Rutgers Business School
Newark and New Brunswick

An Introduction to Search Games

Thomas Lidbetter
Department of Management Science and Information Systems

Monday 25th July 2022

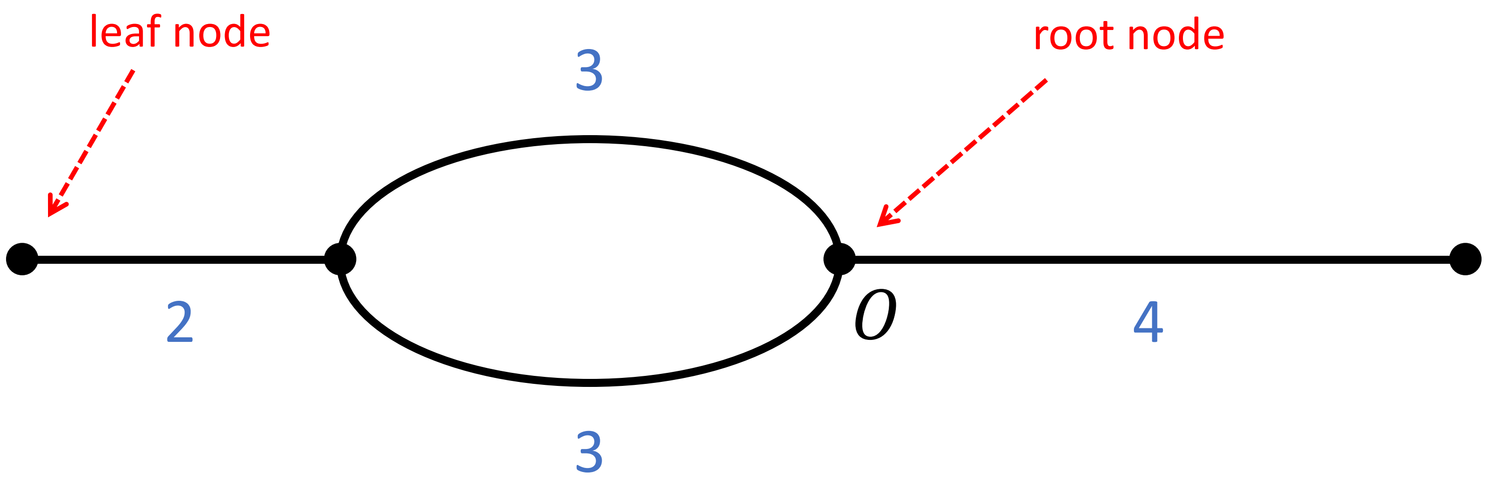
This material is based upon work supported by the National Science Foundation under Grant No. CMMI-1935826. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Part I: Isaac’s problem and Gal’s solution
Hide and seek on a network

- Rufus Isaacs (1965) Differential Games
- Shmuel Gal (1979) Search Games with Mobile and Immobile Hider
- Shmuel Gal (2000) On the Optimality of a Simple Strategy for Searching Graphs
- Steve Alpern (2011) A new approach to Gal's Theory of Search Games on Weakly Eulerian networks
- Steve Alpern, Thomas Lidbetter (2020) Search and delivery man problems: when are depth-first paths optimal?

Search for Immobile Hider on a Network

- Every arc a of a network Q has length $L(a)$
- Total length of Q is $L(Q) = \mu$
- Distance function d on Q is the "shortest path" metric

