MAXMIN-PLUS MODELS OF ASYNCHRONOUS COMPUTATION

A thesis submitted to the University of Manchester
for the degree of Doctor of Philosophy
in the Faculty of Engineering and Physical Sciences

2012

Ebrahim L. Patel
School of Mathematics

Contents

List of Tables

List of Figures

The University of Manchester

Ebrahim L. Patel
Doctor of Philosophy
Maxmin-plus Models of Asynchronous Computation
February 3, 2012

This thesis aims to better represent a framework for asynchrony. Traditional asynchronous models, particularly those used to simulate cellular automata, have used stochasticity or randomness to generate update times. We claim that, while they may make good representations of their application, such asynchronous methods rid the model of the essence of interesting asynchronous processes. Thus, we attempt to better harness the aspects internal to the decision process of such discretely dynamic cells as those in cellular automata.

We propose the maxmin-m model as a suitable model for the asynchronous computation of cellular automata. The model uses maxmin-plus algebra, a special case of which is max-plus algebra. This algebra arises naturally from the cellular automaton requirement that a cell receives the state of its neighbours before updating. The maxmin-m model allows each cell to update after it receives m out of a possible n neighbours' states.

The max-plus model shows that, while update times may be asynchronous in real time, there is no loss of information, since the corresponding asynchronous process is bijectively related to the synchronous model. In turn, the cellular automaton output, measured by the Shannon and word entropies, is shown to vary little from the synchronous model. Moreover, this type of asynchrony is simple, i.e. it is deterministically obtained due to the linearity of max-plus algebra.

Indeed, the maxmin-m model is also shown to be deterministic and always reaches periodic behaviour. In the long time limit, this model is shown to be represented by a max-plus model, supporting its determinism further. Consequently, the complexity of such a model may be thought to be limited. However, we show through large scale experiments that the case $m \approx n/2$ generates most complex behaviour in terms of large periods and transients to the aforementioned periodic orbits. In particular, the complexity is empirically shown to obey a bell form as a function of m $(1 \leq m \leq n)$. The resulting cellular automaton simulations indicate a correspondence from the complexity of the update times. Therefore, cellular automaton behaviour may be predictable with the type of asynchrony employed in this thesis.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright Statement

Acknowledgements

I am honoured to have been supervised by David Broomhead. David's unfortunate absence meant that Mark Muldoon guided me during the critical phase of writing up. I am grateful for Mark's help and constant support in this time. I thank all of the people that have experienced with me the highs and lows of being a PhD student, including friends outside of academia, whom have kept me in touch with reality - their contributions, no matter how small or irrelevant they may have seemed, have been noted and must, therefore, take their place in a most complex space-time pattern of this period of my life. I have especially enjoyed the pleasant company of those that have shared an office with me, and I particularly thank Andrew Irving and James Arnold for the stimulating friendship that was created. Finally, showing my immense gratitude to my family would not do justice to the love that I hold for them. Thank you to my parents for their unwavering support and to my brothers for giving me strength and keeping me positive.

Chapter 1

Introduction

In writing this piece of work, the author was intrigued by an observation by Neil Johnson in [25] in which a feature is expounded that links art, music and even mountain ranges or city skylines. This is the aesthetic feature of such things being most interesting when they are neither too ordered nor too disordered.

Such an observation could also have been made by many academics working in the field of Complex Systems, which comprises of systems that have the remarkable ability to yield a complete spectrum of types of behaviour, from order ("homogeneity") to disorder (or "heterogeneity").

For clarity, the term "Complex Systems" (with capital letters) denotes the subject/field, whereas "complex system" (fully lower case letters) will refer to a particular system contained in the field of Complex Systems.

But what is a complex system? The literature on the subject indicates that there is no strict definition. Indeed, it has been observed that there is a large class of systems that exhibit a few significant traits which qualifies them to be regarded as complex. Although their quantity is small, it is widely accepted that a candidate for a complex system is characterised by these fundamental features, which include the following [25]:

- The system contains a collection of many interacting objects

- These objects' behaviour is affected by feedback

- The system exhibits emergent phenomena which are generally surprising, and may be extreme

- The emergent phenomena typically arise in the absence of any sort of 'invisible hand' or central controller

- The system shows a complicated mix of ordered and disordered behaviour.

The final feature in the above list leads us to think about exactly how this interesting behaviour is produced. In particular, systems with interacting objects abound in the real world, from the microscopic, e.g. the interaction of biological cells, through to the global level, as in the communication of information over the Internet. There is a potential ubiquity of applications, and this leads us to conceive of the growing role and importance of understanding Complex Systems.

1.1 Background

The dynamics on a network can be an archetypal complex system and is indeed the subject of a growing number of research activities and interests. Mathematically, the network is regarded as a *graph*, where directed arrows ("arcs") point from vertex to vertex (a vertex may sometimes also be referred to as a node). The dynamics could represent information transfer, for example.

1.1.1 Random Boolean networks

A fundamental study of dynamics on graphs was carried out by Stuart Kauffman in 1969 [26]. In his paper, he introduced the idea of Boolean networks, whose nodes take value 0 or 1. This value is known as the state of the node, and the collective states form the state of the network. The state is discretely dynamic, in the sense that the state of a node at time $n+1$ is determined by the state of the nodes that provide input to it at time n. This is computed by assigning to each node i a Boolean function $f_i : \{0,1\}^k \to \{0,1\}$, where k is the number of distinct inputs to i; k is fixed for all nodes i, and it might also be termed the "in-degree" of i. The connections between nodes are chosen randomly, as are the Boolean functions, hence the name "random Boolean network" (RBN), and this choice is then fixed to enable the temporal analysis of the state through iterated applications of f_i for all i. Let x_i^n be the state of node i at time n. The RBN of N nodes is then a dynamical system with state vector $X^n = \{x_1^n, x_2^n, \ldots, x_N^n\} \in \{0,1\}^N$ and whose evolution is governed by the map $F : \{0,1\}^N \to \{0,1\}^N$. F is a system of N Boolean functions as above, and can be written

$$\begin{aligned} x_1^{n+1} &= f_1(x_{11}^n, x_{12}^n, \ldots, x_{1k}^n) \\ x_2^{n+1} &= f_2(x_{21}^n, x_{22}^n, \ldots, x_{2k}^n) \\ &\vdots \\ x_N^{n+1} &= f_N(x_{N1}^n, x_{N2}^n, \ldots, x_{Nk}^n). \end{aligned}$$

Element x_{ij} represents the jth variable of f_i, and its existence in f_i depends on the existence of an input from the corresponding node to node i.

Kauffman used his Boolean network to model the gene regulation of a living organism, where the nodes represent genes. The network is known as a genetic regulatory network (GRN). A simple reason given by Kauffman for representing the states of the nodes as Boolean variables is that the effect of one gene on the rate of output of another is probably an 'all or none' process, i.e. it either occurs or not.

The following definitions are required for what follows.

Definition 1.1.1. A *periodic orbit* of length $p \in \mathbb{N}$ is a set of states

$$\mathbf{X} = \{X^1, X^2, \ldots, X^p\} \subseteq \{0,1\}^N$$

such that $X^{n+1} = F(X^n)$ for $1 \leq n < p$ and $F(X^p) = X^1$.

Let $F^m(Y)$ denote the map F applied to the state $Y \in \{0,1\}^N$ m times. Given a periodic orbit $\mathbf{X}$, we define its *basin of attraction* to be the set

$$\mathbf{B} = \left\{Y \in \{0,1\}^N \mid \exists n_\star \in \mathbb{N} \text{ such that } F^m(Y) \in \mathbf{X} \quad \forall m \geq n_\star\right\}.$$

The size of the basin of attraction is $|\mathbf{B}|$ and clearly $\mathbf{X} \subseteq \mathbf{B}$.

Whilst Kauffman demonstrated RBNs to be a useful tool for modelling GRNs, an important result of his work was that such a dynamical system consisting of N coupled randomly chosen Boolean functions is most stable when the number of inputs to each node is 2. To be precise, for $k = 2$, the state space was found to be split into a small number of small periodic orbits (Kauffman called them cycles) having large basins of attraction. As k was increased, the periodic orbits became longer and the basins smaller, i.e. instability was increased.

It can thus be concluded that this RBN is a complex system capable of ordered to disordered behaviour, with the simple variation of the parameter k. Another point to note is that Kauffman favoured such ordered behaviour (found when $k = 2$) since it was more amenable to biology: it correspondingly may be able to feasibly model the GRN. Specifying the state of a node to be Boolean is also popular because of the simplicity of such a minimal number of states. Allied with Kauffman's work, which identifies stability for a small k value, this makes the Boolean model well suited to computational simulation, producing the required results quickly. Although the results may be a simplification of the intended real-life application, such 'minimalistic' models have been found to produce behaviour which not only mimics simple dynamical systems, but also generates insight into the extent of complexity that the systems may reach, e.g. through the exhibition of instability when $k > 2$ above. This has been simulated, reworked and remodelled by changing only a few parameters to even model something as complex as life itself [12, 40]. With the latter comment, we introduce another discrete dynamical system known as a *cellular automaton*, the plural being *cellular automata*(CA)[1].

1.1.2 Cellular automata

CA were introduced by John von Neumann in the 1950s as a possible idealization of biological systems, with the particular purpose of modelling biological self-reproduction [41]. A term that is applied to CA by computer scientists is that of a "universal system", which is a system capable of carrying out arbitrary computations; an infinitely programmable computer. Thus, CA can be regarded as a simple mathematical model of a variety of systems, be they biological, computational or physical. CA consist of a lattice of identical automata, or "cells", where each cell takes one of a finite set of states. Traditionally, the cells would be arranged regularly, in the sense that the connectivity of cells would be uniform and their neighbourhoods the same structure and size [43, 24]. However, this can be extended to include the general case of a lattice with irregular topology and varying neighbourhood size. This consequently gives rise to cellular automata on a network, which we may also refer to as a directed graph (*digraph*), where nodes play the role of cells and incoming arcs indicate the neighbourhood of a cell [31]. By restricting the state space that each node can take to two values, we obtain a Boolean network; Boolean networks can, thus, be regarded as subsets of generalised CA on digraphs.

The foundations of the study of CA were laid by Stephen Wolfram in the 1980s by considering a one-dimensional lattice [43]. Let s_i denote the state of cell i. The index i denotes the position of the cell in the one-dimensional Euclidean plane, so that the

[1] The abbreviation CA will be used to denote both the singular and plural forms, with the context saving confusion

state $\mathbf{s}$ of the CA at any given time can be represented by a string of individual cell states or a 1 by N vector of cell states, where N is the size of the lattice, i.e. the string $s_1 s_2 \ldots s_N$ or the vector $\mathbf{s} = (s_1, s_2, \ldots, s_N)$. Like the RBNs earlier, a function f_i is applied to evolve s_i in discrete time; f_i is called the "CA rule". Kauffman's random Boolean networks assign f_i to be variable with i, whereas here f_i is the same for all cells. Thus, we may replace each f_i by the function f, and it is a function of the states of k neighbourhood cells of each node. Let $s_i(t)$ represent the state of cell i at time t. Then, the state $s_i(t+1)$ is obtained by applying rule f as follows.

$$s_i(t+1) = f(\mathcal{N}(s_i(t)))$$

where $\mathcal{N}(s_i(t)) \in \mathbb{R}^k$ is the vector of real-valued states of the k neighbours of i at time t. It is also referred to as the *neighbourhood of s_i*.

In [43], Wolfram restricted the neighbourhood size to $k = 3$ and considered $\mathcal{N}(s_i(t))$ to be "regular", i.e. not randomly chosen. In particular, the neighbourhood of s_i on time t was chosen to be $\mathcal{N}(s_i(t)) = (s_{i-1}(t), s_i(t), s_{i+1}(t))$. (The index t is removed for a temporally fixed neighbourhood). Thus,

$$s_i(t+1) = f(s_{i-1}(t), s_i(t), s_{i+1}(t))$$

For example, consider

$$f = s_{i-1}(t) + s_i(t) + s_{i+1}(t) \mod 2 \tag{1.1}$$

Likewise, there are other forms of rule f. By considering only Boolean states[2], i.e. 1 or 0, Wolfram devised a way of labelling all possible 3-state neighbourhood rules [43]. As each cell may take one of two states, and there are three such cells in the neighbourhood under consideration, then there are $2^3 = 8$ distinct forms for the neighbourhood of s_i. By representing the neighbourhood in string form, i.e. $s_{i-1}s_is_{i+1}$, we can read it as a binary value, e.g. the neighbourhood represented by the three digit string 110 takes value 6 in usual arithmetic. Together, the eight distinct neighbourhoods determine the Wolfram rule by first reading each neighbourhood as such a binary value. Placing these neighbourhoods in order of size (largest first) and reading the eight outputted values as an eight-digit binary number gives the binary representation of Wolfram's rule number.

Thus, according to (1.1), we obtain the following.

[2] In CA terminology, state 1 is often referred to as "ON", state 0 being "OFF"

$\mathcal{N}(s_i(t))$	111	110	101	100	011	010	001	000
$s_i(t+1)$	1	0	0	1	0	1	1	0.

The Boolean string 10010110 takes value 150 in ordinary arithmetic. Therefore, (1.1) is known as CA rule 150.

Since the application of an arbitrary f to each neighbourhood may yield one of two outputs, there are a total of $2^8 = 256$ such 3-state Boolean CA rules. Wolfram called these rules the "Elementary CA" (or ECA).

The evolution of an initial state according to f can be represented as a space-time diagram as in Figure 1.1 with time running vertically up. OFF cells are represented by a blue square and ON cells are orange, whilst the initial state comprises one cell ON and all others OFF; the rule used is rule 150. The cellular automaton size is 10 cells, where the boundary cells are adjacent, so as to form a closed 'ring' lattice. We refer to this picture as a *space-time plot* or *pattern*.

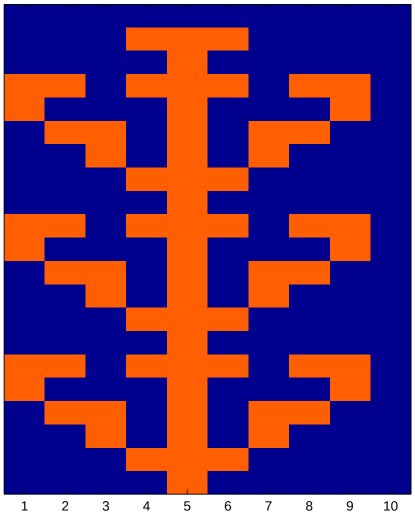


Figure 1.1: Evolution of a one-dimensional CA according to ECA rule 150. Time travels up.

In general, because the size N of the CA is finite, and the set Σ of possible states of each cell is finite ($\Sigma = \mathbb{Z}_2 = \{0, 1\}$ here), the number of possible states of the system is finite. After a maximum of 2^N time steps, such a finite-size CA revisits a previously encountered state. Thus, we would see periodic orbits of states of such a system. Revisiting Figure 1.1 again, notice that the overall pattern produced is highly regular and ordered. One reason for this ordered behaviour is the synchronous mode of applying rule 150, i.e. it is applied to each cell at the same time, so that all cells are updated on each time step. This, in turn, supports the deterministic form of such synchronous CA; given an initial state $\mathbf{s}(0)$, any state $\mathbf{s}(t)$ is found by iterating the ECA rule t times. Therefore, given a finite lattice, a diagram of CA states can be drawn up such that each state may be predicted in time. This is detailed next.

State transition graph

Finiteness of the lattice ensures periodic behaviour. The lattice whose connections are arbitrary is best represented as a digraph. To make the demonstration of this section clearer, we use a small network. Moreover, the arbitrary connectivity turns the network into a Boolean network.

Consider the underlying network of size $N = 4$ as given in Figure 1.2. Let the CA rule that updates the CA state of each cell be

$$f(s_i(t+1)) = \sum \mathcal{N}(s_i(t)) \mod 2$$

i.e. the new state of each cell is the sum of the states of its neighbourhood cells on the previous time step, where the neighbourhood of cell i comprises those cells whose outgoing arc points to i. Note that this rule is an extension of ECA rule 150 to arbitrary lattices, and we shall often refer to it by this name. There are exactly 2^N CA states of a network with N vertices. For small N, as is the case here, it is useful to represent each state as a vertex in a digraph. Thus, there is an arc from CA state $\mathbf{s}^i$ to CA state $\mathbf{s}^{ii}$ if and only if $f(\mathbf{s}^i) = \mathbf{s}^{ii}$. The digraph is known as a *state transition graph* or STG for short. For the $2^4 = 16$ possible CA states of the system in question, the STG is given in Figure 1.3. Each CA state is shown in string form, where the ith digit represents the CA state of the ith node. We follow the arcs to determine the evolution of the CA. It can be seen that an initial CA state

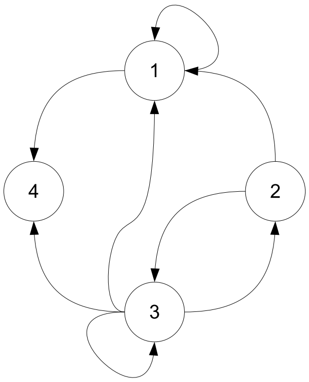


Figure 1.2: Size 4 network of CA cells.

asymptotically evolves into one of four periodic orbits, represented as circuits in the STG. Two of these (states 0000 and 1001) are period-1 orbits (which are termed as *fixed points* in conventional dynamical systems language), and two are period-3 orbits. All other states are transient and they can usually only ever exist as initial states; in terms of a random Boolean network, such states may also be defined to be contained in a basin of attraction of the downstream periodic orbit. In the STG, this implies that these states are never contained in a circuit.

Cellular automaton classification

By conducting extensive computer simulations, Wolfram went on to classify each of the 256 ECA rules by studying the space-time pattern produced. The initial states were randomised and the behaviour empirically examined until a periodic pattern (if any) was observed after some transient time. The eventual classification was known as the "Wolfram Classes" and the conclusion was quite surprising: for most initial states, such a large number of ECA rules could be divided into only four classes. The classes are given below [44].

- Class I: CA evolves to a homogeneous state.
- Class II: CA evolves to simple separated periodic structures.
- Class III: CA shows chaotic aperiodic patterns.

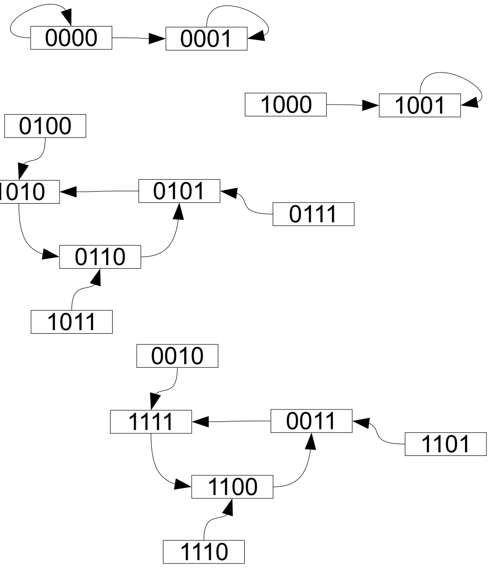


Figure 1.3: State transition graph of the CA rule $f(s_i(t+1)) = \sum \mathcal{N}(s_i(t)) \mod 2$ applied on the size 4 network given in Figure 1.2.

- Class IV: CA yields complex patterns of localized structures.

In the field of dynamical systems, Class I CA may be interpreted as yielding limit/fixed points, Class II CA evolve to periodic orbits/cycles, Class III CA are analogous to strange attractors, and Class IV contains CA that yield long transients that make prediction of the CA almost impossible. In fact, by considering larger nbhd sizes k and larger state set sizes $|\Sigma|$, the four classes were found to be existent in almost all CA. The classes thus appear to be universal [44]. Again, we see a system that has the ability to self-organize into ordered or disordered behaviour from very simple local interactions.

The natural extension to an arbitrary lattice of cells requires methods that are different to those used to classify Wolfram's one-dimensional CA. In particular, when thinking about cellular automata on a digraph, connections between cells (represented as nodes now) may not necessarily be regular, and spatial neighbourhood information may be lost when studying the space-time evolution of the CA such as that in Figure 1.1. Of particular interest then are those ECA rules that are most convenient and carry the most potential to be utilised for arbitrary neighbourhoods such as in a network. From the 256 3-state neighbourhood rules, it turns out that 32 satisfy such conditions [44, 31]. To be precise, the conditions are that

- rules are "legal" [44], i.e. (i) applying a rule to a state of solely OFF cells leaves the state unchanged, and (ii) the rule is reflection symmetric, e.g. for a neighbourhood of three cells, if the middle cell is in question, then the neighbourhood 100 is equivalent to 001 and 110 is equivalent to 011 so that equivalent neighbourhoods induce the same output.

- rules are "totalistic", i.e. a rule depends only on the relative number of ON and OFF cells in a neighbourhood, and not on their order, e.g. neighbourhoods 110, 101 and 011 are equivalent.

A step towards addressing the need to classify CA on graphs is provided by Marr and Hütt in [31]. Two entropy measures $S(T)$ and $W(T)$ are presented which evaluate the time development of single cells that carry Boolean states. Let T denote the length of the time series of a cell (which is also the number of iterations taken). The *Shannon entropy* $S(T)$ takes values in interval $[0, 1]$ and relies on the density $p(s_j)$ of state s_j in the time series. If s_j occurs $\sharp_j$ times along this time series, then

$p(s_j) = \sharp_j/T$. For each individual cell i, the Shannon entropy is defined as

$$S_i(T) = -\sum_{j=1}^{|\Sigma|} p(s_j) \log_2 p(s_j)$$

where Σ is the set of possible cell states ($= \mathbb{Z}_2 = \{0, 1\}$ for Boolean states). Averaging $S_i(T)$ over all N cells yields the average Shannon entropy $S(T) = \sum_{i=1}^{N} \frac{S_i}{N}$ as a function of T.

The *word entropy* $W(T)$ depends on the occurrence of blocks of constant cells of length l (l-words) in the time series of a cell. Thus, if $p(l)$ is the density of an l-word (irrespective of the state (0 or 1) that this word consists of), then

$$W_i(T) = -\sum_{l=1}^{T} p(l) \log_2 p(l)$$

The average word entropy is given by $W(T) = \sum_{i=1}^{N} \frac{W_i}{N}$.

For a fixed number of iterations T, the Shannon and word entropies are denoted S and W respectively. Although the space-time CA patterns now lose the spatial information, patterns are now classified according to their W and S values within the two-dimensional WS plane. This is shown in Figure 1.4. The four Wolfram Classes each correspond to different regions of this plane. A point to note is that the boundaries between regions are not as definite as those shown in Figure 1.4 because the classes are qualitative descriptions, whereas the corresponding W and S values are quantitative approximations. Nevertheless, the Wolfram Classes can now be identified for arbitrary topologies/networks as regions in the WS plane.

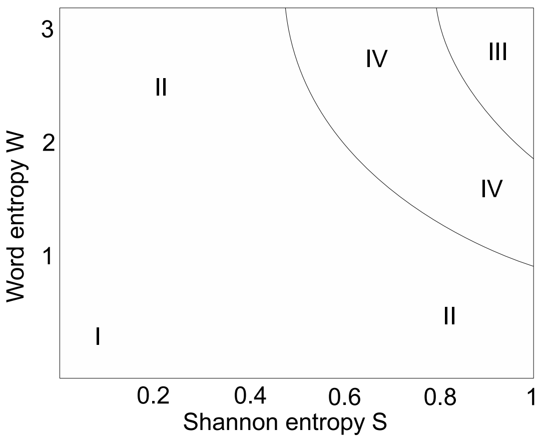


Figure 1.4: The WS entropy plane, signifying regions that correspond to Wolfram's Classes. Image redrawn from [31].

Importantly, the authors observe that transitions between different Wolfram Classes occur with changes in graph topology. Thus, Marr and Hütt demonstrate an approach for analysing the dynamics of CA on arbitrary topologies of cells that aspires to be applicable on any connected graph. Moreover, whereas Wolfram's CA achieved varying types of CA behaviour through the application of different CA rules, Marr and Hütt show that, for a CA rule that is fixed amongst most nodes in the network, the different domains of CA patterns may also be achieved by altering the connections in the graph; the value of $(S, W) \in \mathbb{R}^2$ as a function of change in some connection can then be identified as a path drawn in the WS plane. In some cases, a mixture of behaviour is even observed, where some nodes behave according to one Wolfram Class and some according to another.

In summary, a varying graph topology has been demonstrated as another means of achieving varying types of CA behaviour, from ordered to complex.

It must be noted, however, that identification of Class IV behaviour is difficult as compared to others due to it being comprised of long transients. Thus, after T iterations, a CA pattern may well be mistakenly identified as being in Classes II or III when in fact, taking further iterations would lead to an emergence of Class IV behaviour. Therefore, if we are to definitively identify the Wolfram Classes, then T must be large enough to be able to observe Class IV behaviour. As a consequence, such an experiment might turn out to be impractical, unless constraints are applied to ensure the existence of a desired class. One such constraint is to impose a finite lattice. We show how this impacts on the Wolfram Classes next.

1.1.3 Cellular automata in our model

In this section, the type of cellular automata as introduced above will be detailed with reference to the model that we will demonstrate in the thesis.

Consider the elementary cellular automaton rule 150. For a 3-cell neighbourhood, The rule is $f(\mathcal{N}(s_i(t+1)) = s_{i-1}(t) + s_i(t) + s_{i+1}(t)$, where addition is carried out over the integers modulo 2. The application of f on a size 10 CA has already been shown in Figure 1.1. Such a graphic representation is useful and reveals valuable insight; simple visualisation in this manner also helps to identify the general behaviour of the CA.

The pattern in Figure 1.1 is evidently periodic, which is a consequence of the finiteness of the underlying lattice and the finiteness of the number of states that each cell may take. For each time step t, the CA yields a state $\mathbf{s}(t)$, which is seen again after a few more time steps. Given the initial CA state $\mathbf{s}(0)$ and the CA rule f, an *orbit* of $\mathbf{s}(0)$ is the sequence of states obtained by applying f on $\mathbf{s}(0)$ sequentially. If f is applied k times, we represent this as $f^k(\mathbf{s}(0)) = \underbrace{f(f(\cdots f}_{k \text{ times}}(\mathbf{s}(0))))$. We define periodic behaviour as might be expected from conventional dynamical systems theory.

Definition 1.1.2. Consider the CA rule f and network size N. Let $\mathbf{s}(k) = f^k(\mathbf{s}(0))$ for all $k \geq 0$, where $\mathbf{s}(k)$ is a CA state represented by a $1 \times N$ vector. For some $t \geq 0$, if there exists a finite number $p \in \mathbb{N}$ such that $\mathbf{s}(t+p) = \mathbf{s}(t)$, then the set of states

$$\{\mathbf{s}(t), \mathbf{s}(t+1), \ldots, \mathbf{s}(t+p-1)\}$$

is called *a periodic CA orbit*, where p is the *CA period* of the orbit.

In addition, let $K_C(\mathbf{s}(0))$ be the smallest t for which a periodic CA orbit is obtained after the system is initialised with the initial CA state $\mathbf{s}(0)$. Then $K_C(\mathbf{s}(0))$ is defined as the *CA transient time given* $\mathbf{s}(0)$. When the initial CA state is understood, we may also shorten $K_C(\mathbf{s}(0))$ to K_C.

In Section 1.1.2, we stated that we are interested in those CA rules that provide scope for application to an arbitrary lattice. This is satisfied by those rules that are legal and totalistic. There are 32 legal ECA rules, of which eight are also totalistic. These eight rules separate into Wolfram's Classes I, II and III, but not in IV. As might be expected, Classes I and II have relatively simple behaviour, making them uninteresting to study. Thus, the interesting behaviour is likely to be supplied by those legal, totalistic rules that lie in Class III, which are rules 22, 126 and 150.

Looking ahead to when we will study the effect of asynchronous update times on the CA output, we would like outside effects (as compared to the asynchrony) to be minimal; thus, a rule which synchronously produces both Boolean states with equal probability ensures that any significant features that arise in the CA space-time pattern is attributed to the asynchrony and not the rule. Thus, we impose a condition of fairness in experimentation: we require that a rule f be chosen which does not bias one Boolean state over another, i.e. when considering all eight possible ECA neighbourhoods, the implementation of rule f on these neighbourhoods should yield state 1 four times and state 0 four times. In the literature, this identification is usually characterised by the Langton parameter λ [28], which is the fraction of neighbourhoods yielding the state 1 after one transition of f. Thus, here, we require $\lambda = 0.5$. Now, consider the rule table for each of the rules 22, 126 and 150, as follows.

$\mathcal{N}(s_i(t))$	111	110	101	100	011	010	001	000		
$s_i(t+1)$	0	0	0	1	0	1	1	0	=	**22**

$\mathcal{N}(s_i(t))$	111	110	101	100	011	010	001	000		
$s_i(t+1)$	0	1	1	1	1	1	1	0	=	**126**

$\mathcal{N}(s_i(t))$	111	110	101	100	011	010	001	000		
$s_i(t+1)$	1	0	0	1	0	1	1	0	=	**150**

It is evident that the Langton parameters for rules 22, 126, and 150 are $\lambda = 3/8$, $\lambda = 6/8$, and $\lambda = 4/8$ respectively. Therefore, rule 150 is the only legal and totalistic Class III rule that provides the basis for a fair study of asynchronous CA on a digraph.

A further advantage of rule 150 is its interpretation into matrix-vector form. Consider the size 4 network above again, which is also regarded as a digraph. We can associate to any digraph a square matrix $D \in \mathbb{R}^{N \times N}$ whose $(i, j)^{\text{th}}$ element D_{ij} is 1 if an arc points from node j to node i; D_{ij} is 0 otherwise. Thus, the adjacency matrix of the network in Figure 1.2 is

$$D = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix}.$$

Applying f on state $\mathbf{s}(k)$ is then equivalent to the matrix-vector multiplication $D\mathbf{s}(k)^\top$, where addition is done modulo 2. Consequently, given an initial state $\mathbf{s}(0)$, all subsequent states due to rule 150 may be obtained by the recurrence relation $\mathbf{s}(k) = D^k\mathbf{s}(0)$. A state transition graph is also easily drawn.

In the next subsections, we demonstrate the regions of the WS plane that correspond to Wolfram Classes I and II. As may be expected, on finite networks, Classes III and IV are rare.

Homogeneous CA (Wolfram Class I)

Consider the evolution of a CA pattern such that, after a transient region (in which the states 1 and 0 occur in no fixed pattern), the state $\mathbf{s}(t)$ is fixed (at 0 or 1) for all cells. The transient region is the length of time from the initial time to time $t^* \in \mathbb{R}_+$ such that the state of cell i is $s_i(t)$ for $t \geq t^*$; t^* denotes the CA transient time. To be precise, it is the region corresponding to the time $t \geq t^*$ that we refer to as being homogeneous since this is the attractor of the CA. For now, we loosely define an attractor as the limiting state of the system after a long period of time. The overall CA pattern that evolves to such a homogeneous region is called *eventually homogenous*; the pattern is *eventually periodic* in a similar fashion (later). Note that later on in the thesis, we will dispose of the term "eventually" when describing CA patterns when the asymptotic behaviour of the CA is understood, e.g. an eventually homogenous CA will be termed a homogenous CA.

A typical homogeneous evolution of one such cell is given in Figure 1.5.

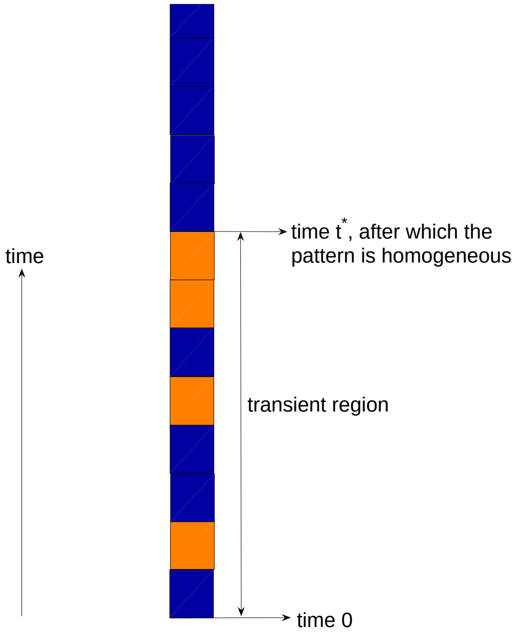


Figure 1.5: Homogeneous cell evolution. The transient time is indicated by the time t^*. Blue squares represent state 0 and orange squares represent state 1.

We can calculate the word entropy as follows. Consider a finite length of time $[t_1, t_2]$ (where $t_1, t_2 \in \mathbb{Z}$) along the CA evolution of a cell. We define a *word* as a block of states along the time series of a cell i such that, for some integer $t \in [t_1, t_2]$, one of the following is true.

- $s_i(t) = s_i(t+1) = \cdots = s_i(t+l-1)$ and $s_i(t) \neq s_i(t+l)$ if $t = t_1$ and $t+l-1 < t_2$ (i.e. if the word in question is the first word in the time interval t_1 to t_2)

- $s_i(t) = s_i(t+1) = \cdots = s_i(t+l-1)$ and $s_i(t-1) \neq s_i(t)$ and $s_i(t) \neq s_i(t+l)$ if $t_1 < t$ and $t+l-1 < t_2$ (i.e. if the word is in the 'middle' of the time interval $[t_1, t_2]$ such that CA states either side are different)

- $s_i(t) = s_i(t+1) = \cdots = s_i(t+l-1)$ and $s_i(t-1) \neq s_i(t)$ if $t_1 < t$ and $t+l-1 = t_2$ (i.e. if the word is the last word in the time interval $[t_1, t_2]$)

- $s_i(t) = s_i(t+1) = \cdots = s_i(t+l-1)$ if $t = t_1$ and $t+l-1 = t_2$ (i.e. if the time interval contains only one type of CA state).

The length of the word is l, so that the word may also be referred to as an l-word. Let $W(T)$ denote the word entropy as a function of T; Recall that T may also be regarded as the number of iterations of the CA. In terms of the time interval, a CA evolution of length T implies that $t_1 = 0$ and $t_2 = T$. Let $T = t^*$ and consider $W(t^*)$. We assume $t^* = 8$, so that the transient region ends after the 2-word consisting of the state 1. Then, in the transient region, there are six words, four of which are 1-words and two are 2-words, so that the density of l-words are $p(1) = 4/6$ and $p(2) = 2/6$. Hence, $W(t^*) = -(\frac{4}{6}log_2(\frac{4}{6}) + \frac{2}{6}log_2(\frac{2}{6})) = 0.9183$. After one more iteration (at time t^*+1), the density of words changes. There are now seven words in total, but each of which are still either 1-words or 2-words, yielding $p(1) = 5/7$ and $p(2) = 2/7$, which gives $W(t^* + 1) = 0.8631$. Continuing in this fashion, we obtain the word entropy as a function of time: $W(t^*+2) = 0.9852$, $W(t^*+3) = 1.3788$, $W(t^*+4) = 1.3788$, and it can be checked that $W(t^* + k) = 1.3788$ for $k \geq 3$.

The above is a somewhat empirical proof to the following lemma, where T is now taken to be t.

Lemma 1.1.1. Consider an eventually homogeneous CA pattern of length t. Let t^* indicate the CA transient time. Then, as $t \to \infty$, the word entropy $W(t)$ of the CA evolution is

$$W(t) = W(t^* + k)$$

where $k = l_{\max} + 1$ and $l_{\max}$ is the largest word length up to t^*.

In other words, as t becomes considerably larger than the CA transient time, the word entropy tends to a steady state, which is slightly larger than the word entropy of the transient region.

The Shannon entropy is simpler to understand. It relies on the density of individual cell states 1 and 0. For the same (homogeneous) output, after the transient region, one state dominates, e.g. in Figure 1.5, the CA is fixed at state 0 after the CA transient time. For Boolean states, the Shannon entropy is defined as $S = -(p(1)\log_2 p(1) + p(0)\log_2 p(0))$, where $p(1)$ and $p(0)$ denote the density of CA states 1 and 0 respectively. Let $S(t)$ denote this Shannon entropy as a function of time, where t is seen to play the same role as T, the number of iterations used to generate the time series of a CA cell. Then, considering the argument that one state dominates in a homogeneous CA output, as $t \to \infty$, $p(s_j) \to 0$ and $p(1 - s_j) \to 1$ for $s_j \in \{0, 1\}$. Therefore, $S(t) \to 0$. Even though $\log_2(0)$ is undefined, we enforce the condition that such a value exists and is zero for our purposes.

To be precise, we obtain the following lemma.

Lemma 1.1.2. Consider an eventually homogeneous CA evolution of length t. Then, as $t \to \infty$, the Shannon entropy $S(t)$ is

$$\lim_{t\to\infty} S(t) = 0.$$

Periodic CA (Wolfram Class II)

We proceed similarly to the previous section by considering an example evolution of a cell, whose state becomes periodic after the transient time t^*. This is shown in Figure 1.6. By assuming each square denotes one time step, the period is seen to be $r = 9$ (while $t^* = 9$ also).

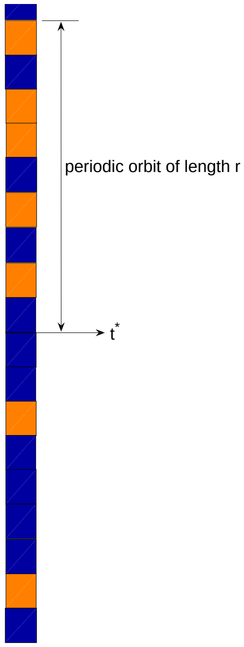


Figure 1.6: CA evolution of a cell that leads to a periodic orbit after CA transient time t^*.

Let $p_s^{t^*}$ denote the density of the CA state s in the transient region; likewise, let p_s^r denote the corresponding density in the periodic CA orbit. Then, in Figure 1.6, the densities in the transient region are $p_0^{t^*} = 7/9$ and $p_1^{t^*} = 2/9$, while the densities in the periodic CA orbit are $p_0^r = 4/9$ and $p_1^r = 5/9$. Let $t^* + r$ denote the first time that the periodic CA orbit is detected, i.e. $t^* + r = 18$. Then, $p_0^{t^*+r} = \frac{7+4}{9+9}$ and $p_1^{t^*+r} = \frac{2+5}{9+9}$.

In general, let $p_0^{t^*} = \frac{\sharp_0^{t^*}}{t^*}$, $p_1^{t^*} = \frac{\sharp_1^{t^*}}{t^*}$, $p_0^r = \frac{\sharp_0^r}{r}$ and $p_1^r = \frac{\sharp_1^r}{r}$, where $\sharp_j^{t^*}$ and $\sharp_j^r$ denote the number of times the state s_j is seen in the transient region and periodic CA orbit respectively. Then,

$$p_0^{t^*+r} = \frac{\sharp_0^{t^*} + \sharp_0^r}{t^* + r}, \quad p_1^{t^*+r} = \frac{\sharp_1^{t^*} + \sharp_1^r}{t^* + r}.$$

We consider further iterations in the form of multiples of r, i.e. we consider the densities each time the periodic CA orbit is repeated. Thus, for $m \in \mathbb{N}$, we obtain the following densities

$$p_0^{t^*+mr} = \frac{\sharp_0^{t^*} + m\sharp_0^r}{t^* + mr}, \quad p_1^{t^*+mr} = \frac{\sharp_0^{t^*} + m\sharp_1^r}{t^* + mr}.$$

By letting $m \to \infty$, we can see that $p_0^{t^*+mr} \to \frac{m\sharp_0^r}{mr} = \frac{\sharp_0^r}{r}$ and $p_1^{t^*+mr} \to \frac{m\sharp_1^r}{mr} = \frac{\sharp_1^r}{r}$, i.e. the densities tend to the densities in the periodic CA orbit. Therefore, the following lemma has been proved.

Lemma 1.1.3. Consider an eventually periodic CA pattern of length t. Then, as $t \to \infty$, the Shannon entropy $S(t)$ of the pattern tends to $S(r)$, where $S(r)$ denotes the Shannon entropy of the periodic CA orbit.

The word entropy for a periodic CA is similarly obtained. Consider again the cellular evolution in Figure 1.6. In the length of time indicated (i.e. $t^* + r$), there are nine 1-words, one 2-word, one 3-word, and one 4-word, giving the word densities $p(1) = 9/12$, $p(2) = 1/12$, $p(3) = 1/12$, and $p(4) = 1/12$. Now consider the densities for a larger time length in which the periodic CA orbit is observed m times. Figure 1.6 indicates the case $m = 1$. Let p_l^t denote the density of l-words after time t. (Figure 1.6 shows $p_1^{t^*+r} = 9/12$, $p_2^{t^*+r} = 1/12$, $p_3^{t^*+r} = 1/12$ and $p_4^{t^*+r} = 1/12$). For larger m, the time $t^* + mr$ asymptotically adds $7m$ 1-words and m 2-words to the fixed number of words in the transient region. On the other hand, since the transient region is such that t^* 'splits' a 3-word, we may view a 'dummy' transient region that has time duration $t^* + r$ instead (and so includes the whole of this 3-word), and the evolution may be regarded as asymptotically adding $7m$ 1-words and m 2-words to the fixed number of words in this region.

Therefore,

$$p_1^{t^*+mr} = \frac{9+7m}{12+8m}, \quad p_2^{t^*+mr} = \frac{1+m}{12+8m}, \quad p_3^{t^*+mr} = \frac{1+0m}{12+8m}, \quad p_4^{t^*+mr} = \frac{1+0m}{12+8m}.$$

In the limit $m \to \infty$, we obtain

$$p_1^{t^*+mr} \to \frac{7}{8}, \quad p_2^{t^*+mr} \to \frac{1}{8}, \quad p_3^{t^*+mr} \to 0, \quad p_4^{t^*+mr} \to 0.$$

Note, however, that the example used above was nice. We can consider an evolution where $t^* + r$ now 'splits' a word in the periodic CA orbit, thereby complicating the identification of words. For example, an eventually periodic cellular evolution may be

$$0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, \ldots$$

in which case $t^* = 4$ and the periodic CA orbit is $\{1, 0, 0, 1\}$, i.e. there are two 1-words and one 2-word, giving the corresponding densities 2/3 and 1/3 respectively in the periodic orbit. However, observe that these densities change when multiples of the periodic CA orbit are considered, e.g. after 11 iterations, the periodic orbit is repeated once. The repeated region is $\{1, 0, 0, 1, 1, 0, 0, 1\}$, in which there are five words, the densities of 1-words and 2-words now being 2/5 and 3/5 respectively, as opposed to 2/3 and 1/3. The cause for concern is the identification of the 'boundary' of the periodic CA orbit. Here, the periodic orbit starts and ends in the same CA state, which consequently creates a new word when the periodic CA orbit is repeated (the new word in question is $\{1, 1\}$). Therefore, to enable us to employ the theory that fixes the density of words for multiples of the periodic CA orbit (as used for the discussion of Figure 1.6), we must revise the identification of the periodic orbit to one where the first CA state is different to the last. For this example, we could take a revised transient time $t^* = 5$, which yields the periodic orbit $\{0, 0, 1, 1\}$. Repeating this periodic orbit once will give the density $4/4 = 1$ of 2-words, the same as for the periodic orbit itself, as required.

Thus, care must be taken when identifying periodic behaviour of a cell. Nevertheless, such identification should be transparent when the CA evolution of all cells is considered as a vector (using Definition 1.1.2). Once such a careful identification of this periodic orbit has been established, we may apply the following lemma, in which we denote by $W(r)$ the word entropy of the periodic CA orbit (taken as the mean word entropy of all cells), i.e. the word entropy of the space-time region between the times t^* and $t^* + r$.

Lemma 1.1.4. Consider an eventually periodic CA pattern whose periodic CA orbit begins and ends in different CA states. Then, as $t \to \infty$, the word entropy $W(t)$ tends to $W(r)$, the word entropy of the periodic CA orbit.

In all cases, i.e. even allowing for the possibility of the first and last states in a periodic CA orbit being equal, the density of l-words in the transient region contribute less and less towards the word entropy of the overall CA space-time pattern as the number of iterations becomes large.

For example, consider the following periodic cellular evolution.

$$1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, \ldots \tag{1.2}$$

The periodic CA orbit is $\{1, 0, 1, 1\}$, in which the density of 1-words is $2/3$ and the density of 2-words is $1/3$. Now consider multiples of this periodic orbit. We introduce the name *periodic region* for a multiple of the periodic CA orbit and denote it by R; thus, (1.2) shows R made up of 5 multiples of the periodic CA orbit. If the periodic CA orbit is repeated once, then the density of l-words in R is $3/5$ if $l = 1$, $1/5$ if $l = 2$, $1/5$ if $l = 3$. In general, a periodic region containing m copies of the periodic CA orbit yields the density $\frac{m+1}{2m+1}$ if $l = 1$, $\frac{1}{2m+1}$ if $l = 2$, $\frac{m+1}{m-1}$ if $l = 3$. As $m \to \infty$, these densities tend to $1/2$, 0, $1/2$ respectively. Note that this is consequently not equal to the densities in (one copy of) the periodic CA orbit. Nevertheless, we can conclude that $W(t) \to W(R)$ as $t \to \infty$, where $W(R)$ denotes the word entropy in the periodic region. In other words, the word entropy of the transient region becomes negligible as the number of iterations becomes large, and the word entropy of the whole CA pattern tends to the word entropy of only the CA region above the transient time.

More complex CA (Wolfram Classes III and IV)

A finite lattice limits the maximal complexity of the resulting cellular automaton model. Thus, we conjecture that such a CA model is unlikely to produce behaviour in Wolfram Classes III and IV. Notwithstanding this, it is useful to understand the entropy measures in these classes in case our asynchronous update time CA (later) shows capability of producing such CA.

It is logical to look at Class IV before Class III since Class III is most complex, as represented by the Euclidean distance of its (W, S) coordinate from the origin in the WS plane; Class IV, while indicating complex behaviour, nevertheless produces periodic CA but after very long transients. This periodic region will be more complex than that of Class II.

Consider the Boolean state $s_j \in \{0, 1\}$. Then, for a complex CA pattern, we expect the density of state 1 and state 0 in the time series of a cell to be almost equal, i.e. $p(s_j) \approx p(1 - s_j)$ in the long time limit $t \to \infty$. Otherwise, one state is more frequent, which implies that $p(s_j) \gg p(1 - s_j)$ for $t \to \infty$. This implies that, as $t \to \infty$, the state s_j is more dominant than state $1 - s_j$. Therefore, $1 - s_j$ becomes negligible in the long time limit, which yields the Shannon entropy $S(t) \to 0$ as $t \to \infty$. This is the limiting behaviour of a homogeneous output. Thus, the states 0 and 1 must occur in almost the same frequency, which implies that $S(t) \to 1$ as

$t \to \infty$.

In both Classes of CA, the number n_w of different wordlengths created by the CA is generally much larger than the number of different wordlengths seen in a Class I or II CA. We assume that $n_w \gg 2$. Therefore, inputting this information into the formula for word entropy, we obtain $W \geq 1$.

The reasoning behind why Class III CA generates larger entropy values than Class IV is due to Class III being more complex. Indeed, it is said to be chaotic and aperiodic, which inevitably leads to even larger disorder, hence entropy.

We have now built up an idea of the regions in the WS plane, as shown in Figure 1.4. Note that, in the homogeneous case, $W \to W(t^*)$, whereas Marr and Hütt obtain $W \approx 0$ [31]. Their method classifies the entropies by omitting from consideration the transient region. Thus, for them, $W_t \approx 0$ since that would be the word entropy in the periodic region, which is dominated by one CA state. For our purposes however, we will retain the transient region. This is because we would like to see the extent to which the transient region affects the value $W(t^*)$.

1.1.4 Asynchronous cellular automata

Notice that classical CA models such as the above have assumed a synchronous mode of operation, where all cells are updated simultaneously. With the extension of the regular lattice to an arbitrary network of nodes, it becomes natural to look at a subtler form of updating, such as asynchrony. This is because such networks may possess irregular connectivity which has implications from a biological as well as a computational perspective. Indeed, the biological view is that, no matter how synchronously individual cells may seem to update, there is no accurate global synchronization in nature. This aspect is surveyed by Cornforth et al in [8] by providing examples such as the behaviour of a neural network and the propagation of bushfires. For example, in a simplified CA model of the spread of a fire through a fire bed, each cell (representing part of the bed), once ignited, would ignite any neighbouring cells on the next time step. However, fire spread is not so simple in reality because the rate of fuel ignition also depends on other factors such as the distance from the flames, fuel moisture etc. Thus, when a cell ignites, its neighbours ignite asynchronously, with the order determined by those external constraints.

Further, Lumer and Nicolis note that there is “a strong dependence of dynamical systems on the model of time being used” [30]. They argue that the synchronous computational methods employed to form models of biological applications exhibit artefacts of the digital model with no correspondence in the physical world. Their work was on coupled map lattices, and they found markedly different dynamics when the standard synchronous model was made asynchronous. The asynchronous methods involved fixed sequences defining the sites to be updated in each time interval.

Whilst the model was not a cellular automaton model, Ruxton and Saravia later used similar ‘step-driven’ methods to simulate CA in order to better model biological applications [35]. Their finding indicated the need for a better model overriding the need for computational efficiency, since, like Lumer and Nicolis argued, the latter may be the cause of effects that are inaccurately attributed to the intended application.

The importance of timing in CA models of social systems was demonstrated by Huberman and Glance in 1993 [23]. They extended a study by Nowak and May of the “Prisoner’s Dilemma”, a model of cooperation between individuals, which assumes a synchronous mode of such cooperation [34]. Huberman and Glance showed that the synchronous pattern, which indicates coexisting defecting and cooperating individuals, is lost when asynchrony is introduced. In particular, the asynchronous CA reached a steady state of only defecting individuals, which is a dramatic departure from the synchronous case.

Similarly contrasting behaviour was observed by Gunji when using CA to form models of pattern evolution on mollusc shells [18]. The study exhibited different patterns, each depending on different asynchronous update methods, leading to their conjecture that “asynchrony is intrinsic to living systems”.

A preliminary study of such asynchronous CA methods was conducted by Ingerson and Buvel in 1984 [24], prior to the aforementioned applications. The authors compared the properties of synchronous CA with two types of CA which iterate asynchronously. The one-dimensional lattice was used, and asynchrony was generated by two models: Model 1 is random iteration, where each cell is assigned a probability of iterating (i.e. updating) so that cells iterate one at a time; Model 2 assigns independent “clocks” to each cell. Here, each cell i iterates after a fixed period k_i of time steps. Because of their favourable dynamical aesthetics, the 32 legal ECA rules talked about earlier were observed under the two simulated asynchronous models as well as the traditional synchronous model for comparison.

It was seen that while some of the rules did not characteristically differ in all three cases, many rules were chaotic and fluctuating in behaviour under asynchrony whilst their synchronous counterparts fell under the generic Wolfram Classes I and II.

Subsequent authors later employed the methods of Ingerson and Buvel as special cases to conduct specific studies into asynchronous CA [3, 11, 10, 29, 36]. Schönfisch and de Roos in particular, carried out the first comprehensive quantitative study of two asynchronous methods which "seem most appropriate for simulating real (biological) processes" [36]. The first method is a step-driven method in which, for each step in the simulation, a cell is selected for updating according to some specified order (which could be random); Ingerson and Buvel's Model 1 fits into this category and corresponds to a "uniform choice" algorithm in [36]. The second group of methods are time-driven methods where the updating of each cell is governed by an exponentially distributed waiting time. Each cell has its own "updating clock" which specifies the time t for the cell to be updated. This method admits t as a real number so that Model 2 in [24] becomes a special case.

The idea in [36] and [29] was to compare the properties of synchronous methods and the various methods that fall under the two asynchronous groups above.

Bersini and Detours [3] and Fatès and Morvan [10] studied robustness of CA by interpreting the asynchrony as a fault or failure in a fraction of the cells. The found that resilience of the CA is dependent on not only the type of asynchrony but also on the CA update model (i.e. lattice and rule). Such robustness will not be a major concern for us in this thesis.

Many of the authors above attested to the best description of asynchrony as being stochastic in nature [24, 36, 3, 11, 10, 29]. This is a general viewpoint in light of their applications: such asynchrony relies on continuous time [36] and is also likely to be more robust [36, 3, 10], thereby aiding a better description of biological phenomena. For example, given a system of coupled cells, the update of cell states depends on a predefined probability (e.g. at time t, the probability that a cell updates in the exponential waiting time method is e^{-t}) [24, 36]. This consequently also led Schönfisch and de Roos to conjecture that, while synchronous updating can produce periodic orbits, asynchronous systems cannot; asynchrony will only yield patterns that converge to a fixed point (where no further change in pattern occurs thereafter) or patterns that are chaotic [36].

There is, however, more to the story of asynchronous CA. Indeed, Capcarrere [6] and Sipper et al [37] constructed asynchronous CA such that the CA behaviour is not studied exclusively. The motivation is an acknowledgement that time is part of the visual information conveyed in a CA. Thus, the authors attempt to restore all information (i.e. CA states) stored in a form of 'memory' in an attempt to solve what is called "the synchronization task". This concerns the development of an asynchronous cellular automaton that exhibits the same computational behaviour as synchronous CA.

Even though this task is achieved, Capcarrere finds that it is not without loss of information [6]. Firstly, the algorithm used to solve the task requires more than the two binary states. Secondly, their algorithm is not exact, but may be regarded as being ‘good enough’, in terms of the visual depiction of the CA. The work in this thesis however, particularly in Chapter 3, will solve the synchronization task.

The argument for asynchronous updating being stochastic (as is supported by the authors mentioned above) was later challenged by Cornforth et al in [15]. The authors claimed that such probabilistic updating schemes are used because of the oversimplification of biologically inspired models. They further argued for mimicking appropriate aspects of nature more closely to create better computational models. Thus, in a related publication, the same authors delved into their claims by drawing attention to a large class of behaviours of natural processes. This class was labelled “ordered asynchronous systems” (OAS) and it is comprised of systems in which the updating is asynchronous but not stochastic (the authors referred to stochasticity as “randomness”) [8]. OAS processes were presented as an alternative to RAS (random asynchronous) updating, which rely on the timing of updating a cell to be stochastic.

Thus, Cornforth et al introduced the “spotlight model”, in which different *regions* of a system update asynchronously to each other (as opposed to only one cell updating asynchronously) [8]. The model is a Boolean network whose nodes are grouped into modules (“blocks”), where each module is associated with a regulator node. A regulator node allows or prevents state updates of nodes in a module depending on its own state, e.g. if the regulator node is ON, then its associated module will update its state, otherwise not; thus, we see ‘spotlights’ in the network, corresponding to those modules that are updating or not updating. Figure 1.7 illustrates this idea for a network with a few nodes. This is ordered asynchrony because regulator nodes (which are themselves part of the network) turn ON/OFF depending on the states of their neighbourhoods, which enforces the asynchronous updating of corresponding modules of nodes. Note that there is no stochasticity involved; update times are determined solely by the states of regulator nodes, as a result of their own interaction with the network. The network itself determines its asynchrony.

Cornforth et al studied the space-time patterns of such a model as the number of spotlights r was varied. It was found that, when r is small, a variety of behaviours occur; this reduces as r increases up to a critical threshold, beyond which the model converges to a homogeneous state. Even though the results of evolving such a model exhibited similar behaviour to a synchronous Boolean network, the space-time diagrams displayed several types of behaviour occurring at the same time, which is believed to be due to the modularity. Thus, the spotlight model enables different processes to occur at different rates without interference. This is a demonstration of an asynchronous system that yields a variety of behaviour, and so may be classed as an interesting complex system, one which is asynchronous yet not stochastic.

The authors concluded by stating that OAS processes had not been identified in the past due to the domination of modularity together with the OAS nature being hidden within the model's workings. Whilst we agree with the latter comment, it is acknowledged that not all OAS behaviour is modular, so there is scope for other avenues of OAS research.

Returning to Boolean networks, in 1997 Harvey and Bossomaier first demonstrated the properties of asynchronous random Boolean networks (ARBNs) in response to the oversimplistic synchronous treatments of related models [21]. The authors used three alternative methods to simulate RBNs: the first was conventional synchronous updating of all nodes, the second and third were two methods of asynchronous updating which relied on randomness to choose the next node to update.

We now define an attractor more formally.

Definition 1.1.3. An *attractor* is a set of states of the system towards which neighboring states in a given basin of attraction asymptotically approach during system evolution.

For us, an attractor is always a periodic orbit, which is consequently also referred to as a cyclic attractor. A dynamical system may have multiple attractors, each with its own basin of attraction. For example, in Figure 1.3 there are four attractors, which are the four periodic orbits, each with its own basin of attraction. However, the two period-3 orbits may be regarded as being 'most attractive' as they have larger basins of attraction.

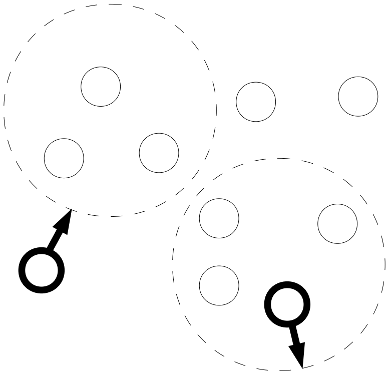


Figure 1.7: Network of nodes divided into two 'spotlights', indicated by the nodes contained within the dashed circles. Nodes outlined in heavier lines indicate regulator nodes.

Harvey and Bossomaier found that, unlike synchronous RBNs, ARBNs do not produce any cyclic attractors due to their indeterminism, but only "point" and "loose" attractors; recall that this would later be supported by the observations made in [36, 24, 23, 11] regarding the asymptotic behaviour of asynchronous CA. A point attractor is a fixed point in state space, i.e. a period-1 orbit. The term "loose attractor" is analogous to the cyclic attractors of synchronous systems. Informally, a loose attractor $\mathcal{A}$ contains those states of the network such that successive states (through asynchronous updates) are also contained in $\mathcal{A}$. It is not cyclic due to the indeterminism of ARBNs but it nevertheless conveniently classifies a region of the state space.

In 2002, Carlos Gershenson provided the first classification of different types of random Boolean networks such as the ARBN and also the GARBN [14]. GARBNs (Generalised ARBNs) were proposed as an extension to ARBNs, in which more than one node is selected to be randomly updated at each time step. CAs of the Wolfram type were proposed as special cases of Boolean networks and deterministic ARBNS (DARBNS) and deterministic GRBNS (DGARBNs) were outlined as new types of RBNs. DARBNs are ARBNs with the exception that they select the node to be updated deterministically, not stochastically. Gershenson talked of DARBNs as being more advantageous because of their modelling capabilities, which are more straightforward than ARBNs that rely on the stochasticity of asynchronous phenomena. This determinism leads to DARBNs having cyclic (as well as point) attractors. Moreover, he found that, although synchronous RBNs qualitatively differ from ARBNs due to their determinism (and the non-determinism of ARBNs), there is no significant deviation between the former and DARBNs; this is because both are deterministic. Gershenson further proposed DARBNs as better representations of Kauffman's genetic regulatory networks as they are asynchronous but do not rely on stochastic methods.

1.2 Modelling asynchrony

An advantage of asynchrony is that it can add an extra level of realism to a model. The type of asynchronous model used is dependent on the application that is intended for being modelled. For example, the ARBN could be appropriate for modelling a social network, where nodes represent people each of whom carry a state that corresponds to an opinion (e.g. "yes"/"no"). The state changes as a result of interactions between people and, since such interactions are randomly timed, the system is more suited to a asynchronous timing that is based on models relying on randomness. Whilst the actual system may model the application well, there is room for improvement, especially as the timings themselves might be better modelled. Following on from Gershenson's idea of using determinism as a more 'model-friendly' form for asynchrony, a goal of this thesis is to exploit this avenue by better modelling not only states (e.g. CA) but the asynchronous update times themselves.

