that this automata has both a predecessor and a successor however this is now in contradiction to the premise that this automaton has no successor and so the algorithm must hold.

□ Figure 3

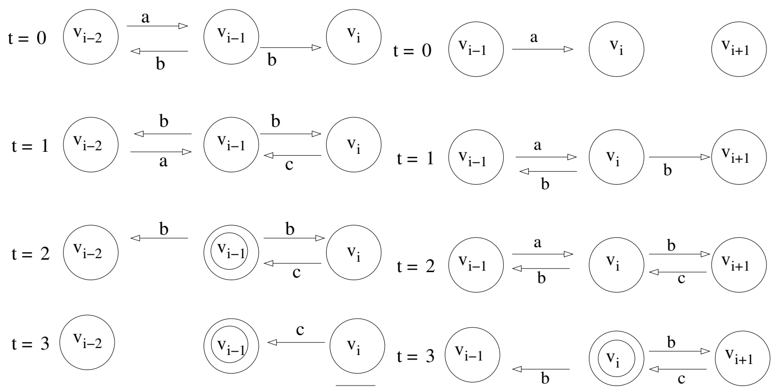

Figure 3: The detection of a none edge case, that has two neighbours (left), and an edge case, with only one neighbour (right)

Lemma 2 *It is possible to elect the automaton with no successor of the shortest branch of a tree with a single fork at the root and* k *branches in time* $3R$*, where* R *is the length of the shortest branch(es) with a distinct alphabet of* $|\Sigma| = 2$.

Proof: The following algorithm provides a constructive proof of the existence of such an algorithm. It is assumed that there is an initial automaton $v_0 \in V_0$ that is at the root vertex of the graph.

1. From the root vertex v_0 the symbol a is broadcast to all neighbours of v_o, propagating through the system as defined by the message passing algorithm for broadcast automata.

2. Once the wave reaches the final vertex in each branch it is broadcast back up towards the root vertex after 2 time steps waiting for the refractory period of the previous vertex to expire.

3. After the root receives the first of the a symbols it begins to broadcast symbol b which is only accepted by those automata which have received two a symbols, which are the automata of the shortest branch(es).

4. Upon receiving symbol b those automata that have already received a, a will be elected as the shortest branch(es).

   All other automata which have at that moment only received a single a will have rejected the b symbol and so are not elected.

The correctness of the algorithm is given by the lengths of the paths in which the transmissions must traverse. Given a set, S, of k branches such that each branch is of length $p_0 \leq p_1 \leq ... \leq p_{k-1}$ there must be some set of least paths which will be denoted S_{min} and contain only those branches that are of equal length to p_0 the branch of least length. Given that the messages are passed at constant speed, and as such are synchronised throughout the system, it must be that each path is traversed in time $2 \cdot p_i$ and those that reach v_0 first, after the initial transmission from v_0, must have taken the shortest path, here given as S_{min}. As only those automata which have received the symbol a twice are elected by the b symbol, other automata, in the set $S \setminus S_{min}$, will reject the symbol, and so only those in the set S_{min} will be elected by b. $\square$ Figure 2

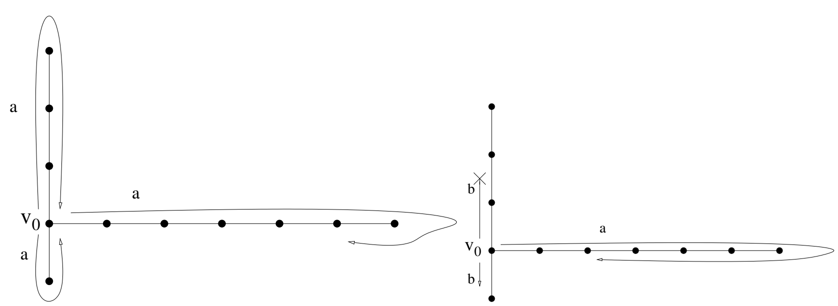

Figure 4: The above two figures illustrate the described algorithm with the left figure showing the initial transmission and on the right showing the transmission whereby the crossed transmission has not propagated through the system.

Nodal Patterns of Discrete Interference

In physics, a standing wave is a wave that remains in a constant position. This phenomenon can occur in a stationary medium as a result of interference between two waves travelling in opposite directions. A standing wave in a transmission line is a wave in which the distribution of current value is formed by the superposition of two waves propagating in opposite directions. The effect is a series of nodes (zero displacement) and anti-nodes (maximum displacement) at fixed points along the transmission line. With standing waves in a two dimensional environment the nodes become nodal lines, lines on the surface at which there is no movement, that separate regions vibrating with opposite phase [7].

In this paper we investigate standing waves in a discrete environment, where the original physical system is generalized in several ways. First, the transmitted waves will be discrete, i.e. the values of the wave which are passing a point p will correspond to a sequence of symbols that are observed in p. Assuming that a point in discrete environment has even simple computational power, like finite memory, the idea of nodal points and nodal lines can be extended. Apart from recognizing points with classical standing waves (i.e. having a constant value over time), it is possible to recognize periodic sequences of values over time. This will eventually lead to a richer computational environment where nodal lines of different periodic values can form patterns in $\mathbb{Z}^n$ and where points may react and pass a wave which contrasts to the physical model.

In order to demonstrate the technique of nodal patterns for geometric computations in non-oriented $\mathbb{Z}^2$ we will start with a simpler abstraction in one dimension line. Let us place two transmitters, T_1 and T_2, on a one dimensional line, each broadcasting words u^*, v^* respectively, where $u, v \in \Sigma$, $|u| = |v| = s$ is the length of the word. The **broadcasting** of symbols begins at transmission points with symbols broadcast away from the source at each time step, cycling through all symbols in the word. We now have two infinite words $(u)^*$ and $(v^R)^*$, the reverse of v, that are broadcast towards each other, from one automata to the next, on every next time step. In this case any Figure 5

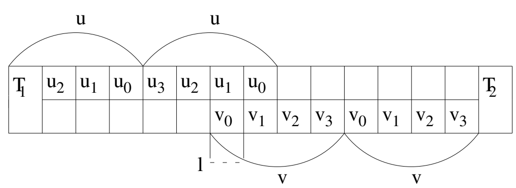

Figure 5: Shows two transmitters T_1 and T_2 broadcasting words u and v respectively.

automata on the line, which has received two symbols, one from u and one from v, forms a pair $p_{i,j} = (u_i, v_j)$. So once a point l is in such a configuration, it is possible to define a sequence of pairs $p_{(i+t) \mod s,(j+t) \mod s}$ contained in l over discrete time $t = 1, \ldots, \infty$. It is easy to see that such sequence will be periodic with a period less or equal to s, where $|u| = |v| = s$, and it represents a history of symbol pairs from u and v which are meeting at the point l over time.