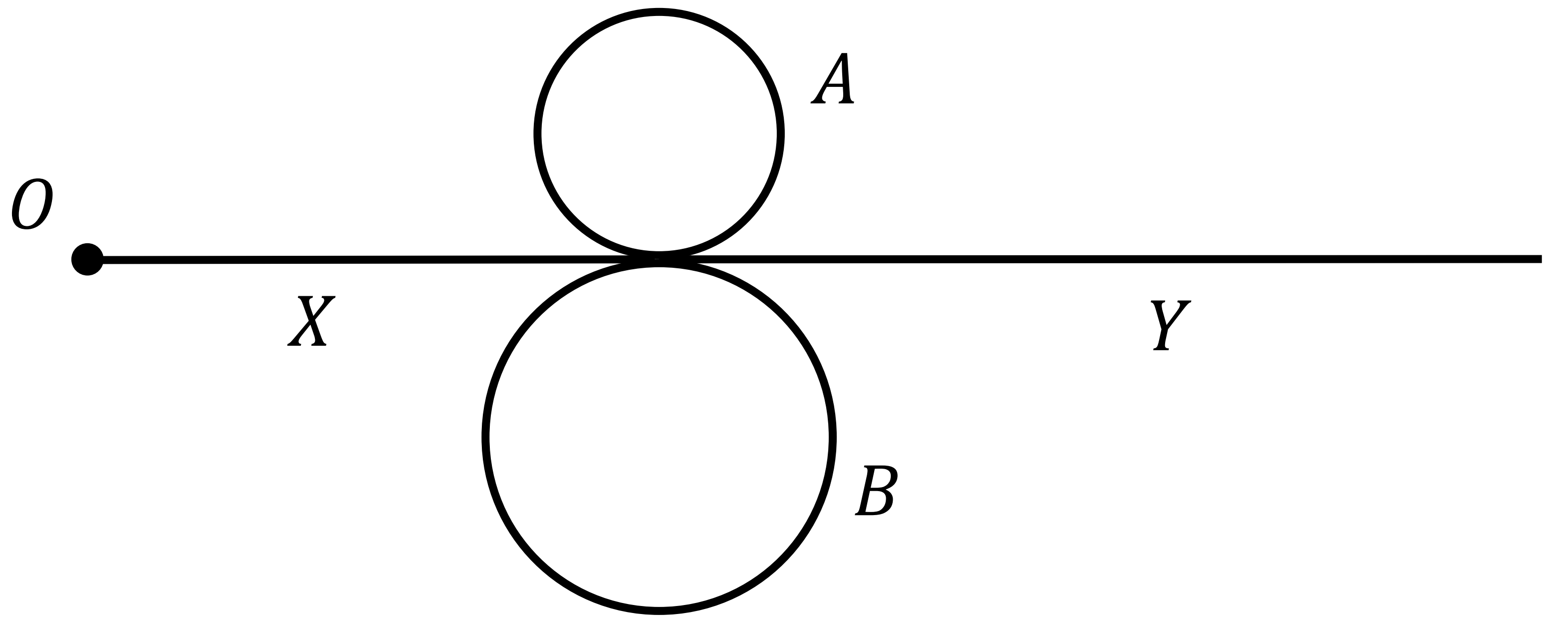


The game $\boldsymbol{G = G(Q, O)}$

- Pure strategy for Hider (maximizer): a point in Q (not necessarily a node)
- Mixed strategy h for Hider is a distribution over Q
- For $A \subseteq Q$, write $h(A)$ for the probability the Hider is in A
- Pure strategy for Searcher (minimizer) is a unit speed path $S(t), t \geq 0$ which covers Q
- Mixed strategy for the Searcher is a probability distribution over such paths
- The payoff is the *search time* $T = T(S, H) = \min\{t: S(t) = H\}$
- The function T is only *lower-semicontinuous* (uniform topology) but the game has a value $V = V(Q, O)$, optimal mixed Searcher strategies and ε-optimal mixed Hider strategies.

Search higher density regions first

For a fixed Q and Hider distribution h, which has a lower expected search time: X, A, B, Y or X, B, A, Y?



It turns out that the answer depends only on the *search density* ρ of A and B, where

$$\rho(C) = h(C)/t(C)$$

and $t(C) =$ time spent in C.

Search higher density regions first

Search density lemma: For a fixed Hider distribution h on a network Q, suppose S_1 is a search of Q that can be written as X, A, B, Y and S_2 is a search that can be written as X, B, A, Y, where X, A and B are disjoint. Then $T(S_1, h) \leq T(S_2, h)$ if and only if $\rho(A) \geq \rho(B)$, with equality if and only if the densities are equal.

Proof: Write T_A for the expected time spent to find the Hider in A, assuming he is in A. Similarly for T_B. Then

$$\begin{aligned} T(S_2, h) - T(S_1, h) &= h(B)(t(X) + T_B) + h(A)(t(X) + t(B) + T_A) \\ &\quad - h(A)(t(X) + T_A) - h(B)(t(X) + t(A) + T_B) \\ &= t(A)t(B)\left(\frac{h(A)}{t(A)} - \frac{h(B)}{t(B)}\right). \end{aligned}$$

Uniform Hider strategy

A mixed strategy always available to the Hider is the uniform strategy $h = u$ which hides in any subset A of Q with probability proportional to its length, that is $u(A) = L(A)/\mu$.

Lemma: For any (Q, O) and any S,

$$T(S, u) \geq \mu/2.$$

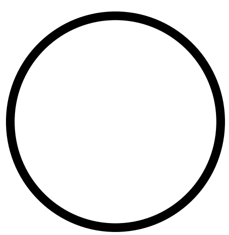
Hence, $V \geq \mu/2$.

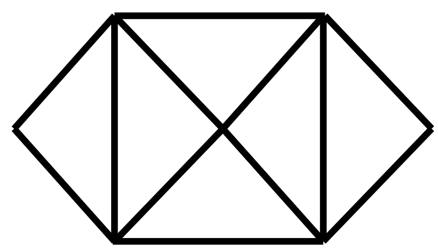
Proof: By time t, max. probability $F(t)$ of finding the Hider is t/μ, so

$$T(S, u) = \int_0^\infty 1 - F(t)\, dt \geq \int_0^\mu 1 - \frac{t}{\mu} dt = \mu/2.$$

Chinese Postman Tours

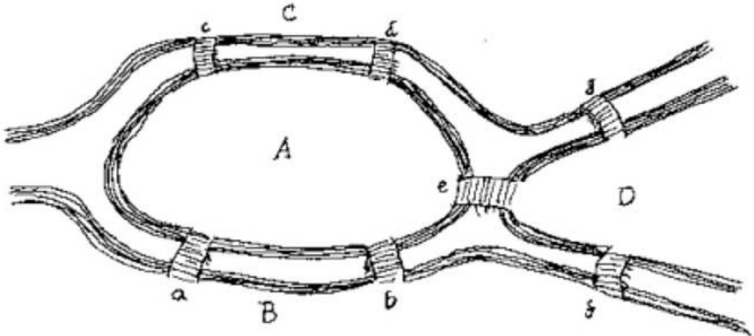
- A covering path S is called a tour if it ends back at O
- If a tour has minimum length, denoted $\bar{\mu}$, it is called a *Chinese Postman Tour*
- A tour is called *Eulerian* if it has length μ (traverses each edge exactly once)
- An Eulerian tour exists if all nodes have even degree (number of incident edges), in which case Q is called Eulerian and $\mu = \bar{\mu}$