We would like to refrain from using such stochastic models as were mentioned in the previous section. No matter how well a probability distribution matches the real system, we claim that the essence of many interesting and important asynchronous processes is lost. As alluded to by Cornforth et al in [15], traditional asynchrony has tended to generalise these dynamics into something which eventually assigns a probability (or the like) of the process updating its state. We want to get to the heart of the matter and study the timing of the actual events that combine to form such probabilities. This section looks at a hand-shaking system in computer science that provides some grounding for our work.

In computer science, a well-known problem has been the synchronisation of events in a *distributed system*. A distributed system is a finite set of distinct processes which are spatially separated, and which communicate with one another by exchanging messages. It can be described by a digraph in which vertices represent processes and arcs represent communication links. An example is a network of interconnected computers. Each processor executes *events* (or "actions") that cause a change of state. A distributed system is defined to be asynchronous if message transmission (hence arrival) times are not known. Events may occur in one of three forms: a processor changes state, receives a message from another processor or sends a message to another processor.

In such an asynchronous system (of a large number of processors say), it is sometimes impossible to distinguish between events happening before or after another in real time. This could be due to the noncommunication between processors, e.g. suppose the processors do not have access to a common real-time clock. A processor A in Aberdeen may execute an event e_A at 12.15 and a processor B in Birmingham may execute event e_B at 12.17, but if these events were not causally related, then it is impossible for either processor to say which event happened first. In 1978, in explaining this causality between events, Leslie Lamport defined the relation "happened before" as a *partial ordering* of events in the system [27], and claimed that it is for this reason that problems arise. Thus, Lamport presented an implementation of "logical clocks", which relies on observable events in the system rather than the usual physical interpretation of (real) time. The rationale behind this is that real clocks may not keep precise physical time and, even if they did, the clocks would have to be perfectly synchronised to a common 'global' clock. This synchronisation of all processors to one common clock, however, becomes an arduous task, especially for a large number of processors. Note that this also supports the argument that opposes considering CA under a synchronous mode of operation.

Lamport considered the events of a processor to form a sequence such that the events are totally ordered. However, as mentioned above, we obtain a partial ordering of events when considering the interaction between processors, and this is when complications begin to emerge. The "happened before" relation is denoted by "$\prec$".

Definition 1.2.1. The relation "$\prec$" on a set of events is defined by the following conditions:

1. If a and b are events in the same processor, and a occurs before b, then $a \prec b$.
2. If a is the sending of a message by processor A and b is the receipt of the message by another processor B, then $a \prec b$.
3. Let b_1 be the receipt of a message by processor B and b_2 be the sending of a message by the same processor such that $b_1 \prec b_2$. If $a \prec b_1$, and $b_2 \prec c$, then $a \prec c$.

We say that two distinct events a and b are "concurrent" if $a \not\prec b$ and $b \not\prec a$. We also assume the properties of irreflexivity, i.e. $a \not\prec a$, and antisymmetry on the times of events, i.e. if a_t and b_t represent the times of events a and b, then $a_t \not\prec b_t, b_t \not\prec a_t \Rightarrow a_t = b_t$. The events in a distributed system are thus defined as being partially ordered. The full picture is best viewed as a space-time diagram of events such as in Figure 1.8. Lamport further introduced logical clocks as a substitute for real-time clocks and this serves as a motivation for our study.

Definition 1.2.2. Suppose E is the set of all events in a distributed system. A (logical) clock C is a function $C : E \to \mathbb{R}$ with the following property.

Clock Condition. For any events a, b: if $a \prec b$ then $C(a) < C(b)$.

A clock is simply a way of assigning a number to an event. We might think of each processor P_i having its own clock C_i which looks only at the occurrence of events - unlike physical clocks, a logical clock need not be continuously running. We commonly take the codomain of C_i to be the set of natural numbers $\mathbb{N}$ as this fits well the counting mechanism that a clock employs.

Definition 1.2.1 along with an intuitive interpretation of Figure 1.8 combine to form the following understanding of logical clocks.

C1 If a and b are events in processor P_i and a occurs before b, then $C_i(a) < C_i(b)$.

C2 If a is the sending of a message by processor P_i and b is the receipt of the same message by P_j, then $C_i(a) < C_j(b)$.

As a conclusion, Lamport presented methods whereby processors would physically implement such clocks, e.g. after the sender assigns a "timestamp" T to a message, the receiver would increment its clock to later than T, thus satisfying C2.

Lamport later used this work as a grounding to develop algorithms that would determine the global state of the network [7], i.e. where each processor has an idea of the states of every other processor to a good enough level of accuracy. In 1987, Neiger and Toueg extended Lamport's algorithms so that the logical clocks would work even when processors did indeed have access to a global real-time clock [33]. Mattern subsequently argued against the linearly ordered structure of time that formed the basis of the algorithms above to present refined algorithms of both time and the global state of processors [32].

Models of these distributed systems deal primarily with the asynchrony associated with the timings of events (in relation to one another) instead of the asymptotic effect of state changes (due to these events). Whilst the latter effect is classically studied in models of asynchrony, as in the asynchronous CA, the former aspect is the central theme for us. Inspired by Lamport's asynchronous distributed system, we shall focus on deterministic systems to potentially provide the solution to how some applications may asynchronously generate their cellular update times.

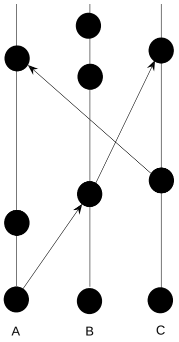


Figure 1.8: Space-time diagram of events in a distributed system of three processors, A, B and C. Time travels vertically up and the horizontal direction represents space (of processors here). Events are represented by dots and messages are represented by arrows connecting the sender to the receiver.

1.2.1 Thesis model of asynchrony

This thesis concerns asynchronous update time as modelled by *maxmin-plus algebra*. Indeed, it is a deterministic system and we notice that the inter-cellular communication of states (CA, Boolean or the like) lends itself nicely to being modelled by such a system. Importantly, the determinism of this asynchrony is generated by the process (cells or nodes) themselves through this exchange of states; this is the most appealing aspect of a maxmin-plus system, which is consequently ideally suited to being a potential model for how a cell in an application might generate its asynchronous update time with no central controller (see the fundamental features of a complex system).

In maxmin-plus algebra, the main operators are maximisation (represented as $\oplus$), minimisation ($\ominus$) and conventional addition. A nice review of the subject is provided by Heidergott et al in [22] in which, along with the mathematical analysis itself, they model the timetabling of trains at railways stations. At many stations, a train is required to wait for the arrival of other trains before departing. This requires the study of the time of arrival of the other trains. Indeed, suppose a train must wait for two trains to arrive before it departs. Then the time of departure is constrained to be no earlier than the latest train to arrive, i.e. the maximum of the two arrival times, which consequently requires the $\oplus$ operator.

In terms of an asynchronous CA model, suppose a cell i (represented by a node in a network) has n neighbours, denoted $i_1, i_2, \ldots, i_n$. Like conventional CA models, the CA state of i on time k is a function of the CA states of its neighbourhood $\mathcal{N}(s_i(k-1))$ on the previous time step $k-1$. The neighbourhood consists of all neighbours, i.e. for the CA function $f : \{0,1\}^n \to \{0,1\}$, the new state $s_i(k)$ is calculated as

$$s_i(k) = f(\mathcal{N}(s_i(k-1)). \tag{1.3}$$

Now suppose that there is a time associated with each input to i, i.e. it takes a finite time for the transmission of the state of each of its neighbours. We call each of these times the *transmission time*, and they can be depicted by labelling the corresponding arcs on the digraph of nodes. Then, i must wait for *all* inputs to arrive before executing (1.3) to process its new state. Thus, the time of this wait is modelled using the $\oplus$ operator as follows. Let $t_{i_1}(k), t_{i_2}(k), \ldots, t_{i_n}(k)$ denote the kth time of arrival of each input. Then all input states will have arrived as soon as the last input has arrived, i.e. the time that all inputs will arrive is the maximum of all input times, which is denoted $\bigoplus_{j=1}^{j=n} t_{i_j}(k-1)$.

Similarly, if node i waits for only the first input before updating its state, then this input time will be modelled by $\ominus_{j=1}^{j=n} t_{i_j(k-1)}$.

The node could wait for neither the first nor the last input, but an intermediate number of inputs, say m $(1 < m < n)$, before updating. This corresponds to operating on the input times with a mix of $\oplus$ and $\ominus$.

For all three cases, a *processing time* ξ_i may be added, indicating the time it takes for i to process the inputs before updating its state $s_i(t)$. Thus, the time of update of each node in such a system is modelled by a maxmin-plus system, which is asynchronous because of the parameters ξ_i and τ_{ij}, the processing time of node i and transmission time from j to i, respectively. Such a system will now deterministically increment the kth update times concurrently with the CA system.

1.3 Criteria for complex behaviour

Another goal of this thesis is to identify complexity in our work. Since the subject is vast, any particular candidate complex system is best analysed independently, in terms of its own parameters and variables, in order to deduce complexity. The history of the subject pinpoints a few fundamental criteria for a system such as CA to display complex behaviour. For our model of CA in maxmin-plus time, we shall look for the following set of arising outcomes:

1. Periodic behaviour with a large period (in terms of both the update times and CA states)

2. Large transient times (in terms of both the update times and CA states)

3. Cellular automaton behaviour with large Shannon and word entropies

Each item is stated in light of the limitations of complexity that a finite lattice is expected to bring. We do not expect to observe Wolfram Classes III or IV. Thus, in terms of the WS plane, the items refer to large periods, transients and entropies for Class II CA.

1.4 Outline of thesis

In this thesis, we fix the neighbourhood size of each node in a CA to n. We study the asynchronous update time model that requires m inputs to arrive at each node before updating the state of each node (where $1 \leq m \leq n$). This is referred to as the *maxmin-m* system.

In Chapter 2, we introduce max-plus algebra, which corresponds to the case $m = n$. It is the primary tool that will be used as a basis to study the maxmin-m system for all other values of m. In particular, the subtleties that surrounds this tool as a result of modelling the asynchronous CA system is explored.

The first demonstration of the effect of a max-plus model on an information exchange system such as CA is carried out in Chapter 3. We display space-time plots that indicate the asynchronous update times as well as the cellular automaton in max-plus time. The results are mainly computational, and we use some intuition backed up by mathematics to formulate the theory that provides the bridge between maxmin-plus algebra and cellular automata.

Chapter 4 details maxmin-plus algebra, in particular an approach that uses max-plus algebra to model the maxmin-m system for $m < n$. This also yields significant insight into the algebra itself, including asymptotic behaviour, which impacts on the corresponding CA model.

Chapter 5 provides a formalism for implementing a CA model on the maxmin-m system. Entropy results for CA on a large network are obtained and analysed for evidence of complex behaviour, with reference to the criteria for complexity in the previous section. A link between maxmin-m update time complexity and CA complexity is thus established.

Chapter 6 concludes the thesis and provides possible avenues of extension. A particular extension is to a maxmin-plus update time model on an arbitrary network where the neighbourhood size of each node is not fixed.

Chapter 2

Max-plus Algebra

This thesis concerns the modelling, analysis and timing of causally related asynchronous events. Such causal relations impose constraints, which in turn affect the way that information is exchanged throughout the network of connected processes (nodes). We examine the effect of such constraints in the next chapters, in which we particularly consider the case whereby a process must receive the first m inputs from a possible n inputs before updating its state. Before that, this chapter introduces the highly useful algebra that will form the basis of the modelling of asynchronous time. Over the course of our work, we shall see that the simplest form of this algebra for the case above arises when $m = 1$ or $m = n$. We will confine the chapter to a survey of preparatory topics which give a suitable grounding for expansion in the next chapters.

Consider a process which receives input from its neighbours. Having received this input, the process computes its new state (which is a function of the input states), then sends a corresponding output to its connected neighbours. This type of information exchange can be represented as the diagram in Figure 2.1, where each circle represents a process and directed arcs between circles indicate the direction of information transfer, i.e. the sender and receiver. The network depicted in this way is known as a *digraph*, where the circles are known as vertices or nodes. We shall explain a digraph in detail later.

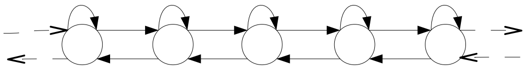
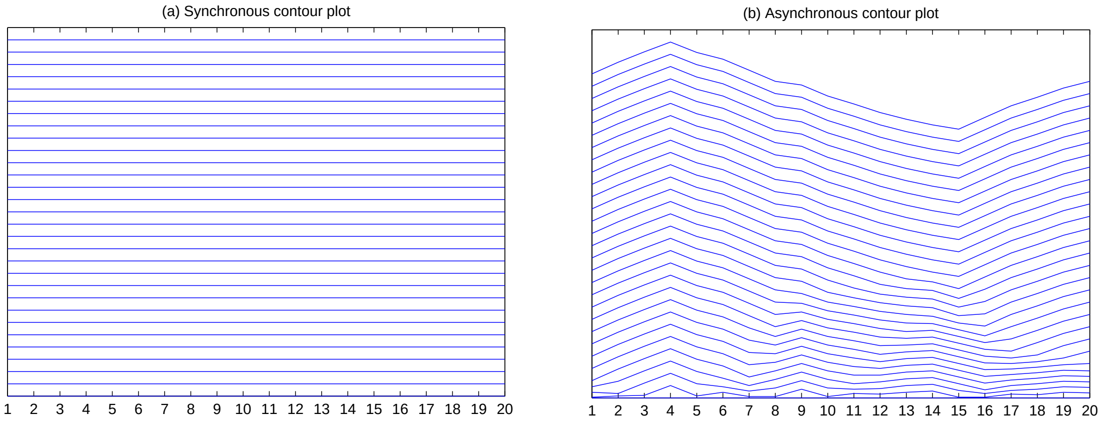


Figure 2.1: Digraph showing information exchange in a regular 3-nbhd network

Figure 2.1 shows three arcs pointing to each vertex, indicating that there are three processes sending information to each process i (which includes i itself). Thus, by letting n denote the neighbourhood size of each node, we see that $n = 3$; we also refer to such a neighbourhood as an *n-neighbourhood (n-nbhd)*. The figure also shows that each process sends output to itself as well as to its left and right neighbours. We refer to this type of neighbourhood as a *symmetric* neighbourhood. If every process has a symmetric neighbourhood, then the resulting network of connected processes will be called a *regular n-nbhd network* or simply a regular network if n is understood. Further, unless explicitly indicated, we shall employ boundary conditions that address the finiteness of the regular n-nbhd network by connecting the first and final nodes so as to form a ring. Such regularity should then ensure that the only sense of the terms "first node", "last node" etc is that of labelling. The information that we think about is specialised to that of the Boolean states (1 and 0) of a classical *cellular automaton* (CA) such that each process may be thought of as a cell. For this reason, we will use the terms *cell*, *vertex*, *node* and *process* to all mean the same thing.

In our work, it is also assumed that there is a *processing time* associated with each nodal computation of a new state. This is the time taken for a new state to be calculated and represents the first divergence from classical CA models since there the computations are assumed to occur instantaneously. The second difference is the incorporation of a *transmission time*, which is the time taken for the computed state to be sent to a neighbourhood node. Having received relevant input then updated its state, the time that this occurs is the *update time*. It should be evident now how a synchronous CA can be 'asynchronised' by the variation of processing and transmission times.

Traditionally, CAs have been studied on a one-dimensional lattice where cells are connected as in the regular n-nbhd network. Such a correspondence between a regular network and the one-dimensional CA lattice provides the primary motivation for basing this section around the regular network; comparisons can then be drawn. As mentioned in Chapter 1, ECA particularly have been studied in synchronous form. This means that each cell updates its state at the same time as all other cells. We can, thus, draw horizontal lines in a space-time plot of the CA evolution, each of which represent the update times of all cells. The CA is a discrete time dynamical system, so each such horizontal line may be drawn in sequence, evenly spaced, as in Figure 2.2(a). We call such a space-time plot a *contour plot*, where each horizontal line is referred to as a *contour*. The contour plot may be thought of as a frame on which the CA states are overlaid and simulated. Now consider altering these contours so that cells do not necessarily update synchronously. The corresponding contour plot may then look like Figure 2.2(b), which shows the contours having variable shapes. The main feature of this chapter is to develop and understand a mathematical framework through which such an asynchronous contour plot is obtained. Unlike many asynchronous update time systems in the literature, our asynchrony will be shown to be obtained deterministically, with corresponding limiting periodic behaviour, as shown in the periodic contours in Figure 2.2(b). In presenting the general mathematical results, we shall borrow most notation and terminology from [22].


(a) Synchronous contour plot

(b) Asynchronous contour plot

Figure 2.2: (a) Synchronous and (b) asynchronous contour plots. The contours indicate update times of cells and act as a frame on which the CA may be evolved. Both lattices are connected as a regular 3-nbhd network on 20 cells. Time is on the vertical axis and the horizontal axis represents the cell positions. In (a), contours are horizontal. In (b), update k of all cells is represented by contour k (counting from the bottom).

2.1 Max-plus algebra

Consider a cell i contained in a regular n-nbhd network of N cells. The cell carries a CA state (1 or 0), which is dynamic, i.e. it changes with time depending on the rules that we employ. Thus, we can plot points on the real line corresponding to when these changes occur. The real line represents time and the points are the update times of the CA state. Let $x_i(k)$ denote the kth update time for cell i. In this chapter, we shall also call $x_i(k)$ the *state* of i at k; there should be no confusion with the CA state of i because cellular automata will not be modelled yet and we shall explicitly distinguish between the two when the need arises. Referring to $x_i(k)$ as a state is also in keeping with traditional dynamical systems convention. We also refer to k as the *cycle number* (or *epoch*)[1]. Cell i updates its state for the $(k+1)^{\text{th}}$ time by first waiting for every cell in its neighbourhood to finish its k^{th} cycle. These neighbourhood cells

[1] We shall refrain from calling k the "time" of the node since time is in fact our state now!

then send their updated CA states to i, taking transmission times $\tau_{ij}(k)$. The update of cell i takes a processing time and it is represented in the k^{th} cycle by $\xi_i(k)$. If $n = 3$, we have the following scheme for the update time of cell i.

$$x_i(k+1) = \max\{x_{i-1}(k)+\tau_{i,i-1}(k), x_i(k)+\tau_{i,i}(k), x_{i+1}(k)+\tau_{i,i+1}(k)\}+\xi_i(k+1) \quad (2.1)$$

We can see this as a map $f_i : \mathbb{R}^3 \to \mathbb{R}$, and cycle k can be thought of as iteration k of f_i. The above sequence of interactions yielding a state change is depicted in Figure 2.3. Notice that this, together with the above description, expands on the simplified notion of inter-cellular communication by highlighting the communication within a cell itself. This is the key to our study of asynchrony.

We shall refer to these internal processes as *events*. In Figure 2.3, there are two significant types of events that we focus on: "receive" and "send". The three times $x_{i-1}(k)$, $x_i(k)$ and $x_{i+1}(k)$ are "send" event times, i.e. when the corresponding CA states are sent. The time $\max\{x_{i-1}(k) + \tau_{i,i-1}(k), x_i(k) + \tau_{i,i}(k), x_{i+1}(k) + \tau_{i,i+1}(k)\}$ is when node i receives the aforementioned "send" states; it is therefore a "receive" event. Once received, node i processes its new CA state (by applying a CA rule on the received states); this takes time duration $\xi_i(k+1)$. Once processed, node i sends its state to connected nodes at time $x_i(k+1)$; this is another "send" event. We shall, however, formally refer to $x_i(k+1)$ as the update time of i, as introduced earlier, since this is our main focus.

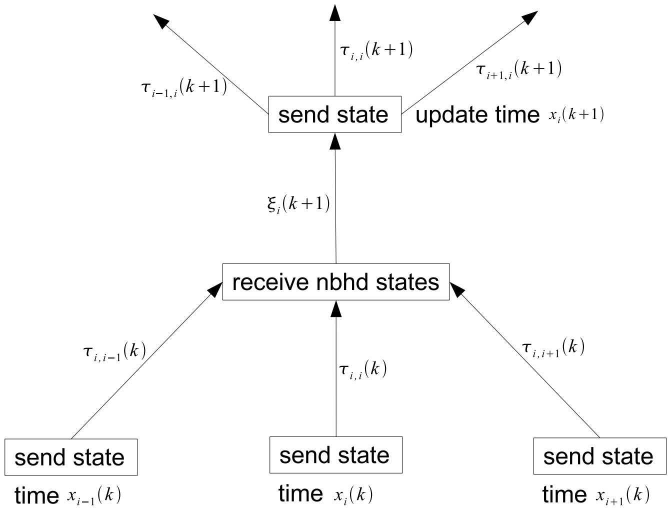


Figure 2.3: The processes internal to the k^{th} state change at cell i. Real time travels vertically upwards. Arrows indicate the destination of the sent state

The events in Figure 2.3 are shown in boxes. Notice that these events form an acyclic graph, i.e. all information travels in one direction (vertically up) and there is no feedback. This is in contrast to the network in Figure 2.1 which does show circuits. Over time, the processes are repeated; we shall thus show that the structure of Figure 2.3, taken together with the same structure in other cells forms such a cyclic graph, thereby lending itself to convenient analysis using results from graph theory. As in the figures above, we shall mainly take a regular 3-nbhd network to formulate examples in this section. If the context is understood, we may also refer to events by their corresponding times.

The max operation leads us to interpret our model in *max-plus algebra*. The effect of this is that a nonlinear system, as seen in ordinary arithmetic, is converted into a linear system in this new algebra, and this linearised system is easier to work with. This is then a useful tool for analysing and optimising phenomena in which the order of events is crucial. Thus, it is well-placed to serve us for this study. In max-plus algebra, we replace max with $\oplus$ and $+$ with $\otimes$.

2.1.1 Definitions

The terms and definitions given in this section will be augmented with their impact on the model.

Elements in max-plus algebra

Define $\varepsilon = -\infty$ and $e = 0$, and denote by $\mathbb{R}_{\max}$ the set $\mathbb{R} \bigcup \{\varepsilon\}$. For elements $a, b \in \mathbb{R}_{\max}$, define operations $\oplus$ and $\otimes$ by

$$a \oplus b = \max(a, b) \qquad \text{and} \qquad a \otimes b = a + b.$$

The set $\mathbb{R}_{\max}$ together with the operations $\oplus$ and $\otimes$ is what we refer to as *max-plus algebra* and is denoted by

$$\mathcal{R}_{\max} = (\mathbb{R}_{\max}, \oplus, \otimes, \varepsilon, e)$$

where $\varepsilon = -\infty$ is the 'zero', i.e. $\forall x \in \mathbb{R}_{\max}$, $\varepsilon \otimes x = x \otimes \varepsilon = \varepsilon$ and $\varepsilon \oplus x = x \oplus \varepsilon = x$. $e = 0$ is the max-plus multiplicative 'unit' element, i.e. $\forall x \in \mathbb{R}_{\max}$, $e \otimes x = x \otimes e = x$.

Max-plus algebra is associative and commutative over both operations $\oplus$ and $\otimes$ while $\otimes$ is distributive over $\oplus$ as in conventional algebra: for all $x, y, z \in \mathbb{R}_{\max}$,

Associativity: $x \oplus (y \oplus z) = (x \oplus y) \oplus z \quad \text{and} \quad x \otimes (y \otimes z) = (x \otimes y) \otimes z$
Commutativity: $x \oplus y = y \oplus x \quad \text{and} \quad x \otimes y = y \otimes x$

Distributivity: $x \otimes (y \oplus z) = (x \otimes y) \oplus (x \otimes z)$

In addition, $\oplus$ is idempotent in $\mathcal{R}_{\max}$:

$$\forall x \in \mathbb{R}_{\max} \quad x \oplus x = x$$

Thus, max-plus algebra is a commutative and idempotent semiring. It is not a ring because there is no inverse element with respect to the $\oplus$ operation, i.e. the following statement is proved false:

$$\forall a \in \mathbb{R}_{\max}, \exists a' \in \mathbb{R}_{\max} \text{ such that } a \oplus a' = a' \oplus a = \varepsilon \tag{2.2}$$

Note that there can only be one element for which (2.2) is true; that is ε, whose inverse is ε itself.

Now writing Equation (2.1) in max-plus algebra gives

$$x_i(k+1) = \{(\tau_{i,i-1}(k) \otimes x_{i-1}(k)) \oplus (\tau_{i,i}(k) \otimes x_i(k)) \oplus (\tau_{i,i+1}(k) \otimes x_{i+1}(k))\} \otimes \xi_i(k+1) \tag{2.3}$$

We will often save space and clarify the presentation by omitting $\otimes$, much as in conventional algebra. Thus, $x \otimes y \equiv xy$ and Equation (2.3) can be written as

$$x_i(k+1) = \xi_i(k+1)\{\tau_{i,i-1}(k)x_{i-1}(k) \oplus \tau_{i,i}(k)x_i(k) \oplus \tau_{i,i+1}(k)x_{i+1}(k)\}$$

Since $\otimes$ is distributive over $\oplus$, we can write this as

$$x_i(k+1) = \xi_i(k+1)\tau_{i,i-1}(k)x_{i-1}(k) \oplus \xi_i(k+1)\tau_{i,i}(k)x_i(k) \oplus \xi_i(k+1)\tau_{i,i+1}(k)x_{i+1}(k) \tag{2.4}$$

To represent a max-plus power, we follow from associativity of $\otimes$ and define, for $x \in \mathbb{R}_{\max}$,

$$x^{\otimes n} \overset{\text{def}}{=} \underbrace{x \otimes x \otimes \cdots \otimes x}_{n \text{ times}} \tag{2.5}$$

for all $n \in \mathbb{N}$ with $n \neq 0$. For $n = 0$, we define $x^{\otimes 0} = e\ (= 0)$. This definition of a max-plus power can, in fact, be generalised to allow n to be any real number by thinking about the $\otimes$ operator in its ordinary arithmetic + form. Thus, $x^{\otimes n} = n \times x$ for all $n \in \mathbb{R}$. This allows negative powers, irrational powers etc.

Matrices and vectors in max-plus algebra

Max-plus algebra also extends naturally to matrices, and this allows the concurrent modelling of the update times for all nodes. Denote the set of $n \times m$ matrices with underlying max-plus algebra by $\mathbb{R}_{\max}^{n\times m}$. The element of a matrix $A \in \mathbb{R}_{\max}^{n\times m}$ in row i and column j is denoted by $[A]_{ij}$ or more agreeably as A_{ij}. The i^{th} row (resp. j^{th} column) vector of A is written as $[A]_{i\cdot}$ (resp. $[A]_{\cdot j}$).

The sum of matrices $A, B \in \mathbb{R}_{\max}^{n\times m}$, denoted by $A \oplus B$, is defined by

$$\begin{aligned} [A \oplus B]_{ij} &= a_{ij} \oplus b_{ij} \\ &= \max(a_{ij}, b_{ij}). \end{aligned}$$

The link to real matrix addition can easily be seen by changing $\oplus$ to $+$. In the same vein, for matrices $A \in \mathbb{R}_{\max}^{n\times l}$ and $B \in \mathbb{R}_{\max}^{l\times m}$, the matrix product $A \otimes B$ is defined by

$$\begin{aligned} [A \otimes B]_{ij} &= \bigoplus_{k=1}^{l} a_{ik} \otimes b_{kj} \\ &= \max_{k\in\{1,\ldots,l\}} \{a_{ik} + b_{kj}\} \end{aligned}$$

For $\alpha \in \mathbb{R}$, the scalar multiple $\alpha \otimes A$ is defined by

$$[\alpha \otimes A]_{ij} = \alpha \otimes a_{ij}.$$

The $n \times m$ matrix with all elements equal to ε is denoted by $\mathcal{E}(n,m)$ and we denote by $E(n,m)$ the $n \times m$ matrix defined by

$$[E(n,m)]_{ij} \overset{\text{def}}{=} \begin{cases} e & \text{for } i = j \\ \varepsilon & \text{otherwise.} \end{cases}$$

If $n = m$, then $E(n,n)$ is the $n \times n$ identity matrix, which may also be referred to as the max-plus matrix 'unit' element, i.e. for any matrix $A \in \mathbb{R}_{\max}^{n\times m}$,

$$A \otimes E(m,m) = A = E(n,n) \otimes A.$$

When their dimensions are clear from the context, $\mathcal{E}(n,m)$ and $E(n,m)$ will also be written as $\mathcal{E}$ and E, respectively.

$\mathcal{E}(n,m)$ is the 'zero' element in max-plus matrix operations, i.e. for $A \in \mathbb{R}_{\max}^{n\times m}$,

$$A \oplus \mathcal{E}(n,m) = A = \mathcal{E}(n,m) \oplus A$$

while, for $k \geq 1$

$$A \otimes \mathcal{E}(m,k) = \mathcal{E}(n,k) \quad \text{and} \quad \mathcal{E}(k,n) \otimes A = \mathcal{E}(k,m).$$

As in classical matrix manipulation, the max-plus matrix addition $\oplus$ is associative and commutative, while the matrix product $\otimes$ is associative and distributive with respect to $\oplus$; it is usually not commutative. Similarly, the operation $\otimes$ has priority over $\oplus$.

The transpose of a matrix $A \in \mathbb{R}_{\max}^{n\times m}$ is defined in the usual way and is denoted by $A^\top$, where $[A^\top]_{ij} = [A]_{ji} = A_{ji}$.

The elements of $\mathbb{R}_{\max}^{n} \overset{\text{def}}{=} \mathbb{R}_{\max}^{n\times 1}$ will be called *vectors*. A vector will usually be written $\mathbf{x}$, thereby distinguishing it from an element x that may be contained in the vector. The vector with all elements equal to e is called the *unit vector* and is denoted by $\mathbf{u}$. We also define the classical addition of vectors $\mathbf{x}$ and $\mathbf{y}$ as a max-plus multiplication:

$$\mathbf{x} + \mathbf{y} \overset{\text{def}}{=} \mathbf{x} \otimes \mathbf{y}$$

where the operation is performed componentwise. We can consequently define a vector power in max-plus algebra as

$$\mathbf{x}^{\otimes k} \overset{\text{def}}{=} \underbrace{\mathbf{x} \otimes \mathbf{x} \cdots \otimes \mathbf{x}}_{k \text{ times}}$$

for all $k \in \mathbb{N}$ with $k \neq 0$. For $k = 0$, we define $\mathbf{x}^0 = \mathbf{u}$.

We can now define matrix-vector products, i.e. the product $A \otimes \mathbf{x}$, where $A \in \mathbb{R}_{\max}^{n\times m}$ and $\mathbf{x} \in \mathbb{R}_{\max}^{n}$ as

$$\begin{aligned} [A \otimes \mathbf{x}]_i &= \bigoplus_{k=1}^{m} a_{ik} \otimes x_k \\ &= \max_{k\in\{1,\ldots,m\}} \{a_{ik} + x_k\}. \end{aligned}$$

Moreover, for the square matrix $A \in \mathbb{R}_{\max}^{n\times n}$, denote the kth power of A by $A^{\otimes k}$ defined by

$$A^{\otimes k} \overset{\text{def}}{=} A \otimes A \otimes \cdots \otimes A \tag{2.6}$$

for all $k \in \mathbb{N}$ with $k \neq 0$. Matrix A appears k times on the right-hand side of (2.6). For $k = 0$, we set $A^{\otimes 0} \overset{\text{def}}{=} E(n,n)$. Indeed, this is a straightforward extension of (2.5) to matrices.

Having established the preliminaries above, a system of N such equations as (2.4) can now be given in the form

$$\mathbf{x}(k+1) = P(k+1) \otimes \mathbf{x}(k) \tag{2.7}$$

where $\mathbf{x}(k) = (x_1(k), x_2(k), \ldots, x_N(k))^\top$. $P(k+1)$ is the $N \times N$ matrix obtained by the max-plus multiplication of the matrices $A_\xi(k+1)$ and $T(k)$, where

$$A_\xi(k+1) = \begin{pmatrix} \xi_1(k+1) & \varepsilon & \cdots & \varepsilon \\ \varepsilon & \xi_2(k+1) & \cdots & \varepsilon \\ \vdots & & \ddots & \vdots \\ \varepsilon & \varepsilon & \cdots & \xi_N(k+1) \end{pmatrix}$$

and

$$T(k) = \begin{pmatrix} \tau_{11}(k) & \tau_{12}(k) & \varepsilon & \varepsilon & \cdots & \varepsilon & \tau_{1N}(k) \\ \tau_{21}(k) & \tau_{22}(k) & \tau_{23}(k) & \varepsilon & \cdots & \varepsilon & \varepsilon \\ \vdots & & & \ddots & & & \vdots \\ \varepsilon & \varepsilon & \cdots & \varepsilon & \tau_{N-1,N-2}(k) & \tau_{N-1,N-1}(k) & \tau_{N-1,N}(k) \\ \tau_{N,1}(k) & \varepsilon & \cdots & \varepsilon & \varepsilon & \tau_{N,N-1}(k) & \tau_{NN}(k) \end{pmatrix}.$$

$A_\xi(k)$ is referred to as the *processing matrix* on epoch k and $T(k)$ is the *transmission matrix* on epoch k. We call (2.7) a *max-plus system* (of dimension N) where the vector $\mathbf{x}(k)$ is the *state of the system.* $P(k)$ will be referred to as the *timing dependency matrix* of the network of cells on epoch k. In the following subsections, we outline the depth of mathematics that is required, and given, by this system.

2.1.2 Graph theory

To a network of cells, we associate a *digraph*, defined as $\mathcal{G} = (V, E)$, which consists of a set V and a set E of ordered pairs (a, b) of V. The elements of V are called *vertices* or *nodes* and those of E are *arcs*. Thus, the cells of Figure 2.1 can be regarded as the nodes of a digraph, where the arrows indicate the arcs.

An arc (a, b) may also be represented as ab and we refer to an arc aa as a *self-loop.* For the arc ab, a is the *start node* and b is the *end node*; a is also referred to as the node *upstream* of b or as a *predecessor* of b. Likewise, b is *downstream* of a and also a *successor* of a. Arc ab is said to be *incident* on b; the number n_i of such incident arcs on b is the *in-degree* of node b. Often, we will refer to the digraph simply as a graph, and we will assume each arc to be uniquely identified by its upstream and downstream nodes, i.e. there will only be at most two arcs connecting two nodes a and b, namely the arcs ab and ba. The number n_o of arcs that share one start node a is termed the *out-degree* of a.

Now denote the *adjacency matrix* of a graph by B, where each element is defined by

$$B_{ij} = \begin{cases} 1 & \text{if } j \text{ is a predecessor of } i \\ 0 & \text{otherwise} \end{cases}$$

By assigning real numbers to the arcs of a graph, we obtain a *weighted graph*, where the number w_{ab}, associated with an arc ab, is the *weight* of ab. A *weighted adjacency matrix* is the matrix W each of whose entries are defined by

$$W_{ij} = \begin{cases} w_{ji} & \text{if } j \text{ is a predecessor of } i \\ 0 & \text{otherwise} \end{cases}$$

Notice the difference in subscripts. We want the ith row of W to indicate node i as being a successor node. This is in anticipation of the type of matrix multiplication that we shall use, which will be multiplication from the left with a vector, each of whose elements indicates a node. We would like predecessor nodes to have an effect on node i, and such a multiplication allows the modelling of this. We can now define a *weighted adjacency matrix over* $\mathbb{R}_{\max}$. It is the matrix $W \in \mathbb{R}_{\max}^{n\times n}$ whose entries are

$$W_{ij} = \begin{cases} w_{ji} \in \mathbb{R} & \text{if } j \text{ is a predecessor of } i \\ \varepsilon & \text{otherwise} \end{cases}$$

When the context is clear, we may also refer to W as a *max-plus adjacency matrix*. Observe that $P(k+1)$ in (2.7) can be viewed as a max-plus adjacency matrix of the regular 3-nbhd network since the network[2] is exactly that shown in Figure 2.1 augmented with arcweights $\xi_i\tau_{ij}$. Indeed, to any graph $\mathcal{G}$, we can associate an adjacency matrix, denoted $A(\mathcal{G})$. Likewise, to any $n \times n$ matrix A over $\mathbb{R}_{\max}$, we can associate a graph, called the *communication graph* of A. This is denoted by $\mathcal{G}(A)$. We define the *timing dependency graph* of a network of cells on epoch k by $\mathcal{G}(P(k))$.

Note 2.1.1. A must be square; it is not conventional to form a communication graph of a non-square matrix in max-plus algebra. For example, consider the following

[2] From now on, the terms “graph” and “network” will be used interchangeably, meaning the same

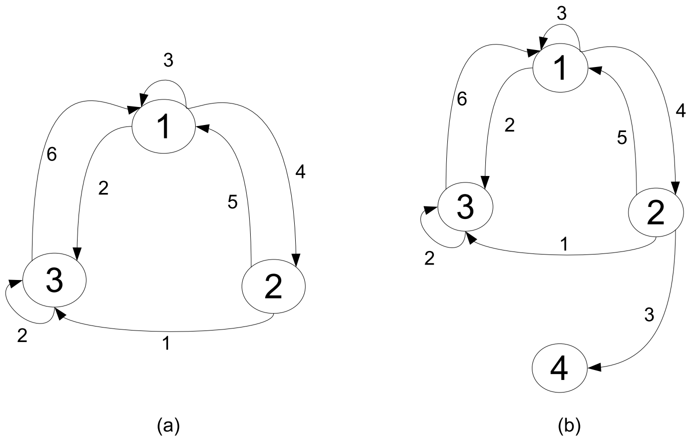
square matrix of size 3, whose communication graph is given in Figure 2.4(a).

$$A_1 = \begin{pmatrix} 3 & 5 & 6 \\ 4 & \varepsilon & \varepsilon \\ 2 & 1 & 2 \end{pmatrix}.$$

It can be seen that element A_{ij} is the weight of the arc from node j to i. Now consider adding a node to $\mathcal{G}(A_1)$, as in Figure 2.4(b), where the new node is node 4. Call this new graph $\mathcal{G}_2$. Then, $A(\mathcal{G}_2)$ is as follows.

$$A(\mathcal{G}_2) = \begin{pmatrix} 3 & 5 & 6 & \varepsilon \\ 4 & \varepsilon & \varepsilon & \varepsilon \\ 2 & 1 & 2 & \varepsilon \\ \varepsilon & 3 & \varepsilon & \varepsilon \end{pmatrix}.$$

Notice that the fourth column is redundant, as a result of node 4 not having any successor nodes. We can remove this column to form a 4×3 matrix, which conveys the same information. However, this prevents the generation of powers of such a matrix and, as we will see later, powers of the adjacency matrix are vital for understanding long-term behaviour of the underlying system.



(a)

(b)

Figure 2.4: Example of a digraph on 3 nodes. (b) displays the same digraph as (a) but with the addition of node 4.

Definition 2.1.1. Let B be the adjacency matrix of a digraph of connected cells. The *neighbourhood* of i is $\mathcal{N}_i = \{j | B_{ij} = 1\}$. It consists of all predecessors of i.

In terms of the n-nbhd, n also signifies the in-degree of node i. A larger n-nbhd ($n > 3$) will thus correspond to a larger in-degree and each row i in $P(k+1)$ will contain more non-zero (i.e. $\neq \varepsilon$) values $\xi_i^{(k+1)}\tau_{ij}^{(k)}$ in columns corresponding to the additional nodes in $\mathcal{N}_i$. The network itself will, as a result, be more connected. What do we mean by such connectivity? This can be deduced from the formal definitions that follow.

Definition 2.1.2. Let $p = \{a_1, a_2, \dots, a_n\}$ be a sequence of arcs. If there are vertices $v_0, v_1, \dots, v_n$ (not necessarily distinct) such that $a_j = v_{j-1}v_j$ for $j = 1, \dots, n$ then p is called a *walk* from v_0 to v_n. A walk for which the a_j are distinct is called a *path.* Such a path is said to consist of the nodes $v_0, v_1, \dots, v_n$ and to have *length* n, which is denoted $|p|_l = n$.

If $v_n = v_0$, then the path is called a *circuit.* If the nodes in the circuit are all distinct (i.e. $v_i \neq v_k$ for $i \neq k$), then it is called an *elementary* circuit.

Just as we define weights w_{ji} for arcs ij, we can define the *weight of a path* as the sum of the weights of all arcs constituting the path. More formally, for the path $p = \{v_1v_2, v_2v_3, \dots, v_nv_{n+1}\}$ (from node v_1 to node v_{n+1}), define the weight $|p|_w$ of p in max-plus notation as

$$|p|_w = \bigotimes_{k=1}^{n} w_{v_{k+1}v_k}.$$

The *average weight* of a path p is $\frac{|p|_w}{|p|_l}$. For a circuit, we refer to this quantity as the *average circuit weight.*

Definition 2.1.3. For a graph $\mathcal{G} = (V, E)$ with node set V and arc set E, node j is said to be *reachable* from node i if there exists a path from i to j. Graph $\mathcal{G}$ is *strongly connected* if, for any two nodes $i, j \in V$, node j is reachable from node i.

Moreover, matrix $A \in \mathbb{R}_{\max}^{n\times n}$ is called *irreducible* if $\mathcal{G}(A)$ is strongly connected; if a matrix is not irreducible, it is called *reducible.*

Those graphs that are not strongly connected will nevertheless contain subgraphs that are strongly connected. There is, thus, a partitioning of the graph into subgraphs, each of whose nodes are not contained in other subgraphs. We classify this next.

We say that node j *communicates with* node i if either

- $i = j$ or
- i is reachable from j and j is also reachable from i.

We use the notation $\leftrightarrow$ to symbolise the relation “communicates with”, i.e. $i \leftrightarrow j$ iff i communicates with j. Note that we allow a solitary node to communicate with itself, even if there is no self-loop attached to it. Then, $\leftrightarrow$ is an equivalence relation. The equivalence class of node i due to $\leftrightarrow$ can then be defined as $[i] \overset{\text{def}}{=} \{j \in V | i \leftrightarrow j\}$.

It is, thus, possible to partition the node set V of a graph into disjoint subsets V_i such that $V = V_1 \cup V_2 \cup \cdots \cup V_q$. Each subset V_i is defined as containing nodes that communicate with each other but not with other nodes of V. By taking V_i together with arc set E_i, each of whose arcs has start node and end node in V_i, we obtain the subgraph $\mathcal{G}_i = (V_i, E_i)$. We define this subgraph as a *maximal strongly connected subgraph* (MSCS) of $\mathcal{G} = (V, E)$. Even though nodes in V_i do not communicate with nodes outside V_i, there can be a path from nodes in V_i to V_j $(j \neq i)$ and vice-versa, but not both. Thus, any graph can be identified as the union of MSCSs. We shall generally denote a MSCS $\mathcal{G}_i = (V_i, E_i)$ by the equivalence class $[i]$ as given above.

The following definition of the cyclicity of a graph will be important.

Definition 2.1.4. Denote the *cyclicity* of a graph $\mathcal{G}$ by $\sigma_{\mathcal{G}}$.

- If $\mathcal{G}$ is strongly connected, then $\sigma_{\mathcal{G}}$ equals the greatest common divisor of the lengths of all elementary circuits in $\mathcal{G}$. If $\mathcal{G}$ consists of only one node without a self-loop, then $\sigma_{\mathcal{G}}$ is defined to be one

- If $\mathcal{G}$ is not strongly connected, then $\sigma_{\mathcal{G}}$ equals the least common multiple of the cyclicities of all maximal strongly connected subgraphs of $\mathcal{G}$.

2.1.3 Max-plus algebra in our system

Recall the evolution equation (2.7), now rewritten as

$$\mathbf{x}(k+1) = A_\xi(k+1) \otimes T(k) \otimes \mathbf{x}(k).$$

In this chapter, we remove the dependence on time k of all parameters so that $T(k) = T$, $A_\xi(k) = A_\xi$ and $P(k) = P$, fixed for all k. Let $\mathbf{x}(0)$ represent the initial state of all cells (which is at epoch $k = 0$). Then we can rewrite (2.7) in its full form as

$$\mathbf{x}(k+1) = P \otimes P \otimes \cdots P \otimes \mathbf{x}(0) = P^{\otimes(k+1)} \otimes \mathbf{x}(0)$$

or equivalently

$$\mathbf{x}(k) = P^{\otimes k} \otimes \mathbf{x}(0). \tag{2.8}$$

This is a neat closed form for the evolution equation, and we generate max-plus powers of the matrix P to evolve and analyse the dynamics. Given $\mathbf{x}(0)$, the sequence of vectors $\{\mathbf{x}(k) : k \in \mathbb{N}\}$, obtained by iterating (2.8), is referred to as the *orbit* of $\mathbf{x}(0)$.

A study of such sequences is provided in [22], Chapters 3 and 4, and the reader is advised to refer to this for greater scope and proofs where not given. We shall detail the topics relevant for this work.

In particular, consider the timing dependency graph of the regular network. A portion of the graph is exactly the graph in Figure 2.1. Notice that each node i is reachable from every other node j, so that $\mathcal{G}(P)$ is strongly connected and P is then irreducible. This is an important property because, as we shall see, irreducibility ensures periodic behaviour of $\mathbf{x}(k)$ in (2.8) which is the same for all nodes.

Asymptotic behaviour of the max-plus system

Definition 2.1.5. Let $A \in \mathbb{R}_{\max}^{n\times n}$. If $\lambda \in \mathbb{R}_{\max}$ is a scalar and $\mathbf{v} \in \mathbb{R}_{\max}^{n}$ is a vector that contains at least one finite element such that

$$A \otimes \mathbf{v} = \lambda \otimes \mathbf{v},$$

then, analogously to conventional linear algebra, λ is called an *eigenvalue* of A and $\mathbf{v}$ is an *eigenvector* of A associated with eigenvalue λ.

We shall occasionally refer to λ and $\mathbf{v}$ as an eigenvalue and eigenvector of the *system*. Our system is given by $\mathbf{x}(k) = P^{\otimes k} \otimes \mathbf{x}(0)$, where P is irreducible. It turns out that for such an irreducible system there is only one eigenvalue, as stated in the following theorem.

Theorem 2.1.1 ([22], Theorem 2.9)**.** Let $A \in \mathbb{R}_{\max}^{n\times n}$ be irreducible. Then A possesses a unique eigenvalue, denoted $\lambda(A)$, which is finite ($\neq \varepsilon$). Moreover, this eigenvalue is equal to the maximal average weight of elementary circuits in $\mathcal{G}(A)$. Let c denote an elementary circuit of $\mathcal{G}(A)$. Denote the set of all elementary circuits of $\mathcal{G}(A)$ by $\mathcal{C}(A)$. Then

$$\lambda(A) = \max_{c\in\mathcal{C}(A)} \frac{|c|_w}{|c|_l}.$$

Proof. See [22], Theorem 2.9. □

A circuit c is called *critical* if its average weight is maximal, i.e. $\lambda = \frac{|c|_w}{|c|_l}$. The *critical graph* of A, denoted by $\mathcal{G}^{cr}(A)$ is the graph consisting of those nodes and arcs that belong to critical circuits in $\mathcal{G}(A)$.

If the initial vector $\mathbf{x}(0)$ is an eigenvector, then we can apply Theorem 2.1.1 to give the sequence of vectors $\mathbf{x}(0), \mathbf{x}(1), \mathbf{x}(2), \mathbf{x}(3), \dots$ such that $\mathbf{x}(k+1) = \lambda \otimes \mathbf{x}(k)$ for $k \geq 0$. The times $\mathbf{x}(k)$, thus, fall into a pattern but, because times are, by conventional definition, monotone increasing, we must define exactly what we mean by this pattern; we relate it to periodic behaviour as follows.

Definition 2.1.6. Let $A \in \mathbb{R}_{\max}^{n \times n}$. For some $k \geq 0$, consider the set of vectors

$$\mathbf{x}(k), \mathbf{x}(k+1), \mathbf{x}(k+2), \ldots \in \mathbb{R}_{\max}^{n}$$

where $\mathbf{x}(k) = A^{\otimes k}\mathbf{x}(0)$ for all $k \geq 0$. The set is called a *periodic regime* if there exists $\mu \in \mathbb{R}_{\max}$ and a finite number $\rho \in \mathbb{N}$ such that

$$\mathbf{x}(k+\rho) = \mu \otimes \mathbf{x}(k).$$

The *period* of the regime is ρ and μ/ρ is the *cycletime*.

For brevity, we will often refer to the periodic regime simply as the regime. The cycletime can be thought of as being a measure of the average delay between event times $\mathbf{x}(k)$ and $\mathbf{x}(k+1)$ in a regime. Contrary to what the follow-up to Theorem 2.1.1 might suggest, it turns out that we need not initialise the system to the eigenvector to ensure periodicity. The remainder of this section is dedicated to showing this and the conditions under which μ always exists. We shall further show that such conditions ensure that the cycletime satisfies $\mu/\rho = \lambda$, the eigenvalue; we shall also relate the period ρ with the cyclicity of P, which is defined as follows.

Definition 2.1.7. Let $A \in \mathbb{R}_{\max}^{n \times n}$ be irreducible. The *cyclicity of* A, denoted $\sigma(A)$, is defined as the cyclicity of the critical graph of A.

When the matrix is understood, we shall sometimes denote the cyclicity by σ. We are now able to state the following crucial theorem of max-plus algebra.

Theorem 2.1.2 ([22], Theorem 3.9)**.** Let $A \in \mathbb{R}_{\max}^{n \times n}$ be an irreducible matrix with eigenvalue λ and cyclicity $\sigma(A)$. Then there is a t such that

$$A^{\otimes(k+\sigma(A))} = \lambda^{\otimes\sigma(A)} \otimes A^{\otimes k}$$

for all $k \geq t$.

Proof. See [22], Theorem 3.9. □

Let $\sigma = \sigma(P)$. To give us an indication of the asymptotic behaviour of our system, we apply this theorem to the state at epoch $k+\sigma$ for $k \geq t$:

$$\begin{aligned} \mathbf{x}(k+\sigma) &= P^{\otimes k+\sigma} \otimes \mathbf{x}(0) \\ &= \lambda^{\otimes\sigma} \otimes P^{\otimes k} \otimes \mathbf{x}(0) \\ &= \lambda^{\otimes\sigma} \otimes \mathbf{x}(k) \qquad (2.9) \end{aligned}$$

where $\lambda^{\otimes\sigma}$ is read as $\lambda \times \sigma$ in terms of classical algebra. This implies that the vector $\mathbf{x}(k+\sigma)$ is found by adding $\lambda \times \sigma$ componentwise to the vector $\mathbf{x}(k)$. From the viewpoint of Definition 2.1.6, we get the following corollary, in which we make use of the term *eventually periodic*, which refers to an orbit of $\mathbf{x}(0)$ which enters a periodic regime for some large enough k.

Corollary 2.1.1. Consider the recurrence relation $\mathbf{x}(k+1) = P \otimes \mathbf{x}(k)$ for $k \geq 0$ and $P \in \mathbb{R}_{\max}^{N\times N}$ irreducible. Then the sequence $\{\mathbf{x}(k)\}$ is eventually periodic with period $\sigma(P)$ and cycletime $(\lambda \times \sigma)/\sigma = \lambda$, where λ is the eigenvalue of P.

It may well be that a regime with smaller period ρ is contained within this regime; in that case, ρ must be a factor of σ since (2.9) must hold true. The exact value of ρ is dependent on $\mathbf{x}(0)$ and so may be written as $\rho(\mathbf{x}(0))$. Thus, we have the following corollary, stating that the period ρ of a periodic regime is bounded by the cyclicity of A.

Corollary 2.1.2. Consider the recurrence relation $\mathbf{x}(k+1) = A \otimes \mathbf{x}(k)$ for $k \geq 0$ and $A \in \mathbb{R}_{\max}^{n\times n}$ irreducible. For all initial conditions $\mathbf{x}(0)$, there exists $r \in \mathbb{N}$ such that

$$r\rho(\mathbf{x}(0)) = \sigma(A).$$

for some $r \in \mathbb{N}$.

This guarantees the periodic behaviour of the max-plus system. Therefore, the vectors $\mathbf{x}(k)$ in a regime turn out to be eigenvectors of $P^{\otimes\sigma}$ associated with eigenvalue $\lambda \times \sigma$, shown as follows.

$$\begin{aligned} \lambda^{\otimes\sigma} \otimes \mathbf{x}(k) &\overset{\text{from above}}{=} P^{\otimes k+\sigma} \otimes \mathbf{x}(0) \\ &= P^{\otimes\sigma} \otimes P^{\otimes k} \otimes \mathbf{x}(0) \\ &= P^{\otimes\sigma} \otimes \mathbf{x}(k). \end{aligned}$$

This may be easier to see by letting $\sigma = 1$, in which case $\mathbf{x}(k+1) = P \otimes \mathbf{x}(k) = \lambda \otimes \mathbf{x}(k)$.

So far, we have observed that, in a periodic regime, the equation $x_i(k+\rho) = \mu \otimes x_i(k)$ is true for all i. It will be useful to consider the conditions under which the cycletime is variable with each node. Denote by $\chi = (\chi_1, \chi_2, \dots, \chi_N)^\top$ the *cycletime vector*, where the cycletime of node i is now denoted by χ_i. Many authors such as [22] define this cycletime asymptotically as in the next definition.

Definition 2.1.8. Let $\{x_i(k) : k \in \mathbb{N}\}$ be an orbit of $x_i(0)$ in $\mathbb{R}_{\max}$. Assuming that it exists, the quantity χ_i, defined by

$$\chi_i = \lim_{k\to\infty} \frac{x_i(k)}{k}$$

is called the *cycletime* of i.

This definition of cycletime is equivalent to our earlier observation where we thought of the cycletime as an average delay between successive states $x_i(k)$ and $x_i(k+1)$ of node i. The delay between update times $x_i(k)$ and $x_i(k+1)$ is $x_i(k+1) - x_i(k)$. Since $x_i(k+1) = [P \otimes \mathbf{x}(k)]_i$, this delay can be written as $[P \otimes \mathbf{x}(k)]_i - x_i(k)$. If we take several states in an orbit of $x_i(0)$, the mean delay is

$$\sum_{j=1}^{k} ([P^j \otimes \mathbf{x}(0)]_i - [P^{j-1} \otimes \mathbf{x}(0)]_i),$$

where $[P^j \otimes \mathbf{x}(0)]_i$ denotes the ith element of the vector $P^j \otimes \mathbf{x}(0)$. Cancellation of terms simplifies the above to $\frac{[P^k \otimes \mathbf{x}(0)]_i - x_i(0)}{k}$. Asymptotically, this reduces to

$$\lim_{k\to\infty} \frac{[P^k \otimes \mathbf{x}(0)]_i}{k},$$

which is indeed the same quantity as in Definition 2.1.8.

The *cycletime vector* χ is the vector of individual cycletimes of each node, i.e. for N nodes,

$$\chi = \left(\lim_{k\to\infty} \frac{x_1(k)}{k}, \lim_{k\to\infty} \frac{x_2(k)}{k}, \dots, \lim_{k\to\infty} \frac{x_N(k)}{k} \right)^\top .$$

We may also represent this as $\lim_{k\to\infty} \frac{\mathbf{x}(k)}{k}$.

Let us state three important properties of our max-plus system defined by (2.8), which can be regarded as a function mapping the state space $\mathbb{R}^N_{\max}$ to itself.

Monotonicity Consider $\mathbf{x}(k), \mathbf{x}(k') \in \mathbb{R}^n_{\max}$ such that $\mathbf{x}(k) \le \mathbf{x}(k')$, i.e. $x_i(k) \le x_i(k')$ for $i = 1, \dots, n$. Then $A \otimes \mathbf{x}(k) \le A \otimes \mathbf{x}(k')$, where these inequalities are interpreted componentwise.

Homogeneity For any scalar $\alpha \in \mathbb{R}$ and any $\mathbf{x}(k) \in \mathbb{R}^n_{\max}$, $A \otimes (\alpha \otimes \mathbf{x}(k)) = \alpha \otimes (A \otimes \mathbf{x}(k))$, where the scalar max-plus multiplication is equivalent to conventional componentwise addition of α.

Nonexpansiveness $\|A \otimes \mathbf{x}(k) - A \otimes \mathbf{x}(k')\|_\infty \leq \|\mathbf{x}(k) - \mathbf{x}(k')\|_\infty$ for arbitrary $\mathbf{x}(k)$, $\mathbf{x}(k') \in \mathbb{R}^n_{\max}$. Here, $\| \cdot \|_\infty$ is the *supremum norm* of a vector, defined as $\|\mathbf{v}\|_\infty = \max_{1 \leq i \leq n} |v_i|$ for $\mathbf{v} \in \mathbb{R}^n$. It is sometimes also referred to as the l^∞-norm.

Functions that satisfy these three properties are called *topical functions*. The properties are fundamental to the proof of the following theorem.

Theorem 2.1.3. Consider the recurrence relation $\mathbf{x}(k+1) = A \otimes \mathbf{x}(k)$ for $k \geq 0$ and $A \in \mathbb{R}^{n \times n}_{\max}$ irreducible. For some $\mathbf{x}_\star(0) \in \mathbb{R}^n_{\max}$ whose elements are all finite, if the limit $\lim_{k \to \infty} \frac{A^k \otimes \mathbf{x}_\star(0)}{k}$ exists, then this limit is the same for any initial condition $\mathbf{x}(0) \in \mathbb{R}^n_{\max}$ whose elements are all finite.

Proof. See [22], Theorem 3.11. □

Notice that the irreducibility of A was not vital. In fact, the condition of irreducibility can be relaxed to that of *regularity*, which means that all rows of A contain at least one non-zero ($\neq -\infty$) element[3]; this corresponds to each node having at least one upstream node in $\mathcal{G}(A)$. It should be clear now that irreducibility of a matrix implies regularity (but not necessarily vice versa). For present purposes, we shall not be too concerned with the particulars of regular matrices, suffice it to say that once the cycletime vector is found to exist, it is unique and independent of the initial vector (even for A reducible), as long as A is regular.

Thus, in the case where A is reducible (yet regular), we obtain a cycletime vector, whose elements may not necessarily be identical (this will be shown in a later chapter). However, for A irreducible, each element of χ turns out to be the same, as hypothesised by the work above; we now state this result formally in the following lemma.

Lemma 2.1.1. For the recurrence relation $\mathbf{x}(k+1) = A \otimes \mathbf{x}(k)$ with $k \geq 0$, let $A \in \mathbb{R}^{n \times n}_{\max}$ be an irreducible matrix having eigenvalue $\lambda \in \mathbb{R}$. Then, for $i = 1, 2, \ldots, n$,

$$\lim_{k \to \infty} \frac{x_i(k)}{k} = \lambda$$

for any initial condition $\mathbf{x}(0) \in \mathbb{R}^n$.

[3] Regularity refers to a *matrix* here, and should not be confused with the regular n-nbhd *network* that forms the underlying network

Proof. Let $\mathbf{v}$ be an eigenvector of A. Initialising the recurrence relation with $\mathbf{x}(0) = \mathbf{v}$ gives $\mathbf{x}(1) = A \otimes \mathbf{x}(0) = A \otimes \mathbf{v}$, where $A \otimes \mathbf{v} = \lambda \otimes \mathbf{v}$. Then, for $k \geq 0$,

$$\begin{aligned} \mathbf{x}(k) &= A^{\otimes k} \otimes \mathbf{x}(0) \\ &= A^{\otimes k} \otimes \mathbf{v} \\ &= \lambda^{\otimes k} \otimes \mathbf{v} \\ &= k \times \lambda \otimes \mathbf{v}. \end{aligned}$$

Then each element of $\mathbf{x}(k)$ may be written as $x_i(k)$ where

$$x_i(k) = k \times \lambda + v_i.$$

Thus,

$$\lim_{k \to \infty} \frac{x_i(k)}{k} = \lambda,$$

independently of i. Theorem 2.1.3 says that once the cycletime vector exists, it is independent of $\mathbf{x}(0)$, so that λ is the cycletime for all nodes for all initial conditions. □

Here, the irreducibility of A is necessary because it ensures the existence of a (unique) eigenvalue of A, and this is crucial to the proof of Lemma 2.1.1. This makes the lemma a tighter result than Theorem 2.1.3, where it is enough for A to be regular.

