Markov Chains And Cheetah Chase Algorithm

M. Goudhaman[1], S. Sasikumar[2], N.Vanathi[3]
[1]*Faculty of CSE, Jeppiaar Engineering College*
[2]*Faculty of CSE, Saveetha Engineering College*
[3]*Faculty of Mathematics, KCG College of Technology*

Abstract. In this paper we discuss the notion of a Markov chain and some of its basic properties. We apply Markov chains to the analysis of Cheetah Chase Algorithm. In particular, we look at the Max-Min Cheetah System for finding shortest path and use the mixing times of Markov chains to provide upper-bounds on running time estimation.
Keyword: Markov Chains, Cheetah Chase Algorithm, Shortest Path Optimization.

I. INTRODUCTION TO MARKOV CHAINS

Markov chains are a large class of stochastic processes characterized by the Markov property which simply means that the next state that the process takes depends only on the current state and none of the previous states. Formally, we can express this description for processes in discrete time in the following way:

Definition 1.1: A stochastic process is said to have the Markov property if:
(1.2) $P\{X_{t+1} = x \mid X_0 = x_0, \ldots X_t = x_t\} = P\{X_{t+1} = x \mid X_t = x_t\}$

Such processes are called Markov Chains. [2]
This definition implicitly restricts Ω to be countable because in the uncountable setting, the probability measure of a single point is necessarily 0. Markov chains over uncountable state-spaces require a more nuanced definition.
We often make the assumption that the transition probabilities do not change over times. If this is the case, then we can describe the conditional probability that $X_{t+1} = x$ given that $X_t = y$ as a function strictly in x and y. This gives rise to the following definition:

Definition 1.3: A Markov chain is said to be time-homogeneous if for all $t > 0$:
$P\{X_{t+1} = x \mid X_t = y\} = p(y, x)$ for some $p: \Omega \times \Omega \to [0, 1]$.
We can interpret p as the probability of jumping from state y to state x in a single time-step.

Definition 1.4: For a Markov chain $\{X_t\}$ with state space Ω, the transition matrix is an $|\Omega|\times|\Omega|$ matrix P (possibly infinite) such that each row index and each column index corresponds to a state and the entries are given by:
$P_{x,y} = p(x, y)$,
where p(x, y) is the transition probability of jumping from state x to state y.
We note that $0 \leq P_{x,y} \leq 1$ since they are probabilities and that $\sum_{y \in \Omega} P_{x,y} = 1$ since some state in Ω with probability 1 and each state is a disjoint event

Definition 1.5: We define the t-step probability denoted $p^t(x,y)$ to be the probability that $X_t = y$ given that $X_0 = x$.

Lemma 1.6: For a Markov chain $\{X_t\}$ with state space Ω and transition matrix P, the (x; y) entry of the matrix P^t is precisely $p^t(x,y)$.
Proof. For t = 1, the statement is true by Definition. So assume that the lemma holds for P^t, then consider:
$P^{t+1}(x, y) = \sum_{w \in \Omega} P^t(x,w)p(w,y)$
$= \sum_{w \in \Omega} P^t_{x,w} P_{w,y}$
$= P^{t+1}_{x,y}$

The first equality follows from the total law of probability and the second equality follows from the inductive hypothesis.
From this point forward, we will use $P^t(x, .)$ to denote the distribution on Ω of the tth step of a chain starting at x.

II. ERGODICITY

So far we have a very broad Definition of Markov chain, but we are interested in a certain class of Markov chains with nice properties. Therefore, we give the following definitions:[2]
Definition 2.1: A state x € Ω is said to communicate with a state y € Ω if and only if there exist s, t > 0 such that:

$P\{X_t = y \mid X_0 = x\} > 0$

$P\{X_s = x \mid X_0 = y\} > 0$:
We denote this as $x \leftrightarrow y$.
Definition 2.2: A communication class for a Markov chain is set of states $W \subseteq \Omega$ such that:
$x \leftrightarrow y,\ \forall x, y \in W$
$w \leftrightarrow z,\ \forall w \in W, \forall z \in \Omega \setminus W$.
Definition 2.3: A Markov chain is irreducible if the sample space Ω is a communication class. It is easy to check that $\leftrightarrow$ forms a well de ned equivalence relation (and thus a partition) on Ω. Knowing this fact, we can interpret an irreducible Markov chain as one which only has the simple partition.