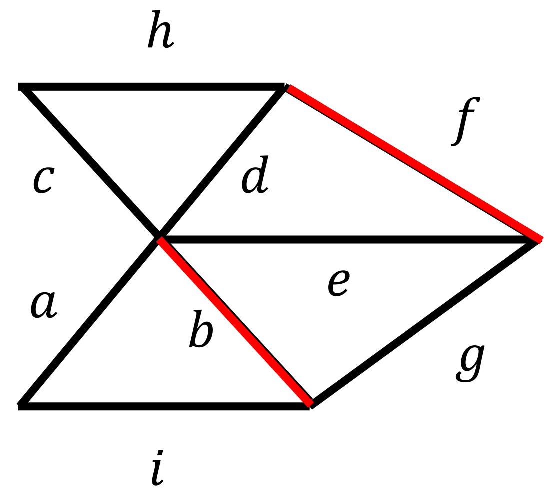




Chinese Postman Tours

Example

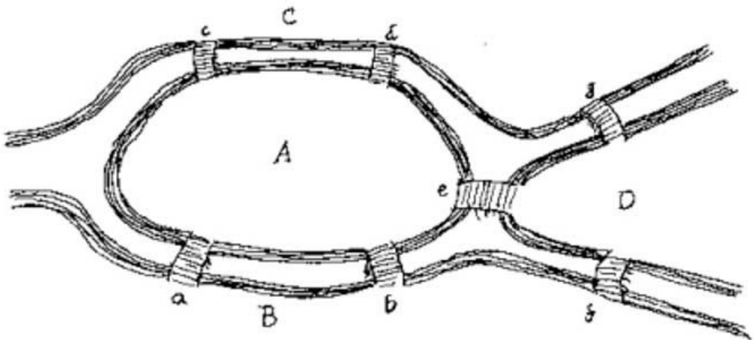


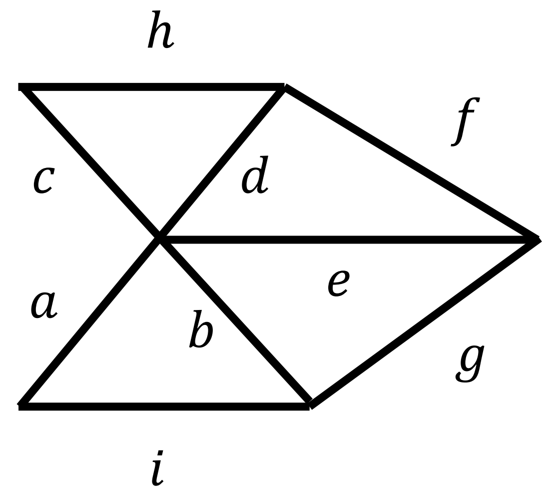


CPT: $achfbgdfebi$

Chinese Postman Tours

Lemma: Any CPT of Q satisfies $\bar{\mu} \leq 2\mu$ with equality only for trees.





Proof:

- Define Q' by doubling every arc of Q (add another arc of the same length with the same endpoints)
- All nodes of Q' have even degree so there is an Eulerian tour of length 2μ
- This is also a covering tour of Q
- If Q is not a tree it contains a circuit (closed path of distinct arcs), whose arcs we do not need to double

Random Chinese Postman Tours

Definition: Suppose that $S: [0, \bar{\mu}] \to Q$ is a CPT. Let S^r denote its reverse, given by $S^r(t) = S(\bar{\mu} - t)$. A Random Chinese Postman Tour (RCPT) s is an equiprobable mix of S and S^r.

Lemma: Let s be a RCPT on a network Q with root O. Then for any $H \in Q$, $T(s, H) \leq \bar{\mu}/2$. Hence $V \leq \bar{\mu}/2$.

Proof: Let t be such that $S(t) = H$. Then $T(S^r, H) \leq \bar{\mu} - t$. So

$$T(s, H) = \frac{1}{2}T(S, H) + \frac{1}{2}T(S^r, H) \leq \frac{1}{2}t + \frac{1}{2}(\bar{\mu} - t) = \bar{\mu}/2.$$

Bounds on $V = V(Q, O)$ for a general network

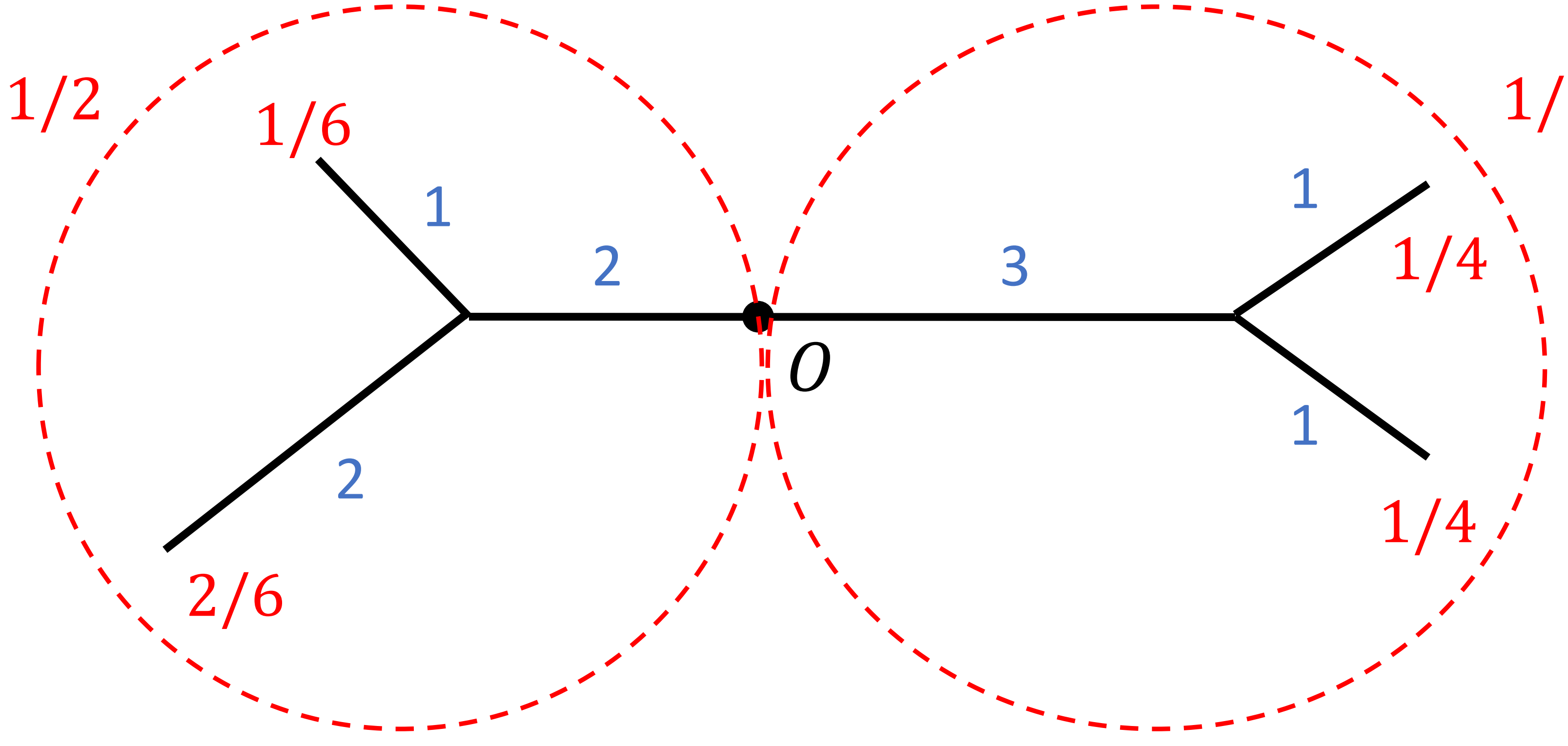
Theorem: For any network Q with root O, the value $V = V(Q, O)$ of the search game for an immobile Hider satisfies