For us, since we have shown that the cycletime vector is independent of the initial condition, we shall relate it to the timing dependency graph by denoting it $\chi(P)$, and referring to it as the *cycletime vector of* P. If P is irreducible, as is the case in most of the thesis, we shall revert to denoting the cycletime as χ, and therefore refer to the cycletime vector of irreducible P simply as the cycletime of P. In either case, the context will make things clear.

As a compact summary, we have shown that, for any initial state $\mathbf{x}(0)$, there exists a K such that, for $k > K$, the irreducible max-plus system enters the periodic regime

$$\mathbf{x}(k + \rho) = \lambda^{\otimes \rho} \otimes \mathbf{x}(k). \tag{2.10}$$

i.e. for all initial vectors $\mathbf{x}(0)$, the cycletime of the regime is the eigenvalue λ of the timing dependency matrix P whilst the period is ρ, which is bounded by the cyclicity of P. Theorem 2.1.1 and Definition 2.1.7 provide procedures by which these quantities can be found.

2.1.4 The transient time in max-plus algebra

Along with this section, the next three sections will describe an eigenvector in max-plus algebra. Here, let us introduce an example. Take $N = 5$ and a regular 3-nbhd max-plus system with $\tau_{ij} = e$ for all i, j. Then,

$$P = \begin{pmatrix} 1 & 1 & \varepsilon & \varepsilon & 1 \\ 2 & 2 & 2 & \varepsilon & \varepsilon \\ \varepsilon & 3 & 3 & 3 & \varepsilon \\ \varepsilon & \varepsilon & 4 & 4 & 4 \\ 5 & \varepsilon & \varepsilon & 5 & 5 \end{pmatrix}.$$

The iteration of $\mathbf{x}(k) = P^{\otimes k}\mathbf{x}(0)$ with $\mathbf{x}(0) = \mathbf{u} = (0,0,0,0,0)^\top$ yields the following sequence of vectors for $\mathbf{x}(k)$.

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix}, \begin{pmatrix} 6 \\ 5 \\ 7 \\ 9 \\ 10 \end{pmatrix}, \begin{pmatrix} 11 \\ 9 \\ 12 \\ 14 \\ 15 \end{pmatrix}, \begin{pmatrix} 16 \\ 14 \\ 17 \\ 19 \\ 20 \end{pmatrix}, \begin{pmatrix} 21 \\ 19 \\ 22 \\ 24 \\ 25 \end{pmatrix}, \cdots.$$

Notice that the first epoch of periodic behaviour is when

$$\begin{pmatrix} 16 \\ 14 \\ 17 \\ 19 \\ 20 \end{pmatrix} = 5 \otimes \begin{pmatrix} 11 \\ 9 \\ 12 \\ 14 \\ 15 \end{pmatrix},$$

i.e. $\mathbf{x}(4) = 5 \otimes \mathbf{x}(3)$ and it is evident that $\mathbf{x}(k+1) = 5 \otimes \mathbf{x}(k)$ for all $k \geq 3$. This gives period 1 and cycletime 5 (which corresponds to the largest of all length 1 elementary circuits in $\mathcal{G}(P)$). Let us define the following.

Definition 2.1.9. Given the initial state $\mathbf{x}(0)$, consider the periodic regime given by

$$\mathbf{x}(k+\rho) = \lambda^{\otimes\rho} \otimes \mathbf{x}(k) \tag{2.11}$$

where ρ is the period and λ the cycletime. Let $K(\mathbf{x}(0))$ be the smallest k for which (2.11) holds. Then $K(\mathbf{x}(0))$ is called the *transient time of the (periodic) regime given* $\mathbf{x}(0)$.

When the initial condition is understood, we shall also denote $K(\mathbf{x}(0))$ simply as K. In the above example, $K(\mathbf{x}(0)) = 3$. For another initial state, $K(\mathbf{x}(0))$ might be different, e.g. take $\mathbf{x}(0) = (11, 9, 12, 14, 15)^\top$ for which $K(\mathbf{x}(0)) = 0$ since the system is instantaneously periodic. Therefore, like the period, the transient time depends on the initial state, whereas the cycletime does not.

We can now give an upper bound for $K(\mathbf{x}(0))$. We need the following definition, which is not too dissimilar to the definition above.

Definition 2.1.10. Let $A \in \mathbb{R}_{\max}^{N\times N}$ be an irreducible matrix with eigenvalue λ and cyclicity $\sigma(A)$. The *transient time of* A is denoted by $t(A)$ and is the smallest k for which the following relation holds.

$$A^{\otimes(k+\sigma(A))} = \lambda^{\otimes\sigma(A)} \otimes A^{\otimes k}$$

Notice that $t(A)$ is exactly the same as t, as given in Theorem 2.1.2.

Theorem 2.1.4. Let $A \in \mathbb{R}_{\max}^{N\times N}$ be an irreducible matrix having transient time $t(A)$. Given $\mathbf{x}(0)$, consider the recurrence relation $\mathbf{x}(k+1) = A \otimes \mathbf{x}(k)$ which yields a periodic regime for $k \geq K(\mathbf{x}(0))$. Then

$$K(\mathbf{x}(0)) \leq t(A).$$

Proof. Let the period of the periodic regime be ρ. For some $r \in \mathbb{N}$, $\sigma = r\rho$, so $\rho \leq \sigma$. The proof follows from the fact that we must see a periodic regime of period ρ before (or at the same time as) we see a regime of period σ. □

The work carried out thus far in this chapter has been geared towards the next section, in which we address the impact on the asynchronous time framework of the contour plot that was introduced earlier.

2.1.5 The Hasse diagram and the contour plot

For each k, we interpret the pair $(i, x_i(k))$ as a two-dimensional coordinate in the Euclidean plane. We shall later see that the elements $x_i(k) \in \mathbf{x}(k)$ are not causally related so that $\mathbf{x}(k)$ can be viewed as an *antichain*. The topic of antichains is surveyed at length in [5], including that of a Hasse diagram. In this section, we define the Hasse diagram formally, before which we will require the “happened before” relation, denoted by $\prec$, that was introduced in Chapter 1. Recall that, for each cell i, the times at which internal processes occur are referred to as events. Consequently, one might replace $\prec$ by $<$ since that is the condition that must be satisfied by predecessor events happening before successor events.

The following defines a partial ordering of events in our system.

Definition 2.1.11. Let X denote the set of all events. Then

(irreflexivity) For all $x \in X, x \not\prec x$;

(antisymmetry) If $x \prec y$ and $y \prec x$ then $x = y$;

(transitivity) If $x \prec y$ and $y \prec z$ then $x \prec z$.

In literature, the irreflexivity is commonly relaxed to reflexivity (allowing xRx, where R is the binary relation) as in [5]. However, we look at the ordering of processes in time, which cannot possibly allow an event to precede itself; we have in fact defined a *strict* partial ordering.

We can now define a Hasse diagram in the context of our system. Note first that, since the set X along with the relation $\prec$ forms a partially ordered set ("poset"), we can make use of the rules and definitions that apply to posets on our system of events. We call the pair $(X, \prec)$ a poset. The following definitions prove useful.

Definition 2.1.12. Let x and y be distinct elements of a poset $(X, \prec)$. y is said to *cover* x if $x \prec y$ but no element z satisfies $x \prec z \prec y$.

Definition 2.1.13. Let the set X_1 of n elements $\{x_1, x_2, \ldots, x_n\}$ be a subset of the poset $(X, \prec)$ such that each element may be totally ordered according to $\prec$ as $x_1 \prec x_2 \prec \cdots \prec x_n$. Then X_1 is a *chain.* The subset $X_2 \in X$ is called an *antichain* if and only if no elements of X_2 may be totally or partially ordered.

These definitions allow us to formulate the following.

Definition 2.1.14. The *Hasse diagram* of a poset $(X, \prec)$ is a graph drawn in the Euclidean plane such that each element of the poset is represented by a unique vertex in the graph. For each covering pair $x \prec y$, the point representing x and y are joined by an arc with arrow pointing to y and the point representing x is 'below' the point representing y (i.e. it has smaller Y-coordinate).

Figure 2.3 can now be seen as a Hasse diagram. By the term "causal relation" we allude to those events which are ordered according to the relation "$\prec$". For example, the "send" events at times $x_{i-1}(k-1)$, $x_i(k-1)$ and $x_{i+1}(k-1)$ are not causally related (since there is no arc connecting any of the three events). We say that $x_{i-1}(k-1)$,

$x_i(k-1)$ and $x_{i+1}(k-1)$ are contained in the same antichain. In this thesis, the study will be elucidated by connecting those elements in the same antichain; we then obtain a piecewise linear plot of the vector $\mathbf{x}(k)$, which we define next.

Definition 2.1.15. Consider the vector $\mathbf{x}(k)$. A *contour* is the plot obtained by connecting $(i, x_i(k))$ to $(i+1, x_{i+1}(k))$ with a straight line for each i, $(i = 1, \ldots, N)$.

The straight lines can be regarded as being drawn between 'neighbouring' coordinates. Repeating the process of creating a contour for each k gives a pictorial representation of vectors $\mathbf{x}(k)$ as a function of k, where the underlying global time t moves vertically upwards. We call this a *contour plot.* Figure 2.2 at the beginning of this chapter displays the contour plots of a size 20 system, where the sequence $\{\mathbf{x}(0), \mathbf{x}(1), \mathbf{x}(2), \ldots\}$ represents the contours (counting k from the bottom). For this reason, we shall interchangeably refer to the vector $\mathbf{x}(k)$ by "the kth contour" from now on.

Between successive contours, we can imagine there being drawn the internal processes such as those in Figure 2.3. Notice that contours do not intersect; this is a consequence of the monotonicity of the max-plus model and will be proved in Chapter 3. Consider the example system given in Section 2.1.4. The contours for this system would represent vectors in the periodic regime $\{\mathbf{x}(k) | \mathbf{x}(k+1) = 5 \otimes \mathbf{x}(k), \mathbf{x}(0) = \mathbf{u}, k \geq 3\}$. The period of a regime and cyclicity are related by $1 \leq \rho \leq \sigma$. Since $\rho = 1$ and $\sigma = 1$ in this example, we find that there is no other period that can be obtained for all initial states $\mathbf{x}(0)$. Thus, the contour plot, in particular the periodic regime, follows a unique evolution. This general idea is proved next. For $\rho = 1$, as it is here, each contour in the periodic regime has the same shape; for a larger period, we obtain a different contour plot of the periodic regime. In particular, we can obtain different plots for the cases $\rho = 1, 2, \ldots, \sigma$, each dependent on the choice of $\mathbf{x}(0)$.

The moving frame

The idea of a 'limiting shape' in contours suggests a change of coordinates: Given the irreducible matrix $P \in \mathbb{R}_{\max}^{N \times N}$ with eigenvalue λ, let

$$\mathbf{x}(k) = \lambda^{\otimes k} \otimes \mathbf{y}(k). \tag{2.12}$$

We can think of $\lambda^{\otimes k}$ as a diagonal matrix, i.e. the product of $\lambda^{\otimes k}$ and the unit matrix E. The advantage of this is that such a diagonal matrix is invertible, its inverse being the diagonal matrix with diagonal entries equal to $\lambda^{\otimes -k}$. Using this property, we rearrange Equation (2.12) to obtain

$$\mathbf{y}(k) = \lambda^{\otimes -k} \otimes \mathbf{x}(k).$$

In other words, $\mathbf{y}(k)$ is the limit to which the vectors $\mathbf{x}(k) - \lambda k$ tend to as $k \to \infty$. By studying the asymptotic behaviour of $\mathbf{y}(k)$ itself, we shall see what the shape of the aforementioned limiting contour looks like.

The original system follows the recurrence relation $\mathbf{x}(k+1) = P \otimes \mathbf{x}(k)$ for some $\mathbf{x}(0) \in \mathbb{R}^N_{\max}$. Substitute (2.12) into this to obtain

$$\lambda^{\otimes(k+1)} \otimes \mathbf{y}(k+1) = P \otimes \lambda^{\otimes k} \otimes \mathbf{y}(k).$$

Interpreting $\lambda^{\otimes k}$ as a diagonal matrix again yields

$$\begin{aligned} \mathbf{y}(k+1) &= \lambda^{\otimes -(k+1)} \otimes P \otimes \lambda^{\otimes k} \otimes \mathbf{y}(k) \\ &= \lambda^{\otimes -1} \otimes P \otimes \mathbf{y}(k) \\ &= \hat{P} \otimes \mathbf{y}(k) \qquad (2.13) \end{aligned}$$

where $\hat{P} = \lambda^{\otimes -1} \otimes P$ represents the *normalised matrix*, which is equivalently obtained by subtracting the eigenvalue of P from each of its entries. The communication graph of $\hat{P}$ is the same as that for P (but with different arcweights); P is irreducible, therefore $\hat{P}$ is too. However, the maximum average circuit weight of $\mathcal{G}(\hat{P})$, hence its eigenvalue, is zero.

We use this property to deduce the limiting behaviour of $\mathbf{y}(k)$. Since $\mathcal{G}(\hat{P})$ is obtained by subtracting λ from each of the arcweights in $\mathcal{G}(P)$, both graphs have the same cyclicity, i.e. $\sigma(\hat{P}) = \sigma(P) = \sigma$. Theorem 2.1.2 tells us of the asymptotic behaviour of the powers of an irreducible matrix. Apply this to $\hat{P}$ to obtain

$$\begin{aligned} \hat{P}^{\otimes(k+\sigma)} &= 0^{\otimes\sigma} \otimes \hat{P}^{\otimes k} \\ &= \hat{P}^{\otimes k} \end{aligned}$$

for $k \geq t$. Thus, using Equation (2.13),

$$\begin{aligned} \mathbf{y}(k+\sigma) &= \hat{P}^{\otimes(k+\sigma)} \otimes \mathbf{y}(0) \\ &= \hat{P}^{\otimes k} \otimes \mathbf{y}(0) \\ &= \mathbf{y}(k). \end{aligned}$$

So the limiting contour $\mathbf{y}(k)$ is periodic with period $\sigma(P)$. (Note that this period now conforms with the traditional dynamical systems definition of a period in that the sequence $\{\mathbf{y}(k)\}$ is not monotonically increasing, whereas the sequence $\{\mathbf{x}(k)\}$ is). In fact, this period $\sigma(P)$ is an upper bound on the actual period obtained for a general initial condition $\mathbf{y}(0)$. This is due to the same fact being true for the corresponding vectors $\mathbf{x}(k)$ that form the underlying periodic regime (see Corollary 2.1.2).

Loosely speaking, this section has shown that there is no unique shape to the limit of a contour plot since it depends on the initial state $\mathbf{x}(0)$. This yields an interesting feature of the max-plus asynchronous model, which relates the asynchrony to not only the timing dependency graph but also at which point in time the system is started.

2.1.6 The eigenspace in max-plus algebra

Consider the example system given in Section 2.1.4 again. Let $\mathbf{x}(k)$ be a vector in the periodic regime. Then $\mathbf{x}(k+1) = 5 \otimes \mathbf{x}(k)$, i.e. $P \otimes \mathbf{x}(k) = 5 \otimes \mathbf{x}(k)$. The eigenvalue of P is obviously 5 and, since $\sigma(P) = 1$, $\mathbf{x}(k)$ is an eigenvector associated with this eigenvalue.

It is evident that eigenvectors are not unique because, as in linear algebra, eigenvectors are defined up to scalar multiplication. The max-plus interpretation is that if $\mathbf{v}$ and $\mathbf{w}$ are eigenvectors of A associated with eigenvalue λ, then, for $\alpha, \beta \in \mathbb{R}_{\max}$, $\alpha \otimes \mathbf{v} \oplus \beta \otimes \mathbf{w}$ is also an eigenvector, shown as follows. Let $\mathbf{v}$ and $\mathbf{w}$ be eigenvectors of A associated with eigenvalue λ. So $A \otimes \mathbf{v} = \lambda \otimes \mathbf{v}$ and $A \otimes \mathbf{w} = \lambda \otimes \mathbf{w}$. Then

$$\begin{aligned} A \otimes (\alpha \otimes \mathbf{v} \oplus \beta \otimes \mathbf{w}) &= \alpha \otimes A \otimes \mathbf{v} \oplus \beta \otimes A \otimes \mathbf{w} \\ &= \alpha \otimes \lambda \otimes \mathbf{v} \oplus \beta \otimes \lambda \otimes \mathbf{w} \\ &= \lambda \otimes (\alpha \otimes \mathbf{v} \oplus \beta \otimes \mathbf{w}). \end{aligned}$$

The set of all eigenvectors of A associated to eigenvalue λ is the *eigenspace* of A. We denote such a set as $\bar{V}(A, \lambda)$.

Let us return to the contour plot with underlying recurrence relation $\mathbf{x}(k+1) = P \otimes \mathbf{x}(k)$. In an earlier section, we established that each contour is an eigenvector of $P^{\sigma(P)}$. Now, since any linear combination of eigenvectors is also an eigenvector, we apply this method for constructing eigenvectors to find the eigenspace of P^{σ}. This eigenspace is the set of all possible periodic regimes (i.e. of all periods ρ, $1 \leq \rho \leq \sigma$, obtained for all initial states $\mathbf{x}(0)$), which corresponds to the set of all contour plots that can be obtained. The next theorem gives a method for constructing the eigenspace of an irreducible matrix such as ours. It uses the definition of the

Kleene star for any $A \in \mathbb{R}_{\max}^{N\times N}$:

$$A^* \overset{\text{def}}{=} \bigoplus_{k=0}^{\infty} A^k.$$

For any square matrix A, the element $[A^{\otimes k}]_{ij}$ is the largest weight for a path of length k from node j to i in the communication graph of A. Thus, if elements of A are positive, then the elements of A^* may tend to infinity. Conversely, if all elements of A are nonpositive, then A^* is finite. In particular, if circuit weights in $\mathcal{G}(A)$ are nonpositive, then the Kleene star of a square matrix over $\mathbb{R}_{\max}$ exists [22].

Recall the critical graph of A as $\mathcal{G}^{cr}(A) = (V^{cr}(A), E^{cr}(A))$ and the normalised matrix $\hat{A} = -\lambda \otimes A$. Then

Theorem 2.1.5 ([22], Theorem 4.5)**.** Let $A \in \mathbb{R}_{\max}^{N\times N}$ be irreducible and consider $\hat{A}^*$ to be the Kleene star of $\hat{A} = -\lambda \otimes A$.

1. If node i belongs to $\mathcal{G}^{cr}(A)$, then $[\hat{A}^*]_{\cdot i}$ is an eigenvector of A.

2. The eigenspace of A is

$$V(A) = \{\mathbf{v} \in \mathbb{R}_{\max}^{N} | \mathbf{v} = \bigoplus_{i\in V^{cr}(A)} a_i \otimes [\hat{A}^*]_{\cdot i} \quad \text{for } a_i \in \mathbb{R}_{\max}\}.$$

3. For i, j belonging to $\mathcal{G}^{cr}(A)$, there exists $a \in \mathbb{R}$ such that

$$a \otimes [\hat{A}^*]_{\cdot i} = [\hat{A}^*]_{\cdot j}$$

if and only if i and j belong to the same MSCS of $\mathcal{G}^{cr}(A)$.

Proof. In [22], Theorem 4.5. □

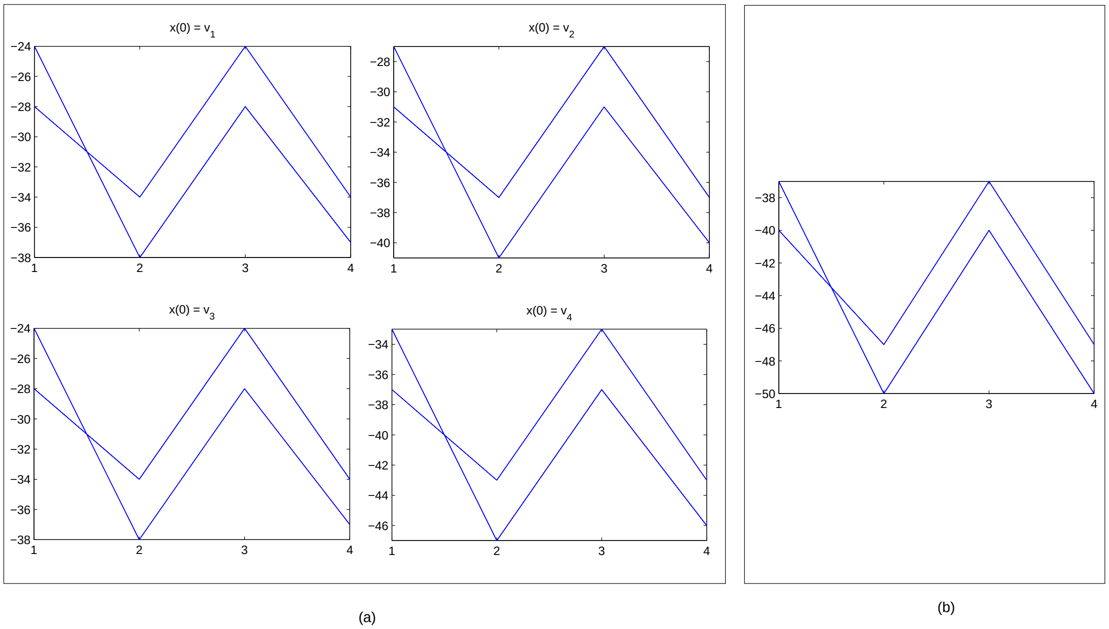
Let $B = P^{\sigma}$. Applying Theorem 2.1.5 to irreducible B will yield its eigenspace and consequently all possible contour plots for the system $\mathbf{x}(k+1) = P \otimes \mathbf{x}(k)$. The biggest step is to calculate the Kleene star of $\hat{B}$ for which we need the eigenvalue $\lambda(B)$ of B to form $\hat{B} = -\lambda(B) \otimes B$. For each ρ $(1 \le \rho \le \sigma)$, we can show that period ρ regimes are not unique, e.g. we may obtain linearly independent eigenvectors of A, likewise for $A^{\otimes \rho}$ etc.

Example 2.1.1. Consider a regular 3-nbhd network of size $N = 4$ with the processing and transmission matrices given as follows. Blank spaces signify ε entries.

$$A_\xi = \begin{pmatrix} 3 & & & \\ & 2 & & \\ & & 8 & \\ & & & 6 \end{pmatrix}, \quad T = \begin{pmatrix} \varepsilon & 18 & 21 & 4 \\ 12 & \varepsilon & 7 & 2 \\ 16 & \varepsilon & 12 & 7 \\ 8 & 5 & 5 & \varepsilon \end{pmatrix}.$$

Then $P = A_\xi \otimes T = \begin{pmatrix} \varepsilon & 21 & 24 & 7 \\ 14 & \varepsilon & 9 & 4 \\ 24 & \varepsilon & 20 & 15 \\ 14 & 11 & 11 & \varepsilon \end{pmatrix}$. The eigenvalue $\lambda = \lambda(P)$ and cyclicity σ of P are found by taking powers or, since P is irreducible, by looking for circuits in $\mathcal{G}(P)$, the timing dependency graph of P. Here, $\lambda = 24$ and $\sigma = 2$. Thus, the highest power of P that we consider when looking for periodic regimes is 2. Consider $Q = P^{\otimes 2} = \begin{pmatrix} 48 & 18 & 44 & 39 \\ 33 & 35 & 38 & 24 \\ 44 & 45 & 48 & 35 \\ 35 & 35 & 38 & 26 \end{pmatrix}$. Let $\lambda(Q)$ be the eigenvalue of Q. Then $\lambda(Q) = 48$ and its cyclicity is one; by studying $\mathcal{G}(Q)$, this is verified by self-loops at nodes 1 and 3 each having arcweight 48. Thus, $\hat{Q}^* = \begin{pmatrix} 0 & -7 & -4 & -9 \\ -14 & 0 & -10 & -23 \\ -4 & -3 & 0 & -13 \\ -13 & -13 & -10 & 0 \end{pmatrix}$.

Each column is linearly independent of the others, a property of part 3 in Theorem 2.1.5 since the cyclicity of Q is established by self-loops, so that all maximal circuits contain only a single node. There are thus at least four linearly independent eigenvectors of Q. However, all four yield the same limiting contours, as shown in Figure 2.5(a), which displays the shape of the two contours in the period-2 regime obtained; the contours satisfy $\lim_{k\to\infty} \mathbf{y}(k) = \lim_{k\to\infty} 24^{\otimes -k} \otimes \mathbf{x}(k)$. Initial vectors are the columns of $\hat{Q}^*$ and we denote column i by $\mathbf{v}_i$. We can nevertheless obtain those eigenvectors that yield different limiting contours, and an example is shown in Figure 2.5(b), where the initial vector is taken as the linear combination $(-13 \otimes \mathbf{v}_1) \oplus (-20 \otimes \mathbf{v}_2) \oplus (-16 \otimes \mathbf{v}_3) \oplus (-18 \otimes \mathbf{v}_4)$, which is $(-13, -20, -16, -18)^\top$. There is a subtle difference in shape, i.e. gradient, between the contours in Figure 2.5(a) and the contours in Figure 2.5(b). In Figure 2.5(a), the limiting contours may be expressed as the periodic orbit of vectors $\{(14, 0, 10, 1)^\top, (10, 4, 13, 4)^\top\}$ whereas the limiting contours in Figure 2.5(a) are $\{(13, 0, 10, 0)^\top, (10, 3, 13, 3)^\top\}$.



(a)

(b)

Figure 2.5: Limiting contour plots of a max-plus system on a regular 3-nbhd network of 4 nodes. (a) Limiting contour plots of four systems with initial vector taken as the four eigenvectors of the max-plus system, where each eigenvector is a column of $\hat{Q}^*$. (b) Limiting contour plot obtained by taking initial vector $(-13 \otimes \mathbf{v}_1) \oplus (-20 \otimes \mathbf{v}_2) \oplus (-16 \otimes \mathbf{v}_3) \oplus (-18 \otimes \mathbf{v}_4)$.

Thus, there is a possibility to obtain more than one non-trivial regime of period $\rho > 1$. The significance of this is that a resultant information transfer system is asymptotically not unique, and dependent on the initial state $\mathbf{x}(0)$; this can affect the time $x_i(k)$ of update k at node i *relative* to $x_j(k)$ (at node j), even though the cycletime (which concerns *consecutive* update times $\mathbf{x}(k)$ and $\mathbf{x}(k+1)$) is independent of $\mathbf{x}(0)$.

We conclude by emphasising the contour plot as a means of visualising the evolution of the max-plus system. Importantly, as will be shown in the rest of the thesis, this contour plot representation is the premise for evolving and analysing cellular automata in max-plus asynchronous time since CA are usually represented as similar space-time plots.

2.2 Min-plus algebra

In this section, we consider the same regular n-nbhd network of cells as in Section 2.1 but the update rule is now altered. Here, cell i updates its CA state for the $(k+1)^{\text{th}}$ time after receiving the *first* input from its neighbourhood cells. This corresponds to waiting for the minimum of all neighbourhood input times on the kth cycle. If $n = 3$, we have the following scheme for the update time of cell i.

$$x_i(k+1) = \min\{x_{i-1}(k)+\tau_{i,i-1}(k), x_i(k)+\tau_{ii}(k), x_{i+1}(k)+\tau_{i,i+1}(k)\}+\xi_i(k+1). \quad (2.14)$$

The min operation will now yield *min-plus algebra*. We will show that this algebra shares fundamental similarities with max-plus algebra; the foremost of these is that a traditionally nonlinear system is interpreted as a linear system in the new algebra. Although there are common properties, the next sections outline the preliminaries and subtleties of the min operation as opposed to a max operation in our system.

2.2.1 Definitions

Define $\varepsilon' = +\infty$ and $e = 0$, and denote by $\mathbb{R}_{\min}$ the set $\mathbb{R}\bigcup\{\varepsilon'\}$. For elements $a, b \in \mathbb{R}_{\min}$, define operations $\ominus$ and $\odot$ by

$$a \ominus b = \min(a, b) \qquad \text{and} \qquad a \odot b = a + b.$$

The set $\mathbb{R}_{\min}$ together with the operations $\ominus$ and $\odot$ is what we refer to as *min-plus algebra* and is denoted by

$$\mathcal{R}_{\min} = (\mathbb{R}_{\min}, \ominus, \odot, \varepsilon', e)$$

where $\varepsilon' = +\infty$ is the 'zero', i.e. $\forall x \in \mathbb{R}_{\min}$, $\varepsilon' \odot x = x \odot \varepsilon' = \varepsilon'$ and $\varepsilon' \ominus x = x \ominus \varepsilon' = x$. $e = 0$ is the min-plus 'unit' element, i.e. $\forall x \in \mathbb{R}_{\min}$, $e \odot x = x \odot e = x$.

Min-plus algebra is obtained from max-plus algebra using the following transform.

Max to min-plus algebra transform Let $x, y \in \mathbb{R}_{\max}$. Then

$$x \ominus y = -(-x \oplus -y).$$

> Similarly, $x \oplus y = -(-x \ominus -y)$. We refer to each of these two ways to transform a sum in $\mathcal{R}_{\max}$ to one in $\mathcal{R}_{\min}$ (and vice versa) as a *min-plus transform* (resp. max-plus transform) and denote them as $\mathcal{R}_{\max} \to \mathcal{R}_{\min}$ (resp. $\mathcal{R}_{\min} \to \mathcal{R}_{\max}$).

Thus, properties that were shown for max-plus algebra are also present in min-plus algebra, and can be checked by simply employing the min-plus transform.

Note that e and $\odot$ are defined exactly as e and $\otimes$ are defined respectively for max-plus algebra. The reason for representing $\otimes$ as $\odot$ for $\mathcal{R}_{\min}$ is mainly for helping to identify the algebra that we work with. Parts of the remaining section may likewise be restatements of the corresponding parts in $\mathcal{R}_{\max}$ but stated in the interests of clarity and completeness.

2.2.2 Min-plus algebra in our system

Recall the vector $\mathbf{x}(k) = (x_1(k), x_2(k), \ldots, x_N(k))^\top$ that denotes the state of the system. Using (2.14), the evolution equation of the system in the min-plus model can be given in matrix-vector form as follows.

$$\mathbf{x}(k+1) = P(k+1) \odot \mathbf{x}(k)$$

where $P(k+1)$ may be written as the product $A_\xi(k+1) \odot T(k)$, where $A_\xi(k+1)$ is the processing matrix given as

$$A_\xi(k+1) = \begin{pmatrix} \xi_1(k+1) & \varepsilon' & \cdots & \varepsilon' \\ \varepsilon' & \xi_2(k+1) & \cdots & \varepsilon' \\ \vdots & & \ddots & \vdots \\ \varepsilon' & \varepsilon' & \cdots & \xi_N(k+1) \end{pmatrix}$$

and $T(k)$ is the transmission matrix given as

$$\begin{pmatrix} \tau_{11}(k) & \tau_{12}(k) & \varepsilon' & \varepsilon' & \cdots & \varepsilon' & \tau_{1N}(k) \\ \tau_{21}(k) & \tau_{22}(k) & \tau_{23}(k) & \varepsilon' & \cdots & \varepsilon' & \varepsilon' \\ \vdots & & & \ddots & & & \vdots \\ \varepsilon' & \varepsilon' & \cdots & \varepsilon' & \tau_{N-1,N-2}(k) & \tau_{N-1,N-1}(k) & \tau_{N-1,N}(k) \\ \tau_{N,1}(k) & \varepsilon' & \cdots & \varepsilon' & \varepsilon' & \tau_{N,N-1}(k) & \tau_{NN}(k) \end{pmatrix}$$

Note that these are the same matrices as in the max-plus case; the 'zero' element now being ε' as opposed to ε. As in the previous section, taking $\tau_{ij}(k) = \tau_{ij}$ implies $T(k) = T$, fixed in time. We also let $A_\xi(k+1) = A_\xi$, i.e. processing times $\xi_i(k)$ are constant in time. Then the evolution equation becomes

$$\mathbf{x}(k+1) = A_\xi \odot T \odot \mathbf{x}(k)$$

and, given the initial set of times $\mathbf{x}(0)$, we can find $\mathbf{x}(k)$ for all $k > 0$ from

$$\mathbf{x}(k) = P^{\odot k} \odot \mathbf{x}(0). \tag{2.15}$$

If we were to compare the stage we are in with the corresponding position in the previous section on max-plus algebra, we would see that it is only the notation that is the difference. The suggestion is that results for max-plus algebra give analogous results in min-plus algebra, and we investigate this next.

2.2.3 Application of the min-plus transform

The matrix P is a *weighted adjacency matrix over* $\mathbb{R}_{\min}$ where ε' signifies no arc connecting the corresponding nodes. To any $n \times n$ matrix A over $\mathbb{R}_{\min}$ we can associate a graph, called the *communication graph* of A, denoted $\mathcal{G}(A)$. Notice that, accounting for the obvious exception of ε', P is defined to be the same as that in max-plus algebra. This helps to yield the following lemma.

Lemma 2.2.1. Let $\mathcal{G}$ be a graph on N nodes. Then we can associate weighted adjacency matrices $A_{\max}(\mathcal{G})$ and $A_{\min}(\mathcal{G})$ in the max-plus and min-plus algebra respectively. Both matrices are the same except for those elements corresponding to no arcs, which are represented by ε in $\mathcal{R}_{\max}$ and ε' in $\mathcal{R}_{\min}$.

Therefore, to any graph, we can assign a max-plus system as well as a min-plus system. Further, properties of strong connectedness and irreducibility are unaltered in min-plus algebra.

We have looked at the max-plus system already. What the above means for the min-plus system is that any asymptotic behaviour, e.g. the cycletime, can be found from the *same* graph as the one we looked at in the max-plus system. This obviously comes to good use when looking at max-plus and min-plus systems over our network, which is now identified as the communication graph of P in both algebras.

The min-plus transform shows the ease by which a min-plus system may be obtained from the aforementioned same graph. Practically speaking, given the timing dependency matrix $P \in \mathbb{R}_{\max}^{N\times N}$, a min-plus system over the graph $\mathcal{G}(P)$ is equivalent to the max-plus system over the graph $\mathcal{G}(-P)$, where all entries in $-P$ are the negative of those in P. This should then be followed by carrying out the required max-plus calculations and taking the negative of the final values. Referring back to Lemma 2.2.1, we can now think of $A_{\min}(\mathcal{G})$ as the min-plus transform of $A_{\max}(\mathcal{G})$.

Thus, along with the necessary manipulation of negatives, the results in $\mathcal{R}_{\max}$ as shown in Section 2.1 apply to $\mathcal{R}_{\min}$ too. We illustrate this in the following sections, which are selected to complement the corresponding topics in Section 2.1.

The eigenspace in $\mathcal{R}_{\min}$

Define the eigenvalue and eigenvector in min-plus algebra as follows.

Definition 2.2.1. Let $A \in \mathbb{R}_{\min}^{N\times N}$. If $\lambda \in \mathbb{R}_{\min}$ is a scalar and $\mathbf{v} \in \mathbb{R}_{\min}^{N}$ is a vector that contains at least one finite element such that

$$A \odot \mathbf{v} = \lambda \odot \mathbf{v},$$

then λ is an *eigenvalue* of A and $\mathbf{v}$ is an *eigenvector* of A associated with eigenvalue λ.

Take the max-plus matrix $P_{\max} = \begin{pmatrix} \varepsilon & 2 & 1 \\ 1 & \varepsilon & 4 \\ 3 & 3 & \varepsilon \end{pmatrix}$, whose communication graph $\mathcal{G}(P_{\max})$ is shown in Figure 2.6. P is obviously irreducible with max-plus eigenvalue 3.5 corresponding to the 2-circuit $\{23, 32\}$ (or cyclicity 2). Given the same graph, let us now carry out the min-plus transform on these quantities: First, take $-P_{\max} = \begin{pmatrix} \varepsilon & -2 & -1 \\ -1 & \varepsilon & -4 \\ -3 & -3 & \varepsilon \end{pmatrix}$. Then $\mathcal{G}(-P_{\max})$ is the same as that in Figure 2.6 except that all arcweights have been multiplied by -1. Since only the signs on the arcweights have changed, the matrix $-P$ is also irreducible but with eigenvalue -1.5 (corresponding to the 2-circuit $\{12, 21\}$. To complete the min-plus transform, we take the final step of taking the negative of the eigenvalue, i.e. the min-plus adjacency matrix P_{min} will have eigenvalue 1.5 (with cyclicity 2, as found).

It is not too difficult to see that this corresponds to the minimal average circuit weight in the original strongly connected graph in Figure 2.6. In fact, working in $\mathcal{R}_{\text{min}}$ doesn't require the two transform steps carried out above. Rather, we simply look for such a 'minimum equivalent' to the max-plus algebraic method of working out the eigenvalue. Formally, we state this in the following theorem.

Theorem 2.2.1. Let $P_{\text{min}} \in \mathbb{R}_{\text{min}}^{N \times N}$ be irreducible. Then P_{min} possesses a unique eigenvalue $\lambda(P_{\text{min}})$, which is finite ($\neq \varepsilon'$). This eigenvalue is equal to the minimal average weight of elementary circuits in $\mathcal{G}(P_{\text{min}})$.

Proof. Apply the max-plus transform on P_{min} to obtain an irreducible matrix in max-plus algebra. Then apply Theorem 2.1.1. □

We apply the min-plus transform on the Kleene star of a matrix $A \in \mathbb{R}_{\text{max}}^{N \times N}$ to define the Kleene star $A^{\odot *}$ of the corresponding matrix A in $\mathcal{R}_{\text{min}}$:

$$A^{\odot *} \overset{\text{def}}{=} \ominus_{k=0}^{\infty} A^{\odot k}.$$

Thus, the eigenspace of A_{min} may be calculated by applying the min-plus transform to Theorem 2.1.5. Note that $\mathcal{G}^{cr}(A)$ where $A \in \mathbb{R}_{\text{min}}^{N \times N}$ is now defined as those circuits in $\mathcal{G}(A)$ having minimal average circuit weight.

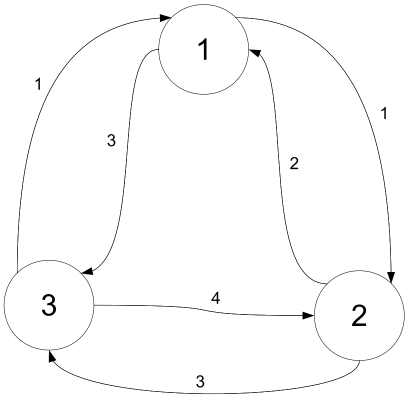


Figure 2.6: $\mathcal{G}(P_{\text{max}})$

Asymptotic behaviour in $\mathcal{R}_{\min}$

We consider the recurrence relation in Equation (2.15) and think about the asymptotic behaviour of the update times $\mathbf{x}(k)$. Define the periodic regime in $\mathcal{R}_{\min}$ as follows.

Definition 2.2.2. Let $P \in \mathbb{R}_{\min}^{N \times N}$. For some $k \geq 0$, consider the set of vectors

$$\mathbf{x}(k), \mathbf{x}(k+1), \mathbf{x}(k+2), \ldots \in \mathbb{R}_{\min}^{N}$$

where $\mathbf{x}(n) = P^{\otimes n} \odot \mathbf{x}(0)$ for all $n \geq 0$. The set is called a *periodic regime* if there exists $\mu \in \mathbb{R}_{\min}$ and a finite number $\rho \in \mathbb{N}$ such that

$$\mathbf{x}(k+\rho) = \mu \odot \mathbf{x}(k).$$

The *period* of the regime is ρ and μ/ρ is the *cycletime*.

This cycletime may also be recovered for each individual node i as the asymptotic limit $\lim_{k\to\infty} \frac{x_i(k)}{k}$ and this can be verified using the same methods as those in Section 2.1.

We define the *cyclicity* of irreducible P as the cyclicity of the critical graph of P. Denote this cyclicity as $\sigma(P)$ or simply σ where the context is understood.

Again, the min-plus transform implies that all results obtained for max-plus algebra in the previous section also apply for min-plus algebra over the same timing dependency graph. In particular, we highlight the following corollary to Theorem 2.1.2.

Corollary 2.2.1. Let $P \in \mathbb{R}_{\min}^{N \times N}$ be an irreducible matrix with eigenvalue λ and cyclicity $\sigma(P)$. Then there is a t such that

$$P^{\odot(k+\sigma(P))} = \lambda^{\odot\sigma(P)} \odot P^{\odot k}$$

for all $k \geq t$.

We follow this result with the implication that, for the recurrence relation $\mathbf{x}(k+1) = P \odot \mathbf{x}(k)$ with $k \geq 0$ and irreducible $P \in \mathbb{R}_{\min}^{N \times N}$ having eigenvalue $\lambda \in \mathbb{R}$, the cycletime of each node i is λ, i.e. for $i = 1, 2, 3, \ldots, N$,

$$\lim_{k\to\infty} \frac{x_i(k)}{k} = \lambda$$

for any initial condition $\mathbf{x}(0) \in \mathbb{R}_{\min}^{N}$.

Thus, a min-plus system over the regular n-nbhd network yields asymptotic behaviour that is governed by the minimal average circuit weight in the network.

2.3 Summary

We have shown that the max-plus system and min-plus system over the regular n-nbhd network of cells share fundamental similarities. The advantage of this is that it is easy to switch from max-plus algebra to min-plus algebra and to deduce asymptotic behaviour. In particular, a contour plot in both systems may be obtained with relative ease even though the two systems represent the two 'extremes' of our asynchronous timing model. The min-plus model corresponds to update after the first input, therefore $m = 1$ and, indeed, it is expected to yield the smallest cycletime in comparison to other values of m, the largest cycletime being obtained for $m = n$, i.e. the max-plus system. Intermediate systems, i.e. those whereby each process updates its state after receiving m out of n inputs, but for which $1 < m < n$, will be explored in later chapters.

Chapter 3

Cellular Automata in Max-plus Time

This chapter demonstrates the effect of a max-plus update time model on an information exchange system such as a cellular automaton. We will show that the max-plus form for asynchrony does indeed yield CA behaviour that looks asynchronous in our traditional view of time. However, what also emerges is that such CA can, in fact, be mapped to the synchronous CA due to a bijection between the two. In Section 3.1, we outline a formalism for the implementation of max-plus algebra that would be suited for simulation of CA. This includes the application of the contour plot as well as conditions that need to be imposed. The theory of this section is assumed in the remainder of the chapter. In Section 3.2, we study the max-plus time model as applied to the regular n-nbhd network on N nodes. Since N (and subsequently n) is large and parameter combinations are vast, the analytical study is assisted with numerical calculations. Section 3.3 gives some quantitative characterisations of simulating cellular automata in max-plus time. Finally, Section 3.4 concludes the chapter by linking Sections 3.2 and 3.3. The main result is that, no matter how asynchronous the update times of each cell becomes, we can still predict the class of CA output since the max-plus model is a bijection from the synchronous case.

3.1 Outline: CA in max-plus time

In this section, we present the first formalism for implementing a cellular automaton asynchronously where update times are determined by a max-plus system. The issues of real time and the partial ordering of events come into play. We deal with each under broader topics in relevant subsections.

3.1.1 Contour plot as a foundation for CA

Recall the N-dimensional max-plus system given as

$$\mathbf{x}(k+1) = P \otimes \mathbf{x}(k) \tag{3.1}$$

where $P \in \mathbb{R}_{\max}^{N\times N}$ is expressed as the max-plus matrix multiplication of the processing matrix A_ξ and the transmission matrix T. The orbit of an initial state $\mathbf{x}(0)$ is the sequence $\{\mathbf{x}(0), \mathbf{x}(1), \mathbf{x}(2), \ldots\}$ arising from (3.1). Such a sequence is graphically depicted in space-time by the contour plot, and one of the advantages of this is the visual identification of periodic behaviour in update times $\mathbf{x}(k)$. In what follows, we couple the contour plot with the CA space-time plot and demonstrate how we update both the times and the CA states simultaneously.

Let $s_i(k)$ denote the CA state of node i at epoch k. We are concerned with Boolean CA states, so that $s_i(k) \in \{0, 1\}$. The unit k is the same unit that is used in the max-plus model which updates the times $\mathbf{x}(k)$. Thus, to be precise, $s_i(k)$ is the CA state of node i at time $x_i(k)$. The *CA state of the system*, i.e. of all nodes, is then represented by the $N \times 1$ vector $\mathbf{s}(k) = (s_1(k), \ldots, s_N(k))$ (or string $s_1(k)s_2(k)\cdots s_N(k)$), which can also be read as the state of all nodes on contour k. As a consequence, just as we represented the vector $\mathbf{x}(k)$ by a contour, we can represent the CA state $\mathbf{s}(k)$ as the same contour but with the addition that the coordinates $(i, x_i(k))$ now display the state $s_i(k)$ (e.g. in coloured form, where two different colours are used to distinguish the two states 1 and 0).

Think back to the previous chapter, where we briefly introduced the main events that are internal to node i, i.e. that occur within cycle k. These events are grouped in two: "receive" and "send". Once the "receive" CA states have all arrived, node i applies a CA rule on this set, to obtain its new state $s_i(k)$. (This calculation of the new state takes processing time ξ_i). All nodes have neighbourhood size n, so the applied CA rule (function) is $f : \{0, 1\}^n \to \{0, 1\}$ and the new state $s_i(k)$ is calculated as

$$s_i(k) = f(\mathcal{N}(s_i(k-1)).$$

So, new CA states are calculated for every contour; in the time between contours, the nodes retain the CA state held on the last contour. We show next how to represent and study this concept.

3.1.2 CA space-time plot

The classical one-dimensional CA is synchronous, so that the kth update time of each cell is the same. Consequently, we can think of such a system as having a contour plot that contains only horizontal contours. Updates of the CA state of the system take place every one time unit, thereby giving the synchronous CA a cycletime of 1. The time between contours in this system is thus of duration one, although no such duration is depicted; for example, if $s_i(k) = 1$ for all k, then this is shown as a continuous vertical coloured block in position i. The CA evolution itself is represented as a space-time plot as in Figure 1.1, where the underlying network of cells is regular and $n = 3$.

So, how do we construct the space-time plot for CA in max-plus time where varying contour shapes dictate the varying time gaps between contours? To make this topic clear, we consider the example of a regular 3-nbhd network with size $N = 10$. Let the positive (diagonal) entries in matrix A_ξ be represented by the vector of processing times ξ; we call it the *processing time distribution.* For the purposes of simulating the example, we choose the entries in ξ at random with equal probability from all integers between 1 and 30, whilst the non-zero entries in T are selected likewise from the integers between 1 and 10. Taking the initial time $\mathbf{x}(0) = \mathbf{u}$, we obtain a contour plot of update times by iterating the max-plus system. We now address the CA state of the nodes by assigning the depicted space between contours as memory: for each node, the CA state remains fixed until the time of update, which corresponds to a contour. The length of time that elapses between contours implies that the storing of the CA state in memory can thus be represented as a vertical block of that length (which is coloured accordingly, depending on the value of the CA state).

It is important to distinguish between the variables t and k. From now on, the term "time" (or "real time") will always refer to a point $t \in \mathbb{R}^+$; it can be thought of as time as we know it. k maintains its role as an epoch. We now have that node i carries a CA state for every point $t \in \mathbb{R}^+$ in real time. Correspondingly, this may be depicted in a space-time plot, the construction of which is shown in three stages in Figure 3.1.

Note 3.1.1. Denote the state of node i at real time t as $s_i^{(t)}$. We now have two ways of understanding the CA state of node i: $s_i(k)$ denotes the CA state on contour k, and it is discretely dynamic, whilst $s_i^{(t)}$ represents the state in a dynamical system with a continuous underlying real time t.

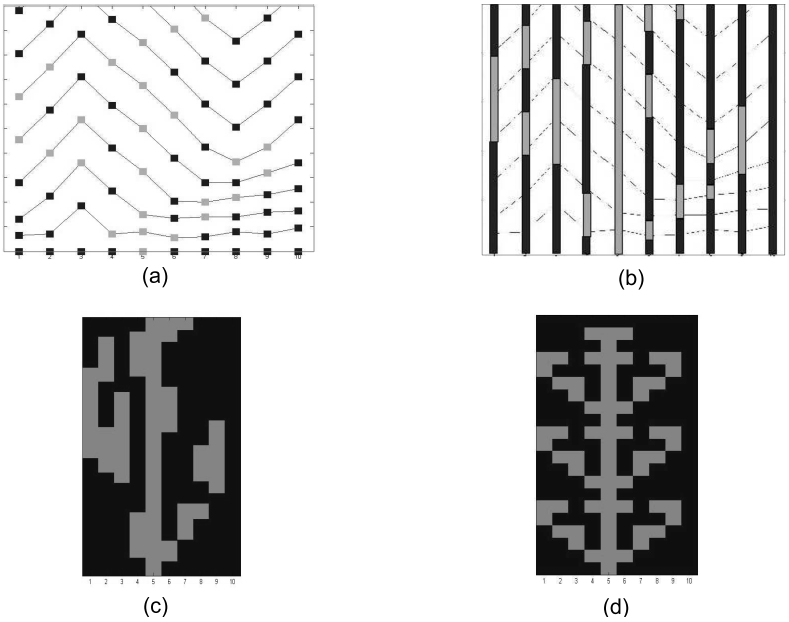


Figure 3.1: Construction of the CA space-time plot in max-plus time. The CA rule is ECA rule 150. The initial CA state on contour 0 is $s_5(0) = 1$, all other nodes are 0. State 0 is coloured black, state 1 is grey. In all figures, the vertical axis denotes real time, travelling up. (a) Contour plot with CA states indicated on each contour. (b) CA states indicated for all time by filling spaces between contours with memory. (c) Contours and space between nodes removed to obtain the CA space-time plot. (d) Classical (synchronous) CA space-time plot.

Figure 3.1(d) is for comparison with Figure 3.1(c) and it shows the classical synchronous CA having the same initial time $\mathbf{x}(0)$, initial CA state $\mathbf{s}(0)$ and CA rule. Whilst the initial conditions and the rule f are the same, for each point t in real time, the state $\mathbf{s}^{(t)}$ in both patterns will generally differ. (This can be seen by simply drawing a horizontal line across both patterns at time t and reading off the state of each node at that time). The difference in pattern is obviously ascribed to the asynchronous form of the max-plus system in Figure 3.1(c). Notwithstanding this, we show next that we can characterise this difference somewhat and, in fact, map the synchronous CA to the max-plus CA via the contour plot.

3.1.3 Bijection

The state transition graph was introduced in Chapter 1. In Figure 3.1(d), the period is 6 and the periodic orbit is the following set.

$$\{0000100000, 0001110000, 0010101000, 0110101100, 1000100010, 1101110110\}.$$

If we consider the CA states on only the contours in Figure 3.1(c) (which is seen better in Figure 3.1(a)), we see that they are exactly the same as the states in Figure 3.1(d). This is a consequence of the max-plus model, outlined in Section 3.1.1, which waits for all neighbourhood states to arrive before processing the new state for each node. The model uses the same CA rule, applied to the same neighbourhoods, the only difference being that the time of application of the rule is different now. Thus, there is a mapping of the CA from the synchronous system to the max-plus system; the max-plus system need not evolve the CA concurrently since the CA plot for the max-plus system may be obtained from this mapping. The mapping is a bijection of CA states on contours, demonstrated as follows.

Given the same initial CA state $\mathbf{s}(0)$ and CA rule, let $\mathcal{S}$ and $\mathcal{M}$ denote the orbit of $\mathbf{s}(0)$ generated in the synchronous system and the max-plus system respectively. Let $\mathbf{s}_{\mathcal{S}}(k)$ denote the CA state after k iterations of the synchronous system; $\mathbf{s}_{\mathcal{M}}(k)$ denotes the CA state after k iterations of the max-plus system. Then, after k iterations of both systems, we clearly have $\mathbf{s}_{\mathcal{S}}(k) = \mathbf{s}_{\mathcal{M}}(k)$. This defines a one-to-one and onto mapping between $\mathcal{S}$ and $\mathcal{M}$, i.e. a bijection, and we say that both systems have the same state transition graph (defined as the transitions between states on contours). Thus, the parameters ξ_i and τ_{ij} have no influence on the CA except in real time t. The extent and impact of this real time effect will be explored in Section 3.3.

In summary, the STG provides a deterministic form for predicting the behaviour of the CA in max-plus time, although each state in the STG does not necessarily correspond to a state in real time t. In fact, the STG in max-plus time may not even be totally ordered when observed in real time. We show this next.

3.1.4 Contour crossing

It is a conventional assumption that real time is monotonic. In particular, the events that occur at one process are assumed to be totally ordered. For a reasonable model to simulate CA, we would thus require that, for each node, the time that its state updates is later than the previous update time. Put mathematically, the following condition needs to hold true for all nodes i.

$$j < k \Rightarrow x_i(j) < x_i(k).$$

Contrary to standard assumptions, it turns out that the max-plus system is not guaranteed this property, which is that of monotonicity for all k.

Recall the statement of monotonicity in Chapter 2. For the max-plus system to be monotonic for all k, it suffices to ensure that $\mathbf{x}(0) < \mathbf{x}(1)$. Remembering that the comparisons are made componentwise, $\mathbf{x}(0) < \mathbf{x}(1)$ means $x_i(0) < x_i(1)$ for all nodes i.

In general, the max-plus iteration scheme for node i is

$$x_i(k+1) = \bigoplus \mathcal{N}_i(k) \otimes \xi_i$$

where $\mathcal{N}_i(k)$ is the set of input times from the neighbourhood nodes of node i and $\bigoplus$ denotes the sum of all elements in $\mathcal{N}_i(k)$. For example, in the regular 3-nbhd network, $\mathcal{N}_i(k) = \{\tau_{i,i-1} \otimes x_{i-1}(k), \tau_{i,i} \otimes x_i(k), \tau_{i,i+1} \otimes x_{i+1}(k)\}$. Thus, for $x_i(0) < x_i(1)$ to be true for all i, it must hold that

$$x_i(0) < x_i(1) = \bigoplus \mathcal{N}_i(0) \otimes \xi_i. \tag{3.2}$$

In the regular network, $\mathcal{N}_i(0)$ contains $\tau_{ii} \otimes x_i(0)$; there is a self-loop attached to node i in the underlying connectivity graph. Therefore, $x_i(0) < \tau_{ii} \otimes x_i(0)$, so that $x_i(0) < \bigoplus \mathcal{N}_i(0)$. Hence, (3.2) is true for the max-plus system on a regular n-nbhd network.

In a general network with arbitrary connectivity, the existence of a self-loop is not guaranteed. In that case, (3.2) must be enforced. Thus, either $x_i(0) \leq \bigoplus \mathcal{N}_i(0)$ or $x_i(0) \leq \xi_i$ (or both) must be true. (We assume that both the transmission times and processing times are positive). However, once $x_i(0)$ is chosen to satisfy $x_i(0) \leq \bigoplus \mathcal{N}_i(0)$, then each member of the neighbourhood $\mathcal{N}_i(0)$ must be chosen in a likewise fashion, leading to a recursively difficult problem of solving a system of linear inequalities. It is easier then to fix the inequality $x_i(0) \leq \xi_i$, which requires a simple choice based on the processing time of node i itself. As will be shown later, in our numerical experiments, we took $x_i(0) = e = 0$ for all i, which indeed satisfies this since all processing times ξ_i are positive.

Let us address the first paragraph of this section: Under what conditions would monotonicity (for all k) not be satisfied? Certainly, one condition that may lead to this is if $x_i(0) > \xi_i$ (and/or if $x_i(0) > \bigoplus \mathcal{N}_i(0)$) as discussed above. Consider the following example.

Example 3.1.1. Let the network and neighbourhood size be $N = 3$ and $n = 2$, with processing and transmission matrices given respectively as

$$A_\xi = \begin{pmatrix} 1 & \varepsilon & \varepsilon \\ \varepsilon & 5 & \varepsilon \\ \varepsilon & \varepsilon & 1 \end{pmatrix}, \quad T = \begin{pmatrix} 7 & 8 & \varepsilon \\ 9 & \varepsilon & 4 \\ 4 & 9 & \varepsilon \end{pmatrix}.$$

Take the initial state $\mathbf{x}(0) = (8, 4, 20)^\top$. Then the first few states in the sequence of update times are

$$\mathbf{x}(0) = (8, 4, 20)^\top, \mathbf{x}(1) = (16, 29, 14)^\top, \mathbf{x}(2) = (38, 30, 39)^\top.$$

Notice that $x_3(0) > x_3(1)$, so monotonicity is not satisfied at node 3. Also, $x_3(1) = 14 = \bigoplus \mathcal{N}_3(0) \otimes 1$ so that $x_3(0)$ is greater than both $\bigoplus \mathcal{N}_3(0)$ and its processing time ξ_3.

Thus, we can deduce that if monotonicity is not satisfied, then $x_i(0)$ is greater than *both* $\bigoplus \mathcal{N}_i(0)$ and ξ_i. (However, the converse need not be true). The contour plot of the system in Example 3.1.1 is drawn in Figure 3.2. It can be seen that the initial contour $\mathbf{x}(0)$ intersects with $\mathbf{x}(1)$ as a result of node 3 not being monotonic. When referring to this in terms of the contour plot, we shall call it a *contour crossing.* The essence of contour crossing is that a future event (on epoch k) affects a past event (on epoch $k' < k$). As a consequence, at least in real time, our notion of causality breaks down. In particular, contour crossing contradicts Definition 1.2.1 in Chapter 1, the condition that an event at node i in real time precedes a later event at the same node. Thus, to simulate a discrete dynamical system such as CA, which should viably model an application that is based on such real time, we would ideally like to preclude contour crossing.

Nevertheless, it would be interesting to study the effect of contour crossing on both the max-plus system and the CA. The first question asks whether the asymptotic behaviour of the update times is affected as a result. Consider Example 3.1.1 again. With the same initial condition, the system yields a periodic regime having period $\rho = 2$ and cycletime $\chi = 11.5$ with transient time $K = 4$. The contour crossing at the start does not persist, and we notice that ρ is exactly the cyclicity of the timing dependency matrix $P = A_\xi \otimes T$. In fact, we can take another initial condition such that there is no contour crossing yet the cycletime is still the same, e.g. $\mathbf{x}(0) = (4, 2, 4)^\top$ yields the monotonic evolution of times $\mathbf{x}(1) = (12, 18, 12)^\top$, $\mathbf{x}(2) = (27, 26, 28)^\top, \ldots$. In other words, even if contour crossing arises, the max-plus system still produces a unique asymptotic behaviour for all initial conditions. This seems obvious in light of Lemma 2.1.1, which says that the cycletime is independent of the initial condition due to the timing dependency graph being strongly connected. However, the current example is significant because it says that the transient to a periodic regime can be so complex as to even allow the seemingly contradictory behaviour of contour crossing; yet the system can still give rise to the same performance measure.

Further, the contour crossing need not be confined to the transient region. Consider the monotonicity property again, which says that once two contours are monotonic, i.e. once one occurs earlier than the other, then the max-plus system remains monotonic. This makes it impossible for contour crossing to occur once two successive contours have been found to be monotonic. However, it does not rule out the possibility of a persistent contour crossing, as shown in the following example.

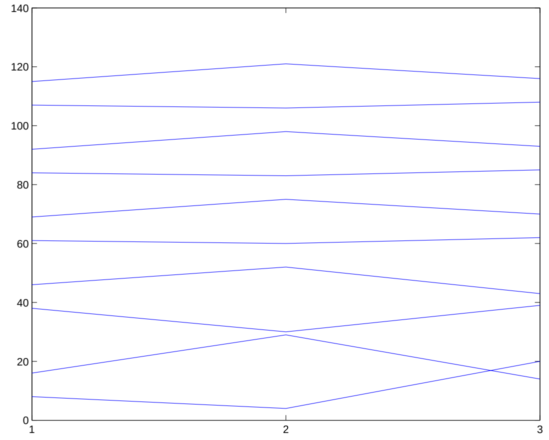


Figure 3.2: Contour plot of a max-plus system with network size $N = 3$.