The **nodal pattern** of a point l is a finite subsequence of s pairs $p_{(i+t) \mod s,(j+t) \mod s}$ over some $t = t'..t' + s$. Let us also assume that nodal patterns are equivalent up to a cyclic shift, so it will not be dependent to the initial time t'. Nodal patterns are labelled using the difference between the indices defined as $P_{|i-j|}$, where i and j are the indices from any two of the pairings of symbols $p_{i,j}$ for some point on the plane at some time. Nodal patterns have s possible labels, $P_0, P_1, \ldots, P_{s-1}$, one for each of the possible pairings, though they may not all be distinct. Such indices are now defined as the **node index** of the automaton, which signify the distinct pairing observed by the automata.

Nodal patterns are exemplified in Figure 5 where $s = 4$, $u = u_0, u_1, u_2, u_3$ and $v = v_0, v_1, v_2, v_3$. At point l the pairs $((u_1, v_0), (u_2, v_1), (u_3, v_2), (u_0, v_3))$ form a nodal pattern corresponding to $p_{(1+t) \mod s,(0+t) \mod s}$ over time t. The nodal pattern is now labelled $P_{|(1+t)-(0+t)|} = P_{|1+t-0-t)|} = P_{|1-0|} = P_1$.

Patterns p_1 and p_2 are called **distinct** if p_1 cannot be constructed from p_2 by applying any cyclic shift to it. Where two patterns are formed by ordered pairs and a word $u = v$ is without repetitive symbols we have s distinct nodal patterns. In the case of unordered pairs such that $(u_i, v_j) = (v_j, u_i)$ the number of distinct nodal patterns is smaller and defines a specific sequence described in the Propositions 2, 3.

Proposition 2 *Given a non-periodic word u where $|u| = s$. The number of possible distinct nodal patterns generated by broadcasting words u and v, where $v = u$, is $s/2 + 1$ if s is even and $(s + 1)/2$ is s is odd.*

Proof: We can represent the pairings that form nodal patterns by the connections of a bipartite graph, each vertex now comprises one of the symbols from each word u and v and edges represent the pairing of symbols, exemplified in Figure 6. In this representation the number of distinct nodes is represented by the number of distinct graphs that can be made from cyclically shifting the set of vertices in v. Figure 6

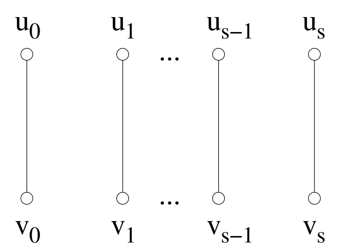

Figure 6: Representing the initial configuration of combinations of words u and v as a graph.

The graphs can now be seen to be of two distinct types. Those that have a symmetric counter part, that is, if the nodes of u and v are cyclically shifted then the graphs are the same, and those which are self symmetric and so are their own inversion, switching u and v generates the same graph. Clearly groups of the latter kind, that are self symmetric, must be distinct.

Only two self symmetric graphs in an even numbered set of u and v and one such graph in an odd numbered set are possible. Whether the number of vertices is odd or even it is always possible to construct the graph which simply associates u_i to v_i, as shown in Figure 6 and as the nodes have been associated with themselves, $u = v$, it is clear that this group is self symmetric. In the even case there is one other self symmetric case when edges are all of the form $(u_i, v_{i+(s/2)})$, exemplified in Figure 7. This may only be of the form $s/2$ as $2(s/2) = s$ which gives 0 over modulo s. In this way each mapping from u to v is reciprocated with the same mapping from v to u. This is only possible with even sets of nodes as it must be possible to pair the nodes from one set which is not possible with an odd number of nodes. Figure 7

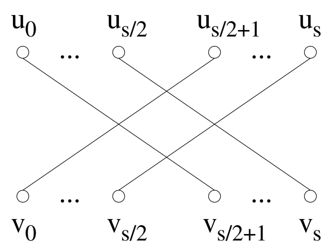

Figure 7: The second self symmetric graph constructed by the combination of words u and v.

All other possible nodes must have a separate symmetric counter part. Taking any none self symmetric combination it is possible to invert the sets u and v and obtain a graph which will be the same as a later permutation of the graph as u_0, and all successive vertices, are mapped to a different vertex to v_0 but both sets are the same we have already mapped both such cases in a single graph.

For the even case we now arrive at $(s-2)/2+2 = (s/2)+1$ halving the number of none self symmetric cases and adding the two that are. For the odd case we do the same only removing the one self symmetric case and adding after halving the none self symmetric cases giving $(s-1)/2+1 = (s+1)/2$ $\square$

Example 1 *Given a sequence* $u = v = 1, 0, -1, 0$ *and so* $|u| = |v| = s = 4$ *it can be shown that there are* $(s/2)+1 = (4/2)+1 = 3$ *distinct nodal points. Enumerating all possible patterns (modulo removed for clarity) and where* t *is over time.*

$P_0 = ((1,1),(0,0),(-1,-1),(0,0))$; $P_1 = ((1,0),(0,-1),(-1,0),(0,1))$
$P_2 = ((1,-1),(0,0),(-1,1),(0,0))$; $P_3 = ((1,0),(0,1),(-1,0),(0,-1))$.

Patterns P_1 and P_3 are the same assuming that the order of the symbols is not relevant i.e. $(-1,0) = (0,-1)$*, as* $P_1 = p_{(0+t),(1+t)} = ((1,0),(0,-1),(-1,0),(0,1))$ *becomes equivalent to* $P_3 = p_{(0+t+3),(3+t+3)} = ((0,1),(-1,0),(0,-1),(1,0))$ *when* P_3 *is cyclically shifted by* 3*.*

Proposition 3 *In the sequence of s nodal patterns* $P_0, P_1, \ldots, P_{s-1}$ *derived from words* $u = v$ *where* $|u| = |v| = s$*, two nodal patterns* P_k *and* $P_{s-k \mod s}$ *are equivalent up to a cyclic shift.*

Proof: As the symmetric counterpart to any nodal pattern can be found by inverting the vertices of u and v any addition in one counterpart equates to subtraction in the other, as exemplified in Figure 8. Figure 8

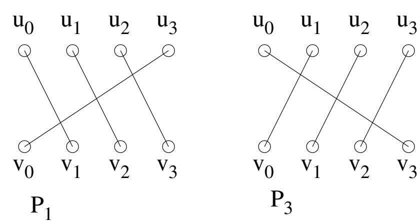

Figure 8: A pair of self symmetric graphs which are equivalent via inversion.

