The following equations can be shown by induction:

$$v_k(a) = \begin{cases} \max\limits_{a \to b} \left(w_{ab} + v_{k-1}(b) \right) & a \in V_{max} \\ \min\limits_{a \to b} \left(w_{ab} + v_{k-1}(b) \right) & a \in V_{min}. \end{cases}$$

This gives an algorithm to compute $\vec{v}_k$ for each k.

$$\underset{\vec{v}_0}{\begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}} \quad \underset{\vec{v}_1}{\begin{bmatrix} \cdot \\ \\ \\ \\ \end{bmatrix}} \quad \cdots \quad \underset{\vec{v}_k}{\begin{bmatrix} \\ \\ \\ \\ \\ \end{bmatrix}}$$

Value iteration.

<u>Claim</u>: Compute $\vec{v}_k$ for $k = 4n^3 W$. The value vector for the mean-payoff game can then be deciphered in constant-time.

<u>Lemma 1</u>: For every $a \in V$,

$$k \cdot v(a) - 2nW \le v_k(a) \le k \cdot v(a) + 2nW$$

<u>Proof</u>: Consider optimal strategy σ in F starting from 'a'. Show that playing that strategy with stack-based extension $\tilde{\sigma}$ gives $v_k(a) \ge k \cdot v(a) - 2nW$

Similarly using τ and $\tilde{\tau}$ show other inequality.

H1, SIPCOT IT Park, Siruseri, Kelambakkam 603 103, INDIA
Phone: +91 44 6748 0900 **Email:** office@algolabs.org.in

~~From~~ In above Lemma, substituting $k = 4n^3 W$ gives:

$$\frac{v_k(a)}{k} \leq v(a) + \frac{2nW}{k}$$

i.e., $$\leq v(a) + \frac{1}{2n^2} \longrightarrow (1)$$

$$n^2 \geq n(n-1)$$

$$\therefore \frac{1}{n^2} \leq \frac{\cancel{n(n-1)}\,1}{n(n-1)}$$

$$\therefore -\frac{1}{n^2} \geq \frac{-1}{n(n-1)} \longrightarrow (2)$$

~~Let~~ $\cancel{v'(a) = \frac{v_k(a)}{k}}$

$\cancel{v(a) \geq v'(a) + \frac{1}{2n^2}}$

~~Similarly~~ $\cancel{\frac{v_k(a)}{k} \geq v(a) + 1}$

Let $$v'(a) = \frac{v_k(a)}{k}$$

(1) gives: $$v(a) \geq v'(a) - \frac{1}{2n^2}$$

(2) gives: $$v(a) > v'(a) - \frac{1}{2n(n-1)} \longrightarrow (3)$$ [strict inequality is $n>1$]

Similarly $$v'(a) \geq v(a) - \frac{2nW}{k}$$ [previous Lemma]

$$v(a) \leq v'(a) + \frac{1}{2n^2}$$ [$k = 4n^3 W$]

From (2): $$v(a) < v'(a) + \frac{1}{2n(n-1)} \longrightarrow (4)$$

Overall: $$v'(a) - \frac{1}{2n(n-1)} < v(a) < v'(a) + \frac{1}{2n(n-1)}$$

AlgoLabs

http://www.algolabs.org.in

From the previous discussion, it follows that if we do the value iteration for $k = 4n^3 W$ steps, we get a vector $v'(a)$: and $v(a)$ is in the $\frac{1}{2n(n-1)}$ ball around it. Can we uniquely spot it?

Observation about value: From the first cycle game, note that the value will be of the form p/q where $q \leq n$, [p, q are ~~tie~~ integers].

Claim: Pick 2 no.s. $\cancel{\frac{a}{k}}, \cancel{\frac{k}{\ell}}$ $\underbrace{\frac{a}{b}, \frac{p}{q}}_{\text{distinct}}$ s.t. $0 \leq q, b \leq n$. $n \neq 1$.

Then $\left|\frac{a}{b} - \frac{p}{q}\right| \geq \frac{1}{n(n-1)}$

Proof: Suppose $b = q$.

$$\left|\frac{a}{b} - \frac{p}{q}\right| = \left|\frac{a-p}{q}\right| = \frac{|a-p|}{q} \geq \frac{1}{q} \geq \frac{1}{n} \geq \frac{1}{n(n-1)}$$

Suppose $b \neq q$.

$$\left|\frac{a}{b} - \frac{p}{q}\right| = \frac{|aq - bp|}{bq} \geq \frac{1}{bq} \geq \frac{1}{n(n-1)}$$

Therefore in the ball $\left(v'(a) - \frac{1}{2n(n-1)}, v'(a) + \frac{1}{2n(n-1)}\right)$ there is atmost one number of the form $\frac{p}{q}$ with $q \leq n$! Moreover we know there exists one such number - which is the value $v(a)$.

How do we find it? $a' = v'(a) - \frac{1}{2n(n-1)}$ $b' = v'(a) + \frac{1}{2n(n-1)}$

If there is an integer between a' and b', then we are done.

Otherwise check if there is some p/q with $q = 2$. Multiply a', b' by 2. → So on.

H1, SIPCOT IT Park, Siruseri, Kelambakkam 603 103, INDIA
Phone: +91 44 6748 0900 **Email:** office@algolabs.org.in

Summary so far:

1. Proof of positional determinacy of finite as well as infinite version for the game starting at a_0 ~~[positional strategy for infinite~~ [proof that was discussed can be applied to both versions. Same strategy works for both cases].

2. Computing values uniformly for all vertices using a value iteration method. - Uri - Zwick's procedure.

Complexity: $4n^3 W \cdot |E|$

$= O(|V|^3 \cdot |W| \cdot |E|)$

$\longrightarrow$ pseudopolynomial.

Weights can be represented in binary. So all weights need $\log W$ bits. An algo with $|W|$ steps will then be exponential in representation.

$\rightarrow$ Same as PRIMEs. If number is represented in unary (no. of bits same as the number) then simple algos work in PTIME. Main question was what happens when we have a binary representation.

Coming next: The above ~~comp:~~ value iteration method computes values. But it does not give a positional strategy immediately.

How to compute uniform positional strategies?

$\rightarrow$ Theorem 3.1 in Zwick - Patterson's paper.