Example 3.1.2. Take

$$A_\xi = \begin{pmatrix} 1 & \varepsilon & \varepsilon \\ \varepsilon & 0.5 & \varepsilon \\ \varepsilon & \varepsilon & 1 \end{pmatrix} \text{ and } T = \begin{pmatrix} \varepsilon & \varepsilon & 3 \\ 0.5 & 3.5 & \varepsilon \\ 3 & 1 & \varepsilon \end{pmatrix}.$$

Then $P = A_\xi \otimes T = \begin{pmatrix} \varepsilon & \varepsilon & 4 \\ 1 & 4 & \varepsilon \\ 4 & 2 & \varepsilon \end{pmatrix}$. The contour plot obtained with the initial contour $\mathbf{x}(0) = (5, 2, 0)^\top$ is shown in Figure 3.3. Notice that contour crossing is continuous in this evolution. However, it still yields a cycletime $\chi = 4$ and period $\rho = 2$. Another initial condition $\mathbf{x}(0) = (1, 4, 3)^\top$ yields a totally monotonic contour plot with the same cycletime and period.

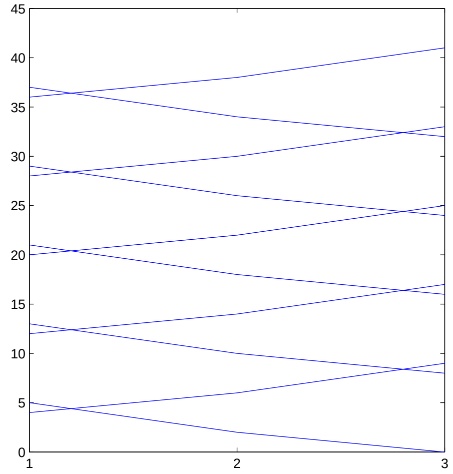


Figure 3.3: Contour plot of a max-plus system of size $N = 3$ and arbitrary connectivity. The plot shows a continuous contour crossing.

We finish this section with the realisation that, contrary to the causal set nature of real time, contour crossing still allows the modelling of CA. This is because max-plus CA is based in ‘contour time’, i.e. the counter k, which proceeds monotonically, and for which a STG may be drawn easily. If, however, the corresponding update times fail to be monotonic, then the CA would have to be implemented such that CA states can be sent ‘back in time’, i.e. real time; those CA states then form the input to other nodes, as in the original CA model, but a real time STG is impossible to draw in this case. Notwithstanding this, we shall avoid contour crossing in our experiments by imposing constraints on the initial condition.

3.1.5 Simplifying assumptions for the max-plus model

In general, the timing dependency matrix of the max-plus system is written as the product $P = A_\xi \otimes T$, where A_ξ is the processing matrix and T is the transmission matrix. We find that this general case is equivalent to employing the condition $\xi_i = \xi$ for all i, which we call the *processing condition.* The reason for this is as follows. The positive entries of A_ξ are all on the diagonal; they are the processing times of each node. Under the processing condition, these non-zero entries are all the same, so that A_ξ can be written as $\xi \otimes E$, where E is the max-plus identity matrix. Thus,

$$P = \xi \otimes E \otimes T = \xi \otimes T.$$

P is then a multiple of ξ, so that the entry P_{ij} is given as the product $= \xi_i \otimes \tau_{ij}$. There are $N \times n$ non-zero entries in P. Likewise, under the processing condition, there would be $N \times n$ non-zero entries in T, hence P. In that case, the max-plus system is iterated through max-plus powers of matrix T and then by simply scaling the resulting entries by ξ.

The transmission condition

We will gain some familiarity with the max-plus system by fixing transmission times to be the same no matter what the successor node, i.e. let τ_{ji} denote the transmission time from node i to some node j. Then the transmission condition fixes τ_{ji} as

$\tau_{ji} = \tau_i$ for all nodes j downstream of i.

Why would such a transmission condition be imposed? Let us take as the simplest example, the case of a two node network, where both nodes communicate with each other and themselves. If there are no restrictions on each τ_{ij}, then the timing dependency matrix P is of the form

$$\begin{pmatrix} \xi_1 \otimes \tau_{11} & \xi_1 \otimes \tau_{12} \\ \xi_2 \otimes \tau_{21} & \xi_2 \otimes \tau_{22} \end{pmatrix}.$$

The network itself is represented by $\mathcal{G}(P)$ in Figure 3.4. Since P is irreducible, the cycletime of this system will be the largest of the average of all elementary circuit weights; there are three here, with average weights $\xi_1\tau_{11}, \xi_2\tau_{22}$ and $\frac{1}{2}(\xi_1\tau_{12} + \xi_2\tau_{21})$. The latter of these weights corresponds to cyclicity $\sigma(P) = 2$, while the others yield cyclicity one. To go from cyclicity one to cyclicity two simply by varying the parameters $\xi_i\tau_{ij}$ can be regarded as a *bifurcation* in the system. Following from our criterion of the period of a regime being a measure of complexity, it is evident here that a small change in $\xi_i\tau_{ij}$ has the potential to double the complexity since the cyclicity is an upper bound of the period. This complexity is exacerbated by the discrete nature of cyclicity, i.e. there is no smooth change in cyclicity; this really is the essence of a bifurcation. So, in terms of the long term behaviour, this smallest of networks can give three possible outcomes. Each of these outcomes can be grouped in two, where the behaviour of one group will be significantly different to the other.

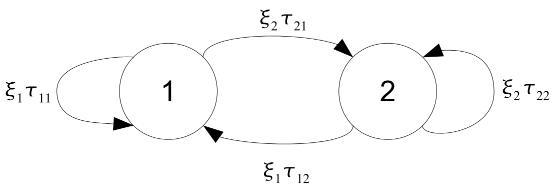


Figure 3.4: Timing dependency graph $\mathcal{G}(P)$ for a two-node network

Now consider a regular 3-nbhd network with $N = 3$. It can be shown that there are three types of elementary circuit, i.e. length-1, length-2 and length-3, the number of each being three, three and two respectively. This gives a total of eight elementary circuits (hence eight types of behaviour). For larger networks, the number of possibilities for cyclic behaviour are also large, and inevitably become harder to enumerate. If we were to characterise complexity in terms of cyclicity, then we could say that complexity scales with N. A transmission condition is thus a step towards restricting the types of behaviour that may be encountered.

Consider the extreme case where $\tau_{ji} = e = 0$ for all i, j. We call this the *zero transmission condition.* Then, the parameters are the processing times ξ_i, of which there are N. As a result, it may be checked that the matrix multiplication $A_\xi \otimes T$ yields the following timing dependency matrix P, which is given for the regular 3-nbhd network.

$$\begin{pmatrix} \xi_1 & \xi_1 & \varepsilon & \varepsilon & \cdots & \varepsilon & \xi_1 \\ \xi_2 & \xi_2 & \xi_2 & \varepsilon & \cdots & \varepsilon & \varepsilon \\ \vdots & & & \ddots & & & \vdots \\ \varepsilon & \varepsilon & \cdots & \varepsilon & \xi_{N-1} & \xi_{N-1} & \xi_{N-1} \\ \xi_N & \varepsilon & \cdots & \varepsilon & \varepsilon & \xi_N & \xi_N \end{pmatrix}. \tag{3.3}$$

In the discussion that follows, we show that the main advantage of the transmission is one of numerics. Whenever we refer to P under the zero transmission condition (as in (3.3)), we shall denote it by Z.

Instead of the transmission condition, we could also employ the processing condition. As can be deduced from earlier on in this section, we note that the processing condition can equivalently be studied by looking at the particular case where $\xi_i = 0$, which we call the *zero processing condition.* This gives $A_\xi = E$, so that $P = T$. So, the parameters of interest are τ_{ij}, of which there are at least $n \times N$ (depending on size of n-nbhd). Notice the difference between Z and T. For n fixed, both matrices would contain the same number of non-zero entries. However, there may be as many as n distinct non-zero entries on each row of T, whereas the n non-zero entries on row i of Z are all ξ_i. Thus, the potential number of distinct parameters in T is $n \times N$, which is obviously more than the N distinct parameters in Z. If the neighbourhood size n is greater than one, then studying T becomes a numerical problem of order greater than the problem of studying Z.

Let us show this by simply raising T and Z to the second power. Assume the most connected case where $n = N$, so that Z and T do not have any elements equal to ε. Then each element of $Z^{\otimes 2}$ is written

$$\begin{aligned}
[Z \otimes Z]_{ij} &= \bigoplus_{k=1}^{N} Z_{ik} \otimes Z_{kj} \\
&= \bigoplus_{k=1}^{N} \xi_i \otimes \xi_k \\
&= \xi_i \otimes \bigoplus_{k=1}^{N} \xi_k .
\end{aligned}$$

To obtain the max, we make $N - 1$ comparisons and, added to the max-plus multiplication of ξ_i, this gives $1 + N - 1 = N$ operations to obtain one element of $Z^{\otimes 2}$. Since $Z \in \mathbb{R}_{\max}^{N \times N}$, the number of operations to work out $Z^{\otimes 2}$ is $N \times N^2 = N^3$.

Now, each element of $T^{\otimes 2}$ is

$$\begin{aligned}
[T \otimes T]_{ij} &= \bigoplus_{k=1}^{N} T_{ik} \otimes T_{kj} \\
&= \bigoplus_{k=1}^{N} \tau_{ik} \otimes \tau_{kj} .
\end{aligned}$$

Since $T \in \mathbb{R}_{\max}^{N \times N}$, there are N such multiplications as $\tau_{ik} \otimes \tau_{kj}$. To obtain their max, we make $N - 1$ comparisons so that one element of $T^{\otimes 2}$ requires $N + N - 1 = 2N - 1$ operations. Therefore, the full $N \times N$ matrix T requires $N^2 \times (2N - 1) = 2N^3 - N^2$ operations. Thus, to obtain the second power of T, it takes $(2N^3 - N^2) - N^3 = N^3 - N^2$

more operations than Z, a significant difference as N becomes large.

We can deduce that, under the processing condition, the max-plus system (which generally requires the generation of powers of T larger than two) is numerically a much larger problem. In fact, we notice that employing the processing condition would be equivalent to the general max-plus system where both parameters ξ_i and τ_{ij} are non-zero and without constraint. The study of this is what we wish to progress towards and is allocated for a later section. In summary, the main advantage of the transmission condition is that it will sharply reduce the size of the numerical problem, especially for a large network size N. Further, it is hoped that this will also act as a foundation for a better understanding of the max-plus system.

3.1.6 An example

We complete Section 3.1 with a small example to illustrate the significance of this chapter. Consider the network size $N = 4$ with arbitrarily chosen connections. For convenience, we consider the network to be that in Figure 1.2. Let the zero transmission condition hold and consider the vector of processing times $\xi = (1, 1, 1, 1)^\top$, which we shall call the *processing time distribution*. This choice of ξ corresponds to a classical synchronous system like the random Boolean networks and ECA. Take the initial time $\mathbf{x}(0) = \mathbf{u}$ and the initial CA state $\mathbf{s}(0) = (0, 1, 0, 0)$. The max-plus system on this network yields period $\rho = 1$ and cycletime vector $\chi = (1, 1, 1, 1)$ after transient time $K = 0$. As for the CA, ECA rule 150 leads to a periodic orbit with CA period $p = 3$ after the CA transient time $K_C = 1$. Note that p and K_C can also be determined by examining the STG of this system in Chapter 1. We will be concerned with the entropies to classify the CA space-time pattern produced. Thus, after 100 (a large number of) iterations, we obtain $S = 0.9174$ and $W = 1.0244$. The CA space-time plot for this system is below the contour plot in Figure 3.5(a).

Now, by altering ξ, the CA pattern will be altered in real time, though the CA period and CA transient time will be the same. Take $\xi = (5, 4, 3, 1)$. Then, under the same initial conditions and number of iterations, this gives $K = 1$, $\rho = 2$ and cycletime vector $\chi = (5, 3.5, 3.5, 5)$. Note that χ is not uniform because the timing dependency matrix P is reducible here. The CA entropies are $S = 0.9161$ and $W = 1.3713$. The contour plot and CA plot for this processing time distribution is in Figure 3.5(b). As discussed in Chapter 1, taking a large number of iterations allows the entropies to settle into a steady state.

The difference between the Shannon entropies above is 0.0013. We would like to characterise whether or not this is a significant difference. This will be shown later,

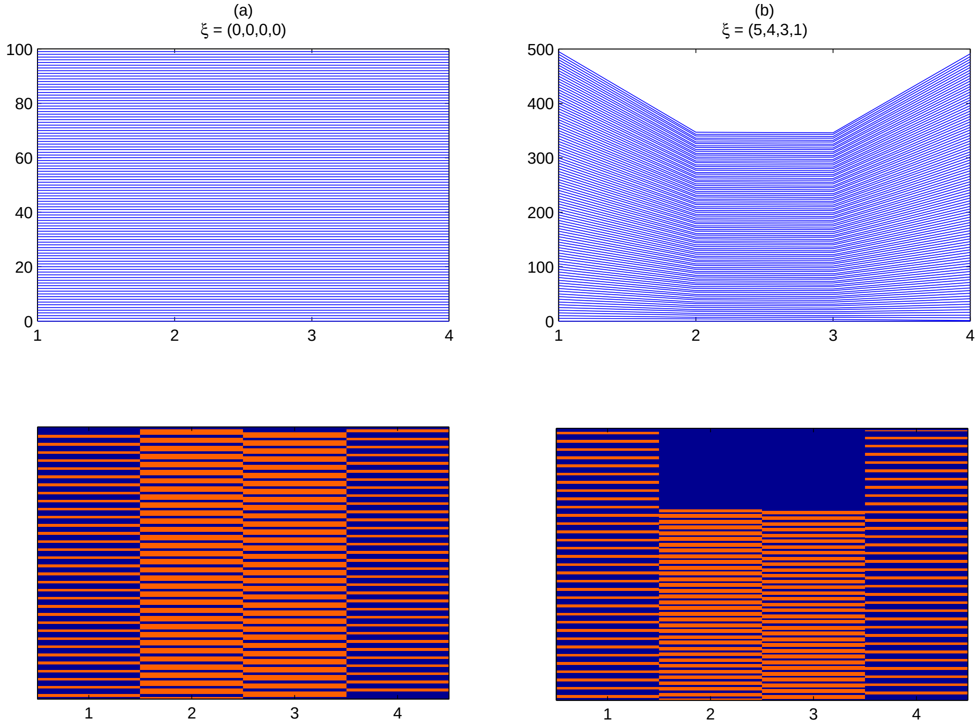


Figure 3.5: Contour plot (above) and CA space-time plot (below) for a 4 node system with arbitrary neighbourhoods. The initial contour is $\mathbf{x}(0) = \mathbf{u}$ and the initial CA state takes $s_2(0) = 1$, $s_i(0) = 0$ for $i \neq 2$. The output is obtained after 100 iterations under the zero transmission condition where (a) $\xi = (1, 1, 1, 1)$ and (b) $\xi = (5, 4, 3, 1)$

and more thoroughly due to a larger set of data. For now, we say that the change in processing time distribution did not significantly alter the temporal Shannon entropy S. Notwithstanding this, different processing time distributions may yield vastly differing entropies. However, it is anticipated that, at least in terms of the Shannon entropy, the CA output will not be too sensitive to processing time. This is because S depends on the occurrence of individual cell states only, each of which occur in almost the same amount due to the bijection between synchronous CA and CA in max-plus time. On the other hand, it is seen that a larger cycletime vector, coupled with a larger period, leads to a larger temporal word entropy W, and this leads to thinking about the effect of ξ on W. We will examine this in more scope after the next section, in which we lay the foundations for the CA in max-plus time.

3.2 Max-plus algebra on the n-nbhd network

In this section, we study the unique type of asynchrony that the max-plus system provides on the regular n-nbhd network, which has classically been used as the lattice for past one-dimensional CA work. Some results will also apply to a general n-nbhd network (that need not be regular), although we give particular focus to the regular network size $N = 20$, on which a cellular automaton will later be studied. The results that we obtain here will prove to be suggestive of the CA work that follows.

3.2.1 The regular n-nbhd network

Recall that Wolfram's ECA was simulated over a regular 3-nbhd network. In this section, we shall see that the regular n-nbhd network allows a natural progression towards the modelling of CA on larger neighbourhood sizes than Wolfram's ECA neighbourhoods. This study should also importantly reveal insight into the relationship between max-plus update times, connectivity of the network and the resulting CA.

Consider the timing dependency matrix P of the regular n-nbhd network, where $P = A_\xi \otimes T$. We have already seen that such a matrix is irreducible since $\mathcal{G}(P)$ is strongly connected. Given an initial state $\mathbf{x}(0)$, the system $\mathbf{x}(k+1) = A_\xi \otimes T \otimes \mathbf{x}(k)$ is represented by $\mathbf{x}(k) = (A_\xi \otimes T)^{\otimes k} \otimes \mathbf{x}(0)$. Thus, by Lemma 2.1.1, the cycletime of this system is the eigenvalue of $A_\xi \otimes T$. The next lemma will give the value of this cycletime under the transmission condition. First, consider the matrix $\bar{P} \in \mathbb{R}_{\max}^{N \times N}$

given by

$$\begin{pmatrix} \xi_1\tau_{11} & \xi_1\tau_{12} & \cdots & \xi_1\tau_{1N} \\ \xi_2\tau_{21} & \xi_2\tau_{22} & \cdots & \xi_2\tau_{2N} \\ \vdots & & & \vdots \\ \xi_N\tau_{N1} & \xi_N\tau_{N2} & \cdots & \xi_N\tau_{NN} \end{pmatrix}.$$

The construction of the regular n-nbhd network implies that the neighbourhood size n is odd (due to the self-loop at each node). Therefore, node i will have $(n-1)/2$ nodes symmetrically connected either side of it, i.e. $(n-1)/2$ neighbours to the right and the same number to the left of i. We can thus obtain P from $\bar{P}$ by removing from each row i, all elements except for $\bar{P}_{ii}$ and the $(n-1)/2$ elements to the right and left of it; the removed elements are replaced by ε. For clarity, let us represent P schematically as the $N \times N$ matrix in Figure 3.6. We can think of P as in Figure 3.6 as being lifted from the surface of a cylinder, where the diagonal non-zero entries forms a continuous diagonal strip on the cylinder. The exact form of P is given in the following lemma.

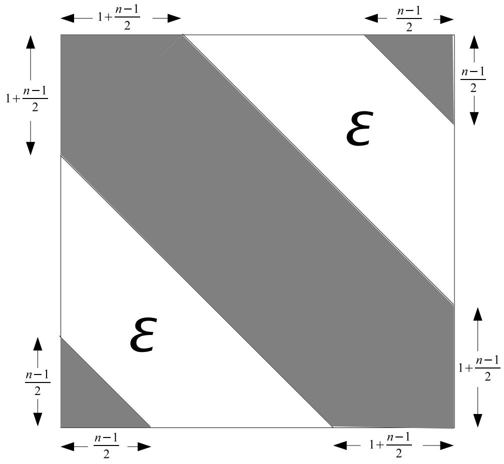


Figure 3.6: Schematic representation of matrix $P \in \mathbb{R}_{\max}^{N\times N}$. The shaded areas contain non-zero entries and zero entries are marked by the white areas labelled ε. Sizes of these areas are as indicated

Lemma 3.2.1. Consider the timing dependency matrix P of the regular n-nbhd network of N cells, where n is odd. Under the transmission condition, the cycletime of P is $\chi = \max_{1\leq i\leq N}\{\xi_i + \tau_i\}$, which is the maximum weight of all self-loops in $\mathcal{G}(P)$.

Proof. Under the transmission condition, P is given by the following matrix, where

$\mathcal{E}_m$ denotes the $1 \times m$ constant vector of ε.

$$\begin{pmatrix} \xi_1\tau_1 & \cdots & \xi_1\tau_{1+(n-1)/2} & \mathcal{E}_{N-n} & \xi_1\tau_{N+1-(n-1)/2} & \cdots & \xi_1\tau_N \\ & \ddots & & & & & \\ \mathcal{E}_{N-n-m} & \xi_i\tau_{i-(n-1)/2} & \cdots & \xi_i\tau_i & \cdots & \xi_i\tau_{i+(n-1)/2} & \mathcal{E}_m \\ & & & & \ddots & & \\ \xi_N\tau_1 & \cdots & \xi_N\tau_{(n-1)/2} & \mathcal{E}_{N-n} & \xi_N\tau_{N-(n-1)/2} & \cdots & \xi_1\tau_N \end{pmatrix}.$$

Let $\overline{\xi\tau} \overset{\text{def}}{=} \max_{1\leq i\leq N} \xi_i + \tau_i$. Then, the following inequality is satisfied for all i, $1 \leq i \leq N$.

$$\overline{\xi\tau} \geq \xi_i + \tau_i \tag{3.4}$$

Matrix P is irreducible. By Theorem 2.1.1 and Lemma 2.1.1, the cycletime of the irreducible matrix P is the maximal average weight of an elementary circuit in $\mathcal{G}(P)$. Figure 3.7 shows part of a 5-nbhd example of $\mathcal{G}(P)$. We show that the average weight of all elementary circuits can never be greater than $\overline{\xi\tau}$. The figure helps to deduce the types of elementary circuits that exist in a general n-nbhd network. For a network size N, there are N lengths of elementary circuit. All length 1 elementary circuits are self-loops, the self-loop at node i taking weight $\xi_i + \tau_i$. Therefore, by definition, $\overline{\xi\tau}$ is greater than or equal to all length 1 elementary circuits. Now suppose that nodes i and $i+2$ are connected in a length 2 elementary circuit. The circuit will have average weight $\frac{1}{2}(\xi_i+\tau_{i+2}+\xi_{i+2}+\tau_i)$. In general, for $1 \leq l \leq N$, it can be verified that a length l elementary circuit will take average weight $\frac{1}{l}((\xi_{i_1}+\tau_{i_2})+(\xi_{i_2}+\tau_{i_3})+\cdots+(\xi_{i_l}+\tau_{i_1}))$

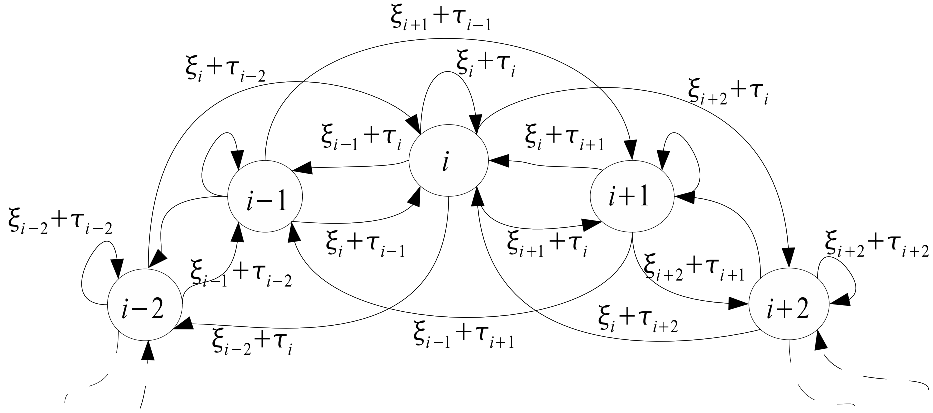


Figure 3.7: 5-nbhd example of $\mathcal{G}(P)$ together with arc weights. For visual clarity, some arcs and weights are omitted. However, the neighbourhood of i is complete.

From the inequality (3.4), we have that, for any pair of nodes i and j,

$$2\overline{\xi\tau} \geq (\xi_i + \tau_i) + (\xi_j + \tau_j) = (\xi_i + \tau_j) + (\xi_j + \tau_i)$$

Thus,

$$\overline{\xi\tau} \geq \frac{1}{2}((\xi_i + \tau_j) + (\xi_j + \tau_i)).$$

Since the right hand side of the above equation is the form of the average weight of any length 2 elementary circuit, $\overline{\xi\tau}$ is greater than or equal to the average weight of all length 2 elementary circuits.

Similarly, for any number l ($1 \leq l \leq N$), we can use the inequality (3.4) to obtain l such inequalities. Adding these l inequalities yields

$$\overline{\xi\tau} \geq \frac{1}{l}((\xi_{i_1} + \tau_{i_2}) + (\xi_{i_2} + \tau_{i_3}) + \cdots + (\xi_{i_l} + \tau_{i_1}))$$

for any l nodes $i_1, \cdots, i_l$. The right hand side of this inequality is the form of the average weight of any length l elementary circuit. Thus, $\overline{\xi\tau}$ is greater than or equal to the average weight of all elementary circuits. □

It can now be deduced that, under the zero transmission condition, the cycletime is $\chi(P) = \max_i\{\xi_i\}$. This is a neat result, indicating that the update time of all nodes in the max-plus system is governed by the largest processing time. A useful numerical implication is that it requires only one calculation.

Observe also that the cycletime $\max_{1\leq i\leq N}\{\xi_i + \tau_i\}$ is the maximum of all self-loop weights in $\mathcal{G}(P)$. Thus, implied by the proof of the lemma is the fact that any length N circuit (or N-*circuit*) has, attached to it, a self-loop whose weight is larger than or equal to the average weight of the N-circuit. This further suggests that the cyclicity of $P = A_\xi \otimes T$ is one (since this is the length of a self-loop). Let us formally state this in the following corollary to Lemma 3.2.1.

Corollary 3.2.1. Under the hypothesis of Lemma 3.2.1, the cyclicity of P is 1.

Proof. Consider $\mathcal{G}^{cr}(P)$, the critical graph of P. By Lemma 3.2.1, it contains those circuits whose average weight is equal to the largest of all self-loop weights. The proof of Lemma 3.2.1 also implies that any circuit c contained within the regular network has an average weight no greater than a self-loop that is attached to (a node in) c. Thus, if c is contained in $\mathcal{G}^{cr}(P)$, then so is a self-loop that is attached to c. The nodes in c (including all arcs attached to it) form a maximal strongly connected subgraph. Let the subgraph take size l and call it MSCS$_l$.

We follow by induction. Due to the self-loop, the cyclicity σ_1 of MSCS_1 is 1. For the inductive hypothesis, assume the cyclicity of MSCS_L is 1, where $L > 1$. Consider $\mathrm{MSCS}_{(L+1)}$. Then it contains circuits, hence maximal strongly connected subgraphs, of sizes 1 to L, each having cyclicity 1 due to the induction hypothesis. Now, since $\mathrm{MSCS}_{(L+1)}$ is strongly connected, it contains a self-loop, which is contained in MSCS_L. Thus, due the strong connectivity, the cyclicity of $\mathrm{MSCS}_{(L+1)}$ is 1. So we have proved by induction that, for $l = 1, \ldots, N$, the cyclicity σ_l of MSCS_l is 1.

Now let $\mathcal{G}^{cr}(P)$ contain p such maximal strongly connected subgraphs. It is sufficient to consider these subgraphs being disjoint. Then the cyclicity of P is the lowest common multiple of their cyclicities $\sigma_1, \sigma_2, \ldots, \sigma_p$, i.e. $\mathrm{LCM}(1, 1, \ldots, 1) = 1$. □

By taking cyclicity as a measure for the complexity of the system, we notice then that this regular max-plus system (constrained by the transmission condition) minimises the complexity; one could argue that complexity is absent if cyclicity was the only criterion.

To summarise this section, we have seen that use of a regular n-nbhd network simplifies the asymptotic behaviour of the max-plus system. The transmission condition adds to this simplification by bounding the cyclicity for such a system.

3.2.2 Cyclicity and transient time of the max-plus system

This section discusses the cyclicity and transient time of the max-plus asynchronous system for general networks, thus providing results that can be applied to the regular n-nbhd network.

Cyclicity

As indicated earlier, the transmission condition leads to simple bounds on the cyclicity, hence to a reduced complexity. In general, the cyclicity has the following upper bound due to the finiteness of a network.

Theorem 3.2.1. Consider the timing dependency matrix $P \in \mathbb{R}_{\max}^{N \times N}$ of a network of N cells which is composed of l maximal strongly connected subgraphs. Let the cyclicity of P be denoted by σ. Then $\sigma \leq e^{\zeta}$, where $\zeta = \sum_{i=1}^{l} \sqrt{N_i \ln N_i}$ and N_i denotes the number of cells in the ith MSCS.

Proof. Consider the connectivity graph $\mathcal{G}(P)$. It can either be strongly connected or not strongly connected, but it is sufficient to consider it being not strongly connected and having l maximal strongly connected subgraphs, which we label MSCSi ($i =$

$1, \ldots, l$). Let the adjacency matrix of MSCSi be denoted M_i and let σ_i denote the cyclicity of M_i. Then σ is the least common multiple (LCM) of $\{\sigma_1, \ldots, \sigma_l\}$.

Now let N_i be the number of nodes in MSCSi. MSCSi is strongly connected, so M_i is irreducible and its cyclicity will be the cyclicity of the critical graph of MSCSi, denoted $\mathcal{G}_i^{cr}$. The critical graph itself may not be strongly connected. If M_{i_j} denotes the adjacency matrix of the j^{th} disjoint maximal strongly connected subgraph in MSCSi, then let σ_{i_j} denote the cyclicity of M_{i_j}. Here, it is sufficient to consider a base case, where each MSCS in $\mathcal{G}_i^{cr}$ is one and only one elementary circuit. Such circuits contain no more than N_i nodes. We take the LCM of all σ_{i_j} to obtain a bound for σ_i, which follows from noting that any set of such cyclicities σ_{i_j} will partition N_i. For example, if $N_i = 4$, then the elementary circuits in $\mathcal{G}_i^{cr}$ (each of which correspond to σ_{i_j}) may have the set of lengths $\{1, 1, 2\}$. Thus, we employ Landau's function $g(N_i)$, which gives the largest LCM of all partitions of N_i [42]. We have

$$\sigma_i \leq g(N_i) \tag{3.5}$$

where Landau's function satisfies

$$\lim_{N_i \to \infty} \frac{\ln g(N_i)}{\sqrt{N_i \ln N_i}} = 1.$$

We rearrange this equation to obtain $\ln(g(N_i)) = \sqrt{N_i \ln N_i}$ as $N_i \to \infty$. This limit is reached from below [42], meaning that, in fact, $\ln(g(N_i)) \leq \sqrt{N_i \ln N_i}$. As a result, we obtain

$$g(N_i) \leq e^{\sqrt{N_i \ln N_i}}. \tag{3.6}$$

The following is true from (3.5).

$$\begin{aligned} \sigma &= \text{LCM}(\sigma_1 \sigma_2 \cdots \sigma_l) \\ &\leq \text{LCM}(g(N_1) g(N_2) \cdots g(N_l)) \\ &\leq g(N_1) g(N_2) \cdots g(N_l). \end{aligned}$$

Applying (3.6) yields

$$\begin{aligned} \sigma &\leq e^{\sqrt{N_1 \ln N_1}} e^{\sqrt{N_2 \ln N_2}} \cdots e^{\sqrt{N_l \ln N_l}} \\ &= e^{\sum_{i=1}^{l} \sqrt{N_i \ln N_i}}. \end{aligned}$$

□

Notice that the proof of Theorem 3.2.1 allowed P to be reducible. As a result, Theorem 3.2.1 applies to any form of matrix P. However, for our asynchronous model, we also require that P is regular, i.e. that all nodes in $\mathcal{G}(P)$ have at least one input (which corresponds to each row in P having at least one element not equal to ε). If P is not regular, then this means that at least one node doesn't have any input; we call these nodes *source* nodes and they can provide input to other nodes yet never take input themselves. In that case, the corresponding rows in P become redundant and implies a reduced form for P, where those rows and corresponding columns are omitted. However, Theorem 3.2.1 is still applicable, but the system would be split: Let $\tilde{P} \in \mathbb{R}_{\max}^{(N-r)\times(N-r)}$ denote the reduced form obtained by eliminating the r redundant rows in $P \in \mathbb{R}_{\max}^{N\times N}$, along with the corresponding columns. Then the new max-plus system is the map $\mathbb{R}_{\max}^{N-r} \mapsto \mathbb{R}_{\max}^{N-r}$, defined by $\tilde{\mathbf{x}}(k+1) = \tilde{P} \otimes \tilde{\mathbf{x}}(k)$, where the update times $x_{q_1}(k), x_{q_2}(k), \ldots, x_{q_r}(k)$ of all source nodes $q_1, q_2, \ldots, q_r$ are eliminated from consideration since they would not be updated and imposed to remain fixed (at $x_{q_i}(0)$ for $i = 1, \ldots, r$). Thus, this upper bound for cyclicity is, in fact, applicable to any matrix P, hence any network of asynchronous processes. This result, therefore, proves to be useful for other types of networks, even those that contain source nodes.

Transient time

Another criterion for complexity is the transient time of a system. Theorem 2.1.4 noted that the transient time K of a max-plus system is bounded by $t(P)$, the transient time of P. The exact value of $K = K(\mathbf{x}(0))$ is dependent on the initial condition $\mathbf{x}(0)$, and may take any integer value from 0 to $t(P)$. In general, if P is a large matrix, then this range of values is large and it becomes a relatively harder problem to find K. In fact, Bouillard and Gaujal noticed that computing $t(P)$ is a NP-complete problem in the number of circuits in $\mathcal{G}(P)$ and consequently constructed an upper bound for $t(P)$ [4]. The upper bound involves an identification of the maximal strongly connected subgraphs of the network as well as critical graphs, so requires a number of computations that is exponentially increasing with N; specifically, the complexity is cubic in N.

Thus, we try to restrict $t(P)$ by applying conditions to the parameters or variables of our system. The parameters in our matrix P are integers, mainly for computational reasons; on MATLAB, the subsequent CA space-time plot is developed as a matrix, of which the columns and rows are integers. For linear systems with such integer parameters, the upper bound for the transient time is reduced. Even and Rajsbaum proved this upper bound to satisfy

$$K(\mathbf{x}(0)) \leq l_\star(\mathcal{G}(P)) + N + 2N^2 \tag{3.7}$$

where $l_\star$ denotes the upper bound on the length l of maximum weight paths that contain those nodes not in the critical graph of $\mathcal{G}(P)$ [9]. The next lemma details the effect on $t(P)$ for a further condition on the parameters, namely the transmission condition, where $\mathcal{G}(P)$ is the regular n-nbhd network; this subsequently provides a tighter upper bound for K.

Lemma 3.2.2. Consider the timing dependency matrix $P \in \mathbb{R}_{\max}^{N\times N}$ of the regular n-nbhd network of N cells. Let the transmission condition hold. Then, if $n = N$, the transient time of P is $t(P) = 1$.

Proof. For $n = N$ under the transmission condition, matrix P is

$$P = \begin{pmatrix} \xi_1\tau_1 & \xi_1\tau_2 & \cdots & \xi_1\tau_N \\ \xi_2\tau_1 & \xi_2\tau_2 & \cdots & \xi_2\tau_N \\ \vdots & & \ddots & \vdots \\ \xi_N\tau_1 & \xi_N\tau_2 & \cdots & \xi_N\tau_N \end{pmatrix}.$$

Due to Corollary 3.2.1, this max-plus system will have cyclicity $\sigma = 1$.

We take powers of P to find $t(P)$. Let $\overline{\xi\tau} = \bigoplus_{i=1}^N \xi_i\tau_i$ and notice that the $(i,j)^{\text{th}}$ element of $P^{\otimes 2}$ is given by

$$\begin{aligned} P_{ij}^{\otimes 2} &= \bigoplus_{k=1}^N P_{ik}P_{kj} \\ &= \bigoplus_{k=1}^N \xi_i\tau_k\xi_k\tau_j \\ &= \xi_i\tau_j \bigoplus_{k=1}^N \xi_k\tau_k \\ &= \xi_i\tau_j \otimes \overline{\xi\tau}. \end{aligned}$$

Using this, we obtain $P^{\otimes 3}$ as follows.

$$\begin{aligned}
P_{ij}^{\otimes 3} &= \bigoplus_{k=1}^{N} P_{ik}^{\otimes 2} P_{kj} \\
&= \bigoplus_{k=1}^{N} \xi_i \tau_k \otimes \overline{\xi\tau} \otimes \xi_k \tau_j \\
&= \overline{\xi\tau} \bigoplus_{k=1}^{N} \xi_i \tau_k \xi_k \tau_j \\
&= \overline{\xi\tau} \otimes \xi_i \tau_j \bigoplus_{k=1}^{N} \xi_k \tau_k \\
&= \overline{\xi\tau} \otimes \xi_i \tau_j \otimes \overline{\xi\tau} \\
&= \overline{\xi\tau}^{\otimes 2} \otimes \xi_i \tau_j
\end{aligned}$$

Therefore, $P^{\otimes 2} = \overline{\xi\tau} \otimes P$ and $P^{\otimes 3} = \overline{\xi\tau}^{\otimes 2} \otimes P$. Similarly, we can show that the following relation holds for $k \geq 1$.

$$P^{\otimes (k+1)} = \overline{\xi\tau} \otimes P^{\otimes k}, \tag{3.8}$$

where $\overline{\xi\tau}$ is just the eigenvalue (and cycletime) of P. Theorem 2.1.2 may now be applied to Equation (3.8) to deduce $t(P) = 1$, the least value of k for which the relation holds. □

Thus, the transient time of a periodic regime in the most connected max-plus system $\mathbf{x}(k+1) = P^{\otimes (k+1)} \otimes \mathbf{x}(0)$ satisfies $K(\mathbf{x}(0)) \leq 1$ for all initial conditions $\mathbf{x}(0)$. We refer to a system with such a transient time as being *minimally transient*. It might be argued that this should be a term used strictly when $K = 0$ for all initial states. However, in that case, it would follow that any arbitrary vector is contained in a periodic regime generated by P (therefore it is a contour), meaning that any arbitrary vector is then an eigenvector of P^{σ}. The case $K = 0$ for all initial states is, thus, impossible. Therefore, applying A to the initial vector at least once allows the system to enter a periodic regime.

Next, we detail the max-plus asymptotic results obtained computationally, where all neighbourhood sizes in the regular n-nbhd network are considered.

3.2.3 Asymptotic results for a large network

The asymptotic behaviour of the regular n-nbhd network was obtained numerically for network size $N = 20$. Initially, we aimed to study the effect of the processing time ξ_i on the max-plus system. Thus, we implemented the zero transmission condition. As a result, the computations were also faster.

Fix n. We choose $r \in \mathbb{N}$, defined as the ξ *radius*. This term is explained in the following algorithm which shows how we collated the required results.

Algorithm 3.2.1.

1. Choose ξ_i from the uniform distribution (with equal probability) of integers between 1 and r.
2. Taking $\mathbf{x}(0) = \mathbf{u} = (0, 0, \ldots, 0)^\top$, run the max-plus system to obtain the transient time K, period ρ and cycletime χ.
3. Carry out above two steps to obtain 500 such results for this value of r.
4. Record the mean of the 500 transient times, periods and cycletimes obtained.

Algorithm 3.2.1 was repeated for each integer r from 1 to 30. (Thus, for one n-nbhd, we obtained a total of $500 \times 30 = 15000$ results).

The results arising from Algorithm 3.2.1 form some approximation to the transient time $t(A)$ and cyclicity $\sigma = \sigma(P)$. Corollary 3.2.1 gave the cyclicity of P under the zero transmission condition as one, characterised by the self-loop(s) with maximal weight. Thus, all periods obtained from the algorithm are one. It might be argued that we could simply have studied powers of P itself and found exact values for $t(P)$ and σ. However, our interest goes further than this section, particularly when we implement a cellular automaton model obeying these max-plus update times. Such a CA model requires initialisation at some time $\mathbf{x}(0)$ and so employing the above method for identifying asymptotic behaviour acts as a foundation for the CA. We also look towards the next chapter, in which we attempt to find the transient time and cyclicity of a maxmin-m system where $1 < m < n$. It turns out that these quantities are almost impossible to obtain in such a system (it is not as straightforward as taking powers of the timing dependency matrix). Thus, the iterative method as above (that looks for a periodic regime given some $\mathbf{x}(0)$) is adopted for that system. It will then be made convenient to draw meaningful comparisons between results for maxmin-m systems for all m ($m = 1, 2, \ldots, n$).

Next, we discuss the impact of the ξ radius on the transient time as above and other quantities.

The likelihood of synchrony

The ξ radius r is regarded as a measure of displacement from 'synchrony' of the max-plus system. We may define a perturbation in our numerical work as the change in processing times by one unit. We might then also regard r as a "perturbation parameter".

In general, for $r \in \mathbb{N}$, each ξ_i may take r values, so there are r^N possible distributions of ξ. From these, the distributions $\xi = (1, 1, \ldots, 1)$, $\xi = (2, 2, \ldots, 2)$, ..., $\xi = (r, r, \ldots, r)$ will yield synchrony in both real time as well as the contour plot. In other words, given initial state $\mathbf{x}(0) = \mathbf{u}$, these r distributions will yield $K(\mathbf{x}(0)) = 0$ and a contour plot where all contours are horizontal.

Thus, for any r, given the ξ distributions are selected randomly with equal probability, the probability that a synchronous ξ distribution is selected is

$$L_s = \frac{r}{r^N} = r^{1-N}.$$

We call L_s the *likelihood of synchrony.* Figure 3.8 shows the graph of L_s as a function of r for a few values of N. It can be seen that, for large r, L_s settles into a steady state ("equilibrium"). In particular, $L_s \to 0$ as $r \to \infty$.

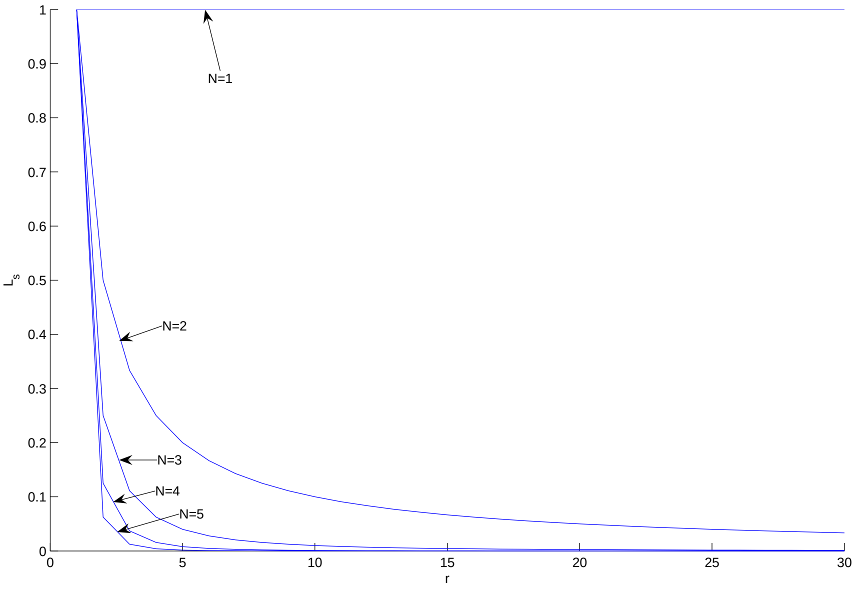


Figure 3.8: Likelihood of synchrony L_s for a few values of network size N

Figure 3.8 further demonstrates that the behaviour of L_s for small r is different to its steady-state behaviour for larger r. For small r, there is a rapid decrease in the likelihood of synchrony, which leads us to conjecture that variables such as the period

ρ and transient time K of a max-plus system will behave similarly. Therefore, we might expect a large change in period and/or transient time for smaller r, which then settles into an equilibrium state as r becomes larger. We expand on this as follows.

Under the zero transmission condition, input arrival times at a node depend asymptotically on ξ_i values. Therefore, it is the difference in ξ_i values that the max-plus system is looking for when ordering such arrival times and finding the maximum.

For small r, a ξ distribution is likely to contain a lot of ξ_i values that are the same, so that the resulting max-plus system has a small transient time. A perturbation in r yields a distribution in which more ξ_i values are now different; the effect is akin to 'kicking' a system to drastically change from synchrony to asynchrony, and the resulting system is, hence, more different. For larger r, a lot of ξ_i in a randomly chosen ξ distribution are already different, so a perturbation in r yields a system which doesn't differ significantly since the resulting system contains a similar number of differing ξ_i values. Thus, the mean transient time of a large number of randomised systems is an increasing function of r, which is expected to settle into some equilibrium value for $r \to \infty$.

Observe the transient time results obtained for each regular n-nbhd in Figure 3.9. Whilst there is an increase in transient time with r, it is not apparent whether mean transient times settle into an equilibrium state for larger r. This is more so the case for smaller n; for larger n, such an equilibrium state for the transient time is clearer. In fact, mean K and mean $t(P)$ are both almost constant for all r for larger neighbourhood sizes. We feel this is due to an increased connectivity that a larger n provides. We can relate this to previous work on transient time. Soto y Koelemeijer [38] provides the following upper bound on K, which we have rewritten according to our work.

$$K(\mathbf{x}(0)) \leq \max\left\{\frac{\|\mathbf{x}(0)\| + N(\Delta - \delta)}{\chi - \chi_1}, 2N^2\right\}$$

where Δ and δ are respectively the largest and smallest arcweights in $\mathcal{G}(P)$. χ_1 is the largest average circuit weight of the "non-critical" graph, which is the subgraph of $\mathcal{G}(P)$ that doesn't contain those nodes and arcs in the critical graph.

For us, $\mathbf{x}(0) = \mathbf{u}$, so $\|\mathbf{x}(0)\| = 0$, and it is likely that, for a given r, we obtain $\Delta = r$ and $\delta = 1$. Similarly, χ is likely to be r (this will be shown shortly). Therefore, $K(\mathbf{x}(0)) \leq \max\left\{\frac{N(r-1)}{r-\chi_1}, 2N^2\right\}$. We chose $N = 20$, which yields

$$K(\mathbf{x}(0)) \leq \max\left\{\frac{20(r-1)}{r-\chi_1}, 800\right\}.$$

We obtain $\frac{20(r-1)}{r-\chi_1} > 800$ when $\frac{r-1}{r-\chi_1} > 40$, which can be rearranged so that

$$\chi_1 > \frac{39r+1}{40}.$$

Table 3.1 compares r with $\frac{39r+1}{40}$ for the values of r used in our experiments, where $r_\star$ denotes $\frac{39r+1}{40}$. It can be seen that, for each r, $r_\star$ is very close to r. It should also be noted that the largest value that χ_1 can take is close but never equal to r; otherwise, the corresponding circuit is critical, which yields a contradiction. However, we estimate that χ_1 is not likely to be as large as $r_\star$. Therefore, Soto y Koelemeijer's upper bound is interpreted as

r	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$r_\star$	1	1.975	2.95	3.925	4.9	5.875	6.85	7.825	8.8	9.775	10.75	11.725	12.7	13.675	14.65
r	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
$r_\star$	15.625	16.6	17.575	18.55	19.525	20.5	21.475	22.45	23.425	24.4	25.375	26.35	27.325	28.3	29.375

Table 3.1: r versus $\frac{39r+1}{40}$ for each r value used for Algorithm 3.2.1

$$K \leq 800$$

for each r used in our experiments. Since 800 is a fixed upper bound on K, and K is generally increasing with r, we can deduce that the mean from our experiments is likely to slow down with r, thereby reaching the expected equilibrium value.

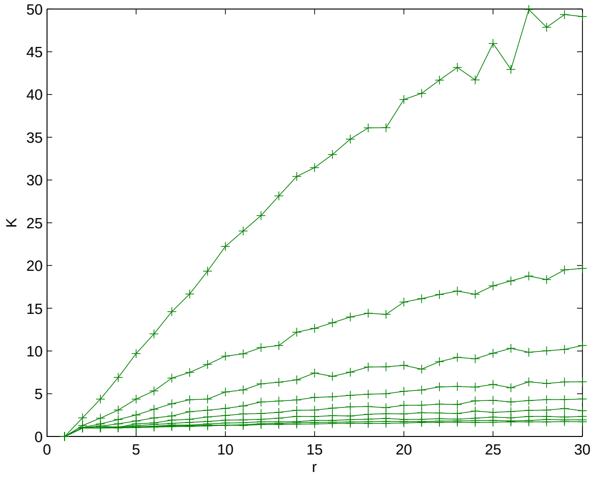


Figure 3.9: Mean transient K as a function of ξ radius r for all regular n-nbhd max-plus systems on 20 nodes. The curves become flatter as n increases. Therefore, for a fixed r, the coordinate (r, K) is smaller as n becomes larger.

The slowing of the rate of increase of K with r is supported by the results for $t(P)$, based on powers of P. This is shown in Figure 3.10, which shows mean $t(P)$

as in Figure 3.9 but now supported with the largest, smallest and modal $t(P)$ value obtained for each r. As a sample, the figure shows the curves for neighbourhood size $n = 7$. It can be seen that, as $r \to \infty$, the range between largest and smallest value of $t(P)$ increases, yet the mode is almost *fixed*. This 'drags' the mean value down, yielding the aforementioned equilibrium state.

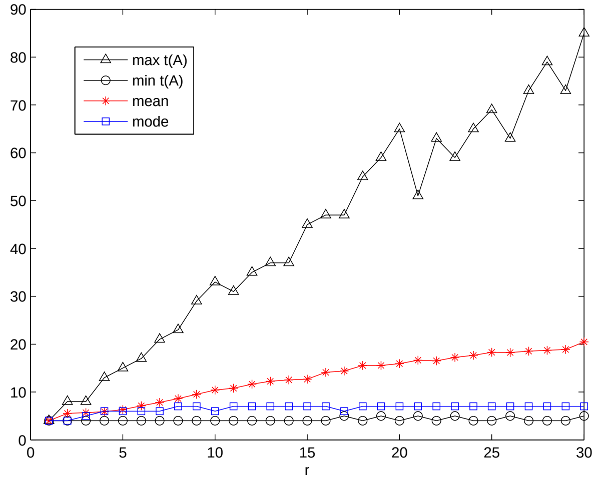


Figure 3.10: Mean, mode and range of transient time $t(P)$ as a function of ξ radius. The network is the regular 7-nbhd network.

To conclude the discussion of the ξ radius, the larger the r, the more asynchronous the max-plus system, but as r tends to infinity, the difference in level of asynchrony between corresponding systems becomes negligible; we could say that, for large r, all max-plus systems are equally asynchronous. The effect on transient time is its increase with r and subsequent slowing down of this increase, which may be manifested by an equilibrium state. The period is fixed at 1 for all r due to Corollary 3.2.1.

Cycletime

Lemma 3.2.1 gives the cycletime to be $\chi = \max_i\{\xi_i\}$ under the zero transmission condition, but what is the largest such processing time exactly? When each processing time is assigned randomly with equal probability (and upper bound r), this might yield a distribution of ξ that only contains small values, thereby yielding a small cycletime. So, for a large r, it is not necessarily the case that a large cycletime is obtained. Given r and network size N fixed, we now look at the probability distribution of cycletime χ.

The probability that the cycletime is c is written $\mathbf{P}[\lambda = c]$, where $c \in \mathbb{N}$ and $1 \leq c \leq r$. $\mathbf{P}[\lambda = c]$ is a function of c, so we denote it compactly as $p(c)$. Since the cycletime is the largest of all processing times, $p(c)$ is the probability that a processing time distribution $\xi \in \mathbb{N}^N$ is generated (randomly with equal probability) that has largest value $\xi_i = c$ and all other values less than or equal to c. Thus, we consider only those distributions of ξ in this discussion of cycletime.

The probability that one element in ξ is equal to c is simply $1/r$, whilst the probability that an element in ξ is not equal to c and also less than c is $(c-1)/r$. Then, the probability of there being exactly j elements in ξ that are equal to c while all other elements are less than c is

$$\binom{N}{j}\left(\frac{1}{r}\right)^j\left(\frac{c-1}{r}\right)^{N-j}$$

which is simplified to

$$\binom{N}{j}\frac{(c-1)^{N-j}}{r^N}. \tag{3.9}$$

$\binom{N}{j}$ is the binomial coefficient $\frac{N!}{j!(N-j)!}$. The required probability $p(c)$ is then the sum of all such probabilities as (3.9), i.e.

$$\begin{aligned} p(c) = \mathbf{P}[\lambda = c] &= \sum_{j=1}^{N}\binom{N}{j}\frac{(c-1)^{N-j}}{r^N} \\ &= \left(\frac{(c-1)}{r}\right)^N\sum_{j=1}^{N}\binom{N}{j}\frac{1}{(c-1)^j}. \end{aligned} \tag{3.10}$$

The sum in (3.10) is simplified to $\left(1+\frac{1}{c-1}\right)^N - 1$ due to a binomial expansion. Therefore,

$$\begin{aligned} p(c) &= \left(\frac{(c-1)}{r}\right)^N\left(\left(1+\frac{1}{c-1}\right)^N - 1\right) \\ &= \left(\frac{(c-1)}{r}\right)^N\left(\left(\frac{c}{c-1}\right)^N - 1\right) \\ &= \left(\frac{c}{r}\right)^N - \left(\frac{c-1}{r}\right)^N. \end{aligned}$$

The result is that $p(c)$ is an increasing function of c. This implies that, when processing times are generated randomly with equal probability, the most likely cycletime is r (since $c \leq r$).

The discrete probability distribution implies that the mean may be calculated as $\mathbf{E}[c] = \sum_{c=1}^{r} cp(c)$, which is

$$\begin{aligned}
\mathbf{E}[c] &= \sum_{c=1}^{r}\left\{c\left(\frac{c}{r}\right)^{N} - c\left(\frac{c-1}{r}\right)^{N}\right\} && (3.11)\\
&= \frac{1}{r^N}\sum_{c=1}^{r} c^{N+1} - \frac{1}{r^N}\sum_{c=1}^{r} c(c-1)^{N} && (3.12)\\
&= \frac{1}{r^N}\sum_{c=1}^{r} c^{N+1} - \frac{1}{r^N}\sum_{c=1}^{r}\{(c-1)^{N+1} + (c-1)^{N}\} && (3.13)\\
&= \frac{1}{r^N}\sum_{c=1}^{r} c^{N+1} - \frac{1}{r^N}\sum_{c=1}^{r}(c-1)^{N+1} - \frac{1}{r^N}\sum_{c=1}^{r}(c-1)^{N} && (3.14)\\
&= \frac{1}{r^N}\sum_{c=1}^{r} c^{N+1} - \frac{1}{r^N}\sum_{c=1}^{r-1} c^{N+1} - \frac{1}{r^N}\sum_{c=1}^{r-1} c^{N}. && (3.15)
\end{aligned}$$

After subtraction of the terms in the second sum from the first sum in (3.15), we obtain

$$\begin{aligned}
\mathbf{E}[c] &= \frac{1}{r^N} r^{N+1} - \frac{1}{r^N}\sum_{c=1}^{r-1} c^{N} && (3.16)\\
&= r - \sum_{c=1}^{r-1}\left(\frac{c}{r}\right)^{N} && (3.17)
\end{aligned}$$

The sum in (3.17) may be approximated by upper and lower bounds as follows. Denote the upper bound of the sum by UB. Then,

$$\begin{aligned}
UB &= \int_0^{r-1}\left(\frac{x+1}{r}\right)^{N} \mathrm{d}x\\
&= \frac{1}{r^N}\int_1^{r} u^N \mathrm{d}u\\
&= \frac{1}{r^N}\left(\frac{r^{N+1}}{N+1} - \frac{1}{N+1}\right) = \frac{r^{N+1}-1}{r^N(N+1)}.
\end{aligned}$$

Similarly, denote the lower bound of the sum by LB. Then,

$$\begin{aligned}
LB &= \int_0^{r-1}\left(\frac{x}{r}\right)^{N} \mathrm{d}x\\
&= \frac{1}{r^N}\frac{(r-1)^{N+1}}{N+1} = \frac{(r-1)^{N+1}}{r^N(N+1)}.
\end{aligned}$$

Since $UB \geq LB$, the lower bound of $\mathbf{E}[c]$ is $r - UB$, whereas the upper bound is $r - LB$. Thus,

$$\frac{Nr^{N+1} - 1}{r^N(N+1)} \leq \mathbf{E}[c] \leq \frac{(N+1)r^{N+1} - (r-1)^{N+1}}{r^N(N+1)}.$$

We obtain almost straight lines for the upper and lower bound of $\mathbf{E}[c]$, as shown in Figure 3.11, which also shows that, even for large r, the cycletime seems predictable to a reasonably accurate level.

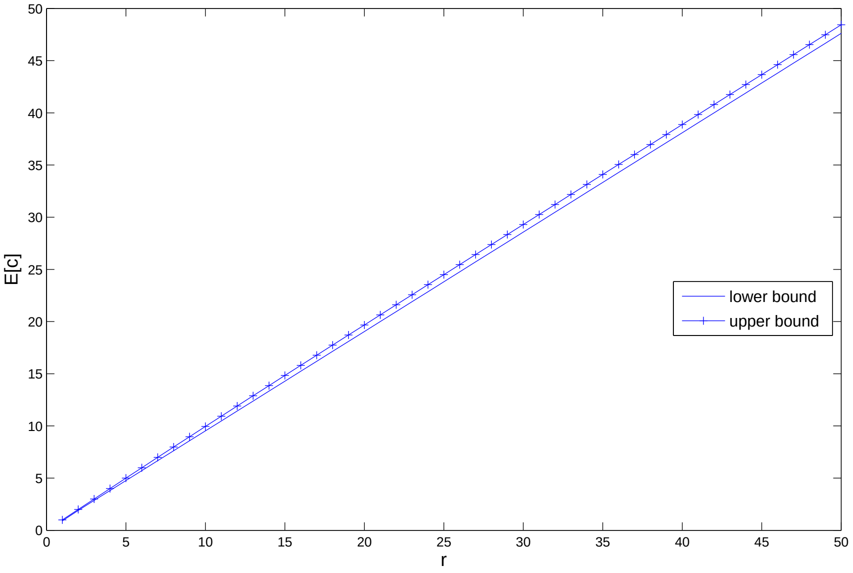


Figure 3.11: Upper and lower bound of the mean cycletime $\mathbf{E}[c]$ of a max-plus system on the regular n-nbhd network as a function of the ξ radius r. The network size is $N = 20$ and transmission times are zero.

Now consider $p(r) = 1 - ((r-1)/r)^N)$, which is a decreasing function of r. For a fixed network size, the cycletime is most likely to be large, but with lower probability as the ξ radius is increased. Consequently, we might expect a larger distribution of cycletimes as the ξ radius increases. Figure 3.12 shows the largest and smallest of the 500 cycletimes obtained by implementing Algorithm 3.2.1. We can see the range of cycletime increasing with r. The implication is that the cycletimes from 500 runs are more distributed over the integers 1 to r. It is further conjectured that the mean of these large number of cycletimes may produce steady state behaviour as $r \to \infty$.

We conclude this section with the observation that the asymptotic behaviour of the large regular network is controlled by the slowest process, i.e. the largest processing time. Moreover, as the range of possible values for processing times becomes large, the mean behaviour of the max-plus system is predictable to a reasonable degree of accuracy. This was highlighted in terms of the transient time, cyclicity and cycletime. Thus, we shall use this result by fixing r to be large in the examination of large parts of the remaining study, in particular when comparing asynchronous systems that differ due to a variation of the max-plus model to the maxmin-m model for $1 < m < n$.