$$\frac{\mu}{2} \leq V \leq \frac{\bar{\mu}}{2} \leq \mu.$$

The lower bound is tight if and only if Q is Eulerian. The bound $V \leq \mu$ can only be tight if Q is a tree.

Equal Branch Density (EBD) Hider Distribution for Trees

Definition: The EBD Hider distribution is concentrated on the leaf nodes and at every branch node the search density of all branches is equal.



Depth-first search is a best response against the EBD

Lemma: Any depth-first search S is a best response against the EBD distribution, h and has expected search time $T(S, h) = \mu$.

Proof

(i) Any two depth-first searches S_1 and S_2 have the same expected search time because S_1 can be transformed into S_2 by successively swapping the order of search of equal density subtrees that share a root.

(ii) If S is any depth-first search and S^r is its time reverse search then for any leaf node v,

$$T(S, v) + T(S^r, v) = 2\mu,$$

so

$$T(S, h) + T(S^r, h) = 2\mu,$$

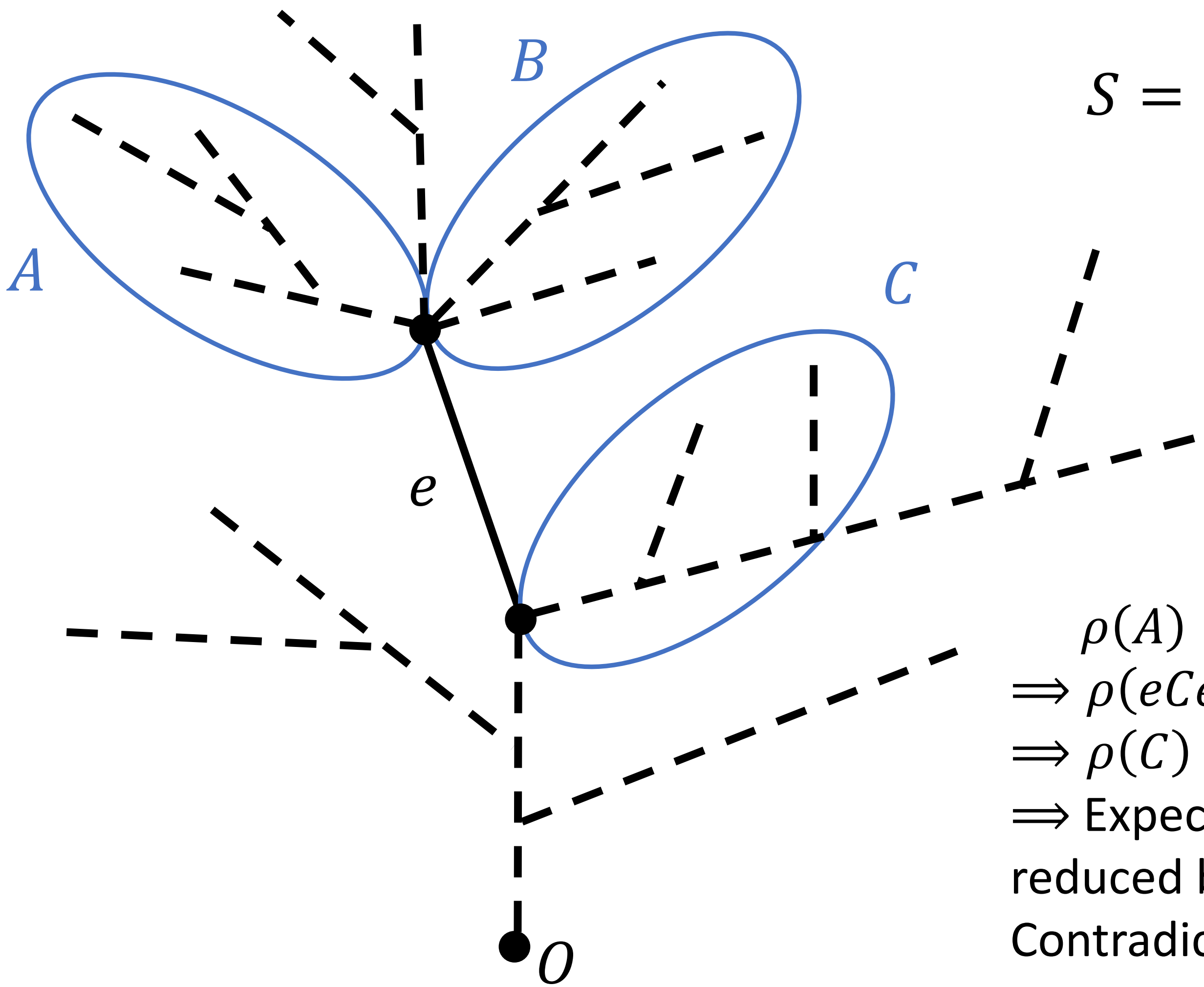
so

$$T(S, h) = T(S^r, h) = \mu.$$

(iii) Proof by contradiction that any depth-first search is a best response…

Depth-first search is a best response against the EBD

(iii) (continued) If a best response S is not depth-first, it must be of the form:



$$S = \cdots eAeCeB \cdots$$

$\rho(A) = \rho(B)$
$\Longrightarrow \rho(eCe) = \rho(A) = \rho(B)$
$\Longrightarrow \rho(C) > \rho(eAe)$
$\Longrightarrow$ Expected search time can be reduced by swapping eAe and C. Contradiction!

