30

Classification of States

In a Markov chain, each state can be placed in one of the three classifications.[1] Since each state falls into one and only one category, these categories partition the states. The secret of categorizing the states is to find the *communicating classes*. The states of a Markov chain can be partitioned into these communicating classes. Two states *communicate* if and only if it is possible to go from each to the other. That is, states A and B communicate if and only if it is possible to go from A to B **and** from B to A.

> **To review** from Section 2: A collection of objects is partitioned if each object falls into one and only one partition class. This is equivalent to an *equivalence relation*. Two objects are equivalent if they fall into the same partition. An equivalence relation has three properties:
>
> - For any object, say A, the relation (A,A) exists. (An object is equivalent to itself).
>   If (A,B) exists, then so does (B,A). (If A is equivalent to B, B is equivalent to A.)
>   If (A,B) and (B,C) exists, so does (A,C). (If A is equivalent to B and B is equivalent to C, then A is equivalent to C.)

We need to see that the relation of communicating is an equivalence relation. First, we arbitrarily (but reasonably) define an object as belonging to the same communicating class as itself. Second, the definition is explicitly symmetric, A and B communicate *if it is possible to go from each to the other.* Lastly, if A communicates with B, and B communicates with C, we have that we can go from A to C, because we can go from A to B and then from B to C.

[1]Again, we are talking only about finite chains. Also, this is one of those unfortunate areas, where there is not standard terminology. So if you go to another source, be careful!

Similarly, we can go from C to B, and then from B to A; hence we can go from C to A. That shows that A and C communicate. Hence, the relation of communicating partitions vertices into communicating classes.

A and B belong to the same communication class if and only if they communicate: that is if it is possible to go from A to B and from B to A.

There are three classifications of states: **transient, ergodic,[1] and periodic.** If two vertices belong to the same communicating class they have the same classification. (The proof of this assertion is most difficult for the periodic case. In the appendix to this chapter it will be shown that if a communicating class has a state of period k, then the communicating class is itself periodic of period k.) To classify a communicating class, find the easiest to classify vertex of that communicating class and classify that vertex. The classifications are as follows:

- Transient: A state is transient if it is possible to leave the state and never return.
- Periodic: A state is periodic if it is not transient, and if that state is returned to only on multiples of some positive integer greater than 1. This integer is known as the *period* of the state.
- Ergodic: A state is ergodic if it is neither transient nor periodic.

A Few Tips

Always find the communicating classes before classifying the states. Since we deal only with finite Markov chains, it can be seen that not all of the states can be transient. An absorbing state (that is a state that returns to itself **immediately** with probability 1) is a special kind of ergodic state. Identify states in the order of the definition: transient states first; periodic states second; and what is left are ergodic states. A state that has a loop cannot be periodic. Lastly, ignore the probabilities: they are irrelevant to the classification. Lastly if a communicating class

[1] Over the years I've noticed that on tests at least half of the students spell ergodic with a "t" like erotic—undoubtedly the sad consequence of too many lit courses.

is periodic, each state has the same period. (This is proven in the appendix to this section.) In a periodic communicating class, find a state for which it is easiest to deduce the period. Then the entire class has that same period.