3.2.4 The effect of neighbourhood size on the regular n-nbhd network

Although the topology of the network is not a major concern in this thesis, it would be interesting to know the variation in behaviour, if any, of the max-plus system on the regular n-nbhd network when the neighbourhood size n is varied. Thus, in this section, we consider all values of n (which are odd by construction). The max-plus system is again constrained by the zero transmission condition. Lemma 3.2.1 gives the period of the max-plus system as 1 for all initial conditions and distributions of ξ_i, no matter the value of n, so we are concerned with the transient time and cycletime. Concerning the cycletime, we refer to the nine graphs in Figure 3.12. It is evident that all graphs share the same approximate shape. This then also supports

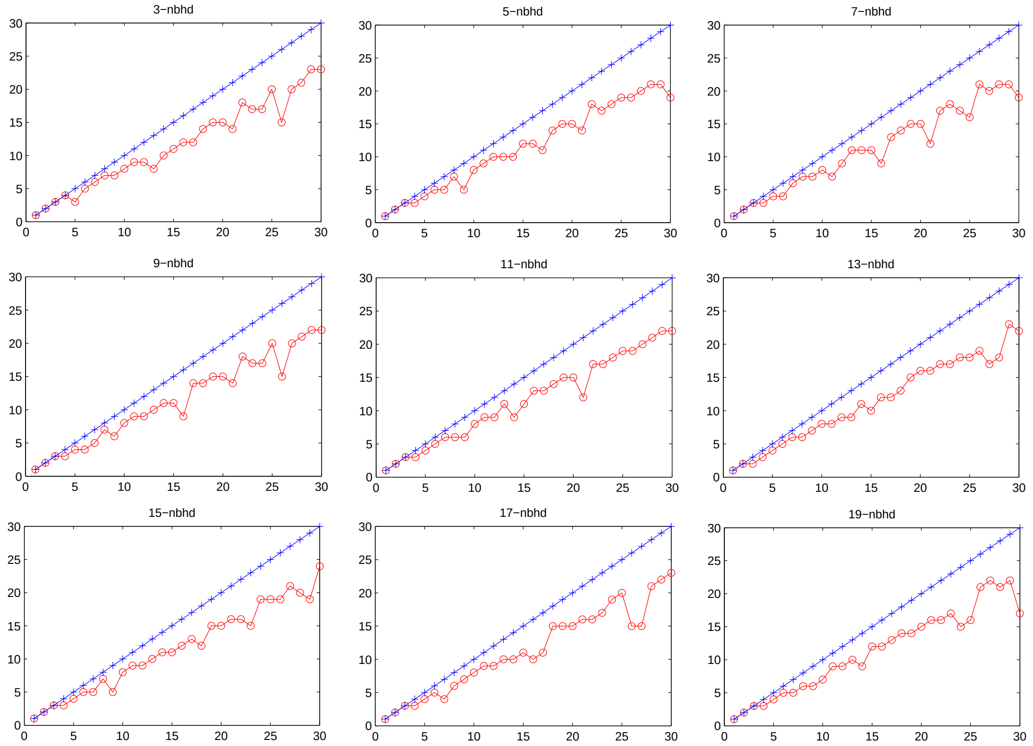


Figure 3.12: Upper and lower bound for cycletime (vertical axis) versus r (horizontal axis) for all regular neighbourhood sizes

Lemma 3.2.1 which states that the cycletime is independent of neighbourhood size. Thus, there remains the study of transient time in this section.

Let us fix the ξ distribution and briefly study two examples, both with $N = 20$, fixed. The first example takes $\xi = (1, 2, 3, \ldots, 19, 20)^\top$. We denote this processing time distribution by ξ^I and find $K(\mathbf{x}(\mathbf{0}))$ for the initial state $\mathbf{x}(0) = \mathbf{u}$, then compare it with $t(P)$. The results are plotted in Figure 3.13(a). The second example uses the same method but the processing time distribution

$$\xi = (4, 2, 13, 9, 10, 11, 1, 5, 18, 7, 17, 16, 20, 3, 8, 12, 14, 19, 15, 6)^\top$$

which we call ξ^{II}. Those results are plotted alongside the first results in Figure 3.13(b). Note that ξ^{II} is simply a 'shuffled' version of ξ^I, and it yields the larger transient times. However, a common feature of the plots in Figures 3.13 is that both K and the respective $t(P)$ values follow a common pattern with varying n. It is conjectured that there is a 'power-law' relation between (either form of) transient time and n. Let us check this by observing the log-log plots of transient time versus n in the insets to the larger graphs in Figure 3.14. For $\xi = \xi^I$, the plot is approximately linear up to the case $n = 11$, which corresponds to the first point where $t(P)$ and K remain unchanged between successive n values. Nevertheless, a linear approximation of the whole log-log plot does not differ significantly from the linear approximation of the plot up to $n = 11$. In particular, we obtain

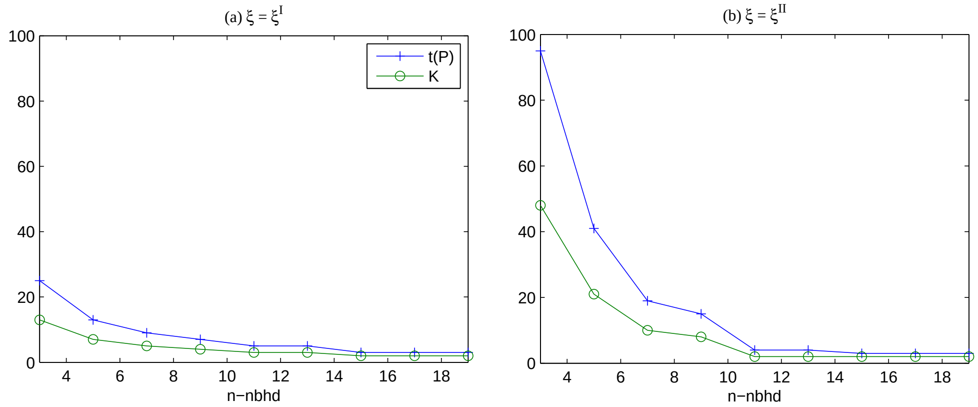


Figure 3.13: Transient times of the regular n-nbhd max-plus system of size 20 under the zero transmission condition. Values of K obtained from initial state zero.

$$t(P) = 90n^{-1.2} \text{ and } K = 40.5n^{-1}.$$

Similarly, for $\xi = \xi^{II}$, we obtain

$$t(P) = 992.3n^{-2.1} \text{ and } K = 403.4n^{-1.9}.$$

The plots of these approximations are shown as the larger graphs in Figure 3.14. The resemblance between these graphs to those in Figure 3.13 is evident and it can be concluded that a power-law is an adequate description for the behaviour of transient time as a function of n.

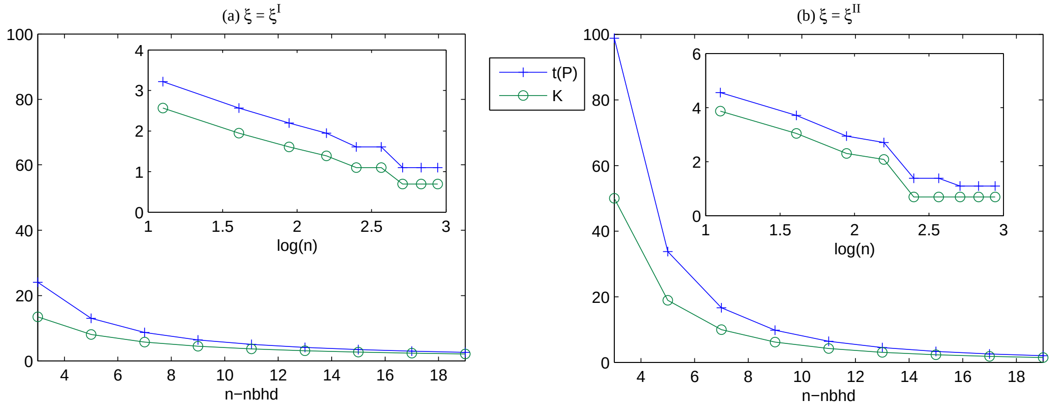


Figure 3.14: (Inset) Log-log plots of the plots in Figure 3.13. (Larger) The corresponding relation between transient times and n that are obtained from the log-log plots.

Further, while we have earlier noted that the creation of a simple formula for transient time is difficult, researchers in max-plus systems have constructed upper bounds for both $t(P)$ and $K(\mathbf{x}(0))$ [20, 38]. However, in these works, the relationship between $t(P)$ and $K(\mathbf{x}(0))$ is not commented on. Here, we observe that such a relationship may be deduced by plotting the values of $K(\mathbf{x}(0))$ against the corresponding values of $t(P)$ on the Euclidean plane.

In order to better understand the relation between the transient time and n, we now construct a lower bound for $t(P)$ in the following lemma. We shall make use of the *ceiling* function $\lceil \cdot \rceil : \mathbb{R} \to \mathbb{Z}$, which yields, for any $x \in \mathbb{R}$, the least integer $\bar{x}$ that satisfies $\bar{x} \geq x$.

Lemma 3.2.3. Consider the timing dependency matrix P of the regular n-nbhd network on N nodes. Let $\underline{t(P)}$ denote the lower bound to $t(P)$. Then $\underline{t(P)} = \left\lceil \frac{N-1}{n-1} \right\rceil$.

Proof. $\underline{t(P)}$ is simply the time it takes for the powers of P to become fully non-zero (i.e. $\neq \varepsilon$). By forming a power $P^{\otimes k}$, we multiply $P^{\otimes(k-1)}$ by P, so that each row of

$P^{\otimes k}$ gains $n-1$ more non-zero elements than $P^{\otimes(k-1)}$. We construct powers of P until we find the first power that has no non-zero elements, i.e. until the number of non-zero elements in each row is N. Therefore, $\underline{t(P)}$ is the smallest solution to

$$N \leq n + (n-1)(\underline{t(P)} - 1).$$

Thus,

$$\underline{t(P)} \geq \frac{N-1}{n-1}.$$

Since we require $\underline{t(P)}$ to be an integer, we use the ceiling function to obtain the smallest solution to the above.

$$\underline{t(P)} = \left\lceil \frac{N-1}{n-1} \right\rceil.$$

□

Under the zero transmission condition, we can deduce that $\underline{t(P)}$ is equal to the transient time $t(P)$ if $\xi_i = j \in \mathbb{N}$ for all nodes i. Notice, however, that there was no condition placed on the processing or transmission times in Lemma 3.2.3. In addition to this, the n-nbhd network need not be regular in the proof. Thus, the lemma is applicable to any type of n-nbhd network that also takes arbitrary transmission times.

If n is fixed, $\underline{t(P)}$ is a linear function of network size N. As a function of n (with N fixed), $\underline{t(P)}$ follows a power-law and tends to 1, the minimal transient of P. Even and Rajsbaum's upper bound in Equation (3.7) can be utilised here to create an intuitive upper bound for $t(P)$. denoted $\overline{t(P)}$, as a function of n. As n increases, the number of arcs in $\mathcal{G}(P)$ increases and, under the zero transmission condition, we are likely to see more occurrences of the largest arcweight $\max\{\xi_i\}$. Therefore, the number of nodes in the critical graph is likely to increase, which implies that the number of nodes in the non-critical graph decreases. Hence, for a fixed N, as n increases, the quantity $l_\star(\mathcal{G}(P))$ is expected to decrease as it depends on the non-critical graph. Consequently, $\overline{t(P)}$ is expected to decrease. Although the exact nature of this decrease proves to be unclear, we can nevertheless combine this argument with the lower bound $\underline{t(P)}$ to say that the actual value of $t(P)$ is at least a decreasing function of n and may well be a negative power-law in n. The arguments formulated for Figures 3.13 and 3.14 are supportive.

Let us return to our study of 500 runs of the size 20 max-plus system with randomised processing times ξ_i. Based on the study of the transient time K as a function of ξ radius r, we assume the existence of an equilibrium value for mean transient time for large r. Thus, we fix r to be 30 (the largest of our studied values) and plot the mean value of K against n for this r. Call this mean value "mean K". The plot is shown in Figure 3.15. There is a similarity in shape of the graph in Figure 3.15 to those graphs of the particular systems in Figure 3.13. However, the power-law behaviour of the transient time as a function of n is notably smoother when taking the mean from a large number of runs.

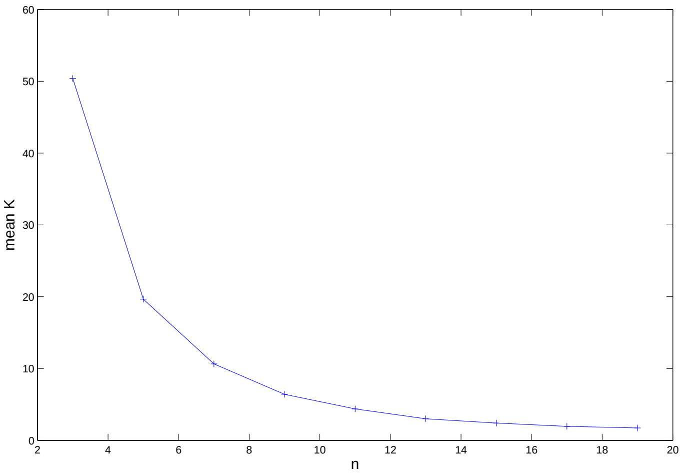


Figure 3.15: Mean transient time mean K as a function of n for a regular n-nbhd network on 20 nodes. The mean is taken from 500 runs with initial condition $\mathbf{x}(0) = \mathbf{u}$, ξ radius 30, and where the zero transmission condition is employed.

Thus, for a fixed neighbourhood size, the transient time in any network is at least linearly increasing with network size N. For fixed N, the transient decreases according to a power law function of n. In other words, a larger connectivity reduces the transient time of the max-plus system and therefore, also minimises any complexity.

3.2.5 General parameter values in the regular n-nbhd network

By removing the transmission condition and allowing transmission times to vary with each successor node, the most general result is obtained. Because the lattice is a regular n-nbhd network, P is still irreducible, but the transient time, and the asymptotic variables of cycletime and cyclicity are not as straightforward to find analytically. Indeed, it is difficult to find expressions for these quantities of interest, which leads to a heavier reliance on numerics than when the transmission condition was imposed. In this section, we shall look at what difference, if any, generalised parameter values provide to the complexity of the max-plus system.

Under the zero transmission condition, we employed the ξ radius, denoted r, as a parameter to generate processing times at random with equal probability. In the general case, we must introduce an analogous parameter for the transmission times. Thus, we let r_2 denote the largest value that transmission times can take, i.e. $1 \leq \tau_{ij} \leq r_2$ for all i, j. We call r_2 the *τ radius.* We also take both processing and transmission times to be non-zero integers.

We numerically study the cycletime χ, period ρ and transient time K as a function of r_2. The process of accumulating the results is identical to Algorithm 3.2.1 with the addition of the following step to step 1:

> Choose τ_{ij} from the uniform distribution (with equal probability) of integers between 1 and r_2.

Along with this additional step and fixing $r = 30$, we take the mean values of χ, ρ and K (denoted mean χ, mean ρ and mean K, respectively) arising from 500 runs of the new algorithm. This algorithm is repeated for various values of r_2. Note that the zero transmission condition work carried out previous to this section corresponds to the case $r_2 = 0$.

General numerical results for a large network

To ease comparison, we take asymptotic results for relatively few values of the τ radius, namely

$$r_2 = 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 100, 200.$$

For each r_2 value stated above, we plot mean K as a function of neighbourhood size n in Figure 3.16. It can be seen that, for small r_2, mean K decreases with n. In fact, this decrease may be characterised by a power-law as was found in the zero transmission case. This seems reasonable in light of the smaller values of r_2 being close to zero. This approximate power-law shape is maintained for larger r_2, although such a shape would now be thought to start at $n = 5$. This is because a 3-nbhd yields smaller transients than the 5-nbhd for large r_2.

The mean cycletime, denoted mean χ, is plotted in Figure 3.17. For small values of r_2, mean χ is approximately constant with n. Again, this follows from the zero transmission case where the cycletime graphs were approximately equal for all values of n. For $r_2 = 100$ and 200, the graph of mean χ displays a 'bump' when $n = 7$ and is otherwise decreasing with n. The range is also relatively large: for $r_2 = 200$, the largest value of mean χ is approximately 155 and the smallest is approximately 69.

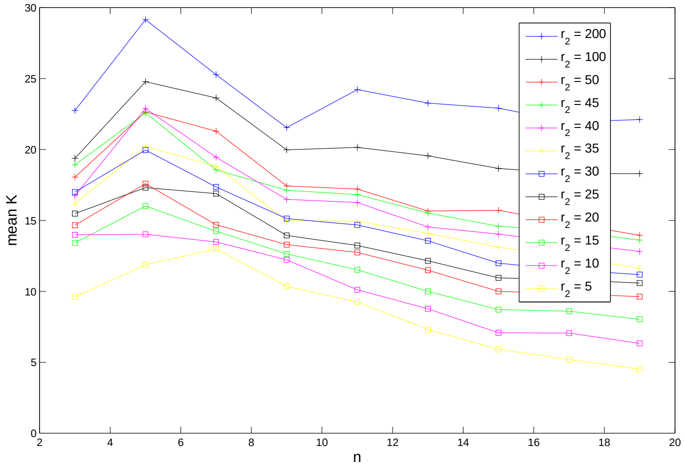


Figure 3.16: Mean transient time mean K as a function of n for a max-plus system on the regular n-nbhd network on 20 nodes. The mean is taken from 500 runs with initial condition $\mathbf{x}(0) = \mathbf{u}$, ξ radius 30, and τ radius r_2 as indicated.

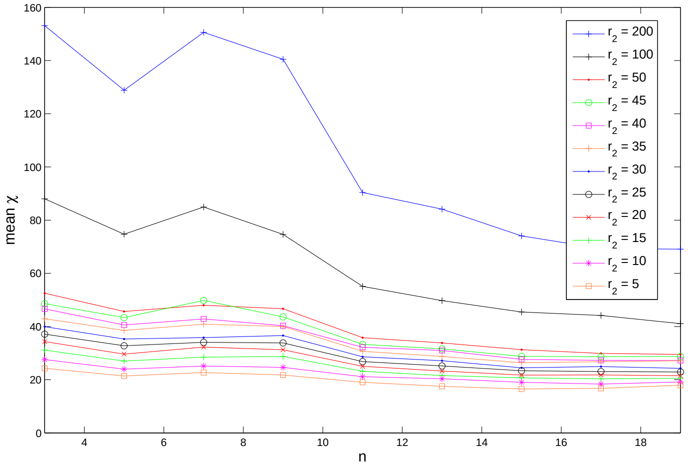


Figure 3.17: Mean cycletime mean χ as a function of n for a max-plus system on the regular n-nbhd network on 20 nodes. The mean is taken from 500 runs with initial condition $\mathbf{x}(0) = \mathbf{u}$, ξ radius 30, and τ radius r_2 as indicated.

Finally, the most interesting of the three results is the plot of mean period, shown in Figure 3.18, where it can be seen that a smaller neighbourhood size yields similar periods, even for large values of the τ radius. Thereafter, mean ρ is almost a monotonically increasing function of n.

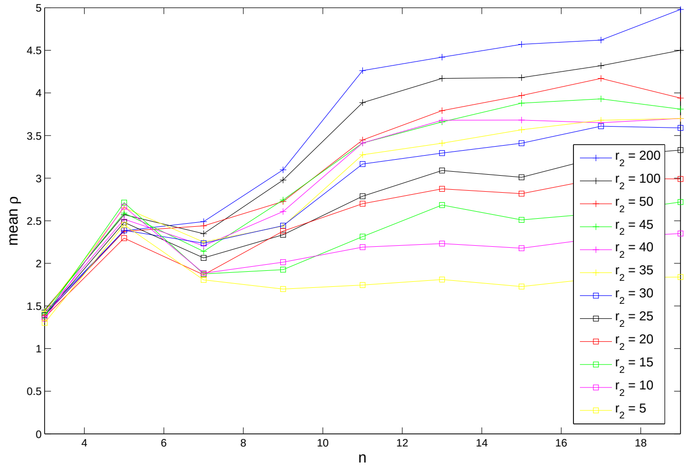


Figure 3.18: Mean period mean ρ as a function of n for a max-plus system on the regular n-nbhd network on 20 nodes. The mean is taken from 500 runs with initial condition $\mathbf{x}(0) = \mathbf{u}$, ξ radius 30, and τ radius r_2 as indicated.

In summary, the transient times are as expected since mean K is uniformly increasing with r_2. Whilst it was unaffected by processing times under the zero transmission condition, the cycletime is affected by neighbourhood size when r_2 is large, with the smaller neighbourhood sizes yielding the largest cycletime and the larger n yielding smaller cycletimes. The period (and cyclicity) depends on elementary circuits, and a large network of size $N = 20$ would contain circuits of varying lengths (up to 20). Thus, the period of the max-plus system on this regular network implies a minimising of complexity since, for all values of r_2, mean ρ never exceeds 5. This is supported by the decreasing transients with n. Together as two of our criteria for complexity, the period and transient time results in this section suggest that a max-plus system on the regular n-nbhd network minimises the complexity of the update times.

3.3 A cellular automaton model in max-plus time

Having studied asynchronous update times generated by the max-plus system in previous sections, this section looks at the results of implementing a cellular automaton model under such timing rules. We focus on ECA rule 150, extended to larger neighbourhood sizes. Wolfram classified this rule as a Class III rule. In Chapter 1, on a finite lattice and under the classical synchronous mode of update times, it was shown to be simple, generating periodic CA space-time patterns at the height of its complexity. In this section, we ask what effect an underlying max-plus timing system can have on this complexity. In Chapter 1, we also demonstrated the techniques that we will use to classify the CA space-time patterns that are produced, i.e. the word and Shannon entropies, as well as the CA period and CA transient times. These quantities will be prominent in the following sections.

3.3.1 CA on the regular n-nbhd network

In Section 3.1, we established the existence of a bijection between the CA in max-plus time and synchronous CA. The implication of this is that, given the same initial CA state, both CAs share the same CA period p and CA transient time K_C. Thus, we need only run the synchronous CA on the regular n-nbhd network to find the asynchronous values for p and K_C. Let the initial CA state be given as follows.

$$s_i(0) = \begin{cases} 1 & \text{if } i = 10 \\ 0 & \text{otherwise} \end{cases} \tag{3.18}$$

For network size $N = 20$, running rule 150 yields the CA periods and CA transient times as in Table 3.2.

n	3	5	7	9	11	13	15	17	19
K_C	0	4	0	0	0	0	4	0	0
p	12	2	12	4	4	12	1	12	2

Table 3.2: CA period p and CA transient time K_C for the regular n-nbhd network of size $N = 20$. The initial CA state is (3.18) and the CA rule is rule 150.

The values in Table 3.2 being fixed for all parameter distributions ξ and τ, our focus now turns to the extent of the effect that these parameters have on the *real time* CA pattern. For example, for the same number of iterations of the regular n-nbhd max-plus system, a large cycletime will yield larger blocks (or spaces) of homogeneity than a smaller cycletime. While both synchronous and max-plus CA outputs share the same CA period and transient time, in real time, the patterns would be qualitatively different. Thus, we classify this difference by calculating the Shannon and word entropies.

Section 3.1.6 provided a taste of CA in max-plus time by considering an arbitrary network of size $N = 4$. In this section, we will look at what happens to the CA on a larger network, particularly the regular n-nbhd network of size $N = 20$.

Entropies for large network size

To compile max-plus update time results, we ran Algorithm 3.2.1 in Section 3.2.3. Not mentioned in that section was the fact that we also implemented ECA rule 150 concurrently to obtain the CA results. Thus, for each run of Algorithm 3.2.1, we obtained the max-plus update time results as well as the mean values of S and W. The initial CA state was the same as (3.18).

Before we state the numerical entropy results produced, we explore the entropies theoretically as far as is possible. Thus, consider the periodic regime in a max-plus system on the regular n-nbhd network. We have proved that, under the zero transmission condition, this network has cyclicity one and cycletime $\chi = \max_i\{\xi_i\}$. Therefore, contours in the regime will be the same shape and separated by a real time interval χ. By thinking about the bijection between the synchronous CA and the max-plus CA, we can say that, for each pair of consecutive contours in the periodic regime, a CA state s appears χ more times along the evolution of node i in the max-plus CA than in the synchronous CA.

Let $p^{\mathcal{S}}(s)$ denote the density of the CA state s along the time evolution of a node in the synchronous CA pattern. Then, $p^{\mathcal{S}}(s) = \frac{n_s}{k}$ where n_s is the number of times state s appears in the time evolution and k is the number of iterations taken. Now assume the same number of iterations are taken in the corresponding max-plus system, which we label $\mathcal{M}$. Then, in the periodic regime of $\mathcal{M}$, real time is stretched by χ, which gives the following value for the density $p^{\mathcal{M}}(s)$ of state s along the time series of a node in the max-plus CA pattern.

$$\begin{aligned} p^{\mathcal{M}}(s) &= \frac{\chi \times n_s}{\chi \times k} \\ &= \frac{n_s}{k} \\ &= p^{\mathcal{S}}(s). \end{aligned} \tag{3.19}$$

We have seen in Chapter 1 that, after a long length of time, the Shannon entropy of a periodic CA tends to the Shannon entropy of its periodic region. This principle can also be applied to a homogeneous CA by simply taking the homogeneous region to be its periodic region. We therefore consider the limit $k \to \infty$ and use (3.19) to state the following lemma.

Lemma 3.3.1. Consider a max-plus system $\mathcal{M}$ on the regular n-nbhd network. Let $S^{\mathcal{M}}(k)$ denote the Shannon entropy of a CA pattern produced under asynchronous time generated by $\mathcal{M}$. If $S^{\mathcal{S}}(k)$ denotes the Shannon entropy of a synchronous CA pattern produced by the same CA rule, then

$$S^{\mathcal{M}}(k) \to S^{\mathcal{S}}(k)$$

as $k \to \infty$,

Now consider the word entropy. In Section 3.1, we saw that, in a max-plus CA, states are fixed (in real time) between contours. For cell i, this creates a word of length equal to the time interval between contours. If the CA rule leaves the state s_i of the cell unchanged on a contour, then $s_i(k-1) = s_i(k)$ and the word comprising state $s_i(k)$ proceeds until time $x_i(k+1)$, i.e. contour $k+1$; this yields a larger word whose length is the sum of the two time intervals between the contours. Continuing in this fashion, it can be deduced that, as $k \to \infty$, wordlengths in the regular n-nbhd max-plus system are multiples of the cycletime. This is because χ is uniform for all nodes and $\sigma = 1$ in this system. Thus, the property of memory between contours in a max-plus system and the bijection from the synchronous CA aids the hypothesis that the word entropy is also approximately equal to the synchronous word entropy as $k \to \infty$.

Figure 3.19 plots the entropy results for all neighbourhood sizes on the regular network. As mentioned in the first paragraph of this section, the entropy results were obtained as part of Algorithm 3.2.1, but not mentioned there. In the interests of clarity, we state those omitted steps in Algorithm 3.3.1.

Fix n and choose the ξ radius $r \in \mathbb{N}$.

Algorithm 3.3.1.

1. Choose ξ_i from the uniform distribution (with equal probability) of integers between 1 and r.

2. Taking $\mathbf{x}(0) = \mathbf{u} = (0, 0, \ldots, 0)^\top$ and initial CA state $\mathbf{s}(0)$ as in (3.18), run 230 iterations of the max-plus system to obtain the entropies S and W.

3. Carry out above two steps to obtain 500 such results for this value of r.

4. Record the mean of the 500 entropy values obtained.

Algorithm 3.3.1 was repeated for each integer r from 1 to 30. (Thus, for example, for one n-nbhd, we obtained a total of $500 \times 30 = 15000$ results for S). Note that 230 iterations were taken as the limit that the computer allowed.

Each point in the displayed WS plane represents one run of Algorithm 3.3.1; thus, there are 30 scatter points in each WS plane shown. More often than not, the larger mean entropy values were obtained for large r.

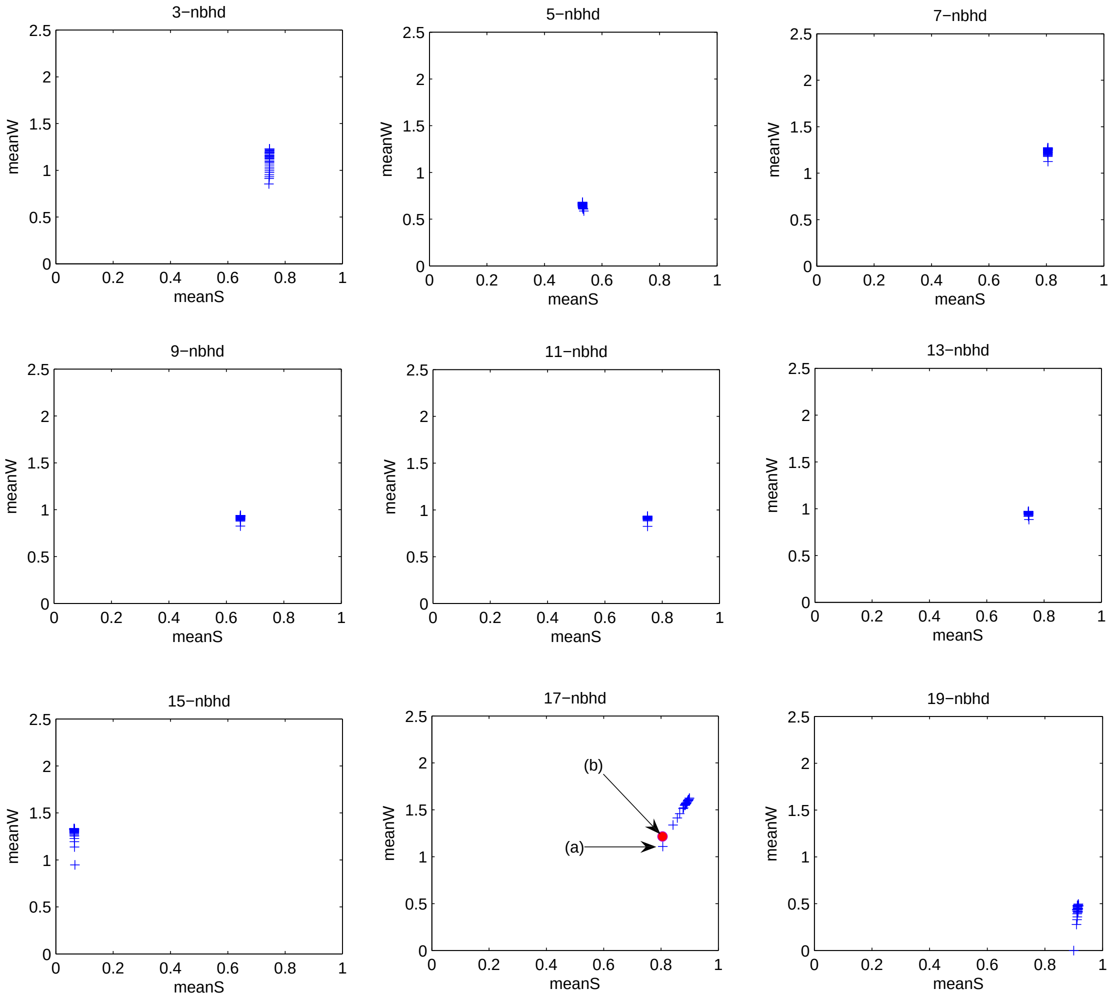


Figure 3.19: WS plane for each neighbourhood size in the regular network on 20 nodes. Each scatter point represents each ξ radius value r (where $1 \leq r \leq 30$). There are thus 30 scatter points, each of which is the mean value (denoted by coordinate $(\text{mean}S, \text{mean}W)$) of 500 runs of the max-plus system on the network arising from Algorithm 3.3.1.

In the nine neighbourhood WS planes, it is evident that the Shannon entropy S of nearly all patterns is almost fixed for all values of r that were tested. The only network that produced significant variation of S with r was the 17-nbhd network. Nevertheless, in terms of Marr and Hütt's domains representing the Wolfram Classes, this neighbourhood size still confines all of its CA in Class II. The evidence thus supports Lemma 3.3.1, which effectively states that the Shannon entropy is almost fixed for all CA on a fixed network. The points indicated by (a) and (b) on the 17-nbhd WS plane refer to the corresponding CA patterns in Figure 3.20. (a) is the pattern obtained for $r = 1$ (so $\xi_i = 1$ for all i) which is equivalent to the synchronous pattern. Point (b) is shaded and denotes the result of taking $r = 20$ (with the exact ξ distribution given in the figure). The entropies of these patterns show a marked difference. However, both patterns qualitatively look strikingly similar to the observer. Thus, even in the 17-nbhd entropies, which show more variance than the other neighbourhoods, we can conclude that the associated CA patterns are not too dissimilar.

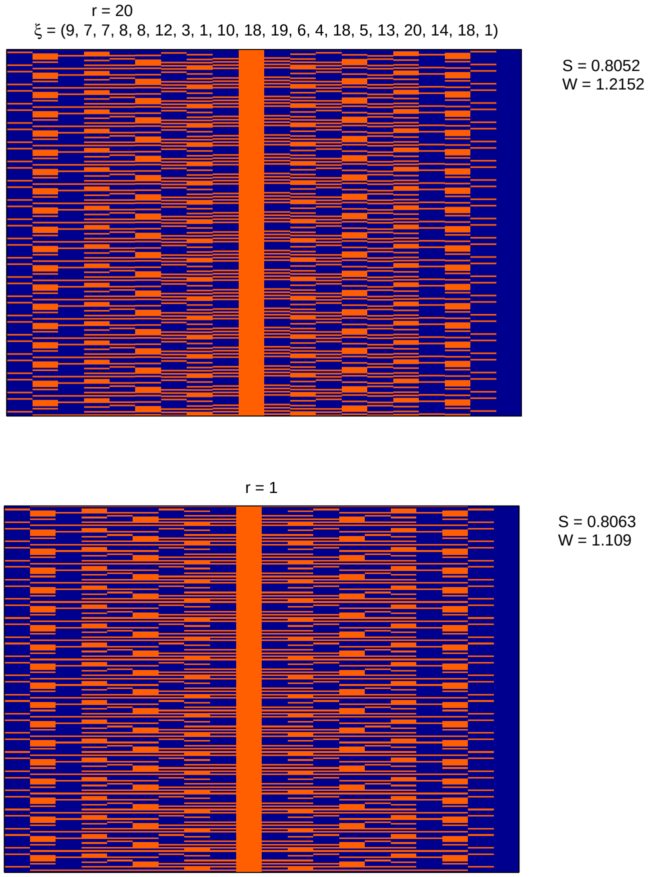


Figure 3.20: 17-nbhd CA patterns with initial CA state as given in (3.18). Both patterns are generated after 230 iterations, where time travels up. $r = 20$ corresponds to a randomised distribution of ξ_i; $r = 1$ corresponds to the synchronous case. Entropy values are as indicated.

A different perspective is provided by the word entropy. Whilst most neighbourhood sizes keep W fixed, the neighbourhood sizes 3, 15, 17, and 19 show relative variation in their word entropy values as a function of r.

We believe the transient region comes into play here. Indeed, any difference in S values is also ascribed to the effect of the transient region, but W is evidently affected more.

Consider Figure 3.21, which shows the transient CA evolution of one cell. With time travelling up, the synchronous transient region may be represented as the sequence 01101 while the max-plus transient region is 001111011. The words in the max-plus case have been highlighted in bold, reflecting the chaotic nature of the transient region contours. Now suppose that, in the synchronous case, the periodic region following this transient is $0110110110110\cdots$, i.e. it is a period 3 CA with transient $K_C = 5$. Consider the following note.

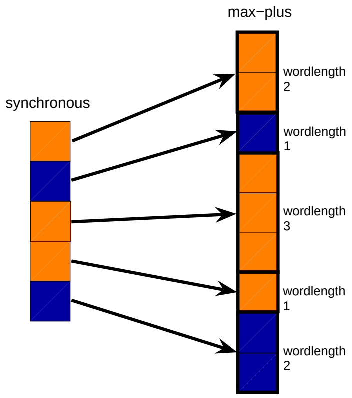


Figure 3.21: Example of the evolution of one cell and its corresponding max-plus evolution. CA state 1 is represented by an orange square, CA state 0 by a blue square.

Note 3.3.1. In the max-plus system, the CA transient K_C may not necessarily coincide with the max-plus transient K. Thus, we may obtain $K_C < K$, in which case the CA becomes periodic in the contour transient region. Here, while the CA is periodic, the density of CA states and wordlengths may not be proportional to the synchronous densities.

If $K_C > K$, then part of the CA transient region is contained in the periodic contour region. In this case, the periodic region densities of CA states and wordlengths will be proportional to the synchronous region and therefore, easier to predict and calculate.

We would ideally like the case

$$K_C \geq K \tag{3.20}$$

since this ensures that the periodic region of the CA is fully contained in the contour periodic region. It subsequently helps the drawing of more accurate predictions, particularly with regards to the entropy of a periodic CA, as the varying real time lengths between contours in the transient region will have been avoided. Thus, we continue with the analysis of Figure 3.21 and assume (3.20) to be true. In the figure, the density of individual cell states is $p^{\mathcal{S}}(0) = 2/5$ and $p^{\mathcal{S}}(1) = 3/5$ in the synchronous pattern, whereas the corresponding densities in the max-plus pattern are $p^{\mathcal{M}}(0) = 6/9$ and $p^{\mathcal{M}}(1) = 3/9$. The two systems yield differing transient densities but, as seen in Chapter 1, Section 1.1.3, their effect is diminished as the number of iterations becomes large.

Now consider the wordlengths in both systems. The synchronous case shows two different wordlengths, three 1-words and one 2-word, whereas the maxplus case shows an extra wordlength: there are two 1-words, two 2-words and one 3-word in the transient region there. If words in the max-plus system were proportional to the synchronous CA (as is the case in the periodic region), then there would be the same number of such wordlengths there and calculation of the word entropy W would involve the sum of two terms corresponding to both wordlengths in the synchronous case. However, Figure 3.21 shows that, in the transient max-plus case, there is an extra wordlength to consider, therefore an extra term to add in the calculation of W. The aperiodic variation of time lengths between contours in the transient region thus can lead to more wordlengths than in the periodic region, especially for different processing and transmission time distributions.

On the other hand, calculation of the Shannon entropy always involves the sum of only two terms, which leads us to conjecture that, in a periodic CA pattern, the Shannon entropy S tends to its limit faster than the word entropy. Put mathematically, we have the following conjecture.

Conjecture 3.3.1. Consider a periodic CA pattern, obtained for k iterations. Let S and W denote the Shannon and word entropies of the pattern. Then, $S \to S(r)$ faster than $W \to W(r)$, where $S(r)$ and $W(r)$ indicate the Shannon and word entropy values of the periodic region.

3.4 Summary

We have seen that the max-plus system on the regular n-nbhd network minimises complexity. Further, we observe that, for $r \gg r_2$, the minimisation of complexity is reinforced. In that case, processing times become much larger than transmission times and transmission times become negligible. In general, i.e. when transmission times are non-zero, we particularly saw that the cyclicity $\sigma(P)$ of the timing dependency matrix P never exceeded mean value 5, even for a large network that would contain a large number of possibilities of behaviour proportional to the number of circuits in the timing dependency graph $\mathcal{G}(P)$. Therefore, no matter how the parameters ξ_i and τ_{ij} are varied, this max-plus system on the regular n-nbhd network never becomes so asynchronous as to be classified as complex.

What looks asynchronous in real time is in fact a mapping from the synchronous CA. This is because there is a *bijection* between classical synchronous CA and max-plus CA. In particular, we can draw up a state transition graph of the CA. This STG will also represent CA in max-plus time, where the CA state on contour k will be equal to the state on time step k in the synchronous CA. Therefore, we can predict the behaviour of the max-plus CA on contours of time, although we can't directly predict the CA states in real time t. The latter problem can be worked around by considering the Shannon and word entropy measures, which can be utilised to indicate such a real time bijection. In general, CA patterns that are qualitatively similar will also share similar such entropy values. However, what is more useful is that the entropy values of synchronous CA will also be approximately equal to the entropy values of corresponding max-plus CA (under the same rule and initial conditions). Thus, entropies also act as an indicator of a bijection.

In conclusion, the type of asynchronous CA in this chapter can in fact be predicted from its synchronous mode. In relation to this, it may also be possible to deduce one of the four Wolfram classes from the max-plus timing model and without running the CA at all.

Finally, we comment on the absence of an exposition of CA in max-plus time when the zero transmission condition is relaxed. Such a CA, in fact, turns out to be surplus to requirements since the regular n-nbhd network makes the bijection clear to see, and makes prediction of CA behaviour simple. Therefore, we can apply the bijection to the general transmission time case also and conclude similarly. In particular, Lemma 3.3.1 applies to such a non-zero transmission case also.

Chapter 4

The Maxmin-m Model of Asynchrony

Thus far, our look at asynchronous time via the max-plus model has focussed on cellular automata as an application. If we abstract away all the particular attributes of CA, we find that, at the heart of it is a network of processes (represented by nodes) that exchange information with their neighbours.

Thus, in this chapter, we demonstrate an implementation of maxmin-plus algebra on such an information exchange system. In particular, we look at the *maxmin-m* model in which the state of a node is updated after it receives the first m inputs from its n-nbhd ($1 \leq m \leq n$).

Although this theory will be illustrated with cellular automata in the next chapter, it turns out that the salient features of the maxmin-m system provide enough scope for research and understanding of any information exchange system (not only CA) without the need to simulate the system itself.

In Section 4.1, we introduce maxmin-plus algebra and establish preliminary ideas for the understanding of asymptotic behaviour of the maxmin-m model. This includes an introduction to the celebrated duality theorem [13], which is expanded upon in Section 4.2, in which we formalise an intuitive approach to examining the behaviour of the maxmin-m system. The work is assisted by an example, in which the timing dependency graph is small, and demonstrates a reduction of the maxmin-m system to a max-plus system after a long period of time. The highlight of the chapter is in Section 4.3, which addresses a unique phenomenon arising out of the maxmin-m system. It proves the nonuniqueness of the aforementioned reduction of maxmin-m which, moreover, cannot be resolved by the duality theorem. Section 4.4 summarises the chapter.

4.1 Maxmin-plus algebra

We refer the reader to Chapter 2 where we introduced our asynchronous update time system as one in which each node in a network waits for the first m out of a possible n inputs before updating its (cellular automaton) state. We have, in fact, been careful with the description of this model in the thesis. The state is not updated *upon* receiving exactly m inputs; rather, it is *after* receiving m inputs. We shall see in this chapter that this distinction is crucial because it allows the possibility for an update after a number of inputs greater than m. For now, we notice that min-plus algebra applies when $m = 1$; max-plus algebra when $m = n$. What about $1 < m < n$? This section looks at a way in which this can be modelled, which turns out to be a mixture of both max-plus and min-plus algebra. We call this *maxmin-plus algebra*, and the topic has been touched upon in [22], which presents a compact exposition of the ideas in [17]. Thus, in presenting the theory, we shall refer to [22], as was done in Chapter 2.

Consider the n-nbhd network of cells where each cell i updates its CA state for the $(k+1)^{\text{th}}$ time after receiving the first m inputs from its neighbourhood cells $(1 \leq m \leq n)$. Then the $(k+1)^{\text{th}}$ update time of cell i is given by the following recurrence relation.

$$x_i(k+1) = x_{(m)}(k) + \xi_i(k+1) \tag{4.1}$$

where $x_{(m)}(k)$ represents the kth time of arrival of the mth input. If k is clear from context, we denote this by $x_{(m)}$ for short. Equation (4.1) formally defines the maxmin-m timing model for node i. When such an equation is specified for all nodes in the network, we refer to the resulting set of equations as the *maxmin-m system*. Suppose the network is regular with $n = 3$. Then, in terms of this 3-nbhd,

$$x_{(m)}(k) \;=\; \begin{cases} x_{i-1}(k) + \tau_{i,i-1}(k) & \text{if } m\text{th input provided by node } i-1 \\ x_i(k) + \tau_{ii}(k) & \text{if } m\text{th input provided by node } i \\ x_{i+1}(k) + \tau_{i,i+1}(k) & \text{if } m\text{th input provided by node } i+1 \end{cases}$$

Note 4.1.1. It is possible that there are a multiple number of inputs that may be taken as the mth input. Let $\mathcal{N}_{x_i} \in \mathbb{R}^{n\times 1}$ denote the vector of arrival times from the neighbourhood of node i with the additional feature that this vector is now sorted in ascending order. Consider an example with $\mathcal{N}_{x_i} = (a, a, b, c, c, c, d, d, e, \ldots) \in \mathbb{R}^{n\times 1}$, where distinct elements are identified with distinct letters. Element h is distinct from element j if the corresponding letters are different. If $m = 4$ here, then the mth input time is c and it may be supplied by one of three nodes (which are the vector elements $\mathcal{N}_{x_i}(4)$, $\mathcal{N}_{x_i}(5)$, and $\mathcal{N}_{x_i}(6)$). We call this simultaneity, and it will be addressed later in this chapter when we think about which nodes provide the inputs to node i. Presently, simultaneity is not a concern since we are working with time; the time $x_{(m)}$ is unchanged no matter how many inputs arrive simultaneously. Thus, we shall proceed with the assumption that a suitable node has been chosen as the mth input when referring to the phrase "the mth input".

For $n = 3$, there is only one value of $m \in \mathbb{N}$ that satisfies $1 < m < n$, namely $m = 2$. To identify $x_{(2)}$ involves sorting all three arrival times then choosing the second smallest. For visual clarity, let us represent in compact form the arrival times $x_{i-1}(k) + \tau_{i,i-1}(k), x_i(k) + \tau_{ii}(k)$ and $x_{i+1}(k) + \tau_{i,i+1}(k)$ as x_a, x_b and x_c respectively. Then it can be verified that

$$x_{(2)} = (x_a \oplus x_b) \ominus (x_b \oplus x_c) \ominus (x_c \oplus x_a).$$

where $\oplus$ and $\ominus$ signify the maximum and minimum operations respectively, as introduced in previous chapters. For $m = 2$, Equation (4.1) becomes

$$\begin{aligned} x_i(k+1) &= x_{(2)}(k) + \xi_i(k+1) \\ &= ((x_a \oplus x_b) \ominus (x_b \oplus x_c) \ominus (x_c \oplus x_a)) \otimes \xi_i(k+1) \end{aligned}$$

where the addition operation $+$ has been replaced by $\otimes$.

As in Chapter 2, we suppress the time dependency of the parameters $\xi_i(k)$ and $\tau_{ij}(k)$. By substituting the original form of x_a, x_b and x_c back in, we now obtain the following evolution equation for the update times of cell i.

$$\begin{aligned} x_i(k+1) &= ((x_{i-1}(k) \otimes \tau_{i,i-1} \oplus x_i(k) \otimes \tau_{ii}) \ominus (x_i(k) \otimes \tau_{ii} \oplus x_{i+1}(k) \otimes \tau_{i,i+1}) \ominus \\ &\quad \ominus (x_{i+1}(k) \otimes \tau_{i,i+1} \oplus x_{i-1}(k) \otimes \tau_{i,i-1})) \otimes \xi_i. \end{aligned} \tag{4.2}$$

4.1.1 Preliminaries

Let us list some identities of maxmin-plus algebra. It can be checked that

$$\text{associativity of } \oplus\text{: } a \oplus (b \oplus c) = (a \oplus b) \oplus c \tag{4.3}$$

$$\text{associativity of } \ominus\text{: } a \ominus (b \ominus c) = (a \ominus b) \ominus c \tag{4.4}$$

$$\text{distributivity of } \oplus \text{ over } \ominus\text{: } c \oplus (a \ominus b) = (c \oplus a) \ominus (c \oplus b) \tag{4.5}$$

$$\text{distributivity of } \ominus \text{ over } \oplus\text{: } c \ominus (a \oplus b) = (c \ominus a) \oplus (c \ominus b). \tag{4.6}$$

Thinking of the min-plus and max-plus algebras, the ordinary addition operation $+$ is a distributive multiplication operation in both algebras. Thus, let us distribute ξ_i in (4.2) to obtain

$$\begin{aligned}
x_i(k+1) &= (x_{i-1}(k) \otimes \tau_{i,i-1} \otimes \xi_i \oplus x_i(k) \otimes \tau_{ii} \otimes \xi_i) \ominus \\
&\quad \ominus (x_i(k) \otimes \tau_{ii} \otimes \xi_i \oplus x_{i+1}(k) \otimes \tau_{i,i+1} \otimes \xi_i) \ominus \\
&\quad \ominus (x_{i+1}(k) \otimes \tau_{i,i+1} \otimes \xi_i \oplus x_{i-1}(k) \otimes \tau_{i,i-1} \otimes \xi_i).
\end{aligned}$$

As previously, we can remove $\otimes$ and write

$$\begin{aligned}
x_i(k+1) &= (x_{i-1}(k)\tau_{i,i-1}\xi_i \oplus x_i(k)\tau_{ii}\xi_i) \ominus (x_i(k)\tau_{ii}\xi_i \oplus x_{i+1}(k)\tau_{i,i+1}\xi_i) \ominus \\
&\quad \ominus (x_{i+1}(k)\tau_{i,i+1}\xi_i \oplus x_{i-1}(k)\tau_{i,i-1}\xi_i).
\end{aligned} \tag{4.7}$$

Let the vector $\mathbf{x}(k) \in \mathbb{R}^N$ represent the state of all N cells on epoch k, i.e. $\mathbf{x}(k) = (x_1(k), x_2(k), \ldots, x_N(k))^\top$. Then the matrix-vector form of (4.7) gives the evolution equation for the whole regular 3-nbhd network, i.e. (4.7) is the maxmin-m system. Here, it is a maxmin-2 system, given as

$$\mathbf{x}(k+1) = P_1 \otimes \mathbf{x}(k) \ominus P_2 \otimes \mathbf{x}(k) \ominus P_3 \otimes \mathbf{x}(k), \tag{4.8}$$

where $P_1, P_2, P_3 \in \mathbb{R}_{\max}^{N\times N}$, i.e. matrices with entries in $\mathbb{R}_{\max}$. They are given as follows.

$$P_1 = \begin{pmatrix}
\xi_1\tau_{11} & \varepsilon & \varepsilon & \varepsilon & \cdots & \varepsilon & \xi_1\tau_{1N} \\
\xi_2\tau_{21} & \xi_2\tau_{22} & \varepsilon & \varepsilon & \cdots & \varepsilon & \varepsilon \\
\vdots & & \ddots & & & & \vdots \\
\varepsilon & \varepsilon & \cdots & \varepsilon & \xi_{N-1}\tau_{N-1,N-2} & \xi_{N-1}\tau_{N-1,N-1} & \varepsilon \\
\varepsilon & \varepsilon & \cdots & \varepsilon & \varepsilon & \xi_N\tau_{N,N-1} & \xi_N\tau_{NN}
\end{pmatrix}, \tag{4.9}$$

$$P_2 = \begin{pmatrix}
\xi_1\tau_{11} & \xi_1\tau_{12} & \varepsilon & \varepsilon & \cdots & \varepsilon & \varepsilon \\
\varepsilon & \xi_2\tau_{22} & \xi_2\tau_{23} & \varepsilon & \cdots & \varepsilon & \varepsilon \\
\vdots & & \ddots & & & & \vdots \\
\varepsilon & \varepsilon & \cdots & \varepsilon & \varepsilon & \xi_{N-1}\tau_{N-1,N-1} & \xi_{N-1}\tau_{N-1,N} \\
\xi_N\tau_{N1} & \varepsilon & \cdots & \varepsilon & \varepsilon & \varepsilon & \xi_N\tau_{NN}
\end{pmatrix}, \tag{4.10}$$

$$P_3 = \begin{pmatrix} \varepsilon & \xi_1\tau_{12} & \varepsilon & \varepsilon & \cdots & \varepsilon & \xi_1\tau_{1N} \\ \xi_2\tau_{2N} & \varepsilon & \xi_2\tau_{23} & \varepsilon & \cdots & \varepsilon & \varepsilon \\ \vdots & & \ddots & & & & \vdots \\ \varepsilon & \varepsilon & \cdots & \varepsilon & \xi_{N-1}\tau_{N-1,N-2} & \varepsilon & \xi_{N-1}\tau_{N-1,N} \\ \xi_N\tau_{N1} & \varepsilon & \cdots & \varepsilon & \varepsilon & \xi_N\tau_{N,N-1} & \varepsilon \end{pmatrix}. \quad (4.11)$$

Whilst the evolution equation is well-defined, this choice of matrices does not provide a unique matrix representation of (4.8). It will be shown later that there is a total of $3^N \times 2^N \times 1 = (3!)^N$ ways of representing this maxmin-2 system.

The system, as written now, is represented in *conjunctive normal form.* Let us describe this by first stating the following definition of a maxmin-plus expression, given in [22].

Definition 4.1.1. A *maxmin-plus expression* is an expression which can be constructed recursively as follows. Variables $x_1, x_2, \ldots, x_N$ taking values in $\mathbb{R}$ are the most basic maxmin-plus expressions. Let $a \in \mathbb{R}$ be a parameter. If f is a maxmin-plus expression, then $a \otimes f$ also is a maxmin-plus expression. If g is another maxmin-plus expression, then $f \oplus g$ and $f \ominus g$ are maxmin-plus expressions. No other expressions are maxmin-plus expressions.

Given some finite $c \in \mathbb{N}$, let f be a maxmin-plus expression, written as

$$f = f_1 \ominus f_2 \ominus \cdots \ominus f_c,$$

where each f_i is a max-plus expression of the form

$$f_i = (a_{i1} \otimes x_1) \oplus (a_{i2} \otimes x_2) \oplus \cdots \oplus (a_{iN} \otimes x_N),$$

with $a_{ij} \in \mathbb{R}_{\max}$. Then we say that f is in *conjunctive normal form.* A maxmin-plus expression f can also be transformed into *disjunctive normal form*, which is written

$$f = f_1 \oplus f_2 \oplus \cdots \oplus f_d$$

for some finite $d \in \mathbb{N}$ and where each f_i is now a min-plus expression, i.e.

$$f_i = (b_{i1} \odot x_1) \ominus (b_{i2} \odot x_2) \ominus \cdots \ominus (b_{iN} \odot x_N),$$

with $b_{ij} \in \mathbb{R}_{\min}$. (Recall that $\odot$ is the same operation as $\otimes$).

The terms "conjunctive" and "disjunctive" are borrowed from [22] in which their relation to the logical "and" ($\wedge$) and "or" ($\vee$) are established. Indeed, $\wedge$ and $\vee$ also refer to the minimum and maximum operators respectively.

Take the following example of an expression.

Example 4.1.1.

$$2x_1 \ominus (4x_2 \oplus 3x_3) \ominus (5x_3 \oplus x_4).$$

This is in conjunctive normal form. By means of the identities (4.3)-(4.6), we transform it into a disjunctive normal form as follows.

$$\begin{aligned}
& ((2x_1 \ominus 4x_2) \oplus (2x_1 \ominus 3x_3)) \ominus (5x_3 \oplus x_4) \\
\overset{(4.6)}{=} \; & ((2x_1 \ominus 4x_2) \ominus (5x_3 \oplus x_4)) \oplus ((2x_1 \ominus 3x_3) \ominus (5x_3 \oplus x_4)) \\
\overset{(4.6)}{=} \; & ((2x_1 \ominus 4x_2) \ominus 5x_3) \oplus ((2x_1 \ominus 4x_2) \ominus x_4) \oplus \\
& \oplus((2x_1 \ominus 3x_3) \ominus 5x_3) \oplus ((2x_1 \ominus 3x_3) \ominus x_4) \\
\overset{(4.4)}{=} \; & (2x_1 \ominus 4x_2 \ominus 5x_3) \oplus (2x_1 \ominus 4x_2 \ominus x_4) \oplus (2x_1 \ominus 3x_3) \oplus \\
& \oplus(2x_1 \ominus 3x_3 \ominus x_4).
\end{aligned}$$

The last expression is in disjunctive normal form, and we can also transform this to the conjunctive normal form through similar steps.

Note that this transformation between conjunctive and disjunctive normal forms seems (and can become) cumbersome, so a more efficient method might be more appropriate.

Definition 4.1.2. A *maxmin-plus function of dimension* N is a mapping $\mathcal{M} : \mathbb{R}^N \to \mathbb{R}^N$, where the components $\mathcal{M}_i$ of $\mathcal{M}$ are maxmin-plus expressions of the N variables $x_1, x_2, \ldots, x_N$.

Thus, the maxmin-2 system in (4.8) is a maxmin-plus function of dimension 3. In Chapter 2, we introduced the idea of a topical function as one which satisfies the properties of monotonicity, homogeneity and nonexpansiveness. It can be checked that $\mathcal{M}$ is also topical, the implication of which will be seen when we discuss the asymptotic behaviour of $\mathcal{M}$.

Definition 4.1.3. Let $\mathcal{M}$ be a maxmin-plus function of dimension N. If the components $\mathcal{M}_i(\mathbf{x})$ are written in conjunctive normal form, then we can formally write $\mathcal{M}(\mathbf{x}) = \min_{j\in J}(P_j \otimes \mathbf{x})$, where J is a finite set and $P_j \in \mathbb{R}_{\max}^{N\times N}$. Such a representation is called a *max-representation* of $\mathcal{M}$. If $\mathcal{M}$ is rewritten as $\max_{j\in J'}(Q_j \odot \mathbf{x})$, with J' being a finite set and $Q_j \in \mathbb{R}_{\min}^{N\times N}$, then such a representation is called a *min-representation* of $\mathcal{M}$. Here, the max and min operations are carried out componentwise.

The matrix P_j in a max-representation of $\mathcal{M}$ will be referred to as a *max-projection* of $\mathcal{M}$; the matrix Q_j in a min-representation of $\mathcal{M}$ will be termed a *min-projection* of $\mathcal{M}$.

4.1.2 The general maxmin-m system

We define the following.

Definition 4.1.4. A *maxmin-plus system* of dimension N is a system with state $\mathbf{x}(k) = (x_1(k), \ldots, x_N(k))^\top$, which evolves according to the equation $\mathbf{x}(k+1) = \mathcal{M}(\mathbf{x}(k))$, $k \geq 0$, where $\mathcal{M}$ is a maxmin-plus function of dimension N.

The maxmin-plus systems arising in our model are the maxmin-m systems for the cases $1 \leq m \leq n$. For a fixed n and N, we obtain a general max-representation of the maxmin-m system by first thinking about $x_{(m)}(k)$ (or simply $x_{(m)}$ for short). Given epoch k, let the n input arrival times from the neighbourhood of node i be compactly labelled $x_{i_1}, x_{i_2}, \ldots, x_{i_n}$, where $x_{i_j} = \tau_{ij} \otimes x_j$. As previously, we refer to this set of input times as $\mathcal{N}_{x_i}$. Then, the mth smallest of these quantities is obtained as follows. From the n input times, first take the largest of each set S_j of m input times; choosing m from n input times ($m \leq n$) means that there are ${}^nC_m$ or $\begin{pmatrix} n \\ m \end{pmatrix} = \frac{n!}{m!(n-m)!}$ such sets and, correspondingly, such 'largest input times'. The required quantity is obtained by taking the smallest of these values. This can also be written as min{max{m input times from the n-nbhd}} which can be interpreted using maxmin-plus notation as follows. First, denote the m elements in the set S_j by $y_1^j, y_2^j, \ldots, y_m^j$ $(1 \leq j \leq {}^nC_m)$. Then

$$x_{(m)} = \min_{1 \leq j \leq {}^nC_m} \{\bigoplus_{h=1}^{m} y_h^j\}. \tag{4.12}$$

The evolution equation for node i is then just Equation (4.1). Note that y_h^j is one of the elements in $\mathcal{N}_{x_i}$; if it represents input from neighbourhood node l, then y_h^j takes the form $\tau_{il} x_l = \tau_{il} \otimes x_l$.