Definition 2.4: The period of a state $x \in \Omega$ is defined to be:
$L(x) = \gcd\{ t \mid P\{ X_t = x \mid X_0 = x \} > 0 \}$.
We say x is aperiodic if and only if $L(x) = 1$.

Definition 2.5: A Markov chain $\{X_t\}$ is said to be aperiodic if and only if every state $x \in \Omega$ is aperiodic.

Lemma 2.6: Let $\{X_t\}$ be a Markov chain over state space Ω, then for any communication class $W \subseteq \Omega$,
$L(x) = L(y); \forall x, y \in W$
Proof. Since x, y are in the same communication class, $x \leftrightarrow y$ and we have: $\exists r; s$ such that $P^r(x,y) > 0$, $P^s(y, x) > 0$
Which implies, $P^{r+s}(x, x) \geq P^r(x,y)P^s(y,x) > 0$.
Therefore, $L(x) \mid r + s$.:
For any $t > 0$ such that $P^t(y,y) > 0$, we have that:
$P^{r+s+t}(x, x) \geq P^r(x,y)P^t(y,y)P^s(y,x) > 0$.
Which means that,
$L(x) \mid r + s + t \Rightarrow L(x) \mid t \Rightarrow L(y) \mid L(x)$.
Then, by a symmetric argument, $L(x) \mid L(y)$. Thus, $L(x) = L(y)$.
In particular, Lemma 2.6 guarantees that if we find that one state in an irreducible Markov chain is aperiodic, the whole chain must be.

Definition 2.7: A state $x \in \Omega$ is said to be recurrent if and only if:
$P\{ X_t = x$ for infinitely many $t > 0\} = 1$.
A Markov chain is recurrent if every state is recurrent.

Definition 2.8: A state $x \in \Omega$ is said to be transient if and only if it is not recurrent. A Markov chain is transient if every state is transient.

Recurrence / transience simply describe whether or not it is possible to leave a state and never return. However, the above Definition is often di cult to directly demonstrate which leads us to an alternative characterization using the notion of return times.

Definition 2.9: The return time of a state x is the random variable T_x which is characterized by:
$T_x = \inf\{ t \mid X_t = x, X_0 = x\}.t$

Theorem 2.10: Let $\{X_t\}$ be a Markov chain with transition matrix P over a state space Ω. Then for any state $x \in \Omega$, the following are equivalent:
i.: x is recurrent.
ii.: $P\{T_x < \infty 1\} = 1$.
iii: $\sum_{t=0}^{\infty} P^t(x,x) = \infty$.
Proof. (i. $\Leftrightarrow$ ii.)
Suppose that x is recurrent. By Definition, the chain must hit state x infinitely often. In particular, the chain must hit the state x at least twice. This guarantees the existence of non-negative integers $s < t$ such that:
$X_s = x$ and $X_t = x$:

By the Markov property and our assumption that the chain is time-homogeneous,
P{ Xt = x; Xs = x } = P { Xt-s = x; X0 = x }.
Which holds for all s < t. Therefore,
P{Tx < ∞} = P {t-s < ∞} = 1.
Conversely, suppose that Tx is always finite. Then, suppose by contradiction that x is a transient state.
$\exists$ t > 0 such that Xt = x and $\forall$ s > t; Xs $\neq$ x:
Then by the Markov Property,
P{ Xt = x; Xs $\neq$ x $\forall$ s < t } = P {X0 = x; Xn $\neq$ x $\forall$ n > 0 }.
However, if this occurs, it must be true that:
Tx = inf { t | Xt = x; X0 = x } = ∞.t
But this contradicts our assumption that the return time is finite.
(ii. $\Leftrightarrow$ iii.)
We define the number of visits to x for a chain starting at x to be:
Vx = $\sum_{t=0}^{\infty}$ 1{Xt = x, x0 = x} .
Then we consider:
$\sum_{t=0}^{\infty}$ Pt (x, x) = $\sum_{t=0}^{\infty}$ P { Xt = x, X0= x}
= $\sum_{t=0}^{\infty}$ E1 { Xt = x, X0= x}
= E $\sum_{t=0}^{\infty}$ 1 { Xt = x, X0= x}
= E Vx