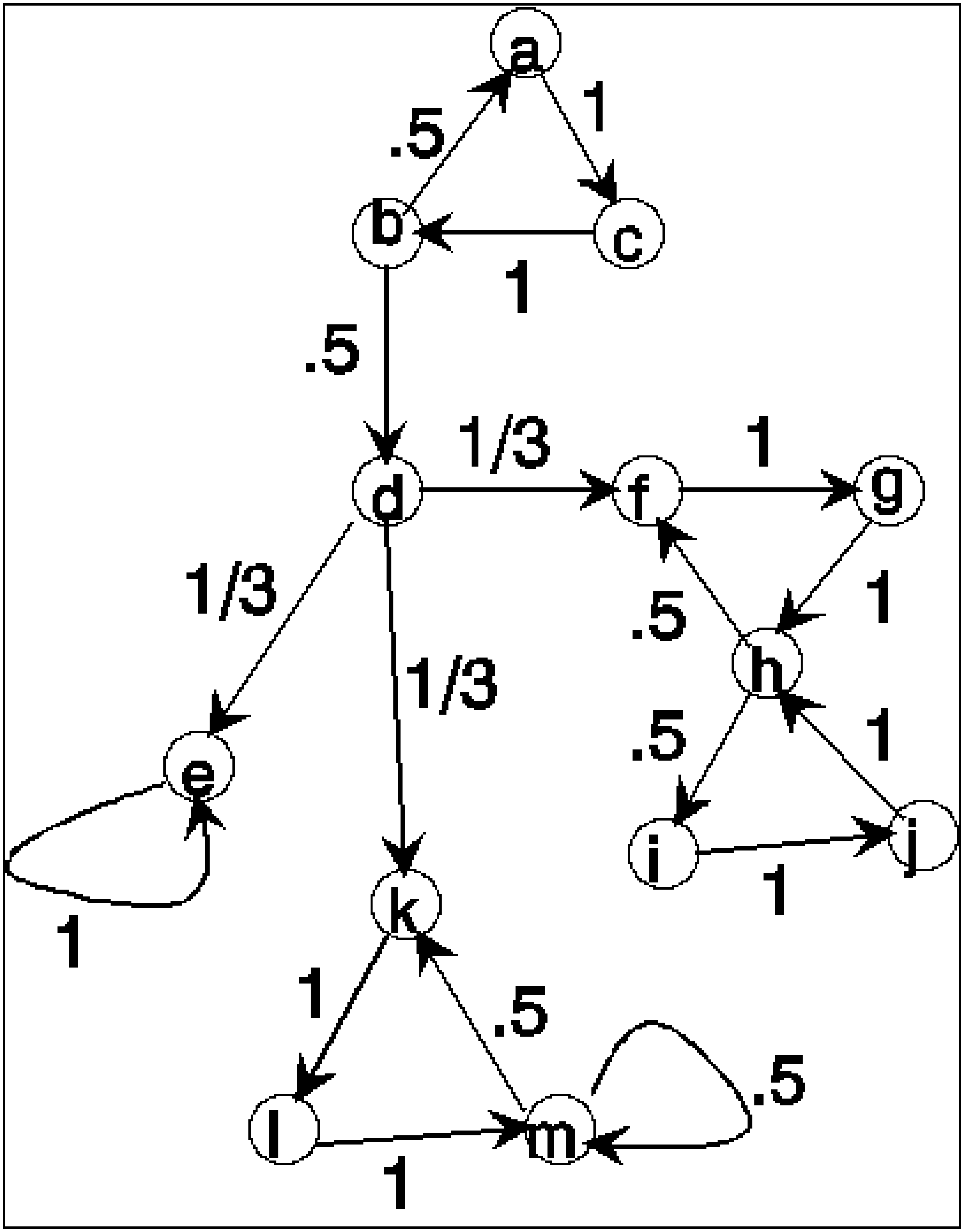

Figure 1 A Graph with Transient, Periodic, and Ergodic States

Example Let us analyze the Markov chain of **Figure 1**. It consists of five communicating classes which we can classify as follows:

States a, b, c are a communicating class. Once the chain goes from a to d, it cannot return to a. Hence, states a, b, c are transient. State d is a communicating class by itself. Once we arrive in d we leave never to return. Hence it is transient. State e is a communicating class and is not transient. Once we arrive there we keep coming back and we go nowhere else. It is absorbing which is a special case of ergodic. States f, g, h, i, j form a communicating class. Once we are in their part of the chain, we cannot leave, hence they are not transient. If we look at state h, we see that once we leave it we will always return in 3 transitions. Therefore, h, and hence the whole class, f-g-h-i-j, are periodic and of period 3. States k, l, m form a communicating class. Once we arrive there we cannot leave, therefore they are not transient. Since state m has a loop, it and its whole class cannot be periodic. Since they are neither transient nor periodic they must be ergodic.

☐ **Exercise 1** Classify the states of **Figure 1**.