$V = \mu = \bar{\mu}$ for trees

Theorem: Let Q be a tree with root O. Then $V = \mu$.

Proof:
(i) $V \leq \bar{\mu}/2 = \mu$ (Searcher uses RCPT)
(ii) $V \geq \mu$ (Hider uses EBD distribution)

Other networks…

Arc-adding lemma: Let Q be a network and let Q' be derived from adding an arc e of length $\ell \geq 0$ between points x and y on Q. Then

1. $V(Q') \leq V(Q) + 2\ell$ so $V(Q') \leq V(Q)$ if we identify x and y (i.e. $\ell = 0$).

2. If $\ell \geq d_Q(x, y)$, then $V(Q') \geq V(Q)$. Any hiding strategy on Q does just as well on Q'.

Other networks…

Arc-adding lemma: Let Q be a network and let Q' be derived from adding an arc e of length $\ell \geq 0$ between points x and y on Q. Then

1. $V(Q') \leq V(Q) + 2\ell$ so $V(Q') \leq V(Q)$ if we identify x and y (i.e. $\ell = 0$).

Proof: Replace every pure S used in an optimal strategy s by S' which follows S until it reaches x, then tours e, then follows S again.

$$T(s, z) \leq V(Q) + \ell \text{ for } z \in e$$

and

$$T(s, z) \leq V(Q) + 2\ell \text{ for } z \notin e.$$

Other networks…

Arc-adding lemma: Let Q be a network and let Q' be derived from adding an edge e of length $\ell \geq 0$ between points x and y on Q. Then

2. If $\ell \geq d_Q(x, y)$, then $V(Q') \geq V(Q)$. Any hiding strategy on Q does just as well on Q'.

Proof: Let h be optimal on Q. Let h' on Q' be same as h (don't hide in e). Note that for $H \in Q$,

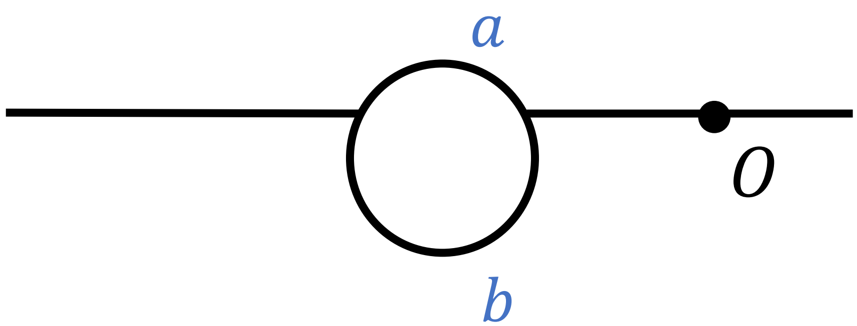
$$T_{Q'}(S', H) \geq T_Q(S, H),$$

where S is like S' but replacing e with the shortest path from x to y in Q.

Other networks…

Proposition: The network Q drawn below has $V(Q) = \bar{\mu}/2$.

Q



Proof:

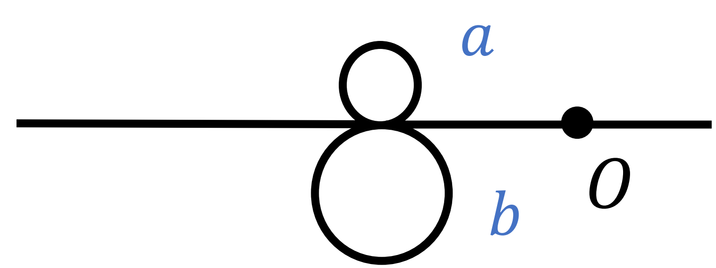
$V(Q^*) \leq V(Q)$ by arc-adding lemma (1).

$V(Q^*) \geq V(Q^{**})$ by arc-adding lemma (2).

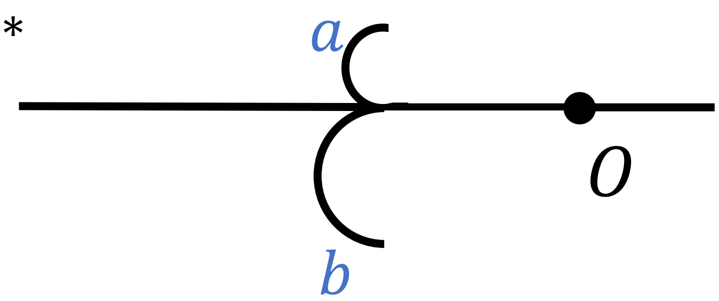
But $V(Q^{**}) = \bar{\mu}/2$ by the tree theorem, so