If the pattern is P_k and we wish to find its counterpart we know that if to find the original pattern we added k to the indices of the vertices of u resulting in vertex pairings of the form $p_{i,i+k}$. Its symmetric counterpart must now be of the form $p_{i,i-k}$ which gives us the nodal pattern P_{-k}. As indices are taken be in a modulo number system $-k$ has a solution $0 \leq k < s$, from the congruence class $[k]_s$ which can be found by adding s, hence P_{s-k}

□

Nodal patterns with two sequential transmitters

Standing waves and nodal patterns may be formed when u^*, where $|u| = s$, is transmitted from one source T_0 and is retransmitted from another source T_1 after reaching it. Any point of a grid having both signals from T_0 and T_1 will have identified its nodal pattern as the 'difference' between them.

Upon moving the waves to two dimensions, $\mathbb{Z}^2$, it is important to define the distance between the points in terms of (square or diamond) wave propagation for proper calculation of the nodal patterns, as it is not the same as in Cartesian geometry.

Definition 1 *Square waves adhere to the distance function d_s for distances from p to q where $d_s : \mathbb{Z}^n \times \mathbb{Z}^n \to \mathbb{Z}$ and is defined by the correspondence $d_s((p_1, \dots, p_n),$ $(q_1, \dots, q_n)) = \max_{1 \le i \le n}\{|p_i - q_i|\}, (p_1, \dots, p_n), (q_1, \dots, q_n) \in \mathbb{Z}^n$.*

Definition 2 *Diamond waves adhere to the distance function d_d for distances from p to q where $d_d : \mathbb{Z}^n \times \mathbb{Z}^n \to \mathbb{Z}$ and is defined by the correspondence $d_d((p_1, p_2, ..., p_n),$ $(q_1, q_2, ..., q_n)) = \sum_{i=1}^{n} |p_i - q_i|, (p_1, p_2, ..., p_n), (q_1, q_2, ..., q_n) \in \mathbb{Z}^n$.*

Informally speaking, the distance functions defined above provide the time taken for the diamond/square wave front to reach an automata at coordinate $(y_1, y_2, ..., y_n)$ from a transmitter at $(x_1, x_2, ..., x_n)$.

Let us consider two ways of computing nodal patterns for this process in terms of synchronous and asynchronous cases with transmitters T_0 and T_1. In both cases we aim to get the same distribution of nodal patterns on $\mathbb{Z}^2$ without the need to continuously propagate values from the sources. The local distribution of these patterns will allow us later to locate points in a non-oriented environment.

Definition 3 *An index i of a nodal pattern P_i in a point $\rho \in \mathbb{Z}^2$ is defined as*

$$i_\rho = |d(T_0, \rho) - (d(T_0, T_1) + d(T_1, \rho))| \mod s$$

where d is understood as d_s or d_d for square or diamond wave respectively.

Synchronous model with a single message. In case of transmitting a single message, waves here are activation waves, which the automata use to start internal clocks.

Transmissions begin from T_0 where the activation wave propagates through the use of a square (or diamond) wave arriving at T_1 which is activated when reached by the first wave after a constant delay s. Any point that has received the first signal will start its internal clock (which counts modulo s), and then after receipt of the second signal the clock is stopped and the value of clocks corresponds to the index of the nodal pattern for this point which is $|(d(T_0, \rho) \mod s) - ((d(T_0, T_1) \mod s) + (d(T_1, \rho) \mod s))|$.

Asynchronous model with multiple messages. In the case of the asynchronous model, the same distribution of nodal patterns can be simulated by sending a wave from T_0 where, on the wave front, every point that receives a symbol u_i immediately transmits the symbol $u_{(i+1) \mod s}$. The pseudo-synchronization of wave propagation is achieved by assuming that every transmission takes the same constant time. Then transmitter T_1 operates in the same way once reached by u_i, transmitting the next symbol corresponding to u_{i+1} but using a different alphabet $\{v_1, ..., v_s\}$ to avoid problems whereby transmitting in the same alphabet could have a blocking effect on the wave. Each node should now contain a pair of symbols $(u_{i'}, v_{i''})$ which is enough to define the pattern $P_{|i'-i''|}$, where $i' = d(T_0, \rho) \mod s$ and $i'' = (d(T_0, T_1) + d(T_1, \rho)) \mod s$.

In the next theorem the properties of nodal pattern distribution is shown resulting in a new approach for partitioning $\mathbb{Z}^2$ via non-oriented transmissions and is one of the core tools for the geometric algorithms discussed in this paper.

Theorem 1 *Let T_0 and T_1 be any two points in $\mathbb{Z}^2$ with coordinates (j_0, k_0) and (j_1, k_1), respectively. Assume that nodal patterns P_i were formed by square waves, i.e. $i = |d_s(T_0, \rho) - (d_s(T_0, T_1) + d_s(T_1, \rho))| \mod s$. For any point $\rho \in \mathbb{Z}^2$ the membership to one of the following sets $\{D_1, D_3, D_5, D_7\}$, $\{D_2, D_4, D_6, D_8\}$ or $\{D_9\}$ is uniquely identified by a number of distinct nodal patterns P_i in the Moore neighbourhood of ρ: $\{D_9\}$ — has two nodal patterns, $\{D_1, D_3, D_5, D_7\}$ — has one nodal* *pattern, $\{D_2, D_4, D_6, D_8\}$ — has three nodal patterns, see Figure 9:*

$D_1 = \{(x, y)|y \geq -x + (k_1 + j_1), y \geq x + (k_0 - j_0)\}$

$D_2 = \{(x, y)|y \geq -x + (k_1 + j_1), y \leq x + (k_0 - j_0), y \geq x + (k_1 - j_1)\}$

$D_3 = \{(x, y)|y \leq x + (k_2 - j_2), y \geq -x + (k_2 + j_2)\}$

$D_4 = \{(x, y)|y \leq -x + (k_1 + j_1), y \leq x + (k_0 - j_0), y \geq x + (k_0 + j_0)\}$

$D_5 = \{(x, y)|y \leq -x + (k_0 + j_0), y \leq x + (k_1 - j_1)\}$

$D_6 = \{(x, y)|y \leq x + (k_0 - j_0), y \leq -x + (k_0 + j_0), y \geq x + (k_1 - j_1)\}$

$D_7 = \{(x, y)|y \geq x + (k_0 - j_0), y \leq -x + (k_0 + j_0)\}$

$D_8 = \{(x, y)|y \geq x + (k_0 - j_0, y \leq -x + (k_1 + j_1), y \geq -x + (k_0 + j_0)\}$

$D_9 = \{(x, y)|y \leq x+(k_0 - j_0, y \geq -x+(k_0+j_0), y \leq -x+(k_1+j_1), y \geq x+(k_1-j_1)\}$.