But Vx is infinite if and only if Tx is finite.
The next lemma demonstrates that recurrence is communication class property:

Lemma 2.11: Suppose {Xt} is a Markov chain with state space Ω, suppose W $\subseteq$ Ω is a communication class. Then given any state x in W:
x € W is transient $\Leftrightarrow$ y € W is transient $\forall$y € W
Proof. Suppose that x € W is transient and fix y € Ω. Since W is a communication class, there exist positive integers s, t such that:
Ps(x, y) > 0 and Pt(y, x) > 0
Then we observe that for all r $\geq$ 0:
Ps+r+t(x, x) $\geq$ Ps(x, y)Pr(y, y)Pt(y, x)
This is true because the left side of the inequality is the probability of the event A = {X0 = x; Xs+r+t = x} and the right side is probability of the event B = {X0 = x, Xs = y, Xs+r = y, Xs+r+t = x}. Then it is clear that B $\subseteq$ A and thus we get the above inequality.
Summing over every r > 0, we get that:
$\sum_{r=0}^{\infty}$ Ps+r+t (x, x) $\geq$ $\sum_{r=0}^{\infty}$ Ps (x, y) Pr (y, y) Pt (y, x)
$\sum_{r=0}^{\infty}$ Pr (y, y) $\leq$ $\frac{1}{Ps\,(x,y)\,Pt\,(y,x)}$ $\sum_{r=0}^{\infty}$ Ps+r+t (x, x)
$\leq$ $\frac{1}{Ps\,(x,y)\,Pt\,(y,x)}$ $\sum_{k=0}^{\infty}$ Pk (x, x) < ∞

The last inequality holds because we assume x to be transient and therefore the sum must be finite by Theorem 2.10. Then, y must be transient as well by Theorem 2.10.
It immediately follows that any state x is recurrent if and only if every other state in the communication class of x is recurrent. In particular, we notice that if a Markov chain is irreducible, then the whole system must either be transient or recurrent.
When dealing with Markov chains with a countable infinite state-space, we must make one further distinction within recurrent states.
Definition 2.12: Let {Xt} be a Markov chain with transition matrix P over a countable infinite state space Ω. A state x € Ω is said to be positive recurrent if and only if it is both recurrent and ETx < ∞.
A state x € Ω is said to be null recurrent if and only if it is both recurrent and ETx = ∞.

Lemma 2.13. Let {Xt} be a Markov chain with transition matrix P over state space Ω. For a communication class W $\subset$ Ω, the following holds:

x € W is positive recurrent ➔ y € W is positive recurrent, $\forall$y € W
Proof. We first define for any x, y € Ω,
Tx,y = inf { t | Xt = y, X0 = x }.t
Let Ty denote the return time to y, then we observe:
Ty $\leq$ Ty,x + Tx,y

Define the event A to be the event that the chain starting at x hits y before returning to x. Then we can split the expected return time of x in the following way:
ETx = E(Tx | A) P{A} + E(Tx | Ac) (1- P{A}).
But if the event A occurs, then the return time for x can be split into the time it takes to reach y before going back, ie:
E(Tx|A) = E(Tx,y + Ty,x).
We know that P{A} and ETx are both finiite and positive values, which gives that:
ETy $\leq$ E(Ty,x + Tx,y) $\leq \frac{ETx}{P\{A\}} < \infty$
So y must be positive recurrent.
The sort of Markov chains we are interested in are precisely the ones which are irreducible, aperiodic and positive recurrent. Markov chains which satisfy these properties have a special name. Therefore, we provide the following Definition.
Definition 2.14: A Markov chain {Xt} is said to be ergodic if it is irreducible, aperiodic, and positive recurrent.