$\bar{\mu}/2 = V(Q^{**}) \leq V(Q^*) \leq V(Q) \leq \bar{\mu}/2.$

Q^*



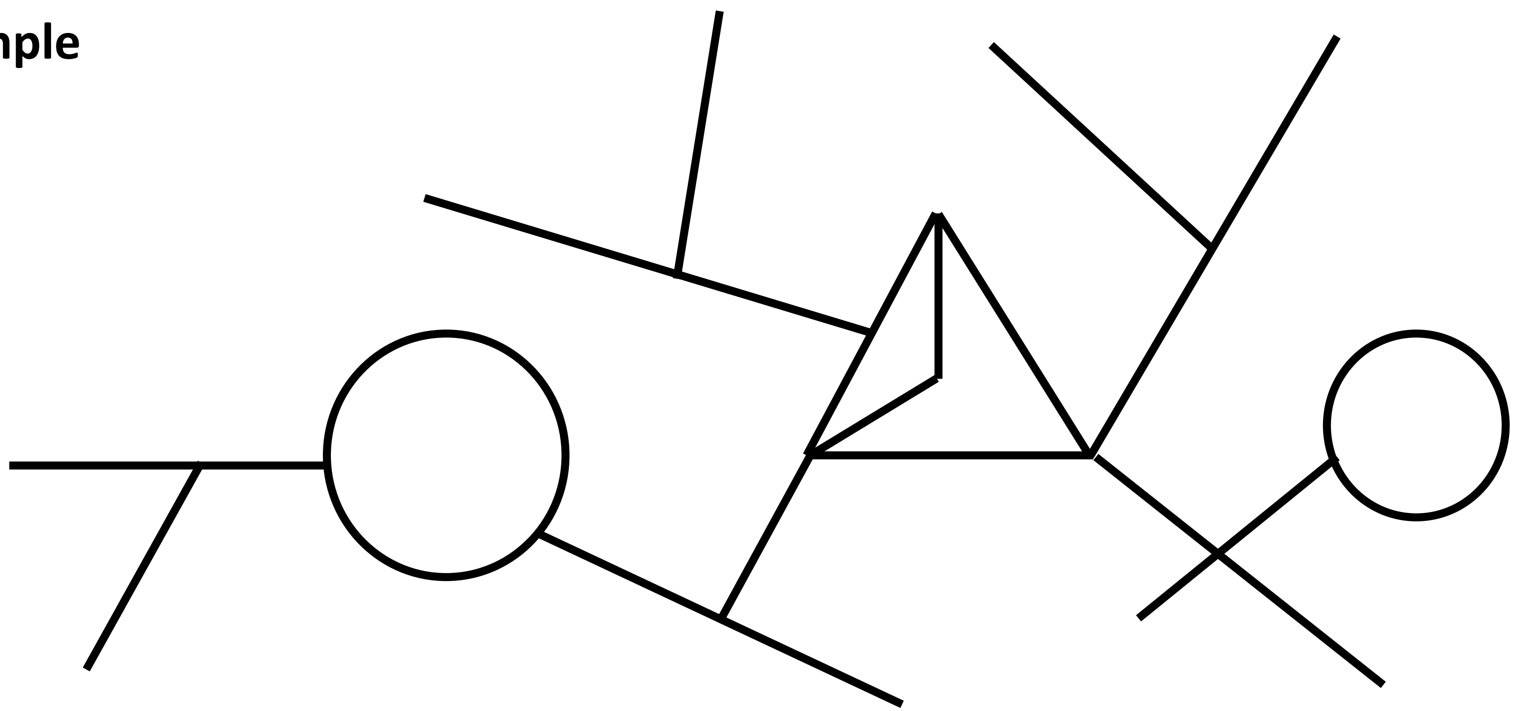
Q^{**}



Weakly Eulerian networks

Definition: A network is weakly Eulerian if it contains a disjoint set of Eulerian networks such that shrinking each to a point transforms the network into a tree.
Equivalently, a network is weakly Eulerian if removing all disconnecting edges leaves a network with only even degree nodes.

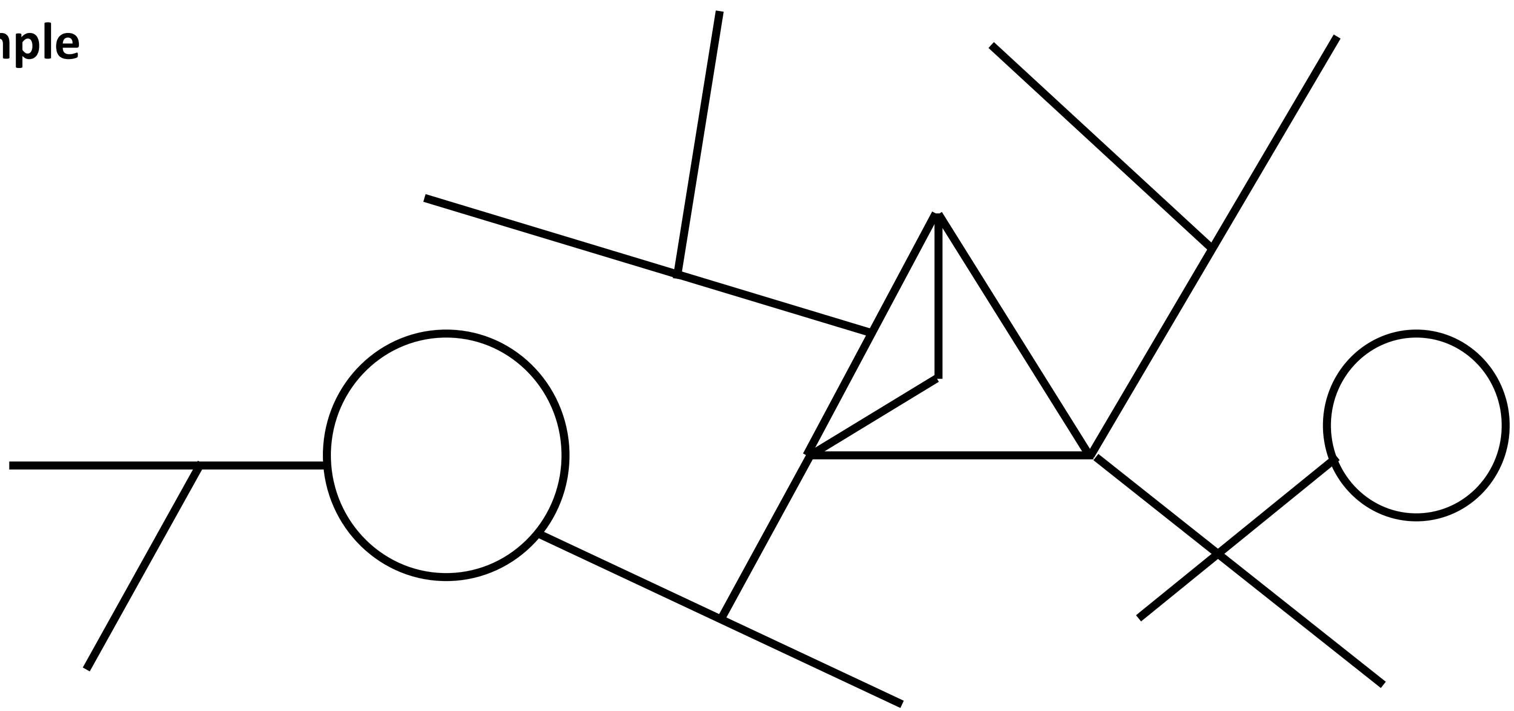
Example



Weakly Eulerian networks

Definition: A network is weakly Eulerian if it contains a disjoint set of Eulerian networks such that shrinking each to a point transforms the network into a tree.
Equivalently, a network is weakly Eulerian if removing all disconnecting edges leaves a network with only even degree nodes.