Proof:

In a system with two transmitters, T_0 and T_1, nodal patterns are formed by broadcasting the square wave from T_0 which, once reached by the wave, will then be broadcast by T_1. The main observation is that broadcasting a square wave generates quadrants defined by the lines $x = y$ and $x = -y$, assuming that the transmitter is the origin, whereby within each quadrant the front of the wave expands such that each element of the orthogonal axis, to the one wholly contained by the quadrant (i.e. x,-x,y,-y), within the quadrant will contain the same member of the alphabet as all the others. This defines the direction of the wave and dictates the structure of the neighbourhoods for the automata here depicted in Figure 9. It is shown that the 3 differing combinations of directions of the waves result in the same number of differing nodal patterns and that the number of distinct combinations of differing wave directions equate to the number of distinct modes that appear on the plane.

The nodal pattern at some point $\rho = (x, y)$ is given in Definition 3 as

$$i = |d(T_0, \rho) - (d(T_0, T_1) + d(T_1, \rho))| \mod s$$

and the distance function for the square wave is given as

$$d_s((x_1,\ldots,x_n),(y_1,\ldots,y_n)) = \max_{1\le i\le n}\{|x_i-y_i|\},(x_1,\ldots,x_n),(y_1,\ldots,y_n)\in\mathbb{Z}^n\ .$$

The statement of the theorem is proved by showing the form of the neighbours in the four distinct cases and illustrating their differences, they cannot be the same assuming some sufficient alphabet, naturally a single alphabet will render the plane in a singular nodal pattern.

If ρ, $\rho_1 = (x+1,y)$ and $\rho_2 = (x,y+1)$ are in the previously defined areas $\{D_2,D_6\}$ then according to the distance function in Definition 3 the difference in distance between $d(T_0,\rho)$ and $d(T_0,\rho_{1,2})$ corresponds to the difference in the abscissa and as such $d(T_0,\rho)+1 = d(T_0,\rho_1)$ and $d(T_0,\rho) = d(T_0,\rho_2)$. Similarly distances for T_1 are restricted to differences in the ordinate resulting in $d(T_1,\rho) = d(T_1,\rho_1)$ and $d(T_1,\rho)+1 = d(T_1,\rho_2)$. As such $i_\rho = i_{\rho_1}+1$ and $i_\rho = i_{\rho_2}-1$. The full neighbourhood may now be inferred:

$i_\rho-2$	$i_\rho-1$	i_ρ
$i_\rho-1$	i_ρ	$i_\rho+1$
i_ρ	$i_\rho+1$	$i_\rho+2$

Cases $\{D_8,D_4\}$ are seen to be the same via the symmetry of the two groups.

By the same reasoning the area D_9 is explored noting that in such an area the distance function shows the abscissa to be greater than or equal to the ordinate for both $T_{\{1,2\}}$ resulting in $d(T_0,\rho) = d(T_0,\rho_1)$ and $d(T_0,\rho) = d(T_0,\rho_2)$ and for T_1 the following $d(T_1,\rho) = d(T_1,\rho_1)$ and $d(T_1,\rho) = d(T_1,\rho_2)$. This give nodal pattern differences as $i_\rho = i_{\rho_1}+2$ and $i_\rho = i_{\rho_2}$.

Further more the areas $\{D_3,D_7\}$ which, by the same reasoning, the distance functions from $T_{\{1,2\}}$ are again bounded by the abscissa. Distances for $d(T_0,\rho)+1 = d(T_0,\rho_1)$ and $d(T_0,\rho) = d(T_0,\rho_2)$. Similarly distances for T_1 are restricted to differences in the ordinate resulting in $d(T_1, \rho) = d(T_1, \rho_1)$ and $d(T_1, \rho) + 1 = d(T_1, \rho_2)$. .This results in nodal pattern neighbourhoods of $i_\rho = i_{\rho_1}$ and $i_\rho = i_{\rho_2}$. By symmetry, neighbourhoods in areas $\{D_1, D_5\}$ are equivalent.

□ Figure 9

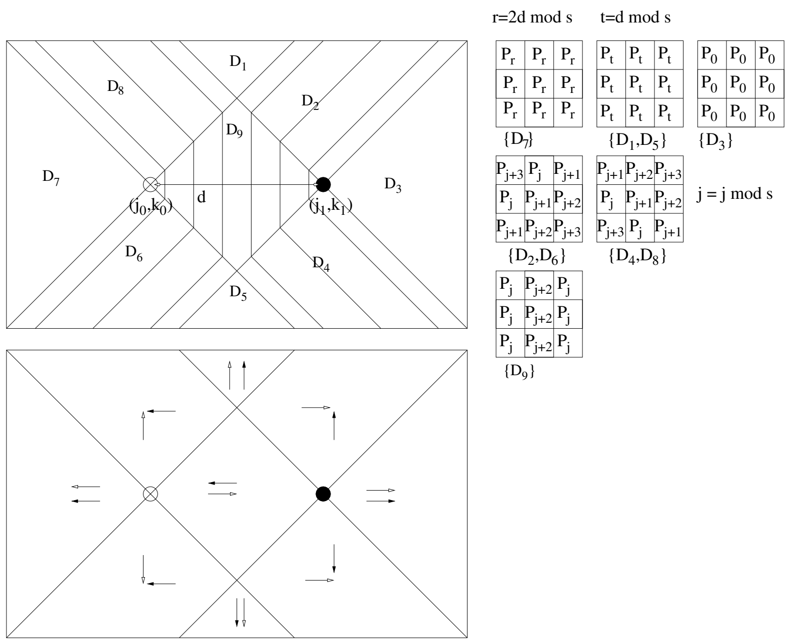

Figure 9: The distribution of three distinct nodal patterns in case of two 'transmitter', where $s = 4$. The bottom diagram showing the 'motion' of the waves, that is it is in these directions that the indices of the messages increases. The top diagram indicates the segmentation and direction of the resulting nodal patterns where the series of diagrams to the right illustrate the categorisation of patterns that are given in the sector D_i