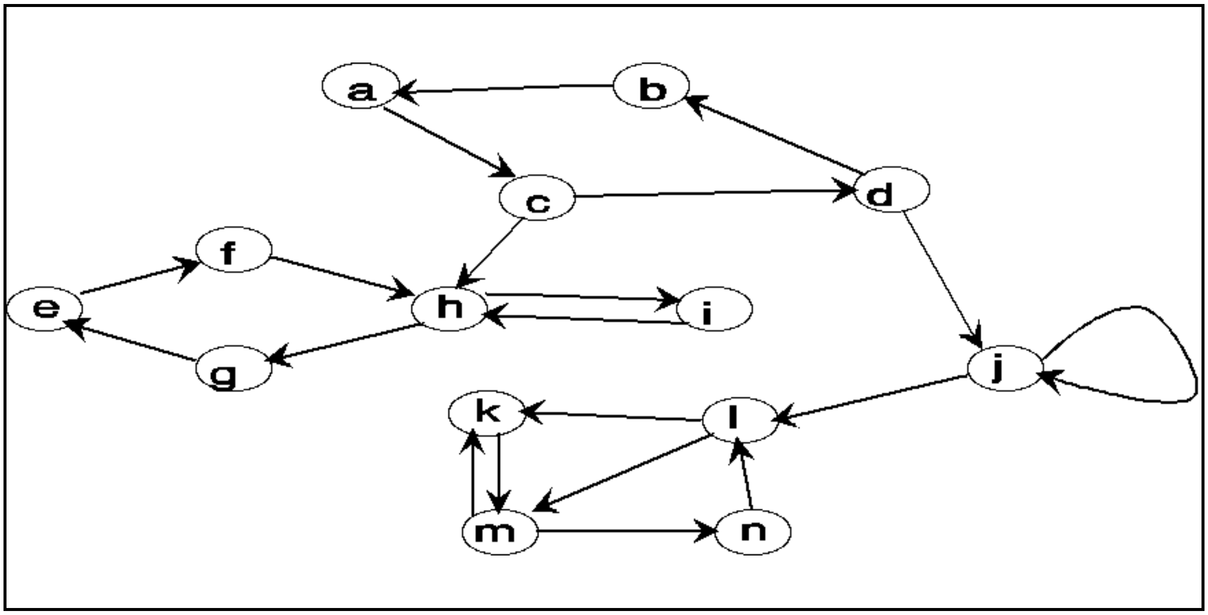

Figure 2 If You Know What is Good for You, You Will Classify These States

Important Considerations On Periodic States

Skip This Section at Extreme Peril

In general in probability, practitioners are only interested in periodic states and graphs in order to avoid them (a graph is periodic if every state is periodic). The reason is quite simple and quite important. Probabilists are interested in long term behavior. In the long term we eventually leave all of the transient states never to return. As time goes to infinity, the proportion of time spent in transient states is 0.[1] Similarly, we will later analyze the long term behavior of ergodic chains. In their case it turns out that we spend a certain average amount of time in each state, regardless of where we started. Typically, if we are in an ergodic chain (or if we have descended to an ergodic sub-chain) we may determine that we will spend proportion, p, of our time in state X. If we are looking far enough ahead, say L transitions, then the

[1]All comments here are, as always, about finite chains.

probability of being in X after L transitions is almost precisely p. Similarly, the probability of being there after L + 1 transitions is also p.

The behavior of periodic states is quite different than transient and ergodic states. Suppose that we are in a chain of period k (or suppose that we have descended to a sub-chain that is periodic and of period k). Then on leaving any state, we will return only on multiples of k. Therefore the probability of being in state X after a million transitions is very dependent on where we started. Consider the Markov chain in **Figure 3**.