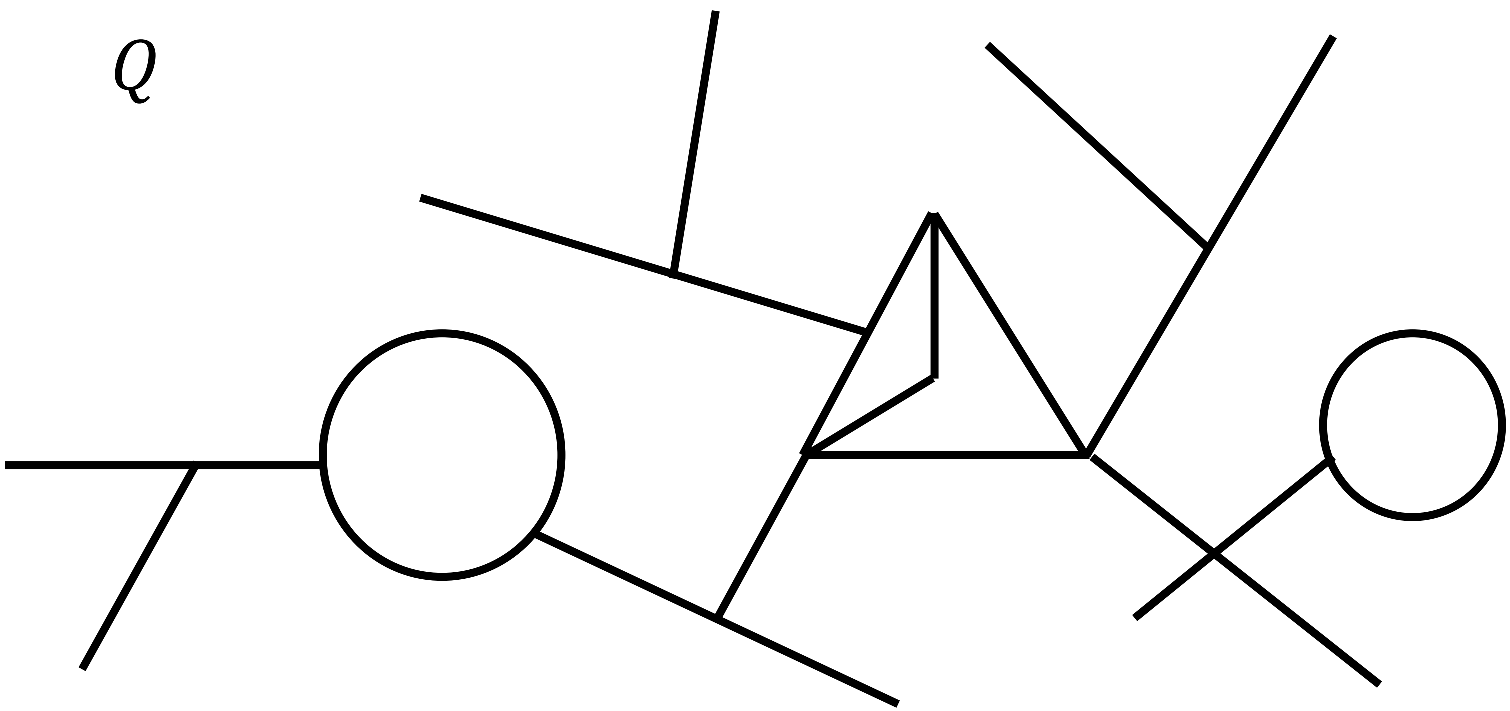
Example



Weakly Eulerian networks

Theorem: The value of the search game on a network Q is $\bar{\mu}/2$ if and only if Q is weakly Eulerian.

Proof ($\Leftarrow$): $\bar{\mu}/2 \geq V(Q)$

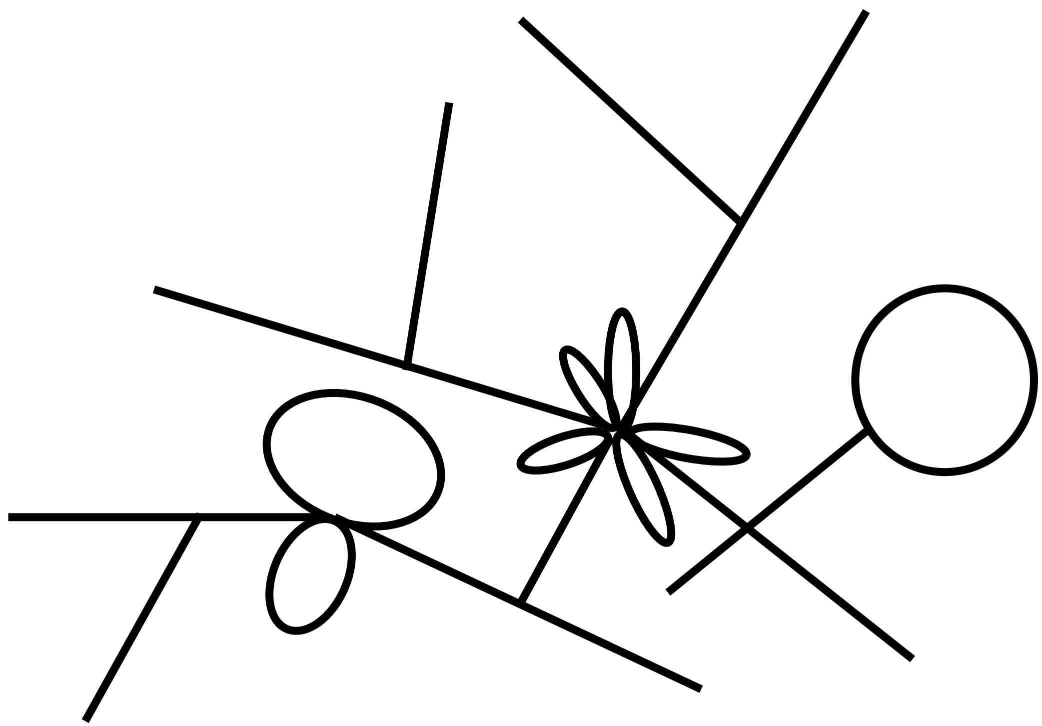


Weakly Eulerian networks

Theorem: The value of the search game on a network Q is $\bar{\mu}/2$ if and only if Q is weakly Eulerian.