Partial Vision of the Neighbourhood

In the current model each element is only aware of its own node index, and information about its neighbours is not directly observable. However, it is possible to organize a partial vision which provides access to a set of distinct values of nodes from the Moore and Von Neumann neighbourhoods via non-oriented broadcast. In order to get this set of distinct values an automaton should send a special message (nodal request) to all neighbourhoods which, in turn, will send their nodal indices on the next step. Alternatively it is possible to initiate an iterative process of such requests through the whole network of broadcasting automata on $\mathbb{Z}^2$ where each next automata will react on this information by transmitting its own nodal value. This type of transmission we call as a **node wave**.

It is possible to test membership to one of the partitions (defined in Theorem 1) generated through waves broadcast by two transmitters. Having partitioned the plane with two transmitters, T_0 and T_1, T_1 waits a constant delay s before broadcasting a nodal wave which allows all automata information about its neighbourhood. After these steps the automaton has enough information to decide to which of the partitions it belongs.

Apart from the above partitioning it is possible to elect vertical, horizontal (set L_0) and diagonal lines (set L_1) starting from a single source by transmitting square and diamond waves after a constant delay.

Proposition 4 *Given a vertex, $T_0 = (j, k)$ on $\mathbb{Z}^2$, it is possible to elect a set of points $L_0 = \{(x, y)|y = k, x \in \mathbb{Z}\} \cup \{(x, y)|x = j, y \in \mathbb{Z}\}$ or the set $L_1 = \{(x, y)|y = x + k - j \in \mathbb{Z}\} \cup \{(x, y)|x = -y + k + j \in \mathbb{Z}\}$ in linear time.*

Proof: Without loss of generality we may assume that the source, T_0, is at $(0, 0)$ and that both the square and diamond waves are broadcast at the same time. Whilst simultaneous broadcast is not possible all values are over modulo equivalent to the size of the alphabet. Hence any delay between the broadcast of the two waves that is equivalent to the size of the alphabet will reduce to 0 resulting in equations that will be the same as if a simultaneous broadcast had taken place. Patterns formed by the transmission of a square wave, shown in Figure 10 a), then a diamond wave, shown in Figure 10 b), can now be expressed as the difference in time between its activation by the diamond wave the point at which its nodal pattern is fixed by the square wave or

$P_{d_d((0,0),(\rho_1,\rho_2))-d_s((0,0),(\rho_1,\rho_2))}$ for any point $\rho = (\rho_1, \rho_2)$ where d_d is the distance function for the diamond wave and d_s is the distance function for the square wave. If points are of the form $\rho = (0, \rho_2)$ for $\rho_2 \in \mathbb{Z}$ or $\rho = (\rho_1, 0)$ for $\rho_1 \in \mathbb{Z}$ then $P_{d(0,\rho_2)-d_s(0,\rho_2)} = P_{\rho_2-\rho_2} = P_0$ or $P_{d_d(\rho_1,0)-d_s(\rho_1,0)} = P_{\rho_1-\rho_1} = P_0$ so all points in L_0 have been elected as P_0, shown in Figure 10 c). This notion can be observed through the composition of the labelled numbers in diagrams Figure 10 a) and Figure 10 b) to get, after addition and through modulo 4 a diagram that resembles Figure 10 c).

The elected nodes will be distinct from the nodes that surround them as can be seen if r is substituted for 0 or $\rho = (r, \rho_2)$ for $\rho_2 \in \mathbb{Z}$ or $\rho = (\rho_1, r)$ for $\rho_1 \in \mathbb{Z}$ then $P_{d_d(r,\rho_2)-d_s(r,\rho_2)} = P_{r+\rho_2-\rho_2} = P_r$ or $P_{d_d(\rho_1,r)-d_s(\rho_1,r)} = P_{r+\rho_1-\rho_1} = P_r$ for $\rho, \rho_2 > r$ or $P_{d_d(r,\rho_2)-d_s(r,\rho_2)} = P_{r+\rho_2-r} = P_{\rho_2}$ or $P_{d_d(\rho_1,r)-d_s(\rho_1,r)} = P_{r+\rho_1-r} = P_{\rho_1}$ for $\rho_1, \rho_2 < r$. Showing the elected points in L_0 to be distinct where $\rho_1, \rho_2, r \neq 0$.

The election of the set L_1 requires the neighbourhood to be checked through the use of a diamond node wave originating from the initial point T_0. The distinct neighbourhoods of the diagonal elements which are detected with a diamond node wave allows the election of all points in L_1, two diagonal lines meeting at point T_0. $\square$ Figure 10

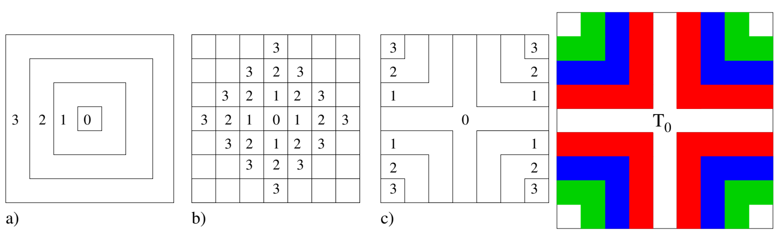

Figure 10: Figures a) and b) show the nodal labellings of the automata from the square and diamond waves respectively which generate the final nodal pattern shown in c). The far right diagram shows the final cross pattern that can be formed by from a single source by transmitting the square and then diamond wave.

4 Geometric Problems on the Digital Disc

A set ζ of points in $\mathbb{Z}^2$ is a **digital disk** if there exists a Euclidean circle, with a centre at an integral point, that encloses the pixels of ζ but excludes its complement. Let us consider a model of broadcasting automata on a digital disk which has a diameter D. We define a procedure for finding the centre of the digital disk in linear time using the notion of waves as described in previous sections.