For fixed $n, N \in \mathbb{N}$, we can now write a max-representation of the general maxmin-m system as:

$$\mathbf{x}(k+1) = P_1 \otimes \mathbf{x}(k) \ominus P_2 \otimes \mathbf{x}(k) \ominus \cdots \ominus P_{{}^nC_m} \otimes \mathbf{x}(k) \tag{4.13}$$

where each non-zero entry in row i of P_j corresponds to an element of S_j. To be precise, element (i, l) of the max-projection $P_j \in \mathbb{R}_{\max}^{N\times N}$ $(j = 1, \ldots, {}^nC_m)$ is

$$P_j(i,l) = \begin{cases} \xi_i \tau_{il} & \text{if } \tau_{il} x_l(k) \in S_j \\ \varepsilon & \text{otherwise.} \end{cases}$$

This is clear to see by observing the entries of the example max-projections P_1, P_2, P_3 in Section 4.1.1.

The same example identified that the maxmin-2 system was not unique. The following lemma is applicable to any n-nbhd network.

Lemma 4.1.1. Consider the maxmin-m system on an n-nbhd network of size N. The number of max-representations of this system is $({}^nC_m!)^N$.

Proof. Consider the max-representation of the system with max-projections P_j for $j = 1, \ldots, {}^nC_m$. We can swap row i of P_j with the same row in $P_{j'}$ $(j \neq j')$ to obtain the same evolution equation for $x_i(k)$ and, indeed, the system. Since there are ${}^nC_m$ max-projections, we find that, in general, there are $({}^nC_m)^N$ ways to represent P_1. Having chosen P_1, there are $({}^nC_m - 1)^N$ ways to represent P_2; similarly, there are $({}^nC_m - 2)^N$ ways to represent P_3, …, 2^N ways to represent $P_{{}^nC_m-1}$ and 1 way of representing $P_{{}^nC_m}$. Thus, the total number of max-representations of the maxmin-m system is

$$({}^nC_m)^N \times ({}^nC_m - 1)^N \times \cdots \times 2^N \times 1 = ({}^nC_m!)^N$$

□

Some of these max-representations will be the same, e.g. we can swap the whole of the matrices P_1 and P_2 to get another representation but one which produces the same maxmin-m system, i.e. the same evolution equations in matrix-vector form. It turns out then that we can obtain the number M of distinct ways of representing this maxmin-m system by dividing by the number of ways of arranging the matrices P_j:

$$M = \frac{({}^nC_m!)^N}{{}^nC_m!} = ({}^nC_m!)^{N-1}.$$

Evidently, M can become impractically large, even for small N. The impact of this is discussed next.

4.1.3 Asymptotic behaviour of the maxmin-m system

The quantities that characterise the asymptotic behaviour of the maxmin-m system are indeed the same as those in the max-plus and min-plus systems, namely the cycletime, cyclicity, eigenvalue and eigenvector. In the same way that we associated the contour plot with these quantities for the max-plus system, we can obtain the contour plot for the maxmin-m system also. As this is a pictorial augmentation, we focus on the work required beforehand, which also turns out to be more intricate than the work done to understand the asymptotic behaviour in the max-plus and min-plus system.

Let us define the cycletime in maxmin-plus algebra. If $\mathcal{M}$ is an N-dimensional maxmin-plus function, we denote by $\mathcal{M}^p$ the action of applying $\mathcal{M}$ to a vector $\mathbf{x} \in \mathbb{R}^N$ a total p times, where p is a positive integer, i.e. $\mathcal{M}^p(\mathbf{x}) = \underbrace{\mathcal{M}(\mathcal{M}(\ldots(\mathcal{M}}_{p \text{ times}}(\mathbf{x})))).$

Definition 4.1.5. Let $\mathcal{M}$ be an N-dimensional maxmin-plus function. If it exists, the *cycletime vector* of $\mathcal{M}$ is $\chi(\mathcal{M})$ and is defined as $\chi(\mathcal{M}) = \lim_{k\to\infty}(\mathcal{M}^k(\mathbf{x})/k)$.

For short, we sometimes denote $\chi(\mathcal{M})$ as χ. So when does the cycletime vector exist? It turns out that a sufficient condition for its existence is if $\mathcal{M}$ has an eigenvalue. Define this as follows.

Definition 4.1.6. The vector $\mathbf{x} \in \mathbb{R}^N$ is called an *eigenvector* for *eigenvalue* $\lambda \in \mathbb{R}$ if $\mathcal{M}(\mathbf{x}) = \lambda \otimes \mathbf{x}$.

The existence of an eigenvalue λ of $\mathcal{M}$ not only ensures the existence of the cycletime vector χ, but it also implies that χ has all elements equal to λ. Indeed, suppose λ is an eigenvalue of $\mathcal{M}$. Then there exists an eigenvector $\mathbf{x}$ such that $\mathcal{M}(\mathbf{x}) = \lambda \otimes \mathbf{x}$. Thus, for a positive integer k, $\mathcal{M}^k(\mathbf{x}) = \lambda^{\otimes k} \otimes \mathbf{x}$. Dividing by k and letting k tend to infinity yields $\lim_{k\to\infty} \mathcal{M}^k(\mathbf{x})/k = \lambda$, where we denote by λ the vector with all elements λ. In such a case as when all elements of χ are equal, we also refer to it simply as the cycletime.

Once the cycletime vector exists, the following result becomes significant.

Theorem 4.1.1 ([22], Theorem 12.7)**.** If the cycletime vector $\chi(\mathcal{M})$ of a maxmin-plus function $\mathcal{M}$ exists for some finite vector $\mathbf{x}$ (i.e. where all elements of $\mathbf{x}$ are finite), then it exists for all finite vectors $\mathbf{x}$ and $\chi(\mathcal{M})$ is independent of the initial condition $\mathbf{x}$.

Proof. See [22], Theorem 12.7. □

Note that the theorem uses a fully finite vector $\mathbf{x}$. This ensures uniqueness of the cycletime vector due to the existence of a limit as $k \to \infty$.

The task is to find this unique cycletime vector for our maxmin-m system. For a max-plus or min-plus system, we could identify χ using the connectivity graph and observing circuits and their weights. This would also yield the cyclicity as a by-product. This is not as straightforward for the maxmin-m system since there is more than just the one underlying connectivity graph. This can be seen in (4.13) where there are ${}^nC_m$ max-projections of the general maxmin-m system. Each of these projections can be identified as the adjacency matrix of a subgraph of the original network. If $\mathcal{G}_j$ is the connectivity graph of the max-projection P_j, then a similar graph theoretical method would compute the cycletime vector as some function of $\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_{{}^nC_m}$. However, even if such a method did exist, there is no guarantee that this would yield the exact cycletime vector since the matrices P_j themselves are not unique. Moreover, the number of distinct max-representations M of the maxmin-m system in the form of (4.13) can become very large, peaking at $m \approx n/2$. As a particular example, take the network size $N = 10$ and neighbourhood size $n = 7$. Then the maxmin-2 system on this network would have $M \approx 2.3719 \times 10^{177}$ max-representations, a large number for a relatively small system.

Despite this potential of a dauntingly large M, we conjecture that at least one of these max-representations corresponds to the exact cycletime vector when calculated analytically. The analytical method used is the maxmin-m analogue to the graph theoretical one used for the max-plus and min-plus systems. We shall describe this after the following.

We can, in fact, use Theorem 4.1.1 and run the maxmin-m system for any initial condition to observe the limiting behaviour and, thus, the cycletime vector (if it exists). The following defines a periodic regime for a maxmin-plus system, thereby enabling us to see how such an iterative method for finding limiting quantities would practically work.

Definition 4.1.7. Let $\mathcal{M}$ be a maxmin-plus function of dimension N. For some $k \geq 0$, consider the set of vectors

$$\mathbf{x}(k), \mathbf{x}(k+1), \mathbf{x}(k+2), \ldots \in \mathbb{R}^N$$

where $\mathbf{x}(n) = \mathcal{M}^n(\mathbf{x}(0))$ for all $n \geq 0$. The set $x_i(k), x_i(k+1), x_i(k+2), \ldots$ (for $i = 1, \ldots, N$) is called a *periodic regime* of i if there exists $\mu_i \in \mathbb{R}$ and a finite number $\rho_i \in \mathbb{N}$ such that

$$x_i(k+\rho_i) = \mu_i \otimes x_i(k).$$

The *period* of the regime is ρ_i and $\chi_i = \mu_i/\rho_i$ is the *cycletime* of i.

Notice the difference between this definition of periodic regime to that given for the max-plus and min-plus systems. There, we expressed the periodic regime in terms of the vector $\mathbf{x}(k)$ and a scalar acting on all the elements. Here, we provide a definition in terms of each element in $\mathbf{x}(k)$ to accommodate the possibility of the cycletime begin different for each node i. Indeed, this might also occur in the max-plus and min-plus systems, particularly when the underlying connectivity graph is not strongly connected (more on this later). Thus, Definition 4.1.7 can be seen to provide the general definition of a periodic regime, even for the max-plus and min-plus systems. The cycletime vector can now be written as $\chi = (\chi_1, \chi_2, \cdots, \chi_N)^\top$.

Although we have talked about the period of the maxmin-plus system $\mathcal{M}$, we have refrained from studying its cyclicity. The definition of the cyclicity of $\mathcal{M}$ follows from Corollary 2.1.2, which gives the cyclicity of the max-plus system as an upper bound for the period obtained with any initial condition. We adopt this here too. However, as mentioned earlier, the cyclicity of a max-plus or min-plus system can be analytically obtained from its underlying connectivity graph; it is not so simple for the maxmin-plus system since there is not one underlying connectivity graph. We propose that an optimal max-representation of the maxmin-m system is one which enables the provision of the exact cycletime analytically.

Definition 4.1.7 provides an iterative means for finding the cycletime vector and period, i.e. we iterate the system until a periodic regime is observed. For a large network size N, this would typically be done computationally. Ideally, we would like to find the cycletime vector and cyclicity (or period) analytically without the need for such computing power. However, recall that the number of max-representations M of maxmin-m is impractically large (even for smallish N) so finding the optimal max-representation analytically would be akin to finding a needle in a haystack, even with some intuition gained from the max-projections. With this in mind, there is an analytical method, known as the *duality theorem*, for obtaining the cycletime vector χ. For the maxmin-m system, it usually finds only upper and lower bounds to χ, but it employs the maxmin-plus analogue of a graph theoretical method since it requires the connectivity graphs $\mathcal{G}(P_j)$ of each max-projection P_j. The cyclicity of the maxmin-m system will follow from this discussion.

Duality theorem

The duality theorem was conjectured by Jeremy Gunawardena in [16] and later proved by the same author in [13]. The theorem uses a max-representation and min-representation of the maxmin-plus function $\mathcal{M}$ as given in Definition 4.1.3, and states that the cycletime vector of $\mathcal{M}$ satisfies

$$\max_{j\in J'}(Q_j \odot \mathbf{x}) \leq \chi(\mathcal{M}) \leq \min_{j\in J}(P_j \otimes \mathbf{x}).$$

In particular, it was conjectured in [16] that there exist finite sets of the matrices P_j and $Q_{j'}$ such that the above inequality becomes an equality.

In this section, we shall detail an outline of the steps that yield the duality theorem and relate it to the maxmin-m system. Due to its frequent use, we shall occasionally denote the duality theorem compactly as DT. Consider the max-representation given in Equation (4.13). By means of the identities (4.3)-(4.6), we can transform this into a min-representation. However, we show here a nice property arising from the maxmin-m model, which is that of a method for obtaining the min-representation from first principles and requiring no transformation from the max-representation.

To make clearer the differences between the max-representation and min-representation, we take a 4-nbhd example and consider the maxmin-2 system . Let the ordered set of input times at node i be $\{x_a, x_b, x_c, x_d\}$. Then, applying the earlier notion in (4.12), the 2nd input time is given by

$$x_{(2)} = (x_a \oplus x_b) \ominus (x_a \oplus x_c) \ominus (x_a \oplus x_d) \ominus (x_b \oplus x_c) \ominus (x_b \oplus x_d) \ominus (x_c \oplus x_d). \quad (4.14)$$

This is a max-representation. We obtain this form by first considering sets of m input times and taking their maxima. Now, we approach it from the 'dual' angle, which is to consider sets of $n-(m-1)$ inputs and taking their minima. It is based on the fact that the mth smallest element in a set of n is also the $(n-(m-1))$th largest element. Thus, we can state the max-representation of the second input time as

$$x_{(2)} = (x_a \ominus x_b \ominus x_c) \oplus (x_b \ominus x_c \ominus x_d) \oplus (x_c \ominus x_d \ominus x_a) \oplus (x_d \ominus x_a \ominus x_b). \quad (4.15)$$

It can be checked that both representations give $x_{(2)} = x_b$, as required.

In general, we proceed as in Section 4.1.2 and let x_{i_j} denote the input time from node j to node i, i.e. $x_{i_j} = \tau_{ij} \otimes x_j$ or more compactly as $\tau_{ij}x_j$. We choose all sets of $n-(m-1)$ elements from $\mathcal{N}_{x_i}$; let S_j denote the jth such set. Choosing $n-(m-1)$ from n elements gives ${}^nC_{(n-(m-1))}$ or $\begin{pmatrix} n \\ n-(m-1) \end{pmatrix} = \frac{n!}{(n-(m-1))!(m-1)!}$ sets. For $j = 1, 2, \ldots, {}^nC_{(n-(m-1))}$, denote the $n-(m-1)$ elements in the set S_j by $y_1^j, y_2^j, \ldots, y_{n-(m-1)}^j$. Then the min-representation for the mth input time at node i

is

$$x_{(m)} = \bigoplus_{j=1}^{{}^nC_{(n-(m-1))}} \{\ominus_{h=1}^{n-(m-1)} y_h^j\}. \tag{4.16}$$

The evolution equation for the update time of node i is then Equation (4.1) where $x_{(m)}$ can now be regarded as either (4.12) or (4.16). If we take the latter form for the mth input time, then we obtain the min-representation of the evolution equation. Remember that each element y_h^j of S_j is also one of the elements in $\mathcal{N}_{x_i}$; if it represents input from neighbourhood node l, then y_h^j takes the form $\tau_{il}x_l$.

For fixed $n, N \in \mathbb{N}$, we can now write a min-representation of the general maxmin-m system as:

$$\mathbf{x}(k+1) = Q_1 \odot \mathbf{x}(k) \oplus Q_2 \odot \mathbf{x}(k) \oplus \cdots \oplus Q_{{}^nC_{(n-(m-1))}} \odot \mathbf{x}(k) \tag{4.17}$$

where each non-zero entry in row i of Q_j corresponds to an element of S_j. To be precise, element (i,l) of the min-projection $Q_j \in \mathbb{R}_{\min}^{N\times N}$ $(j = 1, \ldots, {}^nC_{(n-(m-1))})$ is

$$Q_j(i,l) = \begin{cases} \xi_i\tau_{il} & \text{if } \tau_{il}x_l(k) \in S_j \\ \varepsilon' & \text{otherwise.} \end{cases}$$

Let us highlight the differences between the max-representation and min-representation using the following example.

Example 4.1.2. Consider a 4-nbhd network of N nodes such that $\mathcal{N}_i = \{i-1, i, i+1, i+2\}$ for all i. The timing dependency matrix of this network is

$$P = \begin{pmatrix} \xi_1\tau_{11} & \xi_1\tau_{12} & \xi_1\tau_{13} & \varepsilon & \cdots & \varepsilon & \xi_1\tau_{1N} \\ \xi_2\tau_{21} & \xi_2\tau_{22} & \xi_2\tau_{23} & \xi_2\tau_{24} & \cdots & \varepsilon & \varepsilon \\ \vdots & & \ddots & & & & \vdots \\ \xi_{N-1}\tau_{N-1,1} & \varepsilon & \cdots & \varepsilon & \xi_{N-1}\tau_{N-1,N-2} & \xi_{N-1}\tau_{N-1,N-1} & \xi_{N-1}\tau_{N-1,N} \\ \xi_N\tau_{N1} & \xi_N\tau_{N2} & \cdots & \varepsilon & \varepsilon & \xi_N\tau_{N,N-1} & \xi_N\tau_{NN} \end{pmatrix}.$$

A max-representation of the maxmin-2 system on this network is given by

$$\mathbf{x}(k+1) = P_1 \otimes \mathbf{x}(k) \ominus P_2 \otimes \mathbf{x}(k) \ominus P_3 \otimes \mathbf{x}(k) \oplus P_4 \otimes \mathbf{x}(k) \ominus P_5 \otimes \mathbf{x}(k) \ominus P_6 \otimes \mathbf{x}(k)$$

where P_1, P_2 and P_3 are exactly those matrices in (4.9)-(4.11) and $P_4, P_5, P_6 \in \mathbb{R}_{\max}^{N\times N}$

are given as follows.

$$P_4 = \left(\begin{array}{ccccccc} \xi_1\tau_{11} & \varepsilon & \xi_1\tau_{13} & \varepsilon & \cdots & \varepsilon & \varepsilon \\ \varepsilon & \xi_2\tau_{22} & \varepsilon & \xi_2\tau_{24} & \cdots & \varepsilon & \varepsilon \\ \vdots & & \ddots & & & & \vdots \\ \xi_{N-1}\tau_{N-1,1} & \varepsilon & \cdots & \varepsilon & \varepsilon & \xi_{N-1}\tau_{N-1,N-1} & \varepsilon \\ \varepsilon & \xi_N\tau_{N2} & \cdots & \varepsilon & \varepsilon & \varepsilon & \xi_N\tau_{NN} \end{array} \right),$$

$$P_5 = \left(\begin{array}{ccccccc} \varepsilon & \varepsilon & \xi_1\tau_{13} & \varepsilon & \cdots & \varepsilon & \xi_1\tau_{1N} \\ \xi_2\tau_{21} & \varepsilon & \varepsilon & \xi_2\tau_{24} & \cdots & \varepsilon & \varepsilon \\ \vdots & & \ddots & & & & \vdots \\ \xi_{N-1}\tau_{N-1,1} & \varepsilon & \cdots & \varepsilon & \xi_{N-1}\tau_{N-1,N-2} & \varepsilon & \varepsilon \\ \varepsilon & \xi_N\tau_{N2} & \cdots & \varepsilon & \varepsilon & \xi_N\tau_{N,N-1} & \varepsilon \end{array} \right),$$

$$P_6 = \left(\begin{array}{ccccccc} \varepsilon & \xi_1\tau_{12} & \xi_1\tau_{13} & \varepsilon & \cdots & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \xi_2\tau_{23} & \xi_2\tau_{24} & \cdots & \varepsilon & \varepsilon \\ \vdots & & \ddots & & & & \vdots \\ \xi_{N-1}\tau_{N-1,1} & \varepsilon & \cdots & \varepsilon & \varepsilon & \varepsilon & \xi_{N-1}\tau_{N-1,N} \\ \xi_N\tau_{N1} & \xi_N\tau_{N2} & \cdots & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{array} \right).$$

There are six max-projections since ${}^4C_2 = 6$. A min-representation of this maxmin-2 system is

$$\mathbf{x}(k+1) = Q_1 \odot \mathbf{x}(k) \oplus Q_2 \odot \mathbf{x}(k) \oplus Q_3 \odot \mathbf{x}(k) \oplus Q_4 \odot \mathbf{x}(k)$$

where $Q_1, \ldots, Q_4 \in \mathbb{R}_{\min}^{N \times N}$, given as follows.

$$Q_1 = \left(\begin{array}{ccccccc} \xi_1\tau_{11} & \xi_1\tau_{12} & \varepsilon' & \varepsilon' & \cdots & \varepsilon' & \xi_1\tau_{1N} \\ \xi_2\tau_{21} & \xi_2\tau_{22} & \xi_2\tau_{23} & \varepsilon' & \cdots & \varepsilon' & \varepsilon' \\ \vdots & & \ddots & & & & \vdots \\ \varepsilon' & \varepsilon' & \cdots & \varepsilon' & \xi_{N-1}\tau_{N-1,N-2} & \xi_{N-1}\tau_{N-1,N-1} & \xi_{N-1}\tau_{N-1,N} \\ \xi_N\tau_{N1} & \varepsilon' & \cdots & \varepsilon' & \varepsilon' & \xi_N\tau_{N,N-1} & \xi_N\tau_{NN} \end{array} \right),$$

$$Q_2 = \left(\begin{array}{ccccccc} \xi_1\tau_{11} & \xi_1\tau_{12} & \xi_1\tau_{13} & \varepsilon' & \cdots & \varepsilon' & \varepsilon' \\ \varepsilon' & \xi_2\tau_{22} & \xi_2\tau_{23} & \xi_2\tau_{24} & \cdots & \varepsilon' & \varepsilon' \\ \vdots & & \ddots & & & & \vdots \\ \xi_{N-1}\tau_{N-1,1} & \varepsilon' & \cdots & \varepsilon' & \varepsilon' & \xi_{N-1}\tau_{N-1,N-1} & \xi_{N-1}\tau_{N-1,N} \\ \xi_N\tau_{N1} & \xi_N\tau_{N2} & \cdots & \varepsilon' & \varepsilon' & \varepsilon' & \xi_N\tau_{NN} \end{array} \right),$$

$$Q_3 = \begin{pmatrix} \varepsilon' & \xi_1\tau_{12} & \xi_1\tau_{13} & \varepsilon' & \cdots & \varepsilon' & \xi_1\tau_{1N} \\ \xi_1\tau_{11} & \varepsilon' & \xi_2\tau_{23} & \xi_2\tau_{24} & \cdots & \varepsilon' & \varepsilon' \\ \vdots & & \ddots & & & & \vdots \\ \xi_{N-1}\tau_{N-1,1} & \varepsilon' & \cdots & \varepsilon' & \xi_{N-1}\tau_{N-1,N-1} & \varepsilon' & \xi_{N-1}\tau_{N-1,N} \\ \xi_N\tau_{N1} & \xi_N\tau_{N2} & \cdots & \varepsilon' & \varepsilon' & \xi_N\tau_{N,N-1} & \varepsilon' \end{pmatrix},$$

$$Q_4 = \begin{pmatrix} \xi_1\tau_{11} & \varepsilon' & \xi_1\tau_{13} & \varepsilon' & \cdots & \varepsilon' & \xi_1\tau_{1N} \\ \xi_1\tau_{11} & \xi_1\tau_{12} & \varepsilon' & \xi_2\tau_{24} & \cdots & \varepsilon' & \varepsilon' \\ \vdots & & \ddots & & & & \vdots \\ \xi_{N-1}\tau_{N-1,1} & \varepsilon' & \cdots & \varepsilon' & \xi_{N-1}\tau_{N-1,N-1} & \xi_{N-1}\tau_{N-1,N-1} & \varepsilon' \\ \varepsilon' & \xi_N\tau_{N2} & \cdots & \varepsilon' & \varepsilon' & \xi_N\tau_{N,N-1} & \xi_N\tau_{NN} \end{pmatrix}.$$

Note here that there are four min-projections since ${}^nC_{(n-(m-1)} = {}^4C_3 = 4$. Recall also that the two representations above are not unique since the projections P_j and $Q_{j'}$ are not unique; we could have chosen any one of the $M = ({}^4C_2)!^{N-1} = 720^{N-1}$ number of max-representations of the system. Similarly, the min-representation could have been formed in any one of $({}^4C_3)!^{N-1} = 24^{N-1}$ ways.

Let $\mathcal{P}$ denote the set of all max-projections of the maxmin-plus system $\mathcal{M}$. Since the max-representation of the maxmin-m system as given in Equation (4.13) is not unique, we can, in fact, extend it to include *all* max-projections $P_j \in P$ so that we do obtain one and *only one* max-representation, as ideally required. This doesn't affect the maxmin-m system since taking all max-projections ensures the existence of the minimal ${}^nC_m$ projections required; in the evolution equation of $x_i(k)$ arising from the max-representation (4.13), there will only be ${}^nC_m$ distinct max-plus terms, each corresponding to the ith row of the ${}^nC_m$ max-projections. The ith row of any of those max-projections that were unused in (4.13) will be equal to the ${}^nC_m$ rows mentioned. Thus, the general max-representation of any maxmin-plus system $\mathcal{M}$ is given by

$$\mathcal{M}(\mathbf{x}) = \min_{P_j \in \mathcal{P}} (P_j \otimes \mathbf{x}). \tag{4.18}$$

If $\mathcal{M}$ is the maxmin-m system, then

$$\mathcal{M}(\mathbf{x}) = \mathbf{x}(k+1) = P_1 \otimes \mathbf{x}(k) \ominus P_2 \otimes \mathbf{x}(k) \ominus \cdots \ominus P_{|\mathcal{P}|} \otimes \mathbf{x}(k)$$

where $|\mathcal{P}| = ({}^nC_m)^N$. This can be compared with Equation (4.13), which was the original general evolution equation for the maxmin-m system. Let $\mathcal{Q}$ denote the set of all min-projections of $\mathcal{M}$ (so $\mathcal{Q}$ contains $({}^nC_{(n-(m-1))})^N$ matrices if $\mathcal{M}$ is a maxmin-m system). We shall now use the terms in the following definition.

Definition 4.1.8. A maxmin-plus system $\mathcal{M}$ written in the form $\min_{j\in J}(P_j \otimes \mathbf{x})$ where $|J| < |\mathcal{P}|$ is *a max-representation of* $\mathcal{M}$; if it is written in the form of (4.18), we call it the *full max-representation of* $\mathcal{M}$. If we write

$$\mathcal{M}(\mathbf{x}) = \max_{Q_{j'}\in\mathcal{Q}}(Q_{j'} \odot \mathbf{x}),$$

then we call this the *full min-representation of* $\mathcal{M}$. If the evolution equation is written as $\max_{j\in J'}(Q_j \otimes \mathbf{x})$ where $|J'| < |\mathcal{Q}|$, then this is called *a min-representation of* $\mathcal{M}$.

Thus, (4.13) is a max-representation of the maxmin-m system while (4.18) is the full max-representation of maxmin-m.

Then, for any $P_j \in \mathcal{P}$ and any $\mathbf{x} \in \mathbb{R}^N$, it follows from (4.18) that

$$\mathcal{M}(\mathbf{x}) \leq P_j \otimes \mathbf{x}.$$

Hence, using the monotonicity property of $\mathcal{M}$,

$$\begin{aligned} \mathcal{M}^2(\mathbf{x}) = \mathcal{M}(\mathcal{M}(\mathbf{x})) &\leq \mathcal{M}(P_j \otimes \mathbf{x}) \\ &\leq P_j \otimes (P_j \otimes \mathbf{x}) = P_j^{\otimes 2} \otimes \mathbf{x}. \end{aligned}$$

Continuing in this way gives $\mathcal{M}^k(\mathbf{x}) \leq P_j^{\otimes k} \otimes \mathbf{x}$ so that $\frac{\mathcal{M}^k(\mathbf{x})}{k} \leq \frac{P_j^{\otimes k}\otimes\mathbf{x}}{k}$ and, for $k \to \infty$, $\chi(\mathcal{M}) \leq \chi(P_j)$. This is true for any j, so we take j^* such that P_{j^*} has the smallest cycletime, i.e.

$$\chi(\mathcal{M}) \leq \min_{P_j\in\mathcal{P}} \chi(P_j). \tag{4.19}$$

By applying the same procedure on the full min-representation of $\mathcal{M}$, we obtain another inequality. Thus, the cycletime of $\mathcal{M}$ is an upper bound for the cycletime vector of all $Q_{j'} \in \mathcal{Q}$, i.e.

$$\max_{Q_{j'}\in\mathcal{Q}} \chi(Q_{j'}) \leq \chi(\mathcal{M}). \tag{4.20}$$

Therefore, combining (4.19) and (4.20) gives

$$\max_{Q_{j'}\in\mathcal{Q}} \chi(Q_{j'}) \leq \chi(\mathcal{M}) \leq \min_{P_j\in\mathcal{P}} \chi(P_j). \tag{4.21}$$

The duality theorem consequently proves this to be an equality for at least one max-projection P_j and one min-projection $Q_{j'}$ [13]. Indeed, this is easy to prove when $\mathcal{M}$ has an eigenvalue. Recall from the previous section that if $\mathcal{M}$ has an eigenvalue λ, then this ensures the existence of the cycletime vector, whose elements are all equal to λ.

Theorem 4.1.2. If $\mathcal{M}$ has an eigenvalue, then the duality conjecture holds.

Proof. Let λ and $\mathbf{v}$ denote the eigenvalue and a corresponding eigenvector respectively, i.e. $\mathcal{M}(\mathbf{v}) = \lambda \otimes \mathbf{v}$. Then the cycletime vector $\chi(\mathcal{M})$ exists and has all elements equal to λ. (Also, by Theorem 4.1.1, χ is unique). Thus, for at least one j, $\mathcal{M}(\mathbf{v}) = P_j \otimes \mathbf{v}$ so that $\chi(\mathcal{M}) = \chi(P_j)$. Similarly, $\chi(\mathcal{M}) = \chi(Q_{j'})$ for some j'. This gives equality in (4.21). □

If an eigenvalue of $\mathcal{M}$ exists, then it is also the cycletime. Combined with Theorem 4.1.1, which says that the cycletime vector is unique, this implies that the eigenvalue is also unique, i.e. there is one and only one eigenvalue of a maxmin-plus system if it exists.

We complete this section with a comment on the cyclicity of the maxmin-plus system. Notice that the proof to Theorem 4.1.2 expressed the maxmin-plus system $\mathcal{M}(\mathbf{v})$ as a max-plus system $P_j \otimes \mathbf{v}$ for some max-projection P_j. This implies that the cyclicity of $\mathcal{M}$ is the cyclicity of the max-plus system and this is then straightforward to obtain since we can then apply the methods in Section 2.1. Complications arise when an eigenvalue is not known to exist since that suggests that we can't represent $\mathcal{M}$ as a max-plus system.

We may employ an iterative method. Since it is unique, the cycletime vector can be found simply by running the system for any arbitrary initial condition. However, a similar method for finding the cyclicity is as yet unknown, and so we usually resort to computational methods. Nevertheless, the next section will show an analytical means of obtaining ρ, the period of the maxmin-m system $\mathcal{M}$, which depends on $\mathbf{x}(0)$. Consequently, ρ satisfies $\rho \leq \sigma(\mathcal{M})$, where $\sigma(\mathcal{M})$ is the cyclicity of $\mathcal{M}$.

4.2 Formalism for the maxmin-m system: an example

In this section, we shall employ the duality theorem on the smallest network size that allows a maxmin-m model such that $m \neq 1$ and $m \neq n$. The timing dependency graph is thus a regular 3-nbhd network on 3 nodes, and the only maxmin-m system that fulfils the criteria mentioned is when $m = 2$. This will give some understanding of the issues posed by the impractically large number M of max-representations of the system. Importantly, should equality in the DT be achieved, we shall look at the impact of this on the asymptotic behaviour of the update times, which would then be modelled by a max-plus system.

4.2.1 Duality theorem for the maxmin-m system

Consider the timing dependency graph of this system, depicted in Figure 4.1, for which the timing dependency matrix is

$$P = \begin{pmatrix} 9 & 9 & 2 \\ 4 & 9 & 1 \\ 5 & 5 & 8 \end{pmatrix}.$$

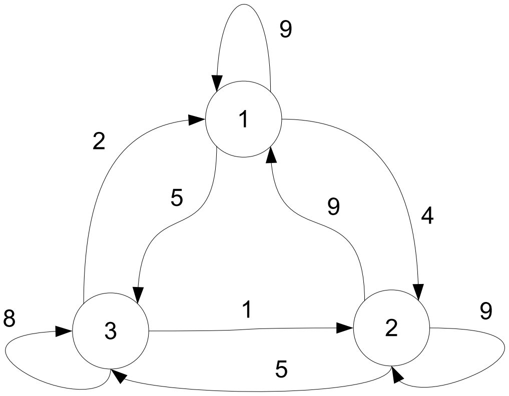


Figure 4.1: Regular 3-nbhd network on 3 nodes.

Let us apply the duality theorem on the maxmin-2 system described above. A max-representation for the evolution equation is

$$\mathbf{x}(k+1) = P_1 \otimes \mathbf{x}(k) \ominus P_2 \otimes \mathbf{x}(k) \ominus P_3 \otimes \mathbf{x}(k), \tag{4.22}$$

where

$$P_1 = \begin{pmatrix} 9 & 9 & \varepsilon \\ \varepsilon & 9 & 1 \\ 5 & \varepsilon & 8 \end{pmatrix}, \quad P_2 = \begin{pmatrix} 9 & \varepsilon & 2 \\ 4 & 9 & \varepsilon \\ \varepsilon & 5 & 8 \end{pmatrix}, \quad P_3 = \begin{pmatrix} \varepsilon & 9 & 2 \\ 4 & \varepsilon & 1 \\ 5 & 5 & \varepsilon \end{pmatrix}.$$

For the case $(n, m) = (3, 2)$, it turns out that the min-representation may be obtained using the same matrices as P_i above but simply changing the ε entries to ε' (i.e. the max-plus zeros to min-plus zeros). This is because the number of max-projections coincides with the number of min-projections, i.e. ${}^nC_m = {}^nC_{(n-(m-1))}$ when $(n, m) = (3, 2)$. For a fixed i, denote by Q_i the min-projection that has the same non-zero entries as P_i. It is easy to find cycletimes of these matrices since both Q_i and P_i share the same communication graph. The three communication graphs are given in

Figure 4.2.

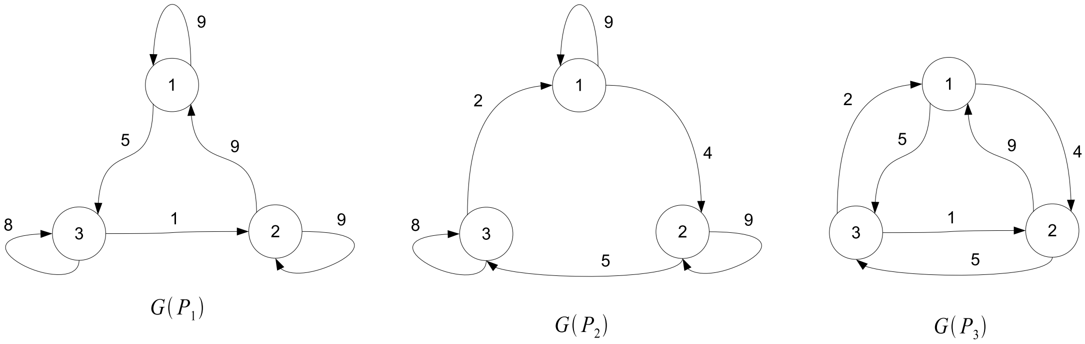


Figure 4.2: Communication graphs of P_1, P_2 and P_3, as given in (4.22)

We now apply the methods from Chapter 2 to find the cycletime vectors of the max-projections. Consider $\mathcal{G}(P_1)$; it is strongly connected, so the cycletime of P_1 is the largest average elementary circuit weight in $\mathcal{G}(P_1)$, i.e. $\chi(P_1) = 9$. Similarly, $\chi(P_2) = 9$ and $\chi(P_3) = 6.5$. By interpreting the graphs in min-plus algebra, we obtain the min-plus cycletimes $\chi(Q_1) = 5$, $\chi(Q_2) = 11/3$, and $\chi(Q_3) = 3$. This gives the following left-hand side to the duality theorem.

$$\chi(Q_1) \oplus \chi(Q_2) \oplus \chi(Q_3) = 5$$

which is characterised by the 3-circuit from node 1 to 3 to 2 back to 1. The right-hand side to the DT is

$$\chi(P_1) \ominus \chi(P_2) \ominus \chi(P_3) = 6.5$$

and this cycletime is characterised by the 2-circuit between nodes 1 and 2. Thus, the cycletime of the maxmin-2 system $\mathcal{M}$ satisfies

$$5 \leq \chi(\mathcal{M}) \leq 6.5. \tag{4.23}$$

We can see that equality is not obtained for this choice of max-representation and min-representation of the system. The reason for this is that we didn't consider the sets of *all* possible max-projections and min-projections of $\mathcal{M}$, i.e. the sets $\mathcal{P}$ and $\mathcal{Q}$, where $|\mathcal{P}| = |\mathcal{Q}| = ({}^3C_2)^N = 27$. Indeed, these 27 matrices could have formed the full max-representation (and min-representation) of the system, for which we would be guaranteed equality in the DT. Even for this small maxmin-m system, we can see that 27 is an impractically large number of matrices to consider before we obtain the exact cycletime $\chi(\mathcal{M})$.

Let us use the alternative and find $\chi(\mathcal{M})$ computationally. Consider the initial condition $\mathbf{x}(0) = (4, 2, 1)^\top$. Then, iterating the maxmin-2 system (4.22) yields the following set of times.

$$\mathbf{x}(0) = \begin{pmatrix} 4 \\ 2 \\ 1 \end{pmatrix}, \mathbf{x}(1) = \begin{pmatrix} 11 \\ 8 \\ 9 \end{pmatrix}, \mathbf{x}(2) = \begin{pmatrix} 17 \\ 15 \\ 16 \end{pmatrix}, \mathbf{x}(3) = \begin{pmatrix} 24 \\ 21 \\ 22 \end{pmatrix}, \mathbf{x}(4) = \begin{pmatrix} 30 \\ 28 \\ 29 \end{pmatrix}, \cdots$$

and it can be seen that, after a transient time $K = 1$, the cycletime is $\chi = 6.5$ with period $\rho = 2$. Since the cycletime has been found to exist for this initial condition, we can apply Theorem 4.1.1 to say that the cycletime is 6.5 for all initial conditions.

We can use this fact now to try and produce equality in the DT even though we have not considered the full set $\mathcal{P}$ (and $\mathcal{Q}$) of all 27 max-projections in (4.22). Consider our current situation in (4.23). We can see that it is only the right-hand side that equals $\chi(M)$ as required, and this was realised from matrix P_3 . Thus, let us keep this matrix fixed and consider a different form for matrices P_1 and P_2. Since the exact cycletime of $\mathcal{M}$ using P_3 was characterised by the 2-circuit connecting nodes 1 and 2, we try to select matrices P_1 and P_2 such that this 2-circuit is obtained. With this is mind, we swap the second rows of P_1 and P_2 to yield new matrices $P_{1'}$ and $P_{2'}$, observing that the aforementioned 2-circuit appears in $\mathcal{G}(P_{1'})$ now. (See also the communication graphs in Figure 4.3).

$$P_{1'} = \begin{pmatrix} 9 & 9 & \varepsilon \\ 4 & 9 & \varepsilon \\ 5 & \varepsilon & 8 \end{pmatrix}, \quad P_{2'} = \begin{pmatrix} 9 & \varepsilon & 2 \\ \varepsilon & 9 & 1 \\ \varepsilon & 5 & 8 \end{pmatrix}. \tag{4.24}$$

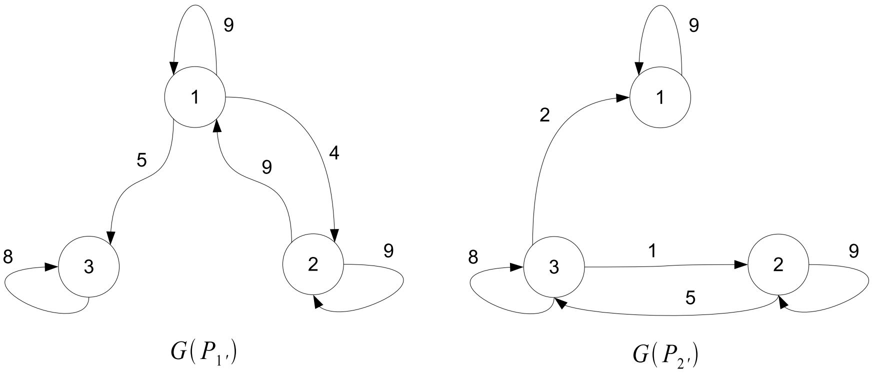


Figure 4.3: Communication graphs of $P_{1'}$ and $P_{2'}$, as given in (4.24)

Notice that the graphs $\mathcal{G}(P_{1'})$ and $\mathcal{G}(P_{2'})$ are not strongly connected. The implication of this is that we cannot simply look for the largest average elementary circuit weight to find the cycletime of such graphs. Indeed, the cycletime vector may not even be uniform now. We shall study this later (in Chapter 5) but, for the purposes of this example, we simply state the cycletime vectors, which do turn out to be uniform here. Thus, $\chi(P_{1'}) = 9$, $\chi(P_{2'}) = 9$ and $\chi(P_3) = 6.5$. Similarly, the corresponding min-projections having the same communication graphs take cycletime $\chi(Q_{1'}) = 6.5$ and $\chi(Q_{2'}) = 3$. Therefore, we have the following left and right hand sides, which now yields equality in the duality theorem, and indeed converges to the cycletime.

$$\begin{aligned}\chi(Q_{1'}) \oplus \chi(Q_{2'}) \oplus \chi(Q_3) &= 6.5\\ \chi(P_{1'}) \ominus \chi(P_{2'}) \ominus \chi(P_3) &= 6.5.\end{aligned}$$

The corollary to this is that we may study the matrices that correspond to this equality to find the cyclicity of the maxmin-m system. In this example, these matrices are P_3 and the new matrix $Q_{1'}$ (obtained as the min-plus transform of $P_{1'}$). By considering either of these matrices, and taking them as max-plus and min-plus matrices in their own right, we can settle on the cyclicity of the maxmin-2 system being 2, as characterised by the 2-circuit between nodes 1 and 2.

The techniques that we employed above was an algorithmic process that converged towards the exact cycletime and cyclicity of $\mathcal{M}$. It was, however, largely based on intuition (e.g. in identifying the circuit that produced the exact cycletime) and therefore, can lead to slower convergence, especially if the network is large. Nevertheless, it provides an analytical method for finding asymptotic behaviour of the maxmin-m system without having to employ the large number M of max-representations, which proves analytically unfeasible.

4.2.2 Reduced max-plus system

The work that has been carried out thus far has determined the cycletime vector $\chi(\mathcal{M})$ of a maxmin-2 system $\mathcal{M}$. The vector obtained is uniform, therefore Theorem 4.1.2 may be applied to say that χ is also the eigenvalue of $\mathcal{M}$. A comment following this theorem suggested that we can express $\mathcal{M}$ as a max-plus system. Let us illustrate this by considering the duality theorem as found in the example above. The max-projection that corresponded to equality is P_3. In other words, for $k \to \infty$, it is implied that

$$\mathcal{M}(\mathbf{x}(k)) = P_3 \otimes \mathbf{x}(k). \tag{4.25}$$

We employ an intuitive approach that attempts to converge towards identifying $\mathcal{M}$ as a max-plus system. Thus, we identify those nodes from the neighbourhood of node i that contribute towards the update time $x_i(k)$ on epoch k. We call these the *affecting nodes* of i. By doing this for all nodes, we obtain N sets of affecting nodes. Let $\mathcal{A}_i(k)$ denote the set of affecting nodes of node i on epoch k. A digraph of affecting nodes on epoch k is obtained by drawing an arc from those nodes in $\mathcal{A}_i(k)$ to node i and repeating this procedure for all nodes. We call such a digraph the *reduced max-plus graph* or simply *reduced graph* of the maxmin-m system on cycle k. The affecting nodes of each node are defined formally as follows.

Definition 4.2.1. Consider the orbit of the initial condition $\mathbf{x}(0)$ for a maxmin-m system on N nodes. Let node j be contained in the neighbourhood of node i. Then, node j is an affecting node of i on epoch k if and only if

$$x_j(k) \otimes \tau_{ij} \leq x_{(m)}(k)$$

where $x_{(m)}(k)$ denotes the k^{th} time of arrival of the mth input at node i, as stated in Equation (4.12).

In other words, on epoch k, the affecting nodes of i are those nodes whose input to i arrived at one of the following two times:

- before the mth input to i
- at the same time as the mth input

Consider the example maxmin-2 system of this section. Up to now, we have not stated transmission and processing times. For clarity then, let us do this here. Let $\xi = (1, 0.5, 2)^\top$ and the transmission matrix be given as

$$T = \begin{pmatrix} 8 & 8 & 1 \\ 3.5 & 8.5 & 0.5 \\ 3 & 3 & 6 \end{pmatrix}.$$

It can be checked that by adding ξ_i to τ_{ij}, we obtain the element P_{ij}. Equivalently, this is the max-plus product of ξ_i and τ_{ij}. Thus, we can multiply the inequality in Definition 4.2.1 by ξ_i to obtain $x_j(k) \otimes \tau_{ij} \otimes \xi_i \leq x_{(m)}(k) \otimes \xi_i$, i.e.

$$x_j(k) \otimes P_{ij} \leq x_{(m)}(k) \otimes \xi_i. \tag{4.26}$$

This property ensures that we do not have to think about the processing and transmission times as separate entities when finding affecting nodes. Thus, the product $A_\xi \otimes T$ yields P and this matrix is sufficient.

Now, consider the orbit obtained for $\mathbf{x}(0) = (4, 2, 1)^\top$. To find the second arrival time at node 1 on the 0^{th} epoch, we consider $\mathcal{N}_1 = \{1, 2, 3\}$, arrange the quantities $x_1(0) \otimes P_{11}$, $x_2(0) \otimes P_{12}$ and $x_3(0) \otimes P_{13}$ in order, then choose the second smallest. Notice that the right-hand side in (4.26) is equal to $x_i(k+1)$. Thus, we are in fact looking for the $(k+1)^{\text{th}}$ update time of node i (node 1 here). Notice also that we have stated here a heuristic method equivalent to the conjunctive and disjunctive operations

$$x_1(k+1) = (x_1(k)\tau_{11} \oplus x_2(k)\tau_{12}) \ominus (x_2(k)\tau_{12} \oplus x_3(k)\tau_{13}) \ominus (x_3(k)\tau_{13} \oplus x_1(k)\tau_{11})$$

and

$$x_1(k+1) = (x_1(k)\tau_{11} \ominus x_2(k)\tau_{12}) \oplus (x_2(k)\tau_{12} \ominus x_3(k)\tau_{13}) \oplus (x_3(k)\tau_{13} \ominus x_1(k)\tau_{11})$$

which were both used to do the algebraic work for this model in the previous section. However, we find that it helps to use a more systematic approach here to enable the identification of affecting nodes. Thus, for node 1 on the initial cycle, the quantities that we arrange in order are $x_1(0) \otimes 9$, $x_2(0) \otimes 9$ and $x_3(0) \otimes 2$, which gives 13, 11 and 3 respectively. The first two of these times are 3 and 11, which implies that the corresponding nodes, 3 and 2, are the affecting nodes of node 1 on the initial cycle. By repeating this process for the other nodes, we obtain a set of affecting nodes for each node on cycle 0 and, moreover, the reduced graph of affecting nodes on cycle 0.

Given the network size N, neighbourhood size n and maxmin-m system, denote by $\mathcal{G}^{(r)}(k)$ the reduced graph of the maxmin-m system on cycle k. We have obtained $\mathcal{G}^{(r)}(0)$ and we may obtain $\mathcal{G}^{(r)}(k)$ for $k > 0$ in a similar way. We may, thus, obtain a sequence of reduced graphs.

Definition 4.2.2. Let $\mathcal{M}$ be a maxmin-m system on N nodes having initial condition $\mathbf{x}(0)$ which yields the reduced graph $\mathcal{G}^{(r)}(0)$. The *orbit of* $\mathcal{G}^{(r)}(0)$ is the sequence $\mathcal{G}^{(r)}(0), \mathcal{G}^{(r)}(1), \mathcal{G}^{(r)}(2), \ldots$ of reduced graphs.

Moreover, we define a set of reduced graphs in which there are repeated graphs as follows.

Definition 4.2.3. Let $\mathcal{M}$ be a maxmin-m system on N nodes having initial condition

$\mathbf{x}(0)$. For some $k \geq 0$, consider the following set of reduced graphs of $\mathcal{M}$.

$$\mathcal{G}^{(r)}(k), \mathcal{G}^{(r)}(k+1), \mathcal{G}^{(r)}(k+2), \ldots$$

This set is called a *periodic orbit of reduced graphs* if there exists $g \in \mathbb{N}$ such that $\mathcal{G}^{(r)}(k+g) = \mathcal{G}^{(r)}(k)$. The *period* of the orbit is g.

Once we know a periodic orbit of reduced graphs, we may denote it as $\mathcal{O} = \{\mathcal{G}_1^{(r)}, \mathcal{G}_2^{(r)}, \ldots, \mathcal{G}_g^{(r)}\}$. This then implies that the period satisfies $g = |\mathcal{O}|$.

Initially, the graphs were referred to as max-plus graphs. This is because on epoch k, the maxmin-m system may be replaced by a max-plus system.

Definition 4.2.4. Let $\mathcal{M}$ be a maxmin-m system on N nodes. Let $P(\mathcal{G}^{(r)}(k)) \in \mathbb{R}_{\max}^{N\times N}$ denote the weighted adjacency matrix of $\mathcal{G}^{(r)}(k)$ in max-plus algebra. We call $P(\mathcal{G}^{(r)}(k))$ the *reduced max-plus matrix* of the maxmin-m system on cycle k.

If the reduced graph is understood, then denote the reduced max-plus matrix as $P^{(r)}(k)$. We may also refer to $P^{(r)}(k)$ simply as a reduced matrix. Moreover, if a sequence of reduced graphs is periodic, then the corresponding sequence of reduced max-plus matrices is also periodic with the same period g. We refer to such a sequence as the *periodic orbit of reduced matrices.* Due to their bijection, we may often denote both the periodic orbit of reduced graphs and the periodic orbit of reduced matrices by the same letter $\mathcal{O}$.

Then, on epoch k, the maxmin-m system may be written as the max-plus system

$$\mathbf{x}(k+1) = P(\mathcal{G}^{(r)}(k)) \otimes \mathbf{x}(k).$$

The implication is that $P(\mathcal{G}^{(r)}(k))$ is expected to be one of the $({}^{n}C_m)^N$ unique max-projections of the maxmin-m system. By relating this to (4.25), we conjecture that $P(\mathcal{G}^{(r)}(k))$ is the matrix P_3 in the example of this section.

State transition diagram

A sequence of reduced max-plus matrices implies a sequence of reduced max-plus systems, referred to simply as the *reduced max-plus system.* In the example maxmin-2 system, the initial condition $\mathbf{x}(0) = (4, 2, 1)^\top$ yields the following set of reduced max-plus matrices.

$$P^{(r)}(0) = \begin{pmatrix} \varepsilon & 9 & 2 \\ 4 & \varepsilon & 1 \\ 5 & 5 & 8 \end{pmatrix}, P^{(r)}(1) = \begin{pmatrix} \varepsilon & 9 & 2 \\ 4 & \varepsilon & 1 \\ 5 & 5 & \varepsilon \end{pmatrix}, P^{(r)}(2) = \begin{pmatrix} \varepsilon & 9 & 2 \\ 4 & \varepsilon & 1 \\ 5 & 5 & \varepsilon \end{pmatrix}, \cdots$$

It is seen that, for $k \geq 1$, the reduced max-plus matrix is equal to $P^{(r)}(1)$. Thus, we obtain a periodic orbit of reduced graphs. This is depicted in Figure 4.4 as part of a sequence of reduced graphs (arcweights omitted), where an arrow points from $\mathcal{G}^{(r)}(k)$ to $\mathcal{G}^{(r)}(k+1)$ for all k. We refer to such a diagram as a *state transition diagram* (STD) of reduced graphs. Since the periodic orbit takes period one, there is an arrow that points from $\mathcal{G}^{(r)}(1)$ to itself. Notice that the period in this STD does not coincide with the period of the regime, which was found earlier to be 2.

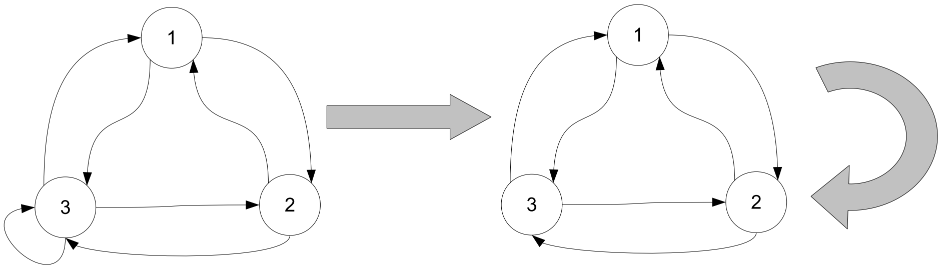


Figure 4.4: State transition diagram of reduced graphs for the maxmin-2 system (4.22) taking $\mathbf{x}(0) = (4, 2, 1)^\top$. The period of the periodic orbit is one.

Now consider another initial condition. We take $\mathbf{x}(0) = (8, 2, 7)^\top$ to obtain a sequence of update times of the maxmin-2 system satisfying $\mathbf{x}(k+2) = 13 \otimes \mathbf{x}(k)$ for $k \geq 2$, i.e. $K = 2$, $\rho = 2$ and cycletime $\chi = 6.5$. The sequence of reduced max-plus matrices of this system is

$$P^{(r)}(0) = \begin{pmatrix} \varepsilon & 9 & 2 \\ \varepsilon & 9 & 1 \\ 5 & 5 & \varepsilon \end{pmatrix}, \quad P^{(r)}(1) = \begin{pmatrix} 9 & \varepsilon & 2 \\ 4 & \varepsilon & 1 \\ 5 & 5 & \varepsilon \end{pmatrix}, \quad P^{(r)}(2) = \begin{pmatrix} \varepsilon & 9 & 2 \\ 4 & 9 & 1 \\ \varepsilon & 5 & 8 \end{pmatrix},$$
$$P^{(r)}(3) = \begin{pmatrix} 9 & 9 & 2 \\ 4 & \varepsilon & 1 \\ 5 & 5 & \varepsilon \end{pmatrix}, \quad P^{(r)}(4) = \begin{pmatrix} \varepsilon & 9 & 2 \\ 4 & 9 & 1 \\ \varepsilon & 5 & 8 \end{pmatrix}, \quad \cdots$$

and we find that $P^{(r)}(k+2) = P^{(r)}(k)$ for $k \geq 2$. Thus, we obtain a periodic orbit of reduced graphs of period 2 for this initial condition. This periodic orbit is shown as part of the STD of reduced graphs in Figure 4.5. In summary, we have seen that the reduced max-plus system of the maxmin-m system is dependent on the initial condition $\mathbf{x}(0)$.

Cyclicity of the maxmin-m system

Consider a maxmin-m system $\mathcal{M}$, where, given the initial condition $\mathbf{x}(0)$, $\mathbf{x}(k) = \mathcal{M}^k(\mathbf{x}(0))$. Let $\mathcal{O}$ denote the periodic orbit that $\mathcal{M}$ reduces to in the limit of large k. Then $\mathcal{O} = \{\mathcal{G}_1^{(r)}, \mathcal{G}_2^{(r)}, \dots, \mathcal{G}_g^{(r)}\}$. In this section, we calculate the period $\rho = \rho(\mathbf{x}(0))$ of $\mathcal{M}$.

If $g = |\mathcal{O}| = 1$, then $\mathcal{O} = \{\mathcal{G}^{(r)}\}$, and we can calculate the cyclicity of $P(\mathcal{G}^{(r)})$, which consequently equals the period ρ. (Note that the cyclicity of $P(\mathcal{G}^{(r)})$ is not the cyclicity of $\mathcal{M}$ since cyclicity must be fixed as the upper bound of the period ρ, which depends on $\mathbf{x}(0)$ and is therefore not fixed). When $g = |\mathcal{O}| > 1$, then ρ is not as straightforward to calculate.

For $g = |\mathcal{O}| > 1$, consider the reduced max-plus matrices of $\mathcal{O}$, i.e. $P_1^{(r)}, \dots, P_g^{(r)}$, where $P_j^{(r)} = P_j(\mathcal{G}_j^{(r)})$ for $\mathcal{G}_j^{(r)} \in \mathcal{O}$. Without loss of generality, let the reduced max-plus system on epoch k of the periodic regime take timing dependency matrix $P_k^{(r)}$. Thus, in the periodic regime, $\mathbf{x}(k+1) = P_k^{(r)} \otimes \mathbf{x}(k)$. Given a state $\mathbf{x}(1)$ in the regime, we apply this equation recursively to yield

$$\mathbf{x}(g+1) = R^{(r)} \otimes \mathbf{x}(1)$$

where $R^{(r)} = P_g^{(r)} \otimes P_{g-1}^{(r)} \otimes \cdots \otimes P_1^{(r)}$. Thus, we can calculate update times $\mathbf{x}(k)$ on every g contours in the periodic regime using the application of one max-plus matrix $R^{(r)}$.

The above discussion leads us to combine intuition with computational evidence to conjecture the following.

Conjecture 4.2.1. Consider a maxmin-m system $\mathcal{M}$, where $\mathbf{x}(k) = \mathcal{M}^k(\mathbf{x}(0))$. Given this initial condition $\mathbf{x}(0)$, let $\mathcal{O} = \{\mathcal{G}_1^{(r)}, \dots, \mathcal{G}_g^{(r)}\}$ denote the periodic orbit

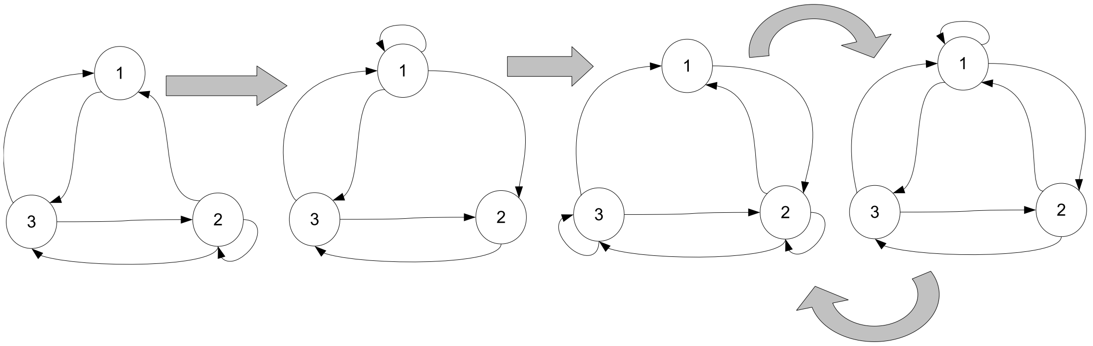


Figure 4.5: STD of reduced graphs for the maxmin-2 system (4.22) taking $\mathbf{x}(0) = (8, 2, 7)^\top$.

of reduced graphs of $\mathcal{M}$ in the limit of large k where $g = |\mathcal{O}| > 1$. Let $R^{(r)} = P_g^{(r)} \otimes \cdots \otimes P_1^{(r)}$, where $P_i^{(r)} = P(\mathcal{G}_i^{(r)})$ for $i = 1, \ldots, g$. Then $R^{(r)}$ will have cyclicity $\sigma(R^{(r)}) = 1$.

Therefore, since $\sigma(R^{(r)}) = 1$, we have $\rho(\mathcal{M}) = g$.

4.3 Simultaneity in the maxmin-m system

By comparing the outcome of the examples in Section 4.2.2 to that of Section 4.2.1, we can notice some contrasting behaviour, especially when carefully inspecting the reduced graphs of period 2 above.

Section 4.2.1 converged towards equality in the duality theorem, and this yielded a reduced max-plus system as a by-product. The reduced max-plus matrix was P_3, where P_3 was equal to one of the 3^N max-projections of the maxmin-2 system. Thus, having obtained equality in the DT for this maxmin-2 system, the suggested periodic orbit of reduced max-plus matrices takes period 1.

For our general maxmin-m model, a max-representation of the evolution equation is (4.13). Our modelling approach was to fix each row in each max-projection to contain m non-zero elements. It can be seen that P_3 satisfies this as it contains exactly 2 non-zero elements in each row. Observe that we do, indeed, obtain this reduced matrix for the initial condition $\mathbf{x}(0) = (4, 2, 1)^\top$ in this section. However, the initial condition $\mathbf{x}(0) = (8, 2, 7)^\top$ yields a reduced max-plus system of period $\rho = 2$. Furthermore, both reduced matrices contain rows with greater than 2 non-zero elements. In other words, both matrices do not lie in the set of all max-projections for this system.