Proof (⟸): $\bar{\mu}/2 \geq V(Q) \geq V(Q^*)$

Q^*

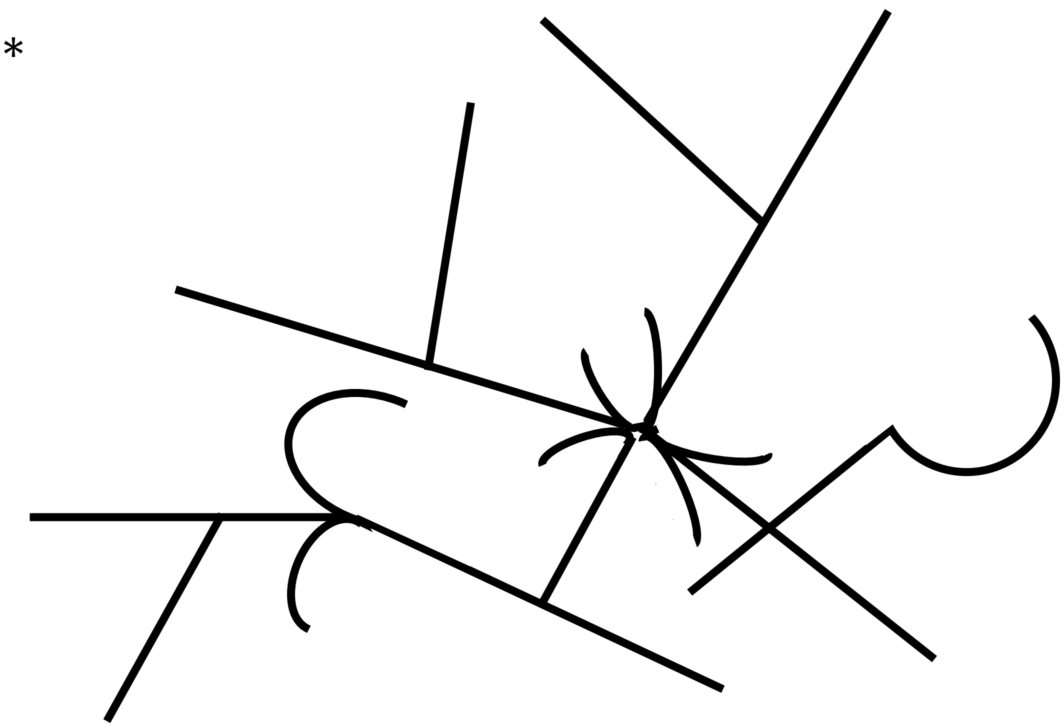


Weakly Eulerian networks

Theorem: The value of the search game on a network Q is $\bar{\mu}/2$ if and only if Q is weakly Eulerian.

Proof (⟸): $\bar{\mu}/2 \geq V(Q) \geq V(Q^{*}) \geq V(Q^{**})$

Q^{**}

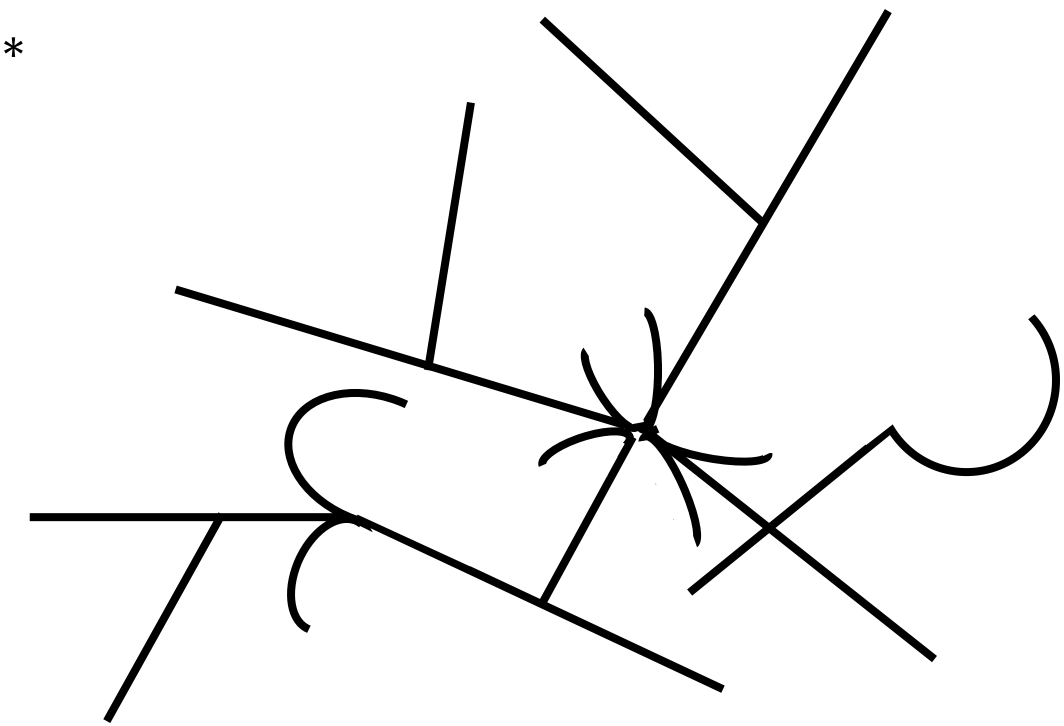


Weakly Eulerian networks