Problem 1 *Beginning from any point on the circumference of the circle it is possible to find the centre of the digital disk as a single point or as a set of two points, depending on whether the radius of the digital disk is odd or even, respectively.*

In this section we abbreviate the partitions previously mentioned to $a = \{D_9\}$, $b = \{D_1, D_3, D_5, D_7\}$ and $c = \{D_2, D_4, D_6, D_8\}$. The algorithm for finding the centre can begin from any arbitrary point, T_1, on the edge of the digital disc and is implementable in both asynchronous and synchronous models.

Depending on the location of the initial point, one of three algorithms is applicable. Finding the correct algorithm to apply is reduced to checking the initial point's neighbourhood to one of three possible sets in the following way.

Definition 4 *Eight points* $\{0, 1, ..., 7\}$ *on the circumference of a digital circle,* ζ *corresponds to the following eight angles* $\{0, 45, 90, 135, 180, 225, 270, 315\}$.

Lemma 3 *Given an automaton on the edge of* ζ *it is possible to check the automaton's membership to one of three sets:* $\{0, 2, 4, 6\}$*,* $\{1, 3, 5, 7\}$ *and all other points on the edge of the circle in a time* $O(D)$ *for both models.*

Proof: From automaton T_1, on the edge of ζ, Proposition 4 is applied via the transmission of a square then diamond wave, resulting in horizontal, vertical lines of elected automata and electing at most two paths. All elected automata on the edge of the disc, ζ, become new points T_N which is a set of points encompassing up to three automata, $\{T_2, T_3, T_4\} \in T_N$. As soon as the automaton or automata on the edge of the disc, ζ, have been elected by T_1, the automata, now denoted as T_N, begin transmission of a square wave. As the transmission of these waves from all automata in T_N, may occur simultaneously on the disc, points at which waves meet each other proceed no further on the disc, due to the automata's inability to receive and broadcast at the same time, cancelling each other. Points of wave cancellation are shown as dotted lines in Figure 11 and Figure 12. The partitions formed by the transmissions from new points in T_N can now be detected by the initial point T_1 through the transmission of a neighbourhood detection wave which gives nodal patterns to automata through the transmission of its own square wave and causing neighbouring automata to transmit their states which allows the detection of T_1's neighbouring nodal patterns. By Theorem 1, possible neighbourhood partitions for the initial point T_1 are now be categorised as $\{a\}$,$\{a, c\}$ and $\{a, b, c\}$ which are the points $\{0, 2, 4, 6\}$ and $\{1, 3, 5, 7\}$ and all other points respectively. The procedure requires only three waves of transmissions, each wave requires an amount of time that is no more then the diameter of the circle as well as some constant time between transmissions and the constant time for the neighbourhood recognition. Figure 11

Figure 12

□

An Algorithm for Locating the centre of a Digital Disk

1. An automaton on the edge of the disc ζ, T_1, checks its location by the creation of the unique local neighbourhood sets: $\{a\}$, $\{a, c\}$ or $\{a, b, c\}$, see Lemma 3.

2. In case of neighbourhood set $\{a, c\}$ apply the algorithm for case 1. In case of neighbourhood set $\{a\}$ apply algorithm case 2.

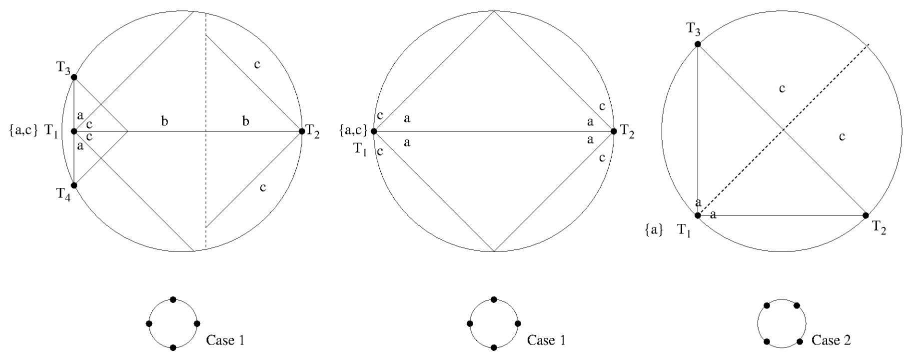

Figure 11: The above Figure shows two of the three possible cases stemming from the five possible variants that require differing solutions based on their location. The two differing sets of the 8 points and those points which lay in none of these. The two diagrams for Case 1 correspond to the situations whereby, for case 1, T_1 generates a 3-branched tree with two equidistant branches or a single chord respectively.

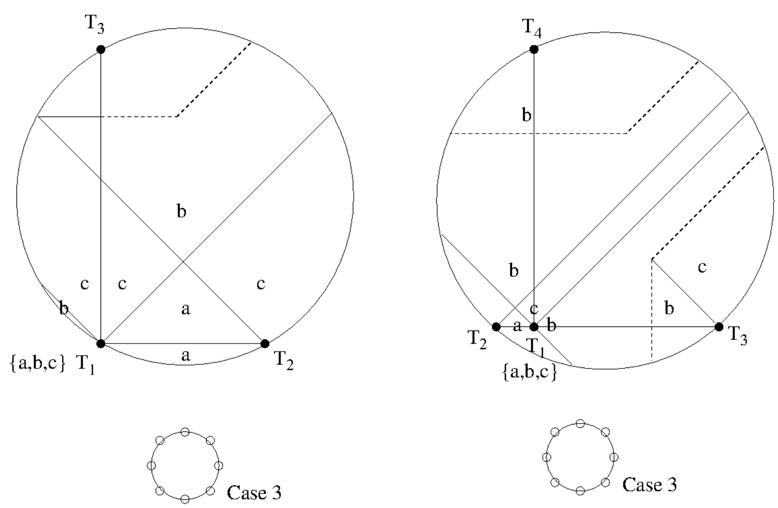

Figure 12: The above figure shows the two possible variations of the final case of the three cases. For case 3 the situations are those such that T_1 generates a 2- or 3-branched tree respectively.

3. In the case of neighbourhood set $\{a, b, c\}$, T_1 is the root of a tree with two or three branches. The third branch may appear if the automata, T_1, finds itself on a 'ledge', such that there are automata on three sides of its Von Neumann neighbourhood, formed from the digitization of the circle. The end point of the shortest branch of such tree, placing the automata in a position whereby there are only two automata in its Von Neumann neighbourhood, is found by Lemma 2 which is relabelled T_1 and then apply case 3.