III. CHEETAH CHASE ALGORITHM (CCA)

The Cheetah is a giant and energetic civet that was once found all through Asia, Africa and certain places of Europe. Cheetahs are one of Africa's most energetic predators and are most famous for their monstrous speed when in a chase. Equipped for achieving speeds of more than 60mph for minimum span of time, Cheetah is the speediest land vertebrate on the earth [12]. The Cheetah is one of a kind among Africa's civets principally on the grounds that they are most dynamic amid the day, which keeps away from rivalry for nourishment from other substantial predators like Lions and Hyenas that chase amid the cooler night. The Cheetah has outstanding visual perception thus chases utilizing sight by first stalking its prey from between 10 to 30 meters away, and after that pursuing it when the time is correct. The light and thin body of the cheetah influences it to appropriate to short, dangerous blasts of speed, hasty acceleration, and a capability to execute extraordinary alters in course while moving at speed. These behaviours represent unique features of the cheetah's capability to capture fast moving prey. [6]
Elliot et al., (1977) gave an applied model to prey securing by earthly carnivores depicting four noteworthy components; look, stalk, assault, and stifle. Of these, the assault is the most power-requesting (Williams et al., 2014), ordinarily including complex fast moves, supported by obviously entangled behavioural alternatives for the both predators and prey. [7]
Goudhaman et al., (2018) [4]gave an optimal algorithm aiming to find better solution for the SPP between the start and end positions with dynamic allocation of nodes between the start and destination nodes. The properties of the present model empower us to sort out a meta-heuristic in accordance with Cheetah Chase Algorithm to illuminate the most limited way outline. The Cheetah Chase Algorithm is developed by the process of Cheetah hunting and chasing of Cheetah to capture its prey with the parameters of high speed, velocity and greater accelerations. The pseudo code for the cheetah chase algorithm is given below. [4]
Algorithms of this sort are attractive for several reasons. First of all, though the algorithms have yet to be understood fully at a theoretical level, there has been empirical evidence of their efficacy in a variety of applications. Furthermore, actual cheetahs are capable of navigating complex environments despite the fact that individual cheetahs have very limited cognitive abilities. The goal of CCA is to replicate the emergent problem solving abilities of a collective group of simple agents. Lastly, the randomized nature of CCA guarantees that the search for an optimum will never strictly converge to a local optimum that is not globally optimal.

IV. MMCS ALGORITHM

We will work with a version of cheetah colony optimization known as the iteration-best MAX-MIN Cheetah System (MMCS) applied to a shortest path problem with a single destination [7]. The precise statement of the problem is the following:[2]

Suppose that we have a directed acyclic graph G(V, E). We will call a connected sequence of vertices and edges from a start vertex u0 to a target vertex v0 a path. For any vertex u € V, we will denote deg(u) to be the number of edges pointing away from u. We will denote Δ as the maximum degree over all the vertices and diam(G) will be the maximum number of edges in any path. Denote the set of paths from a vertex u to v0 by Su. Furthermore, we define:

$S = \bigcup_{u \in V} Su$.

The distance traversed over a given path l = (u0, … , v0) is given by some function f : S → R. The goal is to find a solution x* € Su0 such that:

f(x*) = min { f(x) | x € Su0 }.

Although f maps from the set of all possible paths starting at all vertices, the actual search space is only the set of paths starting at u0.

To make things less cluttered, we will assume that the target v0 can be reached from any vertex u.