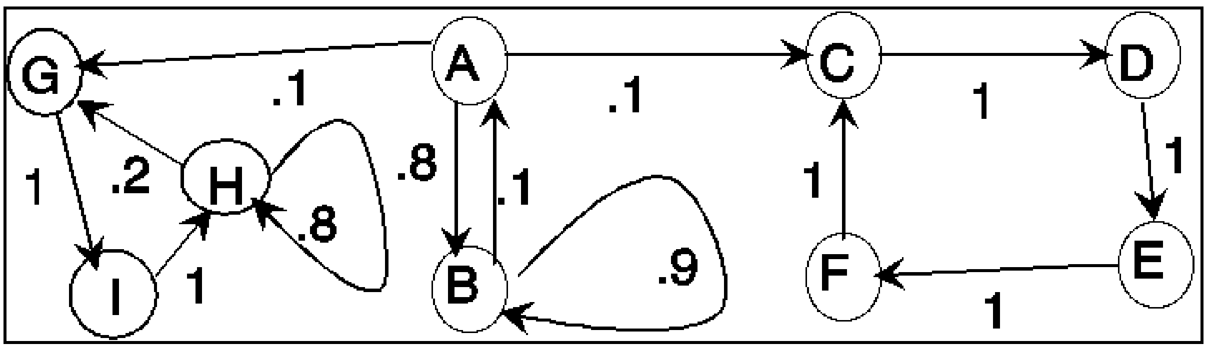

Figure 3 A Mixed Markov Chain

In this chain, states A and B are transient. If we start in them we will wind up in either G-H-I which is ergodic or we will end in C-D-E-F which is periodic. If we start in either the periodic or ergodic sub-chains, we are stuck there. The chain C-D-E-F is periodic of period 4. If we are in state C in transition 10, then in another 1,000,000 transitions the probability of our being back in state C is 1. In another 1,000,001 transitions the probability of being in state C is 0. Our long term probabilities are sensitive to where we are when we start. On the other hand, if we fall into the ergodic sub-chain G-H-I, then in the long run we will spend 1/7'th of our time in state G and 1/7'th in I and 5/7'th in H.[1] If we are in state H in transition 10, the probability of being there 1,000,000 transitions later is nearly precisely 5/7. The probability of being there in 1,000,001 transitions is also nearly exactly 5/7. Once we are in the ergodic sub-chain, our long term probabilities are independent of where we are when we start.

[1]I will show you how to calculate these probabilities in a later section.

Appendix: Analysis of Periodic Chains

Periodic Graphs

A Markov chain is a connected probability graph. However, the qualities of a Markov chain that make it a periodic chain are qualitative properties not involving the probabilities at all. In fact, in nearly every presentation I have seen of periodic Markov chains (we will refer to periodic Markov chains as periodic graphs) they are given a very short, superficial discussion; the idea being that it is only necessary to recognize and then abandon such chains. We will define a periodic graph as having the following properties:

I. The graph is strongly connected. (That is, there is a *directed* path from any vertex to any other.)

II. Each vertex in a periodic graph is periodic in the following sense: Given a vertex V, let us count the number of arcs required to travel from V to itself always keeping with the direction of the arcs. Such a path is a *directed circuit*. Let C be the collection of lengths of directed circuits containing V. Let P be the GCD of all numbers within C. P is called the *period* of the vertex V and is greater than one.[1]

In a periodic graph, whenever you leave a node, you can eventually return to that node (property 1) but you will only return on some multiples of an integer greater than 1 which is known as the *period* of the vertex (property 2). Property 2 implies immediately that a periodic graph has no loops (an arc from a vertex to itself). It is not too hard to show using properties 1 and 2 that the period of each vertex is the same, and therefore we can speak of the period of the graph which must be greater than one.

---

[1]One could argue that the collection of circuit lengths is infinite. However, if we can restrict ourselves to non-redundant circuits (that do not repeat arcs) then this collection is finite.

Theorem: In a periodic graph, every vertex has the same period.

Proof: Let G be a periodic graph, and let U and V be vertices of periods u and v respectively. We will show that $u|v$. It then follows that the same argument can be reversed to show that $v|u$; hence that u = v. Since G is strongly connected (a periodic graph must be strongly connected; see above) there is some (directed) path, P_1, from U to V, and also some path P_2, from V to U. The combined path P_1P_2 is a circuit from U to itself and must be divisible by u. (Actually a path is not visible by a number; its length is divisible by the number. However, it is convenient to abuse the language and speak of the path being divisible by a number.) Now every circuit from V to itself is divisible by v. Pick any such circuit, C. If we consider the path P_1CP_2 (P_1 followed by C and then C followed by P_2) it is a circuit from U to itself. Hence P_1CP_2 must be divisible by u. Since P_1P_2 is divisible by u, so is C. Since C is an arbitrary circuit from V to itself, u must divide all such circuits. However, we know by definition of the period of a vertex that v is the greatest possible divisor of all such circuits. We showed much earlier that a common divisor of a set of numbers must divide the greatest common divisor. Hence $u|v$.