In this thesis, we are heading towards a demonstration of the effect of the maxmin-m system on cellular automata. The reader may well assume that such a CA model is one in which the CA state of node i is a function of the first m input states that arrive at i. However, our approach above has just demonstrated that this is not quite the case, with the model yielding reduced max-plus graphs having more than m inputs to a node. This is in fact a consequence of Definition 4.2.1, which specifies the affecting nodes, and which leads to a crucial corollary that arises from the possibility of other inputs arriving at the same time as the mth input, i.e. the occurrence of simultaneity. Therefore, the implication is that the CA state of node i may be a function of more than m input states.

In the next subsection, we scrutinise this simultaneity and compare it with other approaches that may resolve simultaneously arriving inputs at a node.

4.3.1 Resolving simultaneity

A motivation for researchers of asynchrony is that synchrony can never truly be achieved in the physical world. Therefore, their asynchrony has often been modelled by the scheme whereby only one cell updates on each time step. We have seen in Chapter 1 that this scheme has been referred to by such names as ARBNs [21] and uniform-choice step-driven methods [36, 24]. In real time, the interval between time steps may be minuscule, but an interval is acknowledged nevertheless. For example, Huberman and Glance comment that "a computer simulation ... entails choosing an interval of time small enough so that at each step at most one individual entity is chosen at random to interact with its neighbors" [23]. Therefore, authors commonly select the node to be updated at random.

In our model, we have seen that simultaneity is possible. Simultaneity may be regarded as the synchrony that has been rejected by the authors above. Therefore, to counter this, we could select at random a number $m_\star$ ($< m$) of inputs from all inputs that arrive simultaneously with the mth input such that there are exactly m nodes providing m inputs. This also ensures that the maxmin-m model yields reduced max-plus graphs having exactly m inputs to a node. However, an aim for us is to suppress the usage of such randomness in our model since that involves uncertainty (some cells may never be 'chosen' as the $m_\star^{\text{th}}$ input) and also requires an element of interference (i.e. a random number generator or algorithm would have to be employed on each occurrence of simultaneity). Ideally, we want complete determinism. Indeed, one of the uniform-choice step-driven methods selects nodes to be updated according to a predefined fixed sequence, which implies less interference and also ensures that each node is updated exactly once every N time steps in a network of N nodes. The drawback when we try to apply this technique to simultaneity is that the nodes in question may not even form part of the simultaneous inputs (their input might have arrived before or after the mth input) on some epochs whereas they may be simultaneous with the mth input on others. In this section therefore, we present our method that contrasts with the aforementioned methods. It arises directly from Definition 4.2.1 and ensures that every simultaneous input is retained.

Recall the vector $\mathcal{N}_{x_i} \in \mathbb{R}^{n\times 1}$ from Note 4.1.1 that denotes the ordered vector of arrival times from the neighbourhood of node i. Consider the same example as in the note with $\mathcal{N}_{x_i} = (a, a, b, c, c, c, d, d, e, \ldots) \in \mathbb{R}^{n\times 1}$.

In our approach, the mth input time is taken to be exactly the mth element of $\mathcal{N}_{x_i}$. This is also exactly what Equation (4.12) would yield. Here, the 4th input time is c. By applying Definition 4.2.1, we can deduce that, in a maxmin-4 system, the affecting nodes are those nodes whose input times correspond to the set $\{a, a, b, c, c, c\}$. Here, there are six affecting nodes.

This means that, although the maxmin-m timing model is updated upon arrival of the mth input, the number of affecting nodes may be greater than m. Table 4.1 lists all the possible 5-nbhd inputs and the corresponding affecting nodes that would be obtained using this approach. Notice that the only occasion where all maxmin-m systems yield a different set of affecting nodes is when all input times in $\mathcal{N}_{x_i}$ are distinct, i.e. when $\mathcal{N}_{x_i} = (a, b, c, d, e)$. This reflects the lack of simultaneity and implies that the in-degree of nodes in the reduced graphs of corresponding systems would be m, which is an ideal case, since all max-projections in the maxmin-m system also contain m non-zero entries in each row. For all values of m in the table, only half of the vectors $\mathcal{N}_{x_i}$ yields $|\mathcal{A}_i| = m$, (for example, when $m = 3$, there are eight occurrences of "1:3"). This gives a considerable fraction of behaviour that pertains to reduced graphs having neighbourhood size greater than m.

$\mathcal{N}_{x_i}$	maxmin-1	maxmin-2	maxmin-3	maxmin-4	maxmin-5
(a, a, a, a, a)	all	all	all	all	all
(a, a, a, a, b)	1:4	1:4	1:4	1:4	all
(a, a, a, b, b)	1:3	1:3	1:3	all	all
(a, a, a, b, c)	1:3	1:3	1:3	1:4	all
(a, a, b, b, b)	1:2	1:2	all	all	all
(a, a, b, b, c)	1:2	1:2	1:4	1:4	all
(a, a, b, c, c)	1:2	1:2	1:3	all	all
(a, a, b, c, d)	1:2	1:2	1:3	1:4	all
(a, b, b, b, b)	1	all	all	all	all
(a, b, b, b, c)	1	1:4	1:4	1:4	all
(a, b, b, c, c)	1	1:3	1:3	all	all
(a, b, b, c, d)	1	1:3	1:3	1:4	all
(a, b, c, c, c)	1	1:2	all	all	all
(a, b, c, c, d)	1	1:2	1:4	1:4	all
(a, b, c, d, d)	1	1:2	1:3	all	all
(a, b, c, d, e)	1	1:2	1:3	1:4	all

Table 4.1: The affecting node set $\mathcal{A}_i$ for all the possible 5-nbhd inputs arriving at node i. Without loss of generality, the jth element of $\mathcal{N}_{x_i}$ is referred to as node j. For example, "1:3" implies $\mathcal{A}_i = \{1, 2, 3\}$ and the input times corresponding to these nodes are read left to right from $\mathcal{N}_{x_i}$. "all" denotes nodes 1 to 5.

4.3.2 Avoiding simultaneity

It would be ideal to match the reduced graph obtained through our intuitive method with a max-projection that corresponds to a unique reduced max-plus system. Therefore, let us try and achieve this by trying to avoid simultaneity. In all the work thus far, the cycletime of any maxmin-m system has been realised by the average circuit weight of underlying graphs. The update times $\mathbf{x}(k)$ are thus a multiple of the arcweights of these graphs. Since simultaneity arises when elements $x_i(k)$ and $x_j(k)$ coincide for $i \neq j$, it seems logical then to try and fix the arcweights of the underlying network to values that do not share a common multiple. One way to do this is to let the arcweights be irrational where no arcweight is a rational multiple of another. Thus, consider the following example.

Example 4.3.1. Consider a maxmin-2 system $\mathcal{M}$ with timing dependency matrix

$$P = \begin{pmatrix} \sqrt{13} & \sqrt{5} & \sqrt{11} \\ \sqrt{3} & e^1 & \sqrt{7} \\ \pi & \sqrt{2} & \sqrt{17} \end{pmatrix}.$$

We computationally found the asymptotic behaviour that arises from 500 different initial conditions, where each $\mathbf{x}(0) \in \mathbb{R}^N$ is randomly chosen with uniform probability such that $0 < x_i(0) \leq 10$. As expected, we obtained the unique cycletime $\chi = \frac{\pi+\sqrt{11}}{2}$. All the initial conditions tested also yielded period $\rho = 2$; we could therefore infer that the cyclicity is $\sigma(\mathcal{M}) = 2$. After 30 iterations of each of the 500 runs, one of two reduced max-plus systems are obtained; these are the periodic orbits of reduced matrices $\mathcal{O}_1$ and $\mathcal{O}_2$, where

$$\mathcal{O}_1 = \left\{ \begin{pmatrix} \varepsilon & \sqrt{5} & \sqrt{11} \\ \sqrt{3} & e^1 & \varepsilon \\ \pi & \sqrt{2} & \varepsilon \end{pmatrix} \right\}$$

and

$$\mathcal{O}_2 = \left\{ \begin{pmatrix} \varepsilon & \sqrt{5} & \sqrt{11} \\ \sqrt{3} & e^1 & \varepsilon \\ \pi & \sqrt{2} & \varepsilon \end{pmatrix}, \begin{pmatrix} \sqrt{13} & \sqrt{5} & \sqrt{11} \\ \sqrt{3} & e^1 & \varepsilon \\ \pi & \sqrt{2} & \varepsilon \end{pmatrix} \right\}.$$

$\mathcal{O}_1$ was observed 416 times while $\mathcal{O}_2$ was found to be the orbit of reduced matrices 84 times.

Now consider a max-representation of this system with the following max-projections.

$$P_1 = \begin{pmatrix} \sqrt{13} & \sqrt{5} & \varepsilon \\ \varepsilon & e^1 & \sqrt{7} \\ \pi & \varepsilon & \sqrt{17} \end{pmatrix}, P_2 = \begin{pmatrix} \varepsilon & \sqrt{5} & \sqrt{11} \\ \sqrt{3} & \varepsilon & \sqrt{7} \\ \pi & \sqrt{2} & \varepsilon \end{pmatrix}, P_3 = \begin{pmatrix} \sqrt{13} & \varepsilon & \sqrt{11} \\ \sqrt{3} & e^1 & \varepsilon \\ \varepsilon & \sqrt{2} & \sqrt{17} \end{pmatrix}.$$

We obtain the cycletimes $\chi(P_1) = \sqrt{17}$, $\chi(P_2) = \frac{\pi+\sqrt{11}}{2}$ and $\chi(P_2) = \sqrt{17}$ so that

$\chi(P_1) \ominus \chi(P_2) \ominus \chi(P_3) = \frac{\pi+\sqrt{11}}{2}$. If this forms the right hand side of the duality theorem, then the exact cycletime has been realised by this max-representation. Therefore, according to the duality theorem, the maxmin-2 system may be represented by the max-plus system with underlying timing dependency matrix P_2. Note however, that our computational work did not yield a reduced matrix equal to P_2.

Let us choose different max-projections for use in the duality theorem, i.e. choose

$$P_1 = \begin{pmatrix} \varepsilon & \sqrt{5} & \sqrt{11} \\ \sqrt{3} & e^1 & \varepsilon \\ \pi & \sqrt{2} & \varepsilon \end{pmatrix}, P_2 = \begin{pmatrix} \sqrt{13} & \sqrt{5} & \varepsilon \\ \varepsilon & e^1 & \sqrt{7} \\ \pi & \varepsilon & \sqrt{17} \end{pmatrix}, P_3 = \begin{pmatrix} \sqrt{13} & \varepsilon & \sqrt{11} \\ \sqrt{3} & \varepsilon & \sqrt{7} \\ \varepsilon & \sqrt{2} & \sqrt{17} \end{pmatrix}.$$

In this case, we obtain $\chi(P_1) \ominus \chi(P_2) \ominus \chi(P_3) = \frac{\pi+\sqrt{11}}{2}$, i.e. the exact cycletime is realised again, but this time by P_1, which is the same matrix that formed the periodic orbit $\mathcal{O}_1$ above.

The above example shows a counterexample to the conjecture that the maxmin-m system may be asymptotically represented by a unique max-plus system. Moreover, while the reduced matrices that were found computationally seem to be the only such reduced matrices that may be obtained (for all $\mathbf{x}(0)$), the duality theorem suggests otherwise since it can yield the exact cycletime for a matrix that is not contained in either of $\mathcal{O}_1$ or $\mathcal{O}_2$.

The following example brings to light the problems for identifying asymptotic behaviour as caused by simultaneity, even in such a system that uses pairwise rationally independent arcweights as in Example 4.3.1.

Example 4.3.2. Consider the maxmin-2 system $\mathcal{M}$ of Example 4.3.1. In particular, consider the two periodic orbits of reduced matrices that arise from that example, i.e. $\mathcal{O}_1$ and $\mathcal{O}_2$. Taking initial condition $\mathbf{x}(0) = (2, 11, 8)^\top$, after 30 iterations, the system behaves with periodic orbit of reduced graphs $\mathcal{O}_2$. In fact, this also occurs after $k = 3$ iterations. However, note that the cycletime $\chi(\mathcal{M})$ and period ρ are $\frac{\pi+\sqrt{11}}{2}$ and 2 respectively after the transient time $K = 9$. In other words, although a periodic orbit of reduced graphs is observed when $k = 3$, the update times of the system are not periodic at that time. In fact, when $k = 9$, the system behaves as the orbit $\mathcal{O}_1$, which has period 1. Further, $\mathcal{O}_2$ is again the dominant periodic orbit when $k = 18$, thereafter the reduced system remains in this orbit until the observed $k = 30$ iterations. Let us detail this in Table 4.2, which shows the evolution of the periodic orbits of reduced systems as a function of k. This table may be thought of as a numerical outline of the state transition diagram of the system and it shows that the

k	1	2	3	$\cdots$	8	9	$\cdots$	17	18	$\cdots$	30
reduced graph	transient graph	transient graph	$\mathcal{O}_2$	$\cdots$	$\mathcal{O}_2$	$\mathcal{O}_1$	$\cdots$	$\mathcal{O}_1$	$\mathcal{O}_2$	$\cdots$	$\mathcal{O}_2$

Table 4.2: Evolution of periodic orbit of reduced systems of a maxmin-2 system as function of k. "transient graph" indicates a reduced graph that is not repeated.

periodic orbit of reduced systems is not necessarily an attractor. The reduced systems are clearly periodic, yet may enter (and return to) different periods for different values of k.

The above example shows the flawed usage of computing. The program that was used (MATLAB) approximates irrational numbers with rationals. This indicates that, after a large number of iterations k, simultaneity will be encountered since a rational number is, by its nature, a rational multiple of the integer k. We have shown previous to this section that such a situation leads to simultaneity, and this is indeed what happened in Example 4.3.2.

To counter this, the suggestion is to stop iterating shortly after $k = K$, the transient time and accept the graph $\mathcal{G}^{(r)}(k)$ as the reduced graph of the system for $k \to \infty$. For $k = K = 9$ in Example 4.3.2, the orbit of reduced graphs is periodic with period 1 (see also Table 4.2). Represent this reduced graph by $\mathcal{O}_1$, the same notation as its periodic orbit. The application of $\mathcal{O}_1$ for $k \geq 9$ does not, in fact, change the sequence of update times of the maxmin-2 system. When $k = 9$, we obtain $\mathbf{x}(9) = (29.1495, 27.5649, 28.9745)^\top$ (where the rational numbers have been replaced by their approximations) and further iterations of $\mathcal{M}$ yields the sequence

$$\mathbf{x}(10) = (32.291, 30.882, 32.291)^\top, ..., \mathbf{x}(30) = (96.873, 95.464, 96.873)^\top,$$
$$\mathbf{x}(31) = (100.19, 98.605, 100.015)^\top, \mathbf{x}(32) = (103.332, 101.922, 103.332)^\top, \ldots \quad (4.27)$$

We are particular interested in the update times after the epoch $k = 30$, since that is when the reduced graphs enter the orbit $\mathcal{O}_2$. It turns out that applying only $\mathcal{O}_1$ for $k \geq 30$ yields the same update times as those in (4.27). Therefore, for this example at least, the maxmin-2 system is then seen to be characterised by one max-plus matrix for $k \to \infty$, as we require.

Thus, we can conclude from Example 4.3.2 that the period-2 orbit that was observed in the previous example is in fact *not* representative of the real system, but a result of MATLAB approximating irrationals with rationals. To avoid this orbit, we simply stop a short while after $k = K$ iterations and should observe that the reduced graph is that in $\mathcal{O}_1$ for all 500 runs. In other words, the computer can yield a unique reduced max-plus system whilst the duality theorem may not.

In conclusion to this section, we have seen that equality in the duality theorem does not necessarily yield a reduced max-plus system with a unique max-plus matrix for the maxmin-m system for large k. However, we have shown that an algorithmic (or computational) method using Definition 4.2.1 can find the *unique* reduced graph, and this can in turn give the exact cycletime and period without having to employ the duality conjecture on an impractically large number M of trial systems. This is true at least on occasions when simultaneity is avoided, which can be done by choosing irrational processing and transmission times and, if working computationally, by choosing k to be slightly larger than K.

The issue of simultaneity in the maxmin-m model also causes the reduced graph of the periodic orbit to be dependent on the initial condition $\mathbf{x}(0)$. Moreover, we observe the following conjecture.

Conjecture 4.3.1. Consider a maxmin-m system $\mathcal{M}$. Given $\mathbf{x}(0)$, let the periodic orbit of reduced graphs obtained after k iterations of $\mathcal{M}$ contain $\mathcal{G}_1^{(r)}$ such that $P(\mathcal{G}_1^{(r)})$ has exactly m non-zero elements on each row. For some other initial condition $\mathbf{x}_\star(0)$, let the periodic orbit of reduced graphs after k iterations of $\mathcal{M}$ contain the graph $\mathcal{G}_2^{(r)}$ such that $\mathcal{G}_1^{(r)} \neq \mathcal{G}_2^{(r)}$. Then, $P(\mathcal{G}_2^{(r)})$ contains at least one row with greater than m non-zero elements.

Indeed, if $P(\mathcal{G}_2^{(r)})$ contains rows with greater than m non-zero elements, then $P(\mathcal{G}_2^{(r)})$ is not contained in the set $\mathcal{P}$ of all $({}^nC_m)^N$ distinct max-representations (since all those max-representations contain exactly m non-zero elements on each row); this points to a reduced graph with nodes having larger neighbourhood sizes than m.

4.4 Summary

In general our model is a maxmin-m system; the work in Section 4.1 also addresses the min-plus and max-plus systems since the min-plus system is just the maxmin-1 system, and the max-plus system is the maxmin-n system.

We represented the system in two forms. One is the general maxmin-m form and the other is the full form. The latter is guaranteed to give equality in the duality theorem, thereby yields the exact cycletime vector. However, this would require all $({}^nC_m)^N$ max-projections P_j, whereas the shorter max-representation requires ${}^nC_m$, i.e. only a selection of these max-projections. Computationally, the shorter form is more efficient, although taking only a selection of P_j could imply that the duality theorem does not yield equality and therefore only gives a bound on the cycletime vector. Nevertheless, we conjecture that there must be an optimal selection of P_j such that the shorter max-representation yields the exact cycletime vector. However, this is a difficult task since the number of such max-representations M can become impractically large. The question then becomes one of numerics: which of the two methods is numerically (or computationally) efficient? The first might be better even though it only yields upper and lower bounds because it uses only ${}^nC_m$ matrices; the second guarantees the output of the exact χ but is slower because it uses $({}^nC_m)^N$ matrices. The issue becomes that of a trade-off. An upper and lower bound for the cycletime vector may still be sufficient in terms of performance of a system of interacting processes since it allows some flexibility; it may also be impossible to configure a system to the exact cycletime vector in a real application.

We have shown that, in the long term, some maxmin-m systems are reduced to a max-plus system with underlying max-plus graphs. Such graphs identify the affecting nodes of the CA and, in the next chapter, we shall see that this aids the construction of a state transition graph of CA, enabling prediction of a CA model in maxmin-m time. However, in this chapter, we have also seen that simultaneity causes the reduced graphs to be contained in a periodic orbit of such graphs having period $g > 1$. In this case, the STG is not as straightforward to draw up.

Simultaneity avoidance should thus imply a unique reduced graph, consequently easing the construction of the STG. It has been suggested that taking a number of iterations close to the transient time K helps to overcome simultaneity, especially in cases where the processing and transmission times are approximated as rational by a computer.

Usually, as in most of this thesis, we do not allow processing times and transmission times to be irrational. In this case, simultaneity would be observed after a relatively small number of iterations by the computer, after which no further complications in terms of identifying an attractor should occur. For example, the periodic orbits of reduced graphs observed in the maxmin-2 system in Section 4.2.2 are attractors. Thus, if we are to accept simultaneity in such cases, then we can, in fact, highlight it as a feature that distinguishes the maxmin-m system from other asynchronous timing systems. Indeed, it also highlights the maxmin-m system as a special maxmin-plus system in which, contrary to established thoughts, the initial evolution equation and max-projections may not be indicative of the asymptotic behaviour of the system, especially when $m \neq n$, because the reduced graphs may take neighbourhood size greater than m. Meanwhile, the duality theorem also does not indicate affecting nodes in the long run.

Chapter 5

Cellular Automata in Maxmin-plus Time

In the previous chapter, we saw that the maxmin-m system often behaves as a max-plus system for large k. This property is useful as the max-plus system forms a tractable base for the simulation of cellular automata, as was seen in earlier chapters. As a consequence, we can use max-plus methods described in those chapters to analyse dynamics on a network due to the maxmin-m model.

This chapter is mainly comprised of computational work, whereby the aim is to simulate and analyse cellular automaton behaviour in maxmin-m time. We introduce the work with a small example in Section 5.1, which demonstrates typical behaviour of a CA model in maxmin-m update times for a range of m values. In Section 5.2, we present asymptotic update time results arising from the maxmin-m model on a regular n-nbhd network for $1 \leq m \leq n$. The results suggest a peak in cyclicity and transient time for $m \approx n/2$. Section 5.3 consequently runs a cellular automaton model with update times determined by this maxmin-m model. These results suggest a mapping from maxmin-m complexity to the resulting complexity of the CA. Section 5.4 is a bridge between the previous two sections, and attempts to mathematically explain the findings in relation to the work covered in previous chapters. Finally, Section 5.5 summarises the chapter and suggests possible advantages of the results obtained, in particular for the peak in complexity for $m \approx n/2$.

5.1 A small example

Cellular automata possess a graphic beauty when depicted in colour as a space-time plot. Thus, as a qualitative taster of what is to follow, we present an example of the

CA evolution in maxmin-m time on a relatively small timing dependency graph for all m ($1 \leq m \leq n$). We take $N = 8$, $n = 5$ and employ the following parameter conditions.

$$\begin{gathered} \xi = (5,2,4,3,5,4,3,1)^\top \text{ and } \tau_{ij} = 0 \text{ for all nodes } i \text{ and } j \\ \text{The initial conditions are } \mathbf{x}(0) = \mathbf{u} \text{ and } \quad s_4(0) = 1, s_i = 0 \text{ for } i \neq 4. \end{gathered} \tag{5.1}$$

After 20 iterations of each maxmin-m system, the contour plot and CA plots obtained are presented in Figure 5.1.

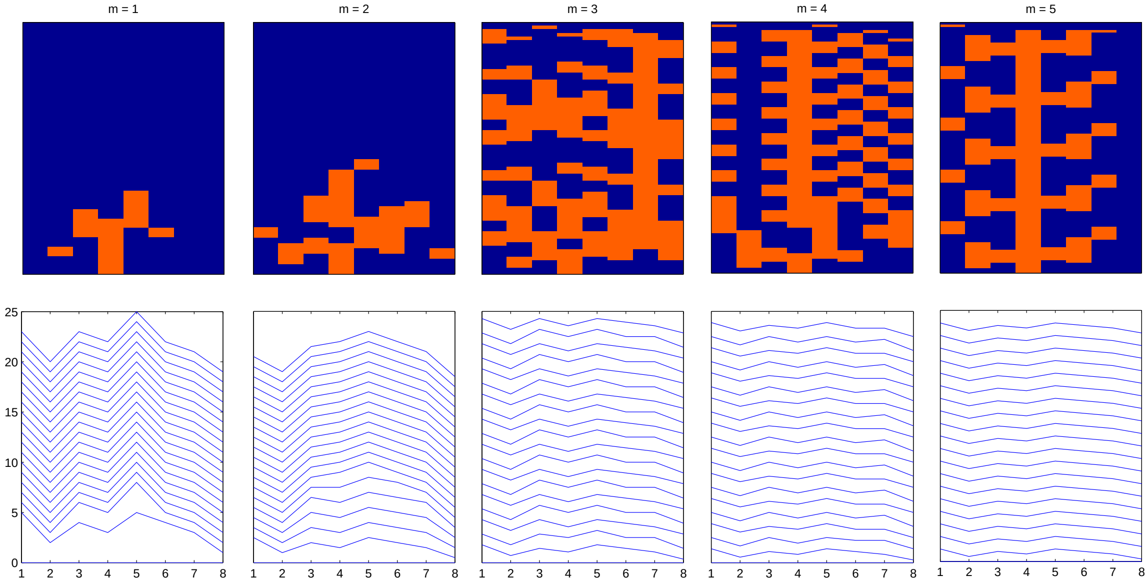


Figure 5.1: The spectrum of CA space-time patterns for a regular 5-nbhd network on 8 nodes and the parameters as in (5.1). The number of iterations taken is 20. For each m, the contour plots are below the corresponding CA patterns.

The figure shows a spectrum of CA patterns, which ranges from homogeneous to more complex. For $m = 1$ and 2, the pattern dies out, therefore is homogeneous, after large time. For $m = 4$ and 5, the patterns are seen to be periodic, where the periodic orbit of CA states can be identified fairly easily by eye. Although the case $m = 3$ also yields a periodic pattern, this periodic behaviour is not as easily identifiable as in the aforementioned patterns for the number of iterations taken. This leads to the thought that this maxmin-m system produces the most complex CA behaviour.

In fact, the quantitative classification of CA patterns supports this. Figure 5.2 shows the plots of the Shannon and word entropy values as functions of m. The plots take highest value for $m = (n + 1)/2$, where the homogeneous patterns (obtained for small m) are characterised by smaller entropy values than for the periodic patterns (observed for larger m).

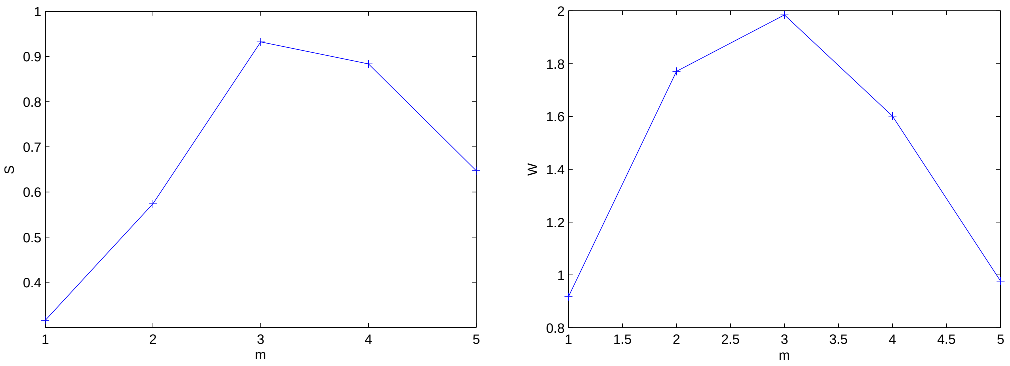


Figure 5.2: Entropy plots as functions of m for a cellular automaton in maxmin-m update time where the timing dependency graph is a regular 5-nbhd network on 8 nodes. The parameters are as in (5.1)

Further, Figure 5.3 plots the CA transient time and CA period as functions of m. In the same figure, the transient time K and period ρ for the update times are also plotted as functions of m; the cycletime χ is plotted next to these. From now on, since we will study various values of m, we also let $\mathcal{M}_m$ denote the maxmin-m system.

Definition 5.1.1. Consider a n-nbhd network. The *middle system* is denoted by $\mathcal{M}_{m_c}$ and is the maxmin-m system on this network when

$$m = m_c = \begin{cases} (n+1)/2 & \text{for } n \text{ odd} \\ n/2 & \text{for } n \text{ even} \end{cases}$$

In Figure 5.3, a large CA transient time and CA period is evident for the middle system. In fact, the middle system produces the largest K and ρ. This coincidence of such a peak for both the transient and cyclicity may be understood by looking at the systems as reduced max-plus systems. Recall the bound (3.7) for the transient time that depends on the choice of parameters used; for us, the parameters are ξ_i and τ_{ij}. In addition, the calculation of cyclicity of a max-plus system depends on the largest average circuit weight, which also depends on the parameters. Therefore, although the transient and cyclicity are independent of each other, for a fixed distribution of the parameters $\xi_\mathbf{i}$ and τ_{ij}, the behaviour of transient time and cyclicity of a maxmin-m system as functions of m may well be expected to coincide.

In the next sections, such a qualitative observation of complexity for $m \approx n/2$ is backed up by quantitative calculations that hint towards a link between complexity of the CA and the complexity of the update time mechanism, i.e. the maxmin-m model.

5.1.1 State Transition Graph of CA in maxmin-m time

In light of the theory that was mainly presented in the previous chapter, when drawing up the state transition graph of a maxmin-m system, we will see that we must think about simultaneity. Recall that simultaneity is when a multiple number of inputs arrive at a node at the same time.

Consider the conditions (a) simultaneity, and (b) no simultaneity. The simpler case to consider first is (b). In Chapter 4, we have conjectured that, where there is no simultaneity, we obtain one and only one reduced graph. Thus, for large enough k, the maxmin-m system $\mathcal{M}_m$ behaves as a max-plus system on a timing dependency graph $\mathcal{G}$ which takes neighbourhood size m. (Recall that such a graph is not necessarily strongly connected).

In this case, for large enough k, there is a bijective relation between synchronous CA on $\mathcal{G}$ and the maxmin-m CA. Consequently, the CA behaviour on contours is predictable and a state transition graph is deterministic and easily drawn.

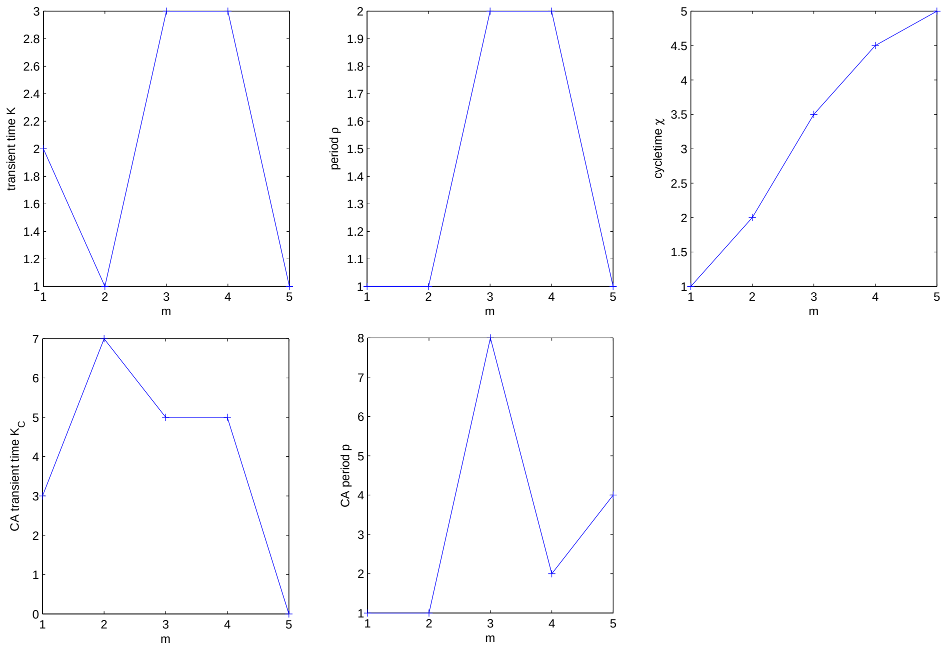


Figure 5.3: Transient and period plots as functions of m for cellular automata in maxmin-m update time where the timing dependency graph is a regular 5-nbhd network on 8 nodes and the parameters are as in (5.1). The cycletime is also plotted as a function of m.

Therefore, in an ideal scenario, simultaneity is avoided. However, arguments were presented in Chapter 4 for the difficulty in avoiding simultaneity. One such argument is the computational approximation of irrational numbers with rationals. Thus, consider case (a), which yields reduced graphs for large k where some nodes take in-degree greater than m, as was observed in Chapter 4. Moreover, these graphs may also be contained in a periodic orbit $\mathcal{O}$ of reduced graphs such that $|\mathcal{O}| \geq 2$.

Let us examine the impact of this on the subsequent STG of such a system. Consider the middle system in the small example of the previous section. Table 5.1 shows the CA states produced on each contour k, which correspondingly may be mapped to the CA output that was shown in Figure 5.1. Row k represents iteration k, and rows are counted from the bottom, i.e. counting from the bottom row, which shows the initial CA state $\mathbf{s}(0)$, the next row represents $\mathbf{s}(1)$, and each subsequent row similarly represents $\mathbf{s}(k)$, $k > 1$. We can imagine arcs being drawn from $\mathbf{s}(k)$ to $\mathbf{s}(k+1)$, as would occur in the full STG of the CA on this maxmin-3 system; the table may be thought of as a section of that STG.

For this initial CA state, the CA is periodic with period $p = 8$ after the transient time $K_C = 5$. Denote this periodic orbit of CA states by $\mathcal{S}$; it is emphasised in heavy print in Table 5.1. For this example, we observe that the periodic orbit of reduced graphs that may yield $\mathcal{S}$ is fixed at $\mathcal{O} = \{\mathcal{G}_1, \mathcal{G}_2\}$ for large k, where the adjacency matrices (in the usual graph theoretical sense) of $\mathcal{G}_1$ and $\mathcal{G}_2$ are the following matrices respectively.

$$P_{1'} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

k	$\mathbf{s}(k)$
21	1 0 0 0 1 1 1 1
20	0 0 1 0 0 0 1 0
19	1 1 1 1 1 0 1 0
18	1 1 1 1 1 1 1 1
17	0 1 1 1 0 1 1 1
16	1 0 0 0 1 1 1 1
15	0 0 0 0 0 0 1 0
14	0 1 0 1 0 0 1 0
13	1 0 0 0 1 1 1 1
12	**0 0 1 0 0 0 1 0**
11	**1 1 1 1 1 0 1 0**
10	**1 1 1 1 1 1 1 1**
9	**0 1 1 1 0 1 1 1**
8	**1 0 0 0 1 1 1 1**
7	**0 0 0 0 0 0 1 0**
6	**0 1 0 1 0 0 1 0**
5	**1 0 0 0 1 1 1 1**
4	1 1 1 1 1 0 1 0
3	0 1 0 1 0 1 1 1
2	1 0 1 0 1 1 1 1
1	0 1 1 1 1 1 0 0
0	0 0 0 1 0 0 0 0

Table 5.1: Section of the state transition graph of a maxmin-3 system on a regular 5-nbhd network on 8 nodes.

and

$$P_{2'} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}.$$

Thus, successive CA states in $\mathcal{S}$ may be obtained by applying the CA rule f on previous states such that the underlying graphs alternate between $\mathcal{G}_1$ and $\mathcal{G}_2$.

Remember that the CA rule f can conveniently be interpreted as the ordinary multiplication of the (non-weighted) adjacency matrix of the underlying network with the CA state $\mathbf{s}(k)$. Therefore, in this example, consider the state $\mathbf{s}(6) = 00100010$. To obtain $\mathbf{s}(7)$, we apply $P_{2'}$. It can be checked that

$$\begin{aligned} \mathbf{s}(7) &= P_{2'}\mathbf{s}(6)^\top \qquad \mod 2 \\ &= 10001111. \end{aligned}$$

The next state is obtained by applying $P_{1'}$ on $\mathbf{s}(7)$:

$$\begin{aligned} \mathbf{s}(8) &= P_{1'}\mathbf{s}(7)^\top \qquad \mod 2 \\ &= 01010010. \end{aligned}$$

Thus, CA states in $\mathcal{S}$ are deterministic. Given the states $\mathbf{s}(k), \mathbf{s}(k+1) \in \mathcal{S}$, all subsequent states may be obtained by the following.

$$\begin{aligned} \mathbf{s}(k+2j) &= (P_{1'}P_{2'})^j\mathbf{s}(k) \\ \mathbf{s}(k+2j+1) &= (P_{2'}P_{1'})^j\mathbf{s}(k+1) \end{aligned}$$

where $j = 0, 1, 2, \ldots$.

In summary, the STG of a maxmin-m system where $|\mathcal{O}| \geq 2$ will not be as easy to draw up as when $|\mathcal{O}| = 1$. This is because the application of different reduced graphs on a state $\mathbf{s}(k)$ can yield a different CA state. In the full STG, we can imagine there being a number $|\mathcal{O}|$ of arcs emanating from each CA state, each arc representing the application of one of the reduced graphs contained in $\mathcal{O}$.

Notwithstanding this, even in a system with $|\mathcal{O}| = 1$, we may encounter problems

k	$\mathbf{s}(k)$
10	**0 0 0 0 0 0 0 0**
9	**0 0 0 0 0 0 0 0**
8	**0 0 0 0 0 0 0 0**
7	**0 0 0 0 0 0 0 0**
6	0 0 0 1 1 0 0 0
5	0 0 0 1 0 0 0 0
4	0 0 1 1 0 0 1 0
3	0 0 1 1 0 1 1 0
2	1 1 0 0 1 1 0 1
1	0 1 1 1 1 1 0 0
0	0 0 0 1 0 0 0 0

Table 5.2: Section of the state transition graph of a maxmin-2 system on a regular 5-nbhd network on 8 nodes.

in detecting periodic behaviour of the CA. Consider the maxmin-2 system on the timing dependency graph of this example. Figure 5.1 shows that the resulting CA pattern is homogeneous for large k. Now consider the STG of this pattern in Table 5.2. The CA goes through some transient states before settling on the homogeneous pattern, characterised by the periodic orbit with period $p = 1$ where $\mathbf{s}(k) = 00000000$ for each contour k in this periodic orbit. However, notice that $\mathbf{s}(5) = \mathbf{s}(0)$, i.e. before the CA transient time $K = 7$, the initial state is repeated. The computer might well define this, along with the intermediate CA states, to be the periodic orbit of CA states here, even though these intermediate CA states are not repeated for $k > 5$. Therefore, our definition for periodic behaviour of CA states in Chapter 1 does not necessarily apply for a maxmin-m system where $m \neq n$, unless the maxmin-m system reduces to one reduced max-plus graph as $k \to \infty$.

It turns out that this maxmin-2 system does indeed asymptotically reduce to one max-plus graph. For $k = 0$, the reduced graph is $\mathcal{G}_1^{(r)} = \mathcal{G}(P)$, the original timing dependency graph, and the reduced graph is fixed at $\mathcal{G}_2^{(r)} \neq \mathcal{G}_1^{(r)}$ for $k \geq 1$. This implies that, to convert state $\mathbf{s}(0)$ to $\mathbf{s}(1)$, the CA is applied on $\mathcal{G}(P)$, whereas the underlying graph is different when the state $\mathbf{s}(5)$ $(= \mathbf{s}(0))$ is evolved into $\mathbf{s}(6)$ $(\neq \mathbf{s}(1))$.

Thus, the contour k on which the CA rule is applied is important in a maxmin-m system since different contours may imply different reduced graphs. It means that the identification of periodic behaviour in such a system requires a preliminary step of identifying periodic behaviour in the reduced graphs first.

Recall from Section 4.3 that the effect of simultaneity is mitigated by accepting that the reduced graphs are periodic with the same transient time K as the maxmin-m system. Thus, Definition 1.1.2 should be refined for a maxmin-m system for $m < n$: the definition is only applicable after the periodic orbit of reduced graphs is obtained.

In addition, let $K_C(\mathbf{s}(0))$ denote the smallest k for which a periodic CA orbit is obtained such that $k \geq K$. Then K_C is defined as the *CA transient time*. Note that the initial CA state is assumed to be understood here (and may not be contained in the CA orbit).

In the maxmin-3 example above, the period of this orbit coincides with the period of the update times $\rho = 2$. A larger period of reduced graphs implies a larger ρ, or at least $\rho \geq |\mathcal{O}|$. This subsequently implies a larger CA period (judging from the comparative plots in Figure 5.3). This result will be supported further in the next sections, where asymptotic update times and CA results are obtained for the maxmin-m model on a large timing dependency graph.

5.2 Cyclicity bell (and related results)

Consider the timing dependency graph of Section 4.2.1 in the previous chapter. The example looked at a maxmin-2 system on this graph. Here, let us now take $m = 1$ and $m = n$, thereby looking at the asymptotic behaviour of the min-plus and max-plus system on the same graph.

In Section 4.2.1, we took the two initial conditions $\mathbf{x}(0) = (4, 2, 1)^\top$ and $\mathbf{x}(0) = (8, 2, 7)^\top$, and obtained different sets of reduced periodic orbits, but for which the update times of the maxmin-2 system were periodic with period $\rho = 2$ and cycletime $\chi = 6.5$ for both initial conditions. For this chapter, having subsequently carried out a large number of numerical simulations with different initial conditions for each, the largest period obtained was 2. Thus, we may apply Corollary 2.1.2 to say that the cyclicity of the maxmin-2 system is likely to be 2. Moreover, we may also apply the theory of Chapter 2 to find the cycletime and cyclicity of $\mathcal{M}_1$, the min-plus system, on this network. The network is strongly connected, therefore the min-plus cycletime and cyclicity are $\chi = 3$ and $\sigma = 2$, as characterised by the 2-circuit between nodes 2 and 3. Similarly, the corresponding max-plus quantities are $\chi = 9$ and $\sigma = 1$, which is realised by the self-loops at nodes 1 and 2. If we collate these results together, we obtain Table 5.3. The feature of these results that we wish to focus on is that the maxmin-2 system yields the largest cyclicity. The significance of this is not immediately apparent since the min-plus system also produces the same cyclicity, but we claim that such a feature is not a special case restricted to this small example only.

	maxmin-1	maxmin-2	maxmin-3
cycletime	3	6.5	9
cyclicity	2	2	1

Table 5.3: Asymptotic results for all maxmin-m systems on a network of size $N = 3$ and neighbourhood size $n = 3$

Indeed, the example in the previous section showed largest period being attained by the middle system, from which we may infer that the cyclicity is also highest when $\mathcal{M}_m$ is the middle system. In this section, we present such results for a larger network size and therefore larger neighbourhood sizes.

5.2.1 Asymptotic maxmin-m results for a large network

We investigate the effect of a maxmin-m system $\mathcal{M}_m$ on a regular n-nbhd network on 20 nodes. The method used is an extension to Algorithm 3.2.1, which collated the max-plus results on the same network. Thus, we let the zero transmission condition hold and fix n before implementing the same steps as Algorithm 3.2.1, the difference in this chapter being that the results are obtained for all m ($m = 1, 2, \ldots, n$). To be precise, we now implement the following algorithm.

Fix n.

Algorithm 5.2.1. For $m = 1$ to n

1. Choose ξ_i from the uniform distribution (with equal probability) as integers between 1 and $r \in \mathbb{N}$.

2. Taking $\mathbf{x}(0) = \mathbf{u} = (0, 0, \ldots, 0)^\top$, run the maxmin-$m$ system to obtain the transient time K, period ρ and cycletime vector χ. If χ is not uniform, then take the mean, i.e. $\chi = \sum_{i=1}^{N} \frac{\chi_i}{N}$.

3. Carry out above two steps to obtain 500 such results for this value of r.

4. Record the mean of the 500 transient times, periods and cycletimes obtained.

This algorithm was repeated for each integer value of the ξ radius r from 1 to 30. (Thus, for one value of m, we obtained a total of $500 \times 30 = 15000$ results). The results are thus, comparative to the max-plus results obtained with Algorithm 3.2.1. In fact, the case $m = n$ here corresponds to that algorithm.

In this chapter, we shall focus on the results for the largest r value that was taken, i.e. $r = 30$. For a more in-depth analysis of the effect of r, we refer the reader to Chapter 3 and claim that the arguments used there may also be applied for all maxmin-m systems ($m = 1, \ldots, n$). Presently, we observe that the largest ξ radius also addresses and minimises the problems posed by simultaneity, which is expanded upon next.

In each of the simulations, we ideally would have liked to have used irrational processing times since that would have ensured no simultaneity. However, as seen in the previous chapter, irrational numbers may cause asymptotic problems. Moreover, the processing times for our simulations were chosen to be integers. Subsequently, the next best way to minimise simultaneity would be to try and ensure that processing times are distinct. See also Section 3.2.3 which showed that a large ξ radius implies a lower probability of generating synchronous behaviour; the same argument may be applied here to say that a large ξ radius also implies a smaller chance of simultaneity.

Notice the algorithm indicates the possibility to obtain a non-uniform cycletime vector in a maxmin-m system. This is particularly the case when we consider the reduced max-plus graph, which disposes the 'non-affecting' arcs of the original network and contains only the affecting node sets of each node i, along with the arcs pointing from a node in $\mathcal{A}_i$ to i. This creates a reduced max-plus graph which is a subgraph of the original timing dependency graph. The reduced graph may not be strongly connected, unlike the original regular n-nbhd network.

We therefore present the following theorem, which is an extension to the methods that we used in previous chapters to calculate the cycletime vector and cyclicity. The proof, given in [22], is substantial, in large part because it makes use of another theorem. Therefore we omit the proof here. We shall, however, need to make use of $\pi^*(i)$, which denotes the set $\{i, j | i \text{ is reachable from } j\}$. Note that node i is also contained in this set.

Theorem 5.2.1 ([22], Theorem 3.17)**.** Consider the recurrence relation $\mathbf{x}(k+1) = P \otimes \mathbf{x}(k)$ for $k \geq 0$, where $\mathbf{x}(0) = \mathbf{x}_\star$ is the initial condition and $P \in \mathbb{R}_{\max}^{N \times N}$ is a square regular matrix. Let $\chi = \lim_{k\to\infty} \frac{\mathbf{x}(k)}{k}$ be the cycletime vector of P. Then

1. For all nodes j,

$$\chi_{[j]} = \bigoplus_{i \in \pi^*(j)} \lambda_{[i]}$$

where the subscript $[j]$ denotes the MSCS containing node j and $\lambda_{[i]}$ is the eigenvalue of the weighted adjacency matrix of the MSCS containing i.

2. For all nodes j and any initial condition $\mathbf{x}(0) \in \mathbb{R}^N$,

$$\lim_{k\to\infty} \frac{x_j(k)}{k} = \bigoplus_{i\in\pi^*(j)} \lambda_{[i]}.$$

The theorem is best illustrated with an example, as shown.

Example 5.2.1. Consider the reduced graph of the maxmin-3 system on 10 nodes as shown in Figure 5.4. The network contains the three maximal strongly connected subgraphs [1], [5] and [8]. Since the associated weighted adjacency matrix is irreducible, the eigenvalue of each MSCS is the largest average circuit weight in the MSCS. Thus, $\lambda_{[1]} = 6$, $\lambda_{[5]} = 5$ and $\lambda_{[8]} = 8/3$. Since [5] is upstream of [1], the cycletime of [1] is $\chi_{[1]} = \max\{\lambda_{[1]}, \lambda_{[5]}\}$, which is 6. Similarly, $\chi_{[8]} = \max\{\lambda_{[8]}, \lambda_{[5]}\} = \lambda_{[5]} = 5$ and [5] has no predecessors, therefore $\chi_{[5]} = \lambda_{[5]} = 5$. Therefore, the cycletime vector for this max-plus system is $\chi = (6, 6, 6, 6, 5, 5, 5, 5, 5, 5)^\top$.

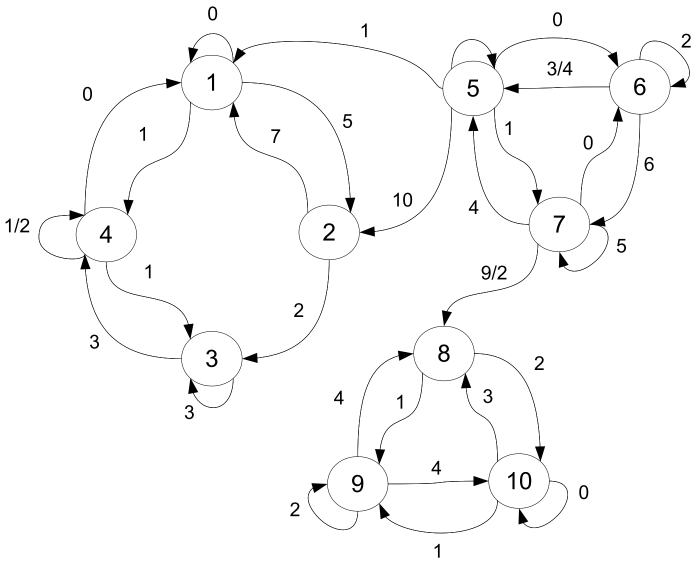


Figure 5.4: Reduced graph of a maxmin-3 system on 10 nodes. The network is not strongly connected.

Thus, the reduced max-plus system may asymptotically settle into a periodic regime whereby some nodes update their state faster than other nodes due to the non-uniform cycletime vector. Therefore, for each m, we take the mean of all elements χ_i in Algorithm 5.2.1 to form some approximation to a uniform cycletime vector, and to enable comparison with maxmin-m systems for other values of m.

For each n, the quantities mean K, mean ρ and mean χ arising from Algorithm 5.2.1 may be plotted as graphs with m on the horizontal axis. However, for each n, the range of values of m is different. Therefore, we replace m with $m/(n+1)$, which ranges from 0 to 1 for all neighbourhood sizes. This now enables comparison between different neighbourhoods because the graph associated to each n-nbhd can be plotted on the same axes, as is done in Figure 5.5. Note that we do not take m/n for reasons which will be made clear shortly.

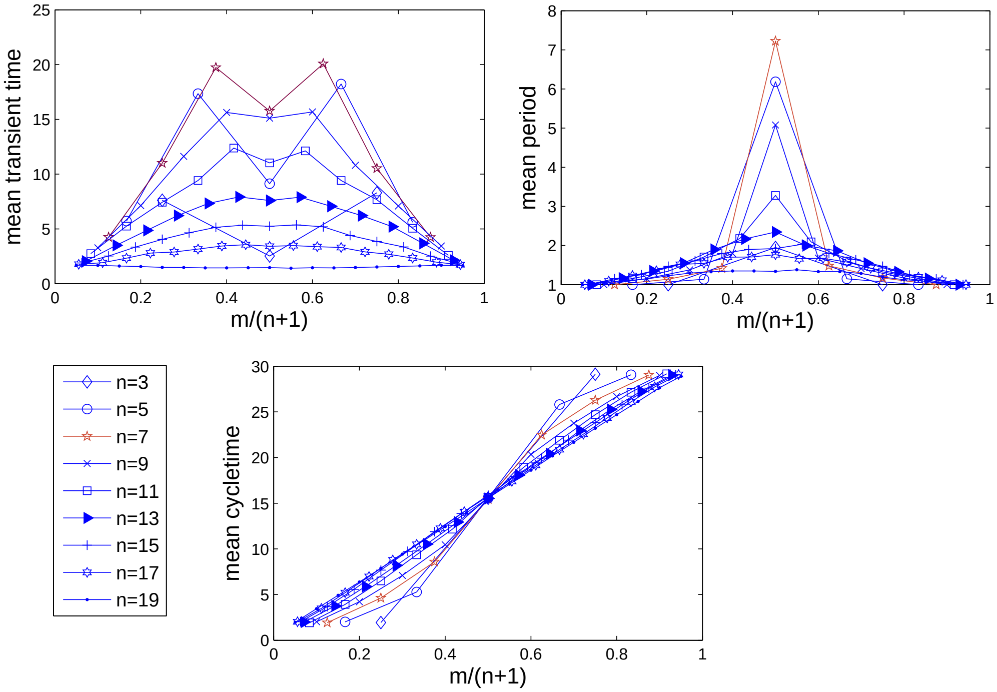


Figure 5.5: Mean transient time, mean period and mean cycletime of the maxmin-m system on the regular n-nbhd network of 20 nodes. The mean values are taken from 500 runs, each with $\mathbf{x}(0) = \mathbf{u}$ and a randomised processing time distribution with ξ radius 30. The zero transmission condition also holds.

Cycletime of the maxmin-m system

Consider a max-plus system $\mathcal{M}_n$ on this regular network under the zero transmission condition. Under these conditions, we can recall from Chapter 3 that, given a fixed ξ radius r, the most likely cycletime of $\mathcal{M}_n$ is $\chi(\mathcal{M}_n) = r$. Thus, in the work above, we expect to see a mean cycletime approximately equal to $r = 30$ when $m = n$. Indeed, we notice in Figure 5.5 that $\chi(\mathcal{M}_n) \approx 29$ for all neighbourhood sizes. Correspondingly, the same argument can be used to deduce that the most likely cycletime of the min-plus system is $\chi(\mathcal{M}_1) = 1$ since 1 is the lower bound of all processing times. The figure supports this by yielding $\chi(\mathcal{M}_1) \approx 2$ for all neighbourhood sizes.

Moreover, for all n, the cycletime graphs are approximately linear with $m/(n+1)$. For the purposes of the proceeding argument, we assume that the graphs are linear with m/n also. Therefore, given a neighbourhood size n, consider the following general form for the linear equation that approximates the mean cycletime graph of a maxmin-m system.

$$\chi(m) = \frac{\alpha}{n}m + \beta \tag{5.2}$$

where $\alpha, \beta \in \mathbb{R}$ to be found. From the mean cycletime graphs in Figure 5.5, we know that

$$\chi(n) \approx 29 \text{ and } \chi(1) \approx 2.$$

Let us replace the approximations with definite equalities then substitute the corresponding values for m and $\chi(m)$ in (5.2) to obtain

$$\begin{aligned} \chi(n) = 29 &= \alpha + \beta \\ \chi(1) = 2 &= \frac{\alpha}{n} + \beta \end{aligned}$$

These are two linearly independent equations with two unknowns. Therefore, a solution exists. Thus, $\alpha = \frac{27n}{n-1}$ and $\beta = \frac{2n-29}{n-1}$.

Hence, the approximate cycletime of a maxmin-m system for any $m \in \mathbb{N}$ $(1 \leq m \leq n)$ may be predicted with relative accuracy on the regular n-nbhd network. Notice that this cycletime is not dependent on the network size, and so the result is applicable to regular networks of all sizes.

Transient time and period of the maxmin-m system

In Figure 5.5, a special feature of the graphs for mean transient time and mean period is the existence of a peak at $m/(n+1) \approx 1/2$. This is present for all n and therefore suggests universality, i.e. for any n, the most complex behaviour is attained for the middle systems. Due to their bell-like shape, we shall refer to each loosely as the *transient bell* and *cyclicity bell* respectively.

Moreover, the corresponding curves for the 7-nbhd yield the highest transients and periods, particularly for the middle systems. Thus, this neighbourhood size produces most complex behaviour.

Let us demonstrate the behaviour produced by the middle system of each n-nbhd in the regular network that was studied, i.e. for the case $m_c(n) = (n+1)/2$ for $n = 3, 5, \ldots, 19$. We plot the mean transient times and periods of the middle systems from Figure 5.5 against n in Figure 5.6. Given n, we let the mean transient and mean period of $\mathcal{M}_{m_c}$ be denoted by K_{m_c} and ρ_{m_c} respectively.

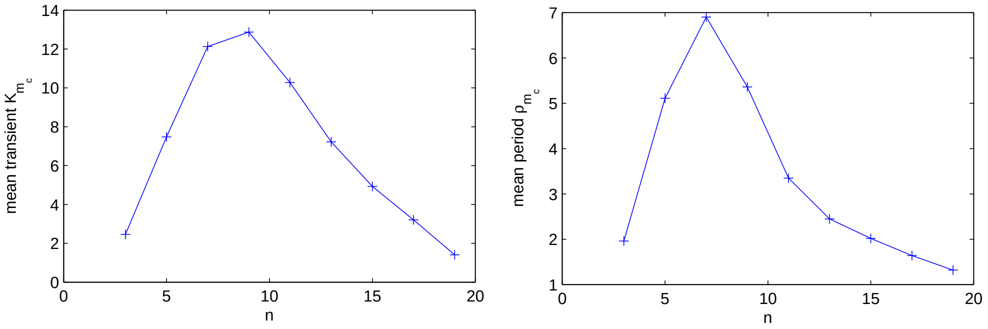


Figure 5.6: Mean transient time and mean period of the middle system of each n-nbhd in the regular network of size $N = 20$. The mean values are taken from 500 runs, each with $\mathbf{x}(0) = \mathbf{u}$ and a randomised processing time distribution with ξ radius 30. The zero transmission condition also holds..

Later we will show that the shape of the graphs in Figure 5.6 is prominent for larger network sizes also.

5.2.2 Non-zero transmission times

What difference does a non-zero transmission time make? Consider the non-zero τ radius r_2. As an example, let us take $r_2 = 120$, so that each transmission time is an integer value satisfying $1 \leq \tau_{ij} \leq 100$. To carry out this study, we implement Algorithm 5.2.1 again, but with the addition to step 1 that transmission times τ_{ij} are now also selected, from the uniform distribution (with equal probability), as integers between 1 and $r_2 \in \mathbb{N}$. Figure 5.7 shows the mean period of maxmin-m systems $\mathcal{M}_n$ and $\mathcal{M}_{m_c}$ arising from this amended algorithm to collate the results. It is seen that, for each n, the middle system produces the larger mean period. In fact, this mean period is the largest from all maxmin-m systems for each n. Therefore, we can conjecture that the cyclicity bell is true for non-zero transmission times also.

Consequently, to understand the effect of non-zero transmission times on the period of the maxmin-m system, it is sufficient to consider only the middle systems in each neighbourhood since those systems will show the highest complexity (in terms of period/cyclicity). The graphs in Figure 5.8 show the mean period of the middle systems for a fixed τ radius $r_2 = 30$ and various values of r. For reference, the inset shows the graph of mean ρ_{m_c} against n under the zero transmission condition for different ξ radii. The striking feature of this plot is that such a graph is almost the same no matter what the ξ radius. The peak for $n = 7$ is particularly maintained.

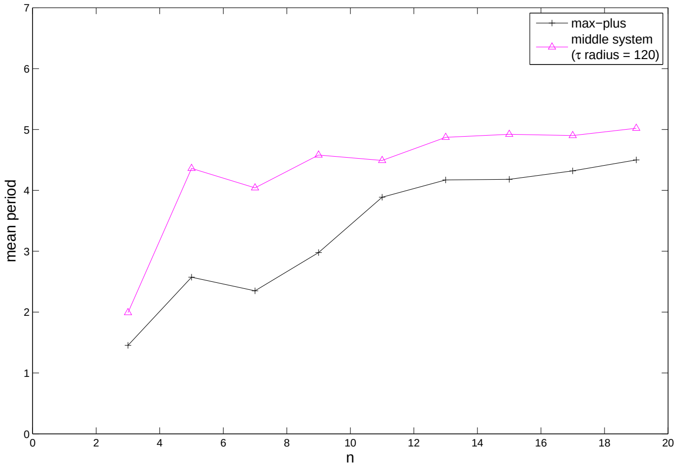


Figure 5.7: Mean period of maxmin-m systems on a regular n-nbhd network as a function of n. The network consists of 20 nodes and means are taken from 500 runs, each of which takes $\mathbf{x}(0) = \mathbf{u}$ and assigns processing and transmission times randomly as positive integers taking maximum value $r = 30$ and $r_2 = 120$, respectively.