Cases 1 and 2 are basic because the location of the point T_1 is known exactly. The least trivial case is Case 3 where further partitioning is required for locating the centre.

Algorithm for Case 1 **(set** $\{0, 2, 4, 6\}$**)**

1. T_1 sends message m_0 to T_2 through the chord which will be sent back from T_2 after some constant delay $k = |\Sigma|$.

2. T_1 sends message m_1 which has a delay of 3 after a constant delay $k = |\Sigma|$.

3. The automata on the chord elected through receipt of both message m_0 and m_1 at the same time will be the centre of the digital disc ζ.

Algorithm for Case 2 **(set** $\{1, 3, 5, 7\}$**)**

1. A new point T_4 is generated along the diagonal through the use of diamond neighbourhood detection wave as described in Proposition 4.

2. T_1 sends message m_0 to T_2 through the chord which will be sent back from T_2 after some constant delay $k = |\Sigma|$.

3. T_1 sends message m_1 which has a delay of 3 after a constant delay $k = |\Sigma|$.

4. The automata on the chord elected, through receipt of both message m_0 and m_1 at the same time, will be the centre of the digital disc ζ.

Figure 13

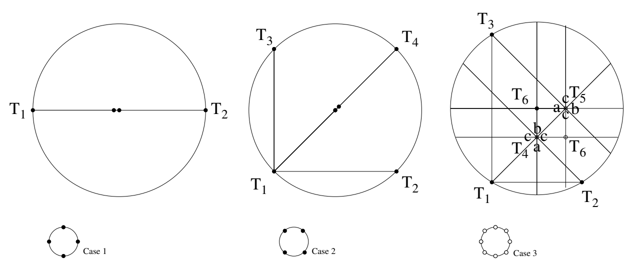

Figure 13: Constructions required to find the centre of the circle for the three differing cases. Centres are indicated by black dots.

Algorithm for Case 3

1. Point T_2 is identified as the shortest chord of the tree constructed from T_1 via Lemma 2.

2. Point T_3 is identified as the longest chord of the tree constructed from T_1 by broadcasting a signal c only to those automata that have received two a's but no b's after Lemma 2, the longest chord.

3. Transmissions from T_3 followed by T_1 elect the point T_5, the only point on the digital disc that has a neighbourhood containing the nodal patterns $\{a, b, c\}$.

4. Transmissions from T_2 followed by T_1 elect the point T_4, the only point on the digital disc that has a neighbourhood containing the nodal patterns $\{a, b, c\}$.

5. Application of proposition 4 from T_4 and T_5 will generate horizontal and vertical lines. The points at which these two line cross are elected as T_6. Only one of the two points marked T_6 will be in partition a given by the initial construction of T_1 and T_3, this is the centre point of the circle.

Theorem 2 *It is possible to find the centre of a digital disc ζ with diameter D starting from a point on the circumference of ζ, T_0, in both models, in $O(D)$ time* [2] .

Theorem 3 directly follows from the construction of the next algorithm for electing the elements of the inscribed square and also holds when starting from a point is on the circumference of ζ.

An Algorithm for electing elements of the inscribed square

1. From point P_1 transmit two waves (square and diamond) to elect the four points A_1, A_2, A_3 and A_4. This follows from Theorem 4 whereby the already mentioned subset is reduced by intersection with the set of those points on the edge of the circle. (see Figure 14).

2. A square wave is transmitted from the points A_1, A_2, A_3 and A_4 as elected in the previous step of the algorithm. The waves form four squares, shown as S_1, S_2, S_3, S_4 but do not intersect each other as once the waves meet all transmission halts. The points at which the waves meet are exactly those that are along the diagonal lines that intersect the central point P_1.

3. A wave is now sent from P_1 forming nodal patterns throughout the circle. It can be noted now that the patterns that form, for each pair, (P_1, A_i), are those which are given in Theorem1.

4. Another wave is now propagated from P_1 which informs the automata of their neighbours and allows the automata to place themselves in the set D_9 if it is within the inscribed square.

Note that the complement of the square will also have a distinct pattern of type $\{D_2, D_4, D_6, D_8\}$ again given by Theorem1.

Theorem 3 *Given an initial transmitting node in the centre of a digital disc. It is possible to elect automata forming an inscribed square in time $O(D)$, where D is the diameter of the digital disk.*

Figure 14

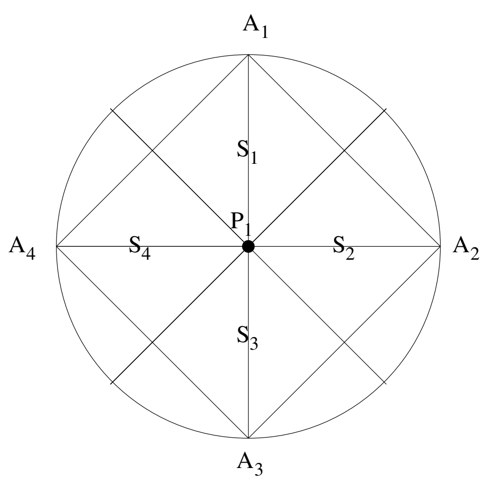

Figure 14: Figure corresponds to production of the inscribed square from centre point.

5 Conclusion and Discussion

We have shown that non-oriented broadcasting of messages on the square grid can form stable interference patterns. Such patterns can be used for efficient partitioning, self-location problems and geometric computations on the static cluster (of robots or automata) via transmission of discrete square and diamond waves, where shapes are defined by the radius of broadcasting and the topology of the grid structures. These shapes can be much more complex then square and diamond waves when we can choose larger radii, higher dimensions and other structures of grid topology. For example, broadcasting with a radius three on the square lattice resulting in the octagon shape of the wave which also, in its turn, provides more complex partitioning of the lattice. Based on our ongoing experimental and theoretical work with lager radii we have observed that they can form quite complex shapes and patterns for square, triangular and hexagon grids as well for high dimensional structures.

Moreover, the sequence of the string that is transmitted by a wave can also be increased in complexity (i.e. different periodic or non-periodic strings, numerical sequences, etc.) along with the corresponding aggregation function. For example, in the case of natural wave intersection the aggregation function is simply the addition of amplitudes which we only extended in this paper through access to a finite history.