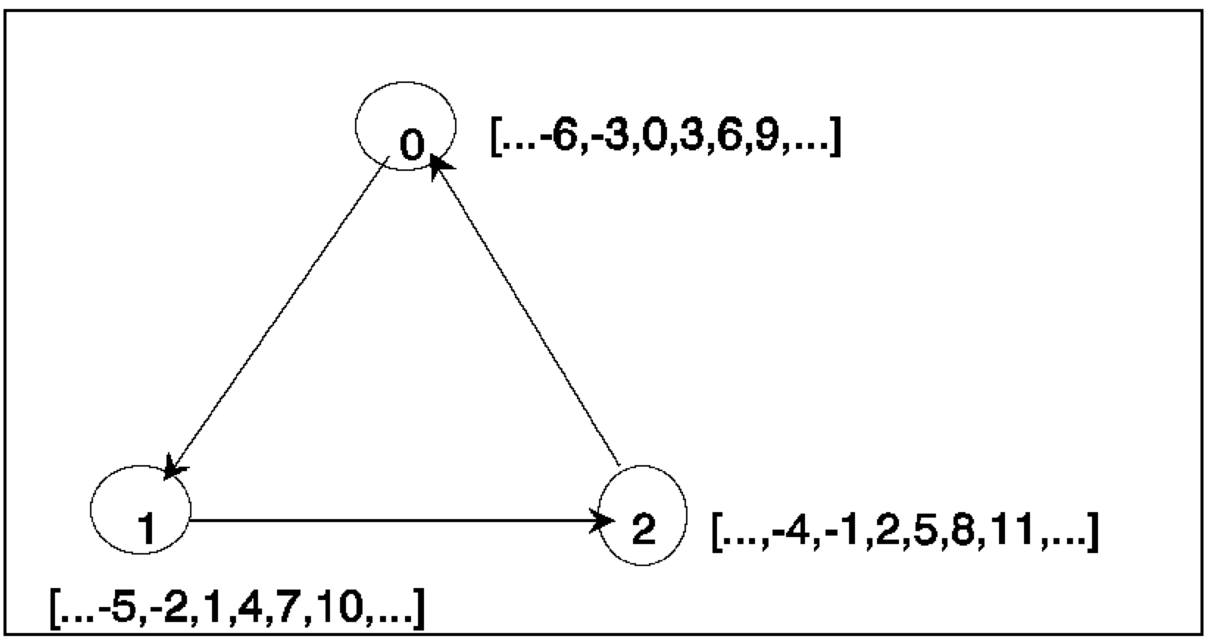

Figure 4 The Graph of the Integers Modulo 3 is of Period 3

Figure 4 is an example of the most elementary type of periodic graph and has period 3. **Figure 5** gives a graph that is of period 2. To see that it is periodic we note that there is a directed path from any vertex to any other, and that the directed circuits from the right most node all have lengths that are multiples of 2. Since 2 is one of the circuit lengths, 2 must be the GCD of the circuit lengths, and the graph is of period 2.