Fix Tmin € (0, 1). Then define an initial function:

$\emptyset 0$: E → [Tmin, 1 - Tmin]

This function gives a vector of length |E|:

($\emptyset$0 (e))e€E

We will refer to this vector as the chase vector. As the algorithm runs, the chase vector will be updated at every iteration by way of updating the function

$\emptyset t$: E → [Tmin, 1 - Tmin] for every time step t

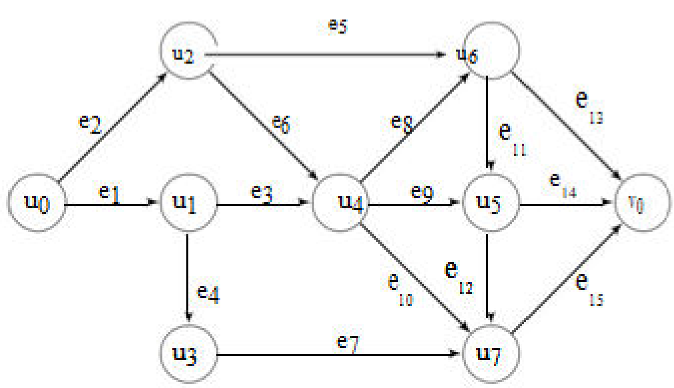
Then at each iteration, λ independent random walks are initiated from every vertex u € V towards the destination v0. If a random chaser is currently at vertex u, then there is a finite set of possible outgoing edges from u, say {e1, …, eK}.

Then the next step that the chaser takes will be chosen from these edges with the following probabilities:

P {chaser chooses ej | chaser is at u } = $\frac{\emptyset(ej)}{\sum_{i=1}^{K} \emptyset(ei)}$

If the chaser chooses an edge ej, then the chaser moves to the corresponding vertex. This process continues until the chaser reaches v0. At the end of the walk, the path that was traversed is stored. We will refer to these random chasers as cheetahs.

Example 4.1. Consider the following graph:



Here u0 is the start vertex and v0 is the target. Suppose we pick the vector $\emptyset$0 such that $\emptyset$0(ei) = 1 - Tmin for all i = 1, ...,15. If we start an cheetah at u0, then the first step is either along edge e1 to the vertex u1 or along e2 to the vertex u2. Based on our choice of $\emptyset$, the probability of an cheetah moving to u1 is given by:

P {cheetah picks e1 | cheetah is at u0 } = $\frac{\emptyset 0(e1)}{\emptyset 0(e1) + \emptyset 0(e2)} = \frac{1}{2}$

If a cheetah is at u4, then the probability of choosing u5 is $\frac{1}{3}$ because the appropriate edge to reach u5 is e9. The conditional probability of choosing e9 while at u4 is computed by:

P {cheetah picks e9 | cheetah is at u4 } = $\frac{\emptyset 0(e9)}{\emptyset 0(e8) + \emptyset 0(e9) + \emptyset 0(e10)} = \frac{1}{3}$

After all the cheetahs starting at any vertex u complete their chase, the paths are evaluated according to the function f and the best path is stored. In precise terms, we can denote the path of the jth cheetah starting from the vertex u as $l_u^{(j)}$ and we store:

l_u^{*} = argmin $f l_u^{(j)}$
$l_u^{(j)}$

This process is done for every vertex u € V. For cheetahs starting at the intended start vertex u0, the best path of the iteration is stored separately and compared against the best path of all previous iterations with respect to f.

The next step is the most crucial aspect of the algorithm: the chase update. Once we have l_u^{*} for each u, we can define the set of first edges of all the best paths in the iteration:

B = {e | e is the first edge of l_u^{*} for some u € V}.

Then, we update each entry of the chase vector in the following way:

$\emptyset_{t+1}$ (e) = { $\frac{\min\{(1-p)\&\emptyset t(e)+p, 1-T\min\} \text{ if } e \in B}{\min\{(1-p)\&\emptyset t(e), T\min\} \text{ otherwise}}$

In the expression above, ρ € (0,1) is a chosen parameter we will call the path fixing factor. The update rule first fixes the path by reducing the chase values for all the edges to simulate the actual path fixing of biological chases. Then, the first edges of the best paths are reinforced to simulate the process of actual cheetahs leaving actual chases as they travel. Notice that the value of the chase vector on a particular edge can only be affected by the cheetahs leaving from the vertex where the edge originates.