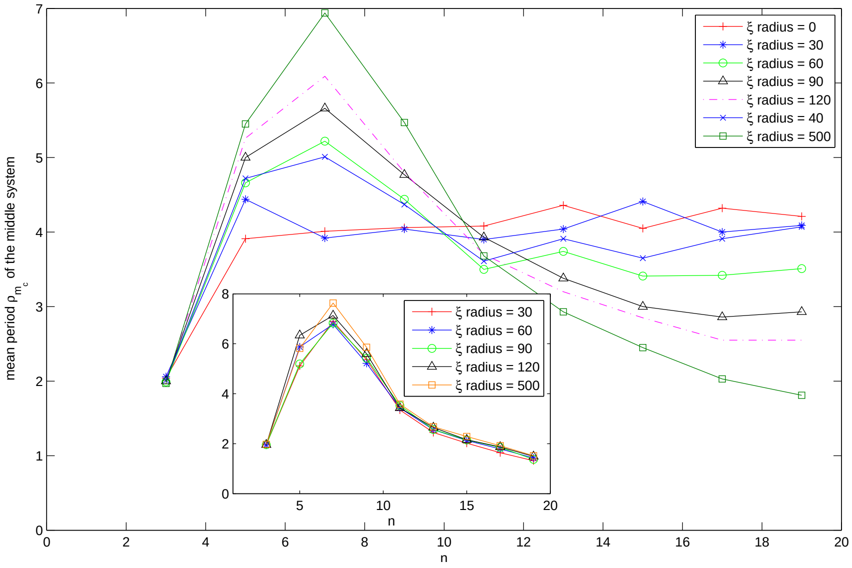


Figure 5.8: Mean period of the middle systems of a regular n-nbhd network as a function of n. The network consists of 20 nodes and means are taken from 500 runs, each of which takes $\mathbf{x}(0) = \mathbf{u}$ and assigns randomly the transmission times as positive integers taking maximum value $r_2 = 30$ and processing times taking maximum value ξ radius. The inset shows the corresponding results when $r_2 = 0$ for various ξ radii.

It is seen that the peak at $n = 7$ is more prominent for $r \gg r_2$. The suggestion is that, for very large ξ_i, the τ_i values become negligible, and the system behaves as if under the zero transmission condition. In this case, prediction of complex behaviour is straightforward as it is evidenced by the peak when $n = 7$. When τ radius becomes much larger than the ξ radius, this prediction is almost impossible since the ρ_{m_c} v n graph doesn't produce such a peak.

Further, for each n in a larger regular network, when we plot K_{m_c} and ρ_{m_c} against n, we obtain similar curves to those observed in Figure 5.6. This is shown in Figure 5.9, which also takes non-zero transmission times but which are relatively small compared to the processing times; we take ξ radius $r = 200$ and τ radius $r_2 = 20$ for all systems. In particular, the graphs for $N = 20, 40, 50$ and 100 take maximum value when $n = 7, 9, 9$ and 11 respectively. This suggests that the middle systems in the corresponding n-nbhd networks behave almost similarly with respect to their transients and periods.

Using this note, we may finish this section by conjecturing that, as $N \to \infty$, the maxmin-m system $\mathcal{M}_m$ on the regular network is most complex when it is the middle system of an 11-nbhd network.

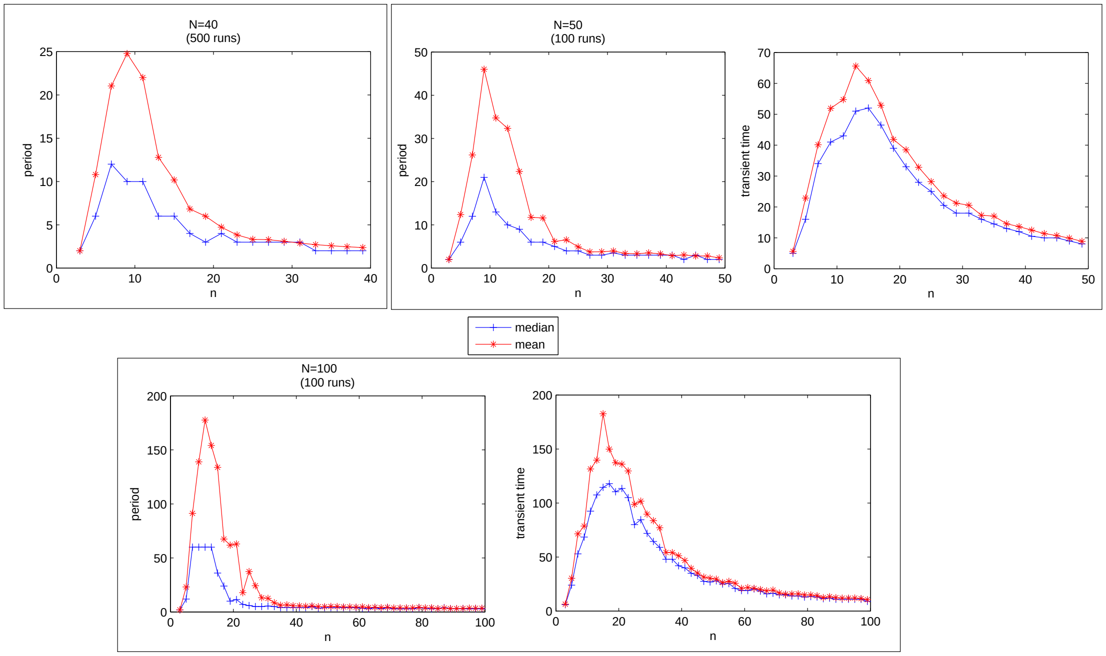


Figure 5.9: Mean transients and mean periods of the middle system in the regular n-nbhd network as functions of n for the network sizes $N = 40, 50$, and 100. The median is also indicated for reference. The ξ radius is 200 and the τ radius is 20 while the initial state is $\mathbf{x}(0) = \mathbf{u}$ in all experiments.

5.3 A cellular automaton model in maxmin-plus time

This section examines the results of implementing cellular automaton rule 150 in asynchronous time generated by the maxmin-m system. The zero transmission condition is employed while the ξ radius is fixed at $r = 30$. For every result obtained from Step 2 of Algorithm 5.2.1, i.e. for every triple (K, ρ, χ) obtained, the corresponding maxmin-m system was also used as the framework of asynchronous times on which a CA simulation is run. Thus, we also calculated the CA transient time K_C, CA period p and the Shannon and word entropies.

Figure 5.10 plots the means of the CA transient time (denoted mean K_C) obtained as a result of carrying out 500 runs, each of which chooses the processing time distribution randomly as integers with uniform probability and satisfying $1 \leq \xi_i \leq r$. The initial update time is $\mathbf{x}(0) = \mathbf{u}$ while the initial CA state is that in (3.18); Indeed, the CA results in Chapter 3 correspond to the case $m = n$ here. For comparison, the mean transient times that we obtained from Algorithm 5.2.1 are also plotted. Likewise, Figure 5.11 plots the mean CA period, denoted mean p on the same axes as the mean period of the update times, mean ρ.

While the CA transient time curves do not follow the maxmin-m transient time curves exactly, there is a general agreement between the two, more so for the case $n = 7$, which produces largest transient times, as was the case for mean K. However, the CA period curves are more bell-like, even though the CA periods themselves are much larger than the update time periods (a zoom in to the update time period plots will reveal a bell shape). The curious case of the 19-nbhd curve in Figure 5.11 is carried over here also. An intuitive explanation for this is that a 19-nbhd yields a complete graph, i.e. one in which each node is contained in the neighbourhood of every other node. Thus, the CA states are the same for each node after the initial time; in particular, the CA space-time plot is homogeneous, which evidently has the fastest CA transient time and smallest CA period. In Figure 5.11, the largest CA periods are produced by the n-nbhds when $n = 9$, 11, and 13.

Finally, Figure 5.12 plots the mean entropies as functions of m. Most neighbourhoods in the regular network yield a bell for the mean entropies with m and, as in the small example in the previous section, this bell is skewed with entropy values for lower m yielding lower entropies than those for the largest values of m. For most n, the graphs reach their peak when m is slightly larger than the middle systems.

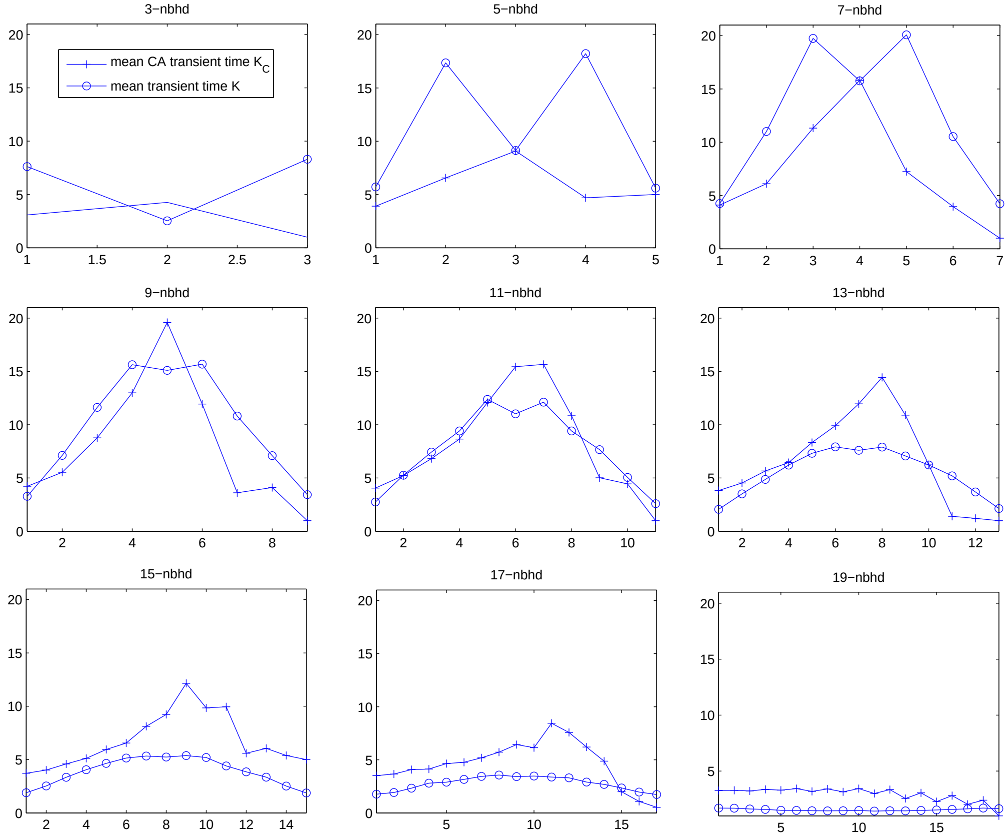


Figure 5.10: Mean CA transient time (vertical axis) as a function of m (horizontal axis) for a regular n-nbhd network on 20 nodes. The mean is taken from 500 runs where, for each run, the processing times are chosen randomly as integers with uniform probability and satisfying $1 \leq \xi_i \leq 30$; the zero transmission condition also holds. For all 500 runs, the initial condition is $\mathbf{x}(0) = \mathbf{u}$ while $\mathbf{s}(0)$ is as in (3.18).

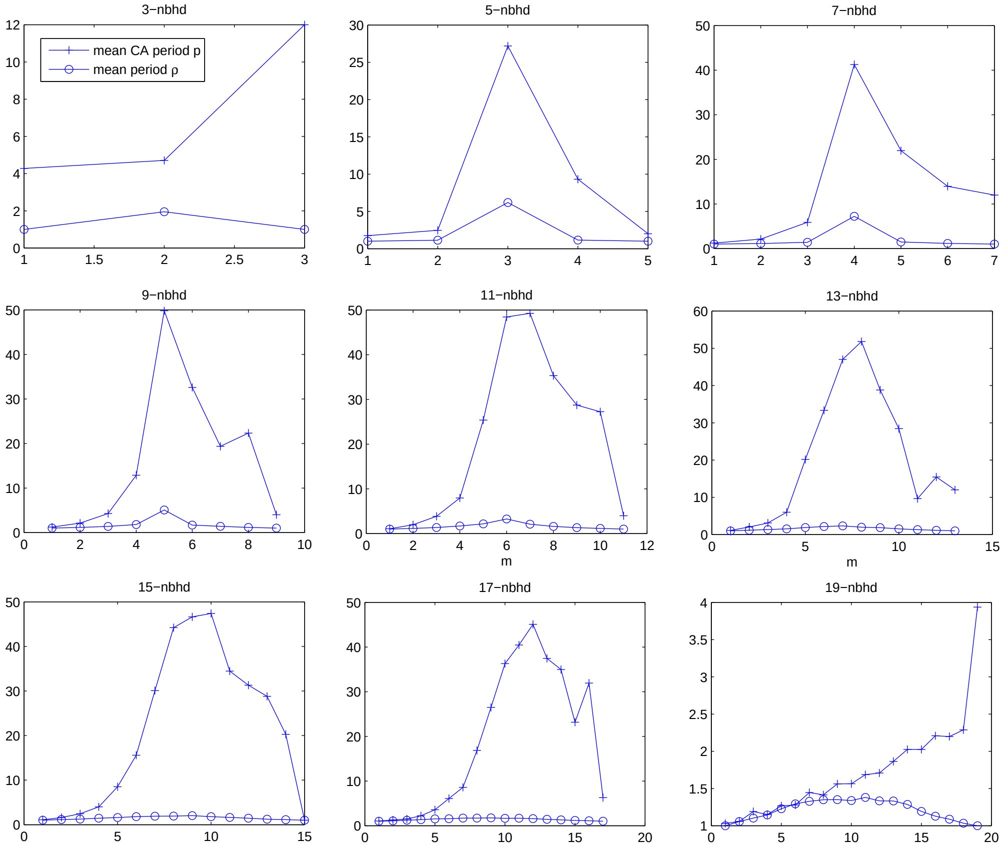


Figure 5.11: Mean CA period (vertical axis) as a function of m (horizontal axis) for a regular n-nbhd network on 20 nodes. The mean is taken from 500 runs where, for each run, the processing times are chosen randomly as integers with uniform probability and satisfying $1 \leq \xi_i \leq 30$; the zero transmission condition also holds. For all 500 runs, the initial condition is $\mathbf{x}(0) = \mathbf{u}$ while $\mathbf{s}(0)$ is as in (3.18).

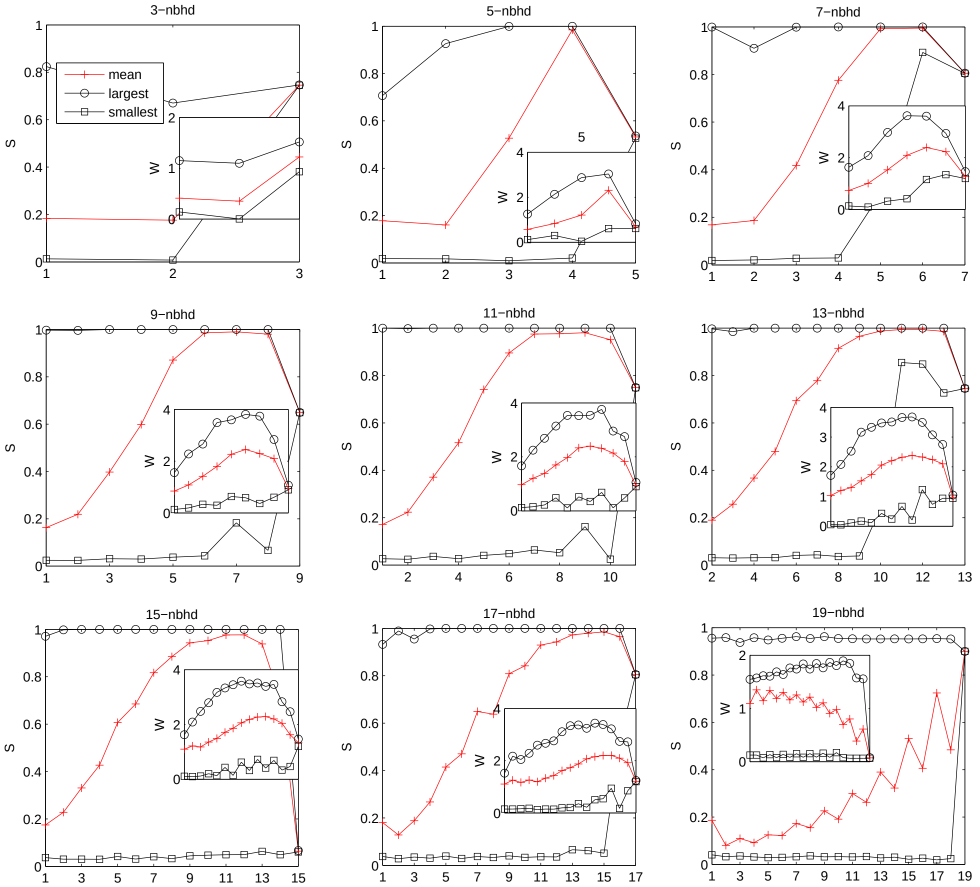


Figure 5.12: Mean entropy plots as functions of m (horizontal axis) for a CA on a regular n-nbhd network on 20 nodes. The mean is taken from 500 runs where, for each run, the processing times are chosen randomly as integers with uniform probability and satisfying $1 \leq \xi_i \leq 30$; the zero transmission condition also holds. The initial condition is $\mathbf{x}(0) = \mathbf{u}$ for all 500 runs, while $\mathbf{s}(0)$ is as in (3.18). The largest and smallest value obtained is also plotted to indicate the spread of results.

5.4 Link between the maxmin-m model and CA

In this section, we attempt to formulate mathematical links between the maxmin-m system and the corresponding cellular automaton model. In particular we would like to see if prediction of CA behaviour, at least in terms of the Wolfram Classes, is possible given the maxmin-m model.

5.4.1 Maxmin-m analysis

The cycletime $\chi(\mathcal{M}_m)$ of a maxmin-m system $\mathcal{M}_m$ has been shown to be predictable. Since $\mathcal{M}_m$ can be reduced to a max-plus system, $\chi(\mathcal{M}_m)$ is characterised by at least one circuit in the reduced max-plus graph. (Recall that this reduced graph may be $\mathcal{G}^{(r)}(R)$ such that $R^{(r)} = P_1^{(r)} \otimes \cdots \otimes P_g^{(r)}$ for some $g \geq 2$. In this case, the period $\rho(\mathcal{M}_m)$ is conjectured to be equal to g). Therefore, once we know $\chi(\mathcal{M}_m)$, we look for the corresponding circuit(s) in the reduced graph to determine the cyclicity $\sigma(\mathcal{M}_m)$ of the maxmin-m system.

The numerical experiments suggest that the middle systems are likely to yield the highest cyclicity. Consider the set $\mathcal{P}$ of all distinct max-projections of $\mathcal{M}_m$. For fixed N and neighbourhood size n, we can plot $|\mathcal{P}| = ({}^nC_m)^N$ as a function of m to see that it is a bell-like curve with highest value when $m = n/2$ if n is even and $m = (n+1)/2$ when n is odd. Figure 5.13 plots this function for $N = 8$ and $n = 7$, and gives an indication towards a link between the bells that we obtained. It is conjectured that a large number of max-projections increases the likelihood of having a reduced graph periodic orbit with very large period g.

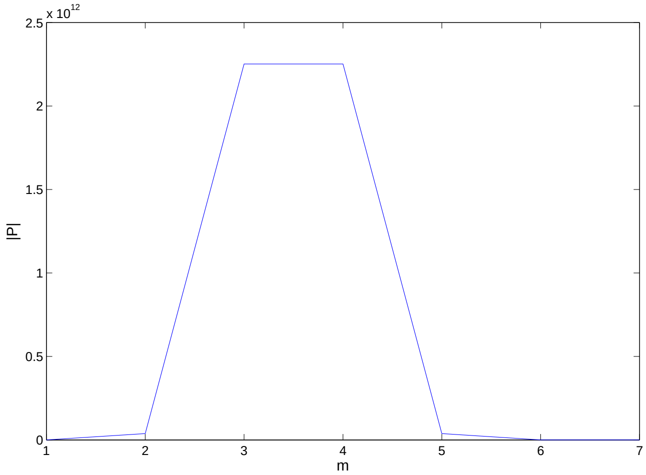


Figure 5.13: The number $|\mathcal{P}| = ({}^nC_m)^N$ of distinct max-projections as a function of m. Here, $N = 8$ and $n = 7$.

5.4.2 Maxmin-m as a predictor of CA behaviour

The bell has been seen to be a common shape to all observed asymptotic quantities, except for the cycletime. In particular, let us look at the cyclicity bell.

Consider a maxmin-m system that yields a periodic regime with period $\rho > 1$. The update times of node i in this regime are $x_i(k), x_i(k+1), \ldots, x_i(k+\rho-1)$. In terms of the contour plot, a large cyclicity (or period) means that the intervals between update times for this node will be more variable than for a maxmin-m system with smaller cyclicity. Thus, without loss of generality, suppose that most differences in consecutive update times in this regime are different, i.e. $x_i(j+1) - x_i(j) \neq x_i(l+1) - x_i(l)$ for most contour pairs j and l in this regime $(j \neq l)$.

Now suppose that $s_i(k) \neq s_i(k+1)$ for all contours in this regime. Then, along the CA evolution of node i, there exist words of the differing lengths $x_i(j+1)-x_i(j)$, $x_i(l+1)-x_i(l)$, etc. A larger period ρ implies a larger number of such different word lengths. Therefore, the word entropy W is highest for such a large period and it is likely to be lowest for the smallest period. Since ρ is likely to be largest for the middle system, then W is likewise. The systems $\mathcal{M}_1$ and $\mathcal{M}_n$ yield the smallest cyclicities, but the numerical experiments indicate a slightly higher word entropy for the CA on $\mathcal{M}_n$; we ascribe this to the rule employed. Indeed, given a node is in CA state s_j, the node is most likely to retain its state due to rule 150 when $m = n$, shown as follows.

Probabilistic study

Since most of the CA work has been qualitative in nature, a quantitative analysis that aims to predict CA behaviour would be an additional benefit. Thus, here, we take a mean field approach, which has been used in the past to conduct similar studies [19, 2, 1]. This approach uses probability to give an indication as to the type of CA behaviour that the maxmin-m system is likely to yield.

Here, we assume that the maxmin-m system is reduced to a unique max-plus system where the reduced graph takes neighbourhood size m. Thus, we forego the possibility of simultaneity, which may lead to this neighbourhood size being greater than m for some nodes (where $m \neq n$). Indeed, during simultaneity, the neighbourhood size would be dependent on the parameters ξ_i and τ_{ij} as well as the initial condition $\mathbf{x}(0)$, making the analysis less tractable than required.

A unique reduced graph implies that, following such a reduction, i.e. after a long period of time, we obtain a deterministic max-plus system with the evolution of CA states on each contour k outlined by its state transition graph. Thus, our analysis proceeds with the assumption that we are working with a fixed max-plus graph with neighbourhood size m.

According to CA rule 150, a node is in CA state 1 on a contour if an odd number of nodes from its neighbourhood of m nodes is in the same CA state on the previous contour. Now let $p(k)$ denote the probability that a node takes state 1 on epoch k. Then, for fixed N and a fixed neighbourhood size m, according to CA rule 150, the probability that the node is in the same state on the next epoch is given by

$$p(k+1) = \sum_{j=0}^{m_\star} {}^{m}C_{2j+1} p(k)^{2j+1}(1-p(k))^{m-2j-1}$$

where

$$m_\star = \begin{cases} \frac{m-1}{2} & \text{if } m \text{ is odd} \\ \frac{m-2}{2} & \text{if } m \text{ is even} \end{cases}$$

Note that $p(k+1)$ is independent of N. Nonetheless, the probability should still reveal some insight into CA behaviour of maxmin-m systems.

Given any $p(k)$, by plotting the graph of $p(k+1)$ as a function of m, we observe that $p(k+1)$ takes highest value when $m = n$. This feature is seen in curves obtained from other combinations of N, m and $p(k)$ also. For example, consider the case $N = 20$, $n = 9$, with $p(k) = 1/20$. We take $p = 1/20$ here to allow comparison with our experiments where the initial CA state took one node out of twenty to be ON and all others OFF. Then, the graph in Figure 5.14(a) reaches its highest value when $m = n$. This probability shows that, in the max-plus system, given that a cell state is equally likely to be 1 or 0, the cell will likewise be in CA state 1 or 0 on the next time step with equal probability. Thus, by interpreting this probability as one of the densities that is used to formulate the Shannon entropy, we obtain a Shannon entropy $S = -(0.5\log_2(0.5) + 0.5\log_2(0.5)) = 1$ here. The arguments that were used in Chapter 1 to analyse CA patterns in terms of S may now be employed to conclude that the maxmin-n (max-plus) system is most likely to produce a periodic CA pattern faster (in terms of number of iterations).

By interpreting the probabilities likewise for the other maxmin-m systems, we can obtain a Shannon entropy curve resembling those found in Figure 5.12 arising from our experiments. Figure 5.14(b) shows the behaviour of this Shannon entropy for this section. In particular, notice that the curve for the 9-nbhd system in Figure 5.12 is similar except for the case $m = n$, where. Thus, we have an idea of the behaviour of CA that arises in our system, and the CA results of this chapter, particularly the Shannon entropy results, go some way towards backing this up by yielding similar curves as functions of m.

5.4.3 Reduced graphs as the underlying graph of CA

As we have seen in this thesis, the in-degree of a node in a reduced max-plus graph is not necessarily equal to m; it may be greater than m for some nodes. Nevertheless, the in-degree of each node is at least one. It easy to verify that this implies that the reduced graph contains at least one circuit.

This reduction to a graph with at least one circuit means that the graph can be seen as a MSCS decomposition such that each MSCS will affect only those nodes downstream of it. E.g. in Figure 5.4, the MSCS [5] (containing nodes 5, 6, and 7) affects all other nodes, i.e. the corresponding CA states will be transmitted to all other nodes. In contrast, the CA states of the MSCS [1] (containing nodes 1, 2, 3, and 4) will only ever affect itself; likewise for MSCS [8] (which contains nodes 8, 9, and 10). Thus, another advantage of a reduced max-plus graph is that it shows which nodes' CA states will be a function of which other nodes as $k \to \infty$. In Figure 5.4 again, $s_{[i]}(k+1) = f(s_{[i]}(k), s_5(k), s_6(k), s_7(k))$ for $i = 1, 2, 3, 4$, where $s_{[i]}(k)$ denotes the CA state of the MSCS $[i]$ at epoch k. Likewise, $s_{[j]}(k+1) = f(s_{[j]}(k), s_5(k), s_6(k), s_7(k))$ for $j = 8, 9, 10$.

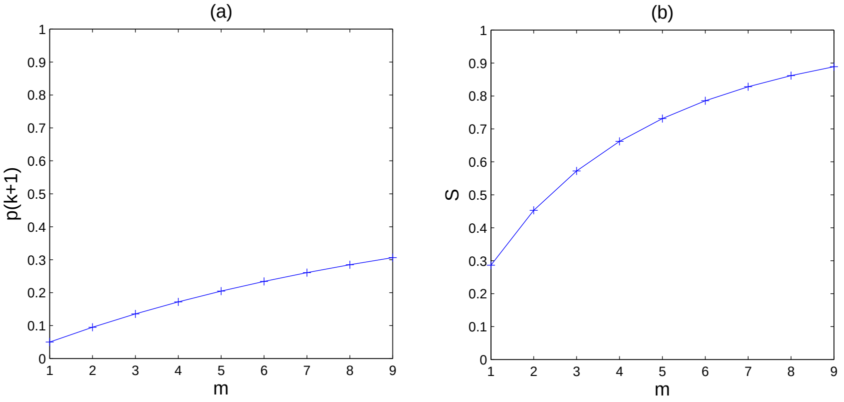


Figure 5.14: (a) The probability $p(k+1)$ that a node is in CA state $s \in \{0, 1\}$ on epoch $k+1$ for the maxmin-m system on a network that takes neighbourhood size $n = 9$. (b) Shannon entropy S as a function of m, where S is obtained by interpreting the probabilities in (a) as densities in the formula. The CA rule is rule 150 and $p(k) = 1/20$ here.

5.5 Summary

In this chapter, we have explored the evolution of cellular automata in the maxmin-m model. As a result of the maxmin-m system $\mathcal{M}_m$ being reduced to a max-plus system as $k \to \infty$, the task of analysing such CA has been reduced to one with which we are more familiar.

The maxmin-m asynchronous update time system has been shown to be fairly easy to predict, especially on the regular n-nbhd network. Indeed, we have shown that $\chi(\mathcal{M}_m)$ is an almost linear function in m. Broadly speaking, for $m_1 < m_2$, the maxmin-m_1 system iterates more quickly than the maxmin-m_2 system. Thus, in Figure 5.1, for example, although each maxmin-m system has iterated the same number of times (i.e. 20), the min-plus system does this in 25 real time units, whereas the max-plus system takes 100 time units.

Figure 5.1 also helps to understand the loss of information due to the inequality $m < n$. We have shown that maxmin-m systems for $m < n$ are likely to lose CA states as the number of iterations becomes large; nodes in the reduced graphs have neighbourhood size ranging from m to n, depending on the presence of simultaneity. Thus, the maximum number of lost states is $n - m$ when simultaneity is avoided. The end result is that there is no simple bijective relation between the synchronous CA and CA in such maxmin-m time. This is unlike the case $m = n$, where such a bijection is easy to identify. Consequently, the periodic orbit of reduced graphs may be so complex as to make difficult the forming of a state transition graph of CA states for those complex maxmin-m systems.

Importantly, it has been shown that prediction of the complexity of CA may nevertheless be possible due to a correspondence with the complexity of $\mathcal{M}_m$ in the form of the 'bells'. These bells indicate least complexity for small m and large m while the most complex CA is attained when $m \approx n/2$, i.e. when each node updates after receiving half of the inputs from its neighbourhood. Prediction of CA behaviour might thus be possible without having to run the CA rule itself.

Chapter 6

Conclusions

6.1 Summary

We have shown max-plus algebra to be a useful tool for the implementation of CA in asynchronous time. The contour plot was introduced, over which a cellular automaton model may be simulated with ease. For the regular n-nbhd network $\mathcal{G}$, the asymptotic behaviour of update times may be obtained from the largest average elementary circuit weight in $\mathcal{G}$ due to the strong connectedness of the network. Thus, while cells may update at different real times (indicated by a non-horizontal contour shape), successive updates are asymptotically carried out at the same interval, the interval being the cycletime. The update times may be regarded as being synchronised after the transient time K; but not synchronised in real time. Likewise, the min-plus system over the same timing dependency graph also yields a uniform cycletime, characterised by the smallest average elementary circuit weight in $\mathcal{G}$.

Both max-plus and min-plus systems produce small cyclicities; allied with their small transient times, the two systems may be regarded as being the least complex from all maxmin-m systems ($1 \leq m \leq n$). We have shown a bijective relation between the synchronous CA and the max-plus CA, particularly in terms of the state transition graphs, where the CA states on contours will be exactly equal to those states in the synchronous CA. Therefore, prediction of CA behaviour on contours is possible in a max-plus system, and this is supported by the Shannon and word entropies since they share similar values with the synchronous CA under the same initial conditions. The min-plus CA is not as predictable due to its long-term behaviour. It has been shown that a maxmin-m system where $m \neq n$ will reduce to a periodic max-plus system in the long time limit. Since this reduction is dependent on the initial condition $\mathbf{x}(0)$, there is no fixed state transition graph for these systems. Therefore, CA behaviour is not easily predicted. Nevertheless, the CA entropies for a min-plus system suggest homogeneous behaviour, and this implies least complexity.

For $m \neq n$, a feature of the maxmin-m system is that the asymptotic reduced graph orbit may have period $g > 1$. In this case, the STG is not as straightforward to construct. This is especially the case if simultaneity is present. Due to the rationally dependent values assigned to the processing and transmission times in our experiments, simultaneity was observed after a relatively small number of iterations. Thus, simultaneity distinguishes the maxmin-m system from other asynchronous timing systems. Indeed, it also highlights the maxmin-m system as a special maxmin-plus system in which, contrary to established thoughts, the initial evolution equation and max-projections may not be indicative of the asymptotic behaviour of the system.

The asymptotic update times of the maxmin-m system have been shown to be fairly straightforward to predict. For the regular n-nbhd network, the cycletime $\chi(\mathcal{M}_m)$ is an approximately linear function of m. By knowing the largest and smallest processing and transmission times, the cycletime of the min-plus and max-plus system can be obtained, and a straight line drawn through the points $(1, \chi(\mathcal{M}_1))$ and $(n, \chi(\mathcal{M}_n))$ enables the prediction of the cycletime for all maxmin-m systems. Once this is known, associated circuits may be identified, from which an approximate value for the period/cyclicity of $\mathcal{M}_m$ may be obtained. This may also give an indication of the cyclicity bell.

The cyclicity bell and, indeed, transient and CA entropy bells indicate that prediction of complexity of CA may be possible, at least in qualitative terms. There is a good match between the CA period bell and CA transient bells to the maxmin-m cyclicity and transient bells. The word entropy also displays a bell shape. These bells indicate least complexity for the smallest and largest m values, while the most complex CA is attained when $m \approx n/2$, i.e. when each node updates after receiving half of the inputs from its neighbourhood. Prediction of CA behaviour might thus be possible without having to run the CA rule itself.

The asynchrony due to the maxmin-m system has thus been shown to be deterministic and not stochastic, yet able to generate a spectrum of behaviour, from homogeneity to heterogeneity. The maxmin-m system therefore, proves to be a complex system with most interest occurring at the halfway point between total homogeneity and total oscillatory behaviour. This compares favourably with the first statement made in Chapter 1 that initiated the thesis.

6.2 Further work

A natural progression from the work in this thesis is to generalise the maxmin-m model for use on arbitrary networks. The research question would ask: If such an extension of the maxmin-m model is possible, then how do the corresponding analyses and results differ from the regular n-nbhd work?

We detail some preliminary experiments in response to this.

6.2.1 Random n-nbhd network

In this section, we study the maxmin-m model where each node in the timing dependency graph takes in-degree n, fixed, but the n neighbours are assigned at random with equal probability from all N nodes. We call such a network the *random n-nbhd network*. This is not a stochastic approach to the maxmin-m model as we fix the network once each neighbourhood is chosen. Therefore, the weighted adjacency matrix P of the network is fixed for all iterations k.

Unlike the regular network, the random network may not be strongly connected. Thus, just as in the argument used for reduced max-plus graphs, in a random n-nbhd network, the max-plus system may asymptotically settle into a periodic regime whereby some nodes update their state more quickly than other nodes. Therefore, we take the mean of the cycletime vector χ for each result, just as we did for the maxmin-m systems in the previous chapter.

We now present the asymptotic behaviour of the random n-nbhd network on 20 nodes. Since randomness further complicates the network, the results are obtained computationally. Moreover, the results complement the corresponding regular n-nbhd results on the same network size in Chapter 5. Thus, the algorithm that is used to process the results is Algorithm 5.2.1. First, we fix the random n-nbhd network via a randomisation process which involves rewiring the regular n-nbhd network. The following algorithm demonstrates this process, which is inspired by a technique used in [39] to generate similar random networks.

Algorithm 6.2.1 (*Rewiring*)**.** Consider a regular n-nbhd network on N nodes, denoted $\mathcal{G}_{\text{reg}}$. For all nodes in $\mathcal{G}_{\text{reg}}$, i.e. for $i = 1 \ldots, N$, carry out the following.

1. From the neighbourhood $\mathcal{N}_i$ of node i, choose γ nodes at random with equal probability. Denote these nodes $\mathcal{N}_i^\gamma$.

2. Select γ nodes at random with equal probability from those nodes in $\mathcal{G}_{\text{reg}}$ that are not in $\mathcal{N}_i$ and denote the set of these γ nodes by Γ_i.

3. Replace $\mathcal{N}_i^\gamma$ with Γ_i. In set theory notation, the new neighbourhood of i may be written as $(\mathcal{N}_i \setminus \mathcal{N}_i^\gamma) \cup \Gamma_i$.

We call γ the *randomisation depth.* It indicates the extent of the deviation from the regular n-nbhd network $\mathcal{G}_{\text{reg}}$. For $\gamma = n$, each neighbourhood $\mathcal{N}_i$ in $\mathcal{G}_{\text{reg}}$ is altered to a totally different neighbourhood. We define this to be the *fully random* n-nbhd network, and therefore bound γ from above by n.

Just as was done in Chapter 5, we implemented the cellular automaton with each run of Algorithm 5.2.1, where the initial CA state was, again, the same as (3.18). For this study, we also implemented the zero transmission condition and fixed the ξ radius to $r = 30$. Thus, for one value of m in Algorithm 5.2.1, we obtained 500 transient times K, cycletimes χ, periods ρ, and likewise 500 Shannon entropies S and word entropies W; the mean of each quantity was subsequently taken. Note that such a random graph ensured no fast detection of a CA period. Therefore, the calculation of CA period p and CA transient time K_C was omitted for this study.

As an example, we display the results obtained for the 7-nbhd in Figure 6.1 for odd values of γ.

As may be expected, for a low randomisation depth, the results are approximately similar to the regular 7-nbhd results. However, what is most significant about these results is that, even for a fully random n-nbhd network, we observe similarities to the regular n-nbhd network. These similarities are the following.

- Bells are obtained for the mean transient time and mean period, with the peak at $m \approx n/2$
- The cycletime is almost linear with m
- The mean word entropy follows a bell shape with m.

Further, for fixed m, the transient time is decreasing with γ. Thus, as the network connections become more randomised, a periodic regime is observed more quickly. The period of the regime itself is increasing with γ, and this increase is approximately by $\rho/4$.

It is noticeable, however, that the mean Shannon entropy increases to 1 rapidly with m for all γ values shown. A random n-nbhd thus increases the complexity of the evolution of CA states 1 and 0.

We conclude therefore, that the cyclicity bell is an expected feature of any n-nbhd network, be it regular or random. Moreover, the cycletime may be calculated with relative accuracy, even if the n-nbhd is assigned at random. In other words,

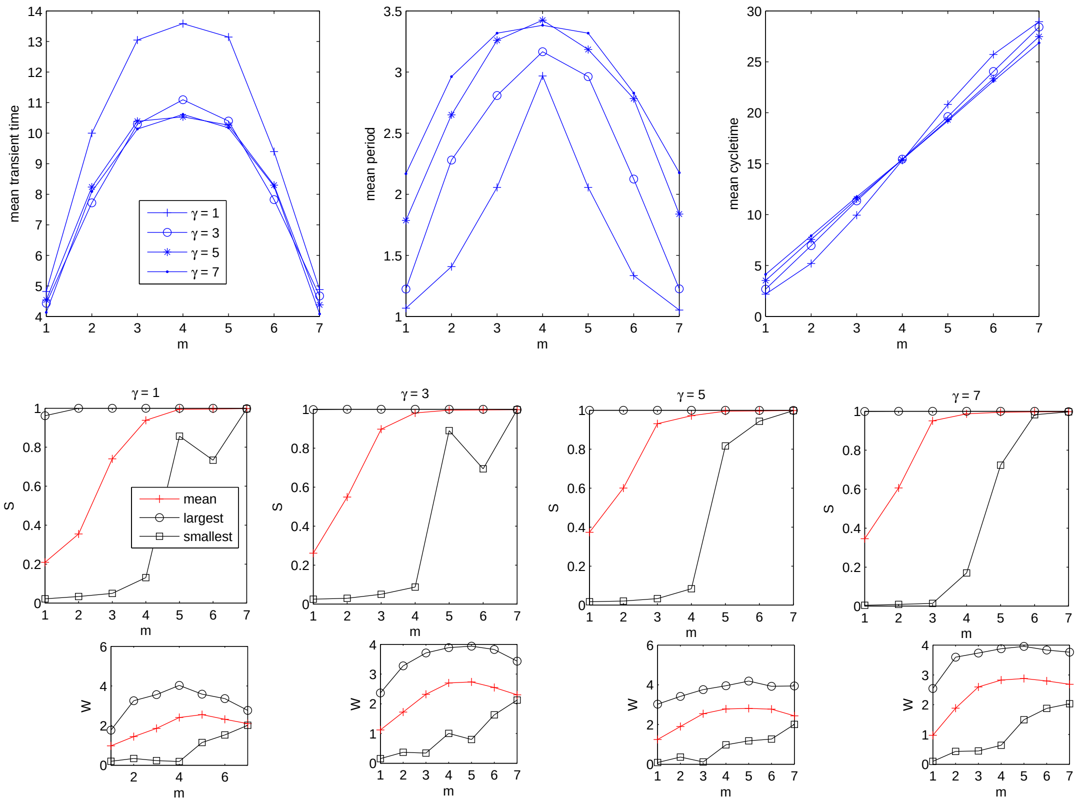


Figure 6.1: Mean asymptotic results for a cellular automaton model on the random 7-nbhd network on 20 nodes. The mean is taken from 500 runs, each choosing processing times randomly from the uniform distribution of integers with ξ radius 30. Transmission times are all zero. The randomisation depth is indicated by γ. The word entropy W is shown as a function of m in the smallest boxes, above which are the Shannon entropy S graphs as a function of m.

given a timing dependency graph that takes neighbourhood size n, we can predict the asymptotic performance of a maxmin-m system relatively easily. The CA results, in particular, the word entropy curves as functions of m, back this up by indicating that the entropy measures of the CA pattern resulting from such an unbiased rule as rule 150 may be developed to a satisfactory extent without having to run the CA itself.

6.2.2 The maxmin-ω system

If we are to aspire towards a model of asynchronous update times on arbitrary networks, then the next step is to think about a variable neighbourhood size since networks that model real-life applications, for example the scale-free networks, are generally not regular. Indeed, flexibility in terms of neighbourhood size also generalises the asynchronous model.

Variable neighbourhood size

In this section, we denote by n_i the neighbourhood size of node i and develop a maxmin-plus model in which the state of each node is updated after receiving a fraction ω of inputs from its neighbourhood. We fix ω and note that this implies that each node will update after receiving a number of inputs m_i that is dependent on node i. Thus, $m_i = \omega n_i$. Since this could result in a non-integer value for m_i, we employ the ceiling operator so that, to be precise, $m_i = \lceil \omega n_i \rceil$.

We shall call this a *maxmin-ω* system. Since neighbourhood size is variable with i, we can represent all the neighbourhood sizes by the vector $\mathbf{n} = (n_1, n_2, \ldots, n_N)$. Likewise, we can also include each m_i in the vector $\mathbf{m} = (m_1, m_2, \ldots, m_N)$. Notice that $\mathbf{m} = \lceil \omega.\mathbf{n} \rceil$. We illustrate the development of the evolution equations for the model using the following example.

Example 6.2.1. Consider a network size $N = 5$ and $\mathbf{n} = (2, 3, 2, 4, 5)$. The network is shown in Figure 6.2. Let $\omega = \frac{1}{2}$. Then $\mathbf{m} = (1, 2, 1, 2, 3)$. Let $x_{ij}(k)$ denote the arrival time of input from node j to node i, i.e. $x_{ij}(k) = x_j(k) \otimes \tau_{ij}$. Then, in matrix-vector form, this maxmin-ω system for $\omega = 1/2$ is written

$$\mathbf{x}(k+1) = P_1 \otimes \mathbf{x}(k) \ominus P_2 \otimes \mathbf{x}(k) \ominus \cdots P_{10} \otimes \mathbf{x}(k)$$

where we list the first few max-projections P_i as follows.

$$P_1 = \begin{pmatrix} \xi_1\tau_{11} & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \xi_2\tau_{21} & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \xi_3\tau_{33} & \varepsilon & \varepsilon \\ \varepsilon & \xi_4\tau_{42} & \xi_4\tau_{43} & \varepsilon & \varepsilon \\ \xi_5\tau_{51} & \xi_5\tau_{52} & \xi_5\tau_{53} & \varepsilon & \varepsilon \end{pmatrix}, P_2 = \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \xi_1\tau_{15} \\ \varepsilon & \varepsilon & \xi_2\tau_{23} & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \xi_3\tau_{34} & \varepsilon \\ \varepsilon & \varepsilon & \xi_4\tau_{43} & \xi_4\tau_{44} & \varepsilon \\ \varepsilon & \xi_5\tau_{52} & \xi_5\tau_{53} & \xi_5\tau_{54} & \varepsilon \end{pmatrix},$$

$$P_3 = \begin{pmatrix} \varepsilon' & \varepsilon' & \varepsilon' & \varepsilon' & \varepsilon' \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \xi_2\tau_{25} \\ \varepsilon' & \varepsilon' & \varepsilon' & \varepsilon' & \varepsilon' \\ \varepsilon & \varepsilon & \varepsilon & \xi_4\tau_{44} & \xi_4\tau_{45} \\ \varepsilon & \varepsilon & \xi_5\tau_{53} & \xi_5\tau_{54} & \xi_5\tau_{55} \end{pmatrix}, P_4 = \begin{pmatrix} \varepsilon' & \varepsilon' & \varepsilon' & \varepsilon' & \varepsilon' \\ \varepsilon' & \varepsilon' & \varepsilon' & \varepsilon' & \varepsilon' \\ \varepsilon' & \varepsilon' & \varepsilon' & \varepsilon' & \varepsilon' \\ \varepsilon & \xi_4\tau_{42} & \varepsilon & \varepsilon & \xi_4\tau_{45} \\ \xi_5\tau_{51} & \varepsilon & \varepsilon & \xi_5\tau_{54} & \xi_5\tau_{55} \end{pmatrix}.$$

The other max-projections can easily be deduced.

At the moment, the exact values of the entries in each matrix P_i are not as important as we wish to highlight only the existence of these entries. Thus, observe that the immediately apparent feature of the system in the example above is the coexistence of ε and ε' within the same matrix. When thinking about employing the duality theorem, we would require the graphs $\mathcal{G}(P_i)$, in relation to which we know no amenable interpretation of ε' within a max-plus matrix (or indeed of ε within a min-plus matrix). To overcome this issue, we first note that ε' never appears alone but in a row filled with the same entry. Therefore, if row j is such a row in P_i, we change it to the row j from any one of the other matrices P_k $(i \neq k)$, as long as that row contains at least one non-zero entry not equal to ε'. The nodal evolution equation is unchanged when such elements are repeated due to idempotency, for example,

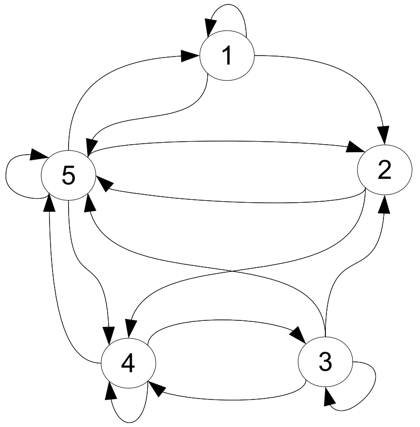


Figure 6.2: Network on 5 nodes taking arbitrary neighbourhood size.

$(x_1 \oplus x_2) \ominus (x_1 \oplus x_2) = (x_1 \oplus x_2)$.

The duality theorem may now be employed without difficulty, just as was done in the previous chapter. Reduced graphs may also be obtained likewise.

Example 6.2.2. Let us now consider such a variable neighbourhood graph. We take $N = 10$ and $\mathbf{n} = (9, 8, 10, 6, 4, 4, 10, 9, 2, 3)$, where the actual neighbourhood nodes are given in the following list.

i	$\mathcal{N}_i$
1	$\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$
2	$\{1, 2, 4, 6, 7, 8, 9, 10\}$
3	$\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$
4	$\{1, 4, 6, 7, 9, 10\}$
5	$\{1, 4, 7, 10\}$
6	$\{3, 4, 5, 6\}$
7	$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$
8	$\{1, 2, 4, 5, 6, 7, 8, 9, 10\}$
9	$\{4, 5\}$
10	$\{4, 6, 9\}$

Given initial condition $\mathbf{x}(0) = \mathbf{u}$, we gather asymptotic quantities in the form of the mean of all transient times, periods, cycletimes and the mean of the entropies from the resulting cellular automaton model. The mean is taken from 20 runs of the system, each with processing and transmission times selected randomly from the uniform distribution of integers with ξ radius $r = 5$ and τ radius $r_2 = 5$. Figure 6.3 shows the graphs of the mean quantities, which are plotted as functions of ω, where $0 \leq \omega \leq 1$ and ω is incremented in step sizes 0.05. The CA entropies were taken from 200 iterations of the system.

Whilst the cycletime is almost linearly increasing with ω, the other quantities take largest value for $\omega \approx 0.5$.

The above example shows that the bells that classify complexity in the long time limit can be obtained even on a graph with variable neighbourhood.

6.2.3 Maxmin-m variable with node i

The work of the previous section is easily extended to an update time system which takes $m = m_i$, variable with each node i. Due to familiarity, we consider a regular 3-nbhd network on 5 nodes. Suppose $\mathbf{m} = (1, 1, 2, 2, 3)$. Then, applying the same procedure as in the previous section gives the following evolution equations in matrix-vector form.

$$\mathbf{x}(k+1) = P_1 \otimes \mathbf{x}(k) \ominus P_2 \otimes \mathbf{x}(k) \ominus P_3 \otimes \mathbf{x}(k)$$

where

$$P_1 = \begin{pmatrix} \xi_1\tau_{11} & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \xi_2\tau_{21} & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \xi_3\tau_{32} & \xi_3\tau_{33} & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \xi_4\tau_{43} & \xi_4\tau_{44} & \varepsilon \\ \xi_5\tau_{51} & \varepsilon & \varepsilon & \xi_5\tau_{54} & \xi_5\tau_{55} \end{pmatrix}, P_2 = \begin{pmatrix} \varepsilon & \xi_1\tau_{12} & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \xi_2\tau_{22} & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \xi_3\tau_{33} & \xi_3\tau_{34} & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \xi_4\tau_{44} & \xi_4\tau_{45} \\ \xi_5\tau_{51} & \varepsilon & \varepsilon & \xi_5\tau_{54} & \xi_5\tau_{55} \end{pmatrix},$$

$$P_3 = \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \xi_1\tau_{15} \\ \varepsilon & \varepsilon & \xi_2\tau_{23} & \varepsilon & \varepsilon \\ \varepsilon & \xi_3\tau_{32} & \varepsilon & \xi_3\tau_{34} & \varepsilon \\ \varepsilon & \varepsilon & \xi_4\tau_{43} & \varepsilon & \xi_4\tau_{45} \\ \xi_5\tau_{51} & \varepsilon & \varepsilon & \xi_5\tau_{54} & \xi_5\tau_{55} \end{pmatrix}.$$

Therefore, we can create evolution equations for this variable m case too. In fact, we can now regard this system as a *maxmin-*ω_i system, where ω_i is variable with each

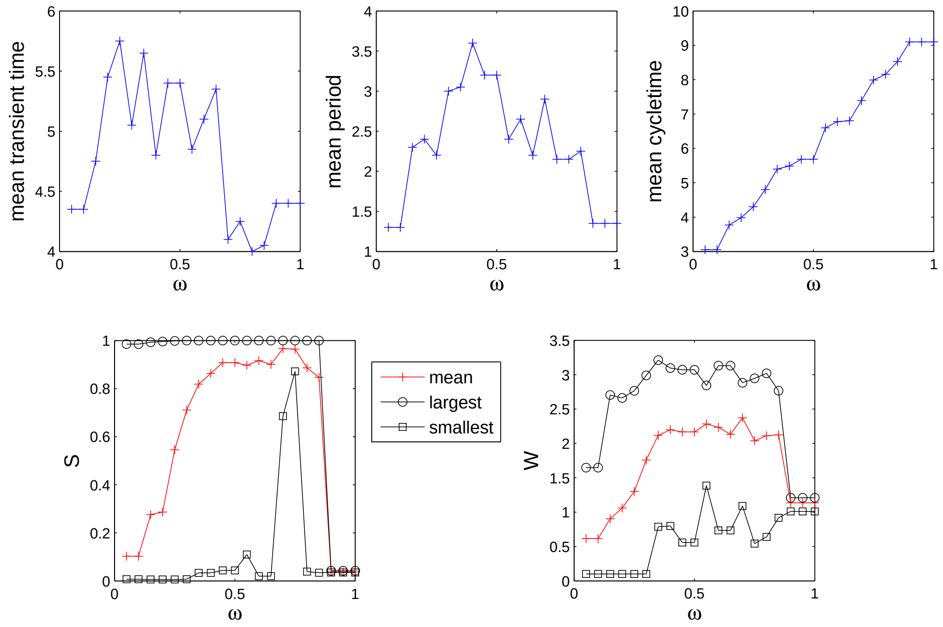


Figure 6.3: Asymptotic results of the maxmin-ω system as a function of ω. The network consists of 10 nodes, each possessing an arbitrary neighbourhood.

node. Thus, in general,

$$\omega_i = \frac{m_i}{n_i}.$$

We can fix any of the three variables in this equation to easily produce evolution equations for the corresponding system. The example in this section took n fixed, so $\omega = \frac{m_i}{n}$

In summary, we have shown that extension of the maxmin-m model to arbitrary networks is straightforward. If we are to apply the model to a real network of interest, then a possible difficulty in implementing would be due to having to know the neighbourhood of each node, a problem whose complexity scales with the number of nodes N. However, once this is established, the formation of evolution equations and identifying of asymptotic behaviour becomes methodical.

Even though the generalisations in this chapter are a big departure from regular n-nbhd networks and even from random n-nbhd networks, the asymptotic results obtained point towards a sense of universality, i.e. that identification of main features of the behaviour is predictable on any arbitrary network.

In particular, we have observed transient and cyclicity bells, as well as similar shapes for the graphs of Shannon entropy and word entropy as functions of ω. This suggests that maxmin-ω on many arbitrarily connected networks is most complex when $\omega \approx 1/2$, i.e. when the nodes update their states having received approximately half of their neighbourhood input.

Bibliography

[1] M. Andrecut. Mean field dynamics of random Boolean networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2005:P02003, 2005.

[2] M. Andrecut and M K Ali. Chaos in a simple boolean network. *International Journal of Modern Physics B*, 15(1):17–24, 2001.

[3] H. Bersini and V. Detours. Asynchrony induces stability in cellular automata based models. In *Artificial Life IV*, pages 382–387. MIT Press, MA, 1994.

[4] A. Bouillard and B. Gaujal. Coupling time of a (max, plus) matrix. *Rapports de recherche-INRIA*, 2000.

[5] Peter J. Cameron. *Combinatorics: topics, techniques, algorithms.* Cambridge University Press, Cambridge, 1994.

[6] M. Capcarrere. Evolution of asynchronous cellular automata. *Parallel Problem Solving from NaturePPSN VII*, pages 903–912, 2002.

[7] K.M. Chandy and L. Lamport. Distributed snapshots: Determining global states of distributed systems. *ACM Transactions on Computer Systems (TOCS)*, 3(1):63–75, 1985.

[8] D. Cornforth, D. Green, D. Newth, and M. Kirley. Ordered asynchronous processes in natural and artificial systems. In *Proceeding of the Fifth Australia-Japan Joint Workshop on Intelligent and Evolutionary Systems*, pages 105–112, 2001.

[9] S. Even and S. Rajsbaum. The use of a synchronizer yields maximum computation rate in distributed networks. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 95–105. ACM, 1990.

[10] N. Fatès and M. Morvan. An experimental study of robustness to asynchronism for elementary cellular automata. *Complex systems*, 16(1):1–28, 2005.

[11] N. Fatès, É. Thierry, M. Morvan, and N. Schabanel. Fully asynchronous behavior of double-quiescent elementary cellular automata. *Theoretical Computer Science*, 362(1-3):1–16, 2006.

[12] M. Gardner. Mathematical games: The fantastic combinations of John Conway's new solitaire game 'Life'. *Scientific American*, 223(4):120–123, 1970.

[13] S. Gaubert and J. Gunawardena. The duality theorem for min-max functions. *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics*, 326(1):43–48, 1998.

[14] C. Gershenson. Classification of random Boolean networks. *Artificial Life*, 8:1, 2003.

[15] D.G. Green, D. Newth, D. Cornforth, and M. Kirley. On evolutionary processes in natural and artificial systems. In *Proceeding of the Fifth Australia-Japan Joint Workshop on Intelligent and Evolutionary Systems*, pages 1–10, 2001.

[16] J. Gunawardena. Cycle times and fixed points of min-max functions. In *11th International Conference on Analysis and Optimization of Systems Discrete Event Systems*, pages 266–272. Springer, 1994.

[17] J. Gunawardena. Min-max functions. *Discrete Event Dynamic Systems*, 4(4):377–407, 1994.

[18] Y. Gunji. Pigment color patterns of molluscs as an autonomous process generated by asynchronous automata. *Biosystems*, 23(4):317–334, 1990.

[19] H. Gutowitz and C. Langton. Mean field theory of the edge of chaos. *Advances in Artificial Life*, pages 52–64, 1995.

[20] M. Hartmann and C. Arguelles. Transience bounds for long walks. *Mathematics of operations research*, pages 414–439, 1999.

[21] I. Harvey and T. Bossomaier. Time out of joint: Attractors in asynchronous random Boolean networks. In *Proceedings of the Fourth European Conference on Artificial Life*, pages 67–75. MIT Press, Cambridge, 1997.

[22] B. Heidergott, G.J. Olsder, and J.W. van der Woude. *Max Plus at work: modeling and analysis of synchronized systems: a course on Max-Plus algebra and its applications*, volume 13 of *Princeton Series in Applied Mathematics*. Princeton Univ Press, 2006.

[23] B.A. Huberman and N.S. Glance. Evolutionary games and computer simulations. *Proceedings of the national academy of sciences of the United States of America*, 90(16):7716–7718, 1993.

[24] T.E. Ingerson and R.L. Buvel. Structure in asynchronous cellular automata. *Physica D: Nonlinear Phenomena*, 10(1-2):59–68, 1984.

[25] Neil F. Johnson. *Two's company, three is complexity: a simple guide to the science of all sciences.* Oxford : Oneworld, 2007.

[26] Stuart A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22(3):437–467, March 1969.

[27] L. Lamport. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 21(7):558–565, 1978.

[28] C.G. Langton. Computation at the edge of chaos: Phase transitions and emergent computation. *Physica D: Nonlinear Phenomena*, 42(1-3):12–37, 1990.

[29] G. Le Caër. Comparison between simultaneous and sequential updating in $2^{n+1}-1$ cellular automata. *Physica A: Statistical Mechanics and its Applications*, 157(2):669–687, 1989.

[30] E.D. Lumer and G. Nicolis. Synchronous versus asynchronous dynamics in spatially distributed systems. *Physica D: Nonlinear Phenomena*, 71(4):440–452, 1994.

[31] C. Marr and M.T. Hütt. Topology regulates pattern formation capacity of binary cellular automata on graphs. *Physica A: Statistical Mechanics and its Applications*, 354:641–662, 2005.

[32] F. Mattern. Virtual time and global states of distributed systems. *Event London*, pages 215–226, 1989.

[33] G. Neiger and S. Toueg. Substituting for real time and common knowledge in asynchronous distributed systems. In *Proceedings of the sixth annual ACM Symposium on Principles of distributed computing*, pages 281–293. ACM, 1987.

[34] M.A. Nowak and R.M. May. Evolutionary games and spatial chaos. *Nature*, 359(6398):826–829, 1992.

[35] G.D. Ruxton and L.A. Saravia. The need for biological realism in the updating of cellular automata models. *Ecological Modelling*, 107(2-3):105–112, 1998.

[36] B. Schönfisch and A. de Roos. Synchronous and asynchronous updating in cellular automata. *BioSystems*, 51(3):123–143, 1999.

[37] M. Sipper, M. Tomassini, and M.S. Capcarrere. Evolving asynchronous and scalable non-uniform cellular automata. In *Proceedings of International Conference on Artificial Neural Networks and Genetic Algorithms (ICANNGA97)*, pages 67–71. Citeseer, 1997.

[38] G. Soto y Koelemeijer. *On the behaviour of classes of min-max-plus systems.* PhD thesis, Delft University of Technology, 2003.

[39] Steven H. Strogatz and Duncan J. Watts. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.

[40] Rachel Thomas. Games, life and the game of life, May 2002.

[41] John von Neumann. *Collected works. Vol. V: Design of computers, theory of automata and numerical analysis.* General editor: A. H. Taub. A Pergamon Press Book. The Macmillan Co., New York, 1963.

[42] Eric W Weisstein. "Landau's function". from MathWorld–a Wolfram web resource.

[43] Stephen Wolfram. Statistical mechanics of cellular automata. *Rev. Modern Phys.*, 55(3):601–644, 1983.

[44] Stephen Wolfram. Universality and complexity in cellular automata. *Phys. D*, 10(1-2):1–35, 1984. Cellular automata (Los Alamos, N.M., 1983).