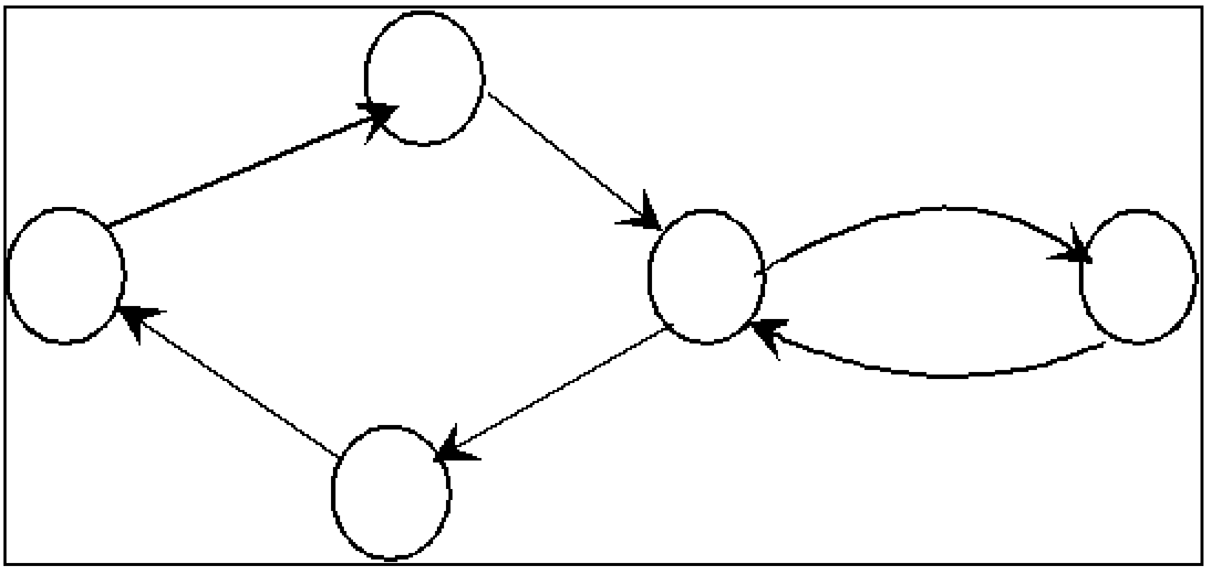

Figure 5 A Graph of Period Two.

The graph in **Figure 6** is not periodic since it contains circuits of length 3 and 4, but GCD(3,4) = 1. Let us call a circuit *fundamental* if it does not have a sub-circuit containing fewer nodes.[1] It can be shown that a graph is periodic of period n, if n = GCD(a,b,...,m) where a,b,... ,m are the lengths of the directed fundamental circuits and n is greater than 1. Note that it is not trivial to write an algorithm to recognize all fundamental circuits in a graph.

[1]A sub-circuit is a proper sub-graph that is itself a circuit. (By *proper* sub-graph we mean that it is not the original graph.)

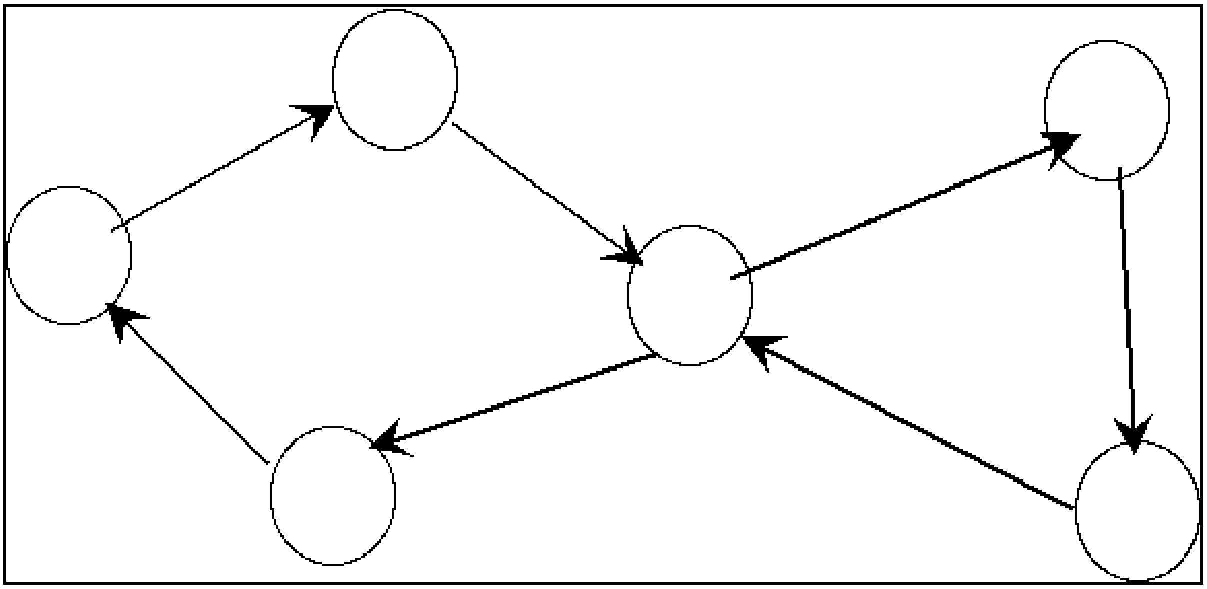

Figure 6 A Non-periodic Graph.

1. Transient: a, b, c, d, j
   Periodic: e, f, g, h, i (period 2)
   Ergodic: k, l, m, n