Example 4.2. Consider the graph from 4.1. Assume that $\emptyset$0 is initiated as before and ρ € (0,1). Suppose that the best path from u0 is $l_u^{(*)}$ = (u0, e2, u2, e7, u6, e13, v0). Then in the next iteration, the chase value on the edge e2 will be given by:

$\emptyset 1(e2)$ = (1- ρ) (1- Tmin) + ρ.

On the other hand, the chase value on the edge e1 will be given by:

$\emptyset 1(e1)$ = (1- ρ) (1- Tmin)

After the chasing path update is completed, the algorithm moves to the next iteration and the cheetahs make their random chase with respect to the newly updated chase vector. Then the iterations continue until some stopping condition is met. The easiest example is simply fixing a number of iterations T .

Table 5.1 Mmas Algorithm

Formally, the algorithm is given by:
for iter from 1 to T do
    for u in V do
        for j from 1 to λ do
            i = 0, $p_0^{(j)} \leftarrow$ u, $V_1^{(j)} \leftarrow$ { p € V | ($p_0^{(j)}$, p) € E }
            While $p_i^{(j)} \neq$ v0 and $V_{i+1}^{(j)} \neq$ 0 do
              i $\leftarrow$ i + 1
              Choose a vertex p(ij) € Vi(j) according to the probabilities:
              $\frac{\emptyset\left(\left(p_{t-1}^{(j)}, p_t^{(j)}\right)\right)}{\sum_{p \in Vi(j)} \emptyset(p_{t-1}^{(j)}, p)}$
              $V_{i+1}^{(j)} \leftarrow$ { p € V | ($p_0^{(j)}$ … ($p_i^{(j)}$) ($p_i^{(j)}$, ($p^{(j)}$) € E }
            end

```
current_path ← (p_0^(j) … (p_i^(j) ) ;
if f(best_path) < f(stored_path) then,
        stored_path ← best_path
end
  end
  if u = = u0  then
if f(current_path) < f(best_path) then,      best_path ← current_path
      end
    end
  end
  update chase path
end
```

The variable stored path simply stores the best-so-far solution over all of the iterations from the desired start point u0 to the destination v0.

V. CONVERGENCE OF MMCS

Since the algorithm puts a lower bound on the amount of chase on any given edge of the graph, there is a minimum positive probability of choosing an optimum solution at any given time-step t. Therefore the probability of never observing an optimal solution is zero and we will almost surely find an optimum.[2]

However, this doesn't give any idea of precisely how long this process will take. Knowing that the algorithm will find the optimum infinite steps is unhelpful if the time it takes to reach those steps is longer than it takes to reach the heat death of the universe. Now, we will appeal to the stochastic process induced by the iterations of the algorithm to get a sense of the running time of the algorithm. In particular, if we denote the sequence of random chase vectors as {Yt} we will show that {Yt} is an ergodic Markov chain with a countable state space with a slight restriction to the initial chase vector.

We will assume the same problem as before with a finite directed acyclic graph G(V, E) with start u0 and destination v0 with the function f under the same assumptions as before. We note that for any path x € Su0, there is a unique set of edges associated with the vertices in the path. For convenience, we will say that an edge, e, connecting adjacent vertices, u1; u2 € x is in the path.

Now we provide some restrictions to the chase vectors which will be important in proving the ergodicity of the chase vectors. First, we define a vector $\emptyset$(x) for any path x € Su0 to be a chase vector such that:

$\emptyset$(x) (e) = {1- T min if e is an edge in the path x
Tmin Otherwise

Then we can define:

C = { $\emptyset$(x) | x € Su0 }

Lastly, we define:

Υ={ $\emptyset$ € R|E| |P { Yt = $\emptyset$, Y0 = $\emptyset$(x) } > 0 for some t > 0 and for some x € Su0

Observe that if we initialize the algorithm in the set V, then we actually have a countable infinite number of possible chase vectors that the algorithm can actually achieve since there is only a finite number of a new configuration available at each iteration. Now we need to show that Yt satisfies the Markov property and demonstrate ergodicity.