The proposed algorithms based on digital waves and nodal patterns can also be extended by using iterative application of informational waves on already generated nodal pattern. So in each subsequent round waves can be transmitted with different delays, speed and shapes based on already formed pattern. Currently we are exploring many other generalization and constraints of the proposed approach which will be presented in future publications by the authors.

The current paper extends the existing area of wave algorithms by introducing new methods, framework and models for further theoretical analysis and practical implementation of complex self-orientation and self-organization mechanisms.

References

[1] Alderighi, Sechi, Mazzei, Tisato: Broadcast Automata: a Parallel Scalable Architecture for Prototypal Embedded Processors for Space Applications. HICSS (5) 1997: 208-217.

[2] Jurek Czyzowicz, Leszek Gąsieniec, Andrzej Pelc , Gathering Few Fat Robots in the Plane, Principles of Distributed Systems, LNCS, Volume 4305/2006, 350-364 pp, 2006.

[3] David Peleg, Asaf Efrima, Distributed algorithms for partitioning a swarm of autonomous mobile robots, Theoretical Computer Science,Vol. 410 , Issue 14, 1355-1368 pp, 2009.

[4] Martijn Hendriks: Model Checking the Time to Reach Agreement. FORMATS 2005: 98-111.

[5] Lee, Geunho and Chong, Nak: A geometric approach to deploying robot swarms. Annals of Mathematics and Artificial Intelligence 2008: 257-280.

[6] J. Kari, Theory of cellular automata: a survey, Theor. Comput. Sci. 334 (2005)

[7] Standing Waves http://en.wikipedia.org/wiki/Standing_wave, January 2010.

[8] Mondada, F., Pettinaro, G.C., Guignard, A., Kwee, I.V., Floreano, D., Deneubourg, J.-L., Nolfi, S., Gambardella, L.M., and Dorigo, M. 2004. SWARM-BOT: A New distributed robotic concept. Autonomous Robots, 17(2/3):193-221.

[9] Efrima, A. and Peleg, DT. 2007. Distributed models and algorithms for mobile robot systems. In Proceedings of SOFSEM (1). Lecture Notes in Computer Science, vol. 4362. Springer, 70-87.

[10] Beni, G.: From swarm intelligence to swarm robotics. In Sahin, E., Spears, W., eds.: Swarm Robotics: State-of-the-art Survey. Lecture Notes in Computer Science 3342, Springer-Verlag (2005) 1-9

[11] J. McLurkin and J. Smith. Distributed algorithms for dispersion in indoor environments using a swarm of autonomous mobile robots. In Proceedings of Distributed Autonomous Robotic Systems Conference, 2004.

[12] Bayindir, L. and Sahin, E. (2007). A review of studies in swarm robotics. Turkish Journal of Electrical Engineering & Computer Sciences, 15(2):115-147.

[13] Trianni, v., Nolfi, s., and Dorigo, M. 2006. Cooperative hole avoidance in a swarm-bot. Robotics Auton. Syst. 54, 2, 97-103.

[14] S. Nouyan, A. Campo, and M. Dorigo, Path formation in a robot swarm: Self-organized strategies to find your way home, Swarm Intell., vol. 2, no. 1, pp. 1-23, Mar. 2008.

[15] S. Nouyan and M. Dorigo. Chain formation in a swarm of robots. Technical Report TR/IRIDIA/2004- 18, IRIDIA, Universite Libre de Bruxelles, March 2004.

[16] R. Groy, M. Bonani, F. Mondada, M. Dorigo, Autonomous self-assembly in a swarmbot, in: K. Murase, K. Sekiyama, N. Kubota, T. Naniwa, J. Sitte (Eds.), Proceedings of the Third International Symposium on Autonomous Minirobots for Research and Edutainment, Springer Verlag, Berlin, 2006, pp. 314-322

[17] R. Olfati-Saber, Flocking for multi-agent dynamic systems: Algorithms and theory, IEEE Trans. Autom. Control.

[18] K. Passino, Biomimicry of bacterial foraging for distributed optimiza- multirobot control, in Proc. Int. Conf. Intelligent Robots Systems, Sept.

6 Legends

Figure 1: A variety of transmission radii are shown (l-r) squared radii $r^2 = \{1, 2, 4, 5, 8, 9, 10, 13, 16\}$. Crosses represent the centre of the respective discrete disc.

Figure 2: Diagram $a)$ represents the propagation pattern for a diamond wave (Von Neumann neighbourhood) and diagram $b)$ shows the propagation pattern for a square wave (Moore neighbourhood). Also wave propagation with the asynchronous model is shown on the square grid: Moore neighbourhood (left) and Von Neumann neighbourhood (right).

Figure 3: The detection of a none edge case, that has two neighbours (left), and an edge case, with only one neighbour (right).

Figure 5: Shows two transmitters T_1 and T_2 broadcasting words u and v respectively.

Figure 6: Representing the initial configuration of combinations of words u and v as a graph.

Figure 7: The second self symmetric graph constructed by the combination of words u and v.

Figure 8: A pair of self symmetric graphs which are equivalent via inversion.

Figure 9: The distribution of three distinct nodal patterns in case of two 'transmitter', where $s = 4$.

Figure 10: Figures a) and b) show the nodal labellings of the automata from the square and diamond waves respectively which generate the final nodal pattern shown in c). The far right diagram shows the final cross pattern that can be formed by from a single source by transmitting the square and then diamond wave.

Figure 11: The above Figure shows two of the three possible cases stemming from the five possible variants that require differing solutions based on their location. The two differing sets of the 8 points and those points which lay in none of these. The two diagrams for Case 1 correspond to the situations whereby, for case 1, T_1 generates a

3-branched tree with two equidistant branches or a single chord respectively.

Figure 12: The above figure shows the two possible variations of the final case of the three cases. For case 3 the situations are those such that T_1 generates a 2- or 3-branched tree respectively.

Figure 13: Constructions required to find the centre of the circle for the three differing cases. Centres are indicated by black dots. Figure on the right corresponds to production of the inscribed square from centre point.

Notes

[1]It is also possible to show, in a more or less straightforward way, that broadcasting automata on $\mathbb{Z}^n$ (for any $n > 0$), with a single initial source of transmission, two radii of broadcasting (1 and 1.5) and a large alphabet of messages, can simulate a Turing Machine.

[2]As there are a finite number of passing of waves which all time-bounded by at most $O(D)$, the algorithm cannot exceed linear growth by D.