Lemma 5.1. Let {Yt}be a sequence of random variables over V such that Yt is the vector of chase values at time t of the MMCS algorithm. Then {Yt}is a Markov chain.

Proof. Returning to the Definition, we want to show that:

P{Yt+1 = y | Y0 = y0, …, Yt = yt} = P{Yt+1 = yt+1 | Yt = yt }.

But this is as simple as realizing that the random mechanism by which Yt+1 is generated if we are at time t is entirely determined by the behaviour of the cheetahs randomly chasing across the area. The random chases of the cheetahs are in turn governed strictly by a function of the current chase values at time t.

Lemma 5.2. The chase vector of the MMCS algorithm is aperiodic.

Proof. For aperiodicity, we can consider the state in which an optimal path x* € S has taken all maximal chase values and edges not in x have taken all minimal chase values at time t. Denote this state y*. Then in time t + 1, there is a positive probability that all edges of x* are reinforced. But the bounds on chase values mean that this event leaves all edges of x* at 1- Tmin and all other edges at Tmin. Therefore, we have that:
gcd {t | P{Yt = y* | Y0 = y* }} = 1.
Hence, the state y* is aperiodic. Then since we know that Yt is irreducible (at least based on our chosen initialization), we can use Lemma 2.6 to conclude that the whole Markov chain is aperiodic.

Theorem 5.3: The chase vector of the MMCS is ergodic and thus has a stationary distribution.
Proof. Irreducibility and aperiodicity are shown in the previous two lemmas. Only positive recurrence remains.
Consider an arbitrary path x* € Su0 and consider the associated chase vector where every edge in x* has chase value 1-Tmin and Tmin everywhere else. Denote this vector as $\emptyset$*. Then for any other $\emptyset \in \gamma$, our goal is to bound the expected number of steps until $\emptyset$* is reached again.
First, reconstruct M as in the proof of Lemma 8.2, where M is the minimum number of consecutive iterations of reinforcing only the edges in x* to reach $\emptyset$* from any state. Then we observe that the event {$\emptyset$* is reached within M steps} has a fixed minimum probability p* of occurring from any state.
If we consider the geometric random variable Z where a trial is M iterations with probability of success given by p*, we see that that the expected time it takes to reach $\emptyset$* from any vector $\emptyset$ is bounded above by:

$$E T_{\emptyset,\emptyset*} \leq M * EZ = \frac{M}{p*}$$

This means that the expected hitting time from any state $\emptyset$ is finite. Therefore, the expected return time must be finite and $\emptyset$* must be positive recurrent. By Lemma 2.13, Lemma 5.1, and Lemma 5.2, we have that the chase vector is ergodic.

VI. CONCLUSION

In this study, we discussed the notion of a Markov chain and some of its basic properties. We applied Markov chains to the analysis of Cheetah Chase Algorithm. In particular, we prove at the Max-Min Cheetah System for finding shortest path and use the mixing times of Markov chains to provide upper-bounds on running time estimation.

VII. REFERENCES

[1] David A. Levin, Yuval Peres and Elizabeth L. Wilmer. Mixing Times and Markov Chains.
[2] Wayne Yang, Markov Chains And Ant Colony Optimization, August 2013.
[3] James Norris. Markov Chains. Cambridge Press 1998.
[4] Goudhaman.M, Vanathi. N, Sasikumar.S, (2018), Semantic Approach for Dynamic Shortest Path Problem (SPP) By Cheetah Chase Algorithm (CCA)";, IAETSD JOURNAL FOR ADVANCED RESEARCH IN APPLIED SCIENCES VOLUME 5, ISSUE 8, August/2018 ISSN NO: 2394-8442, PAGE NO:229-237, UGC Indexed Journal - August 2018.
[5] https://cheetah.org/about-the-cheetah/
[6] http://c21.phas.ubc.ca/article/cheetah-chase
[7] Elliot JP, McTaggart C, Holling CS. 1977. Prey capture by the African lion. Canadian Journal of Zoology 55: 1811– 1828. doi: 10.1139/z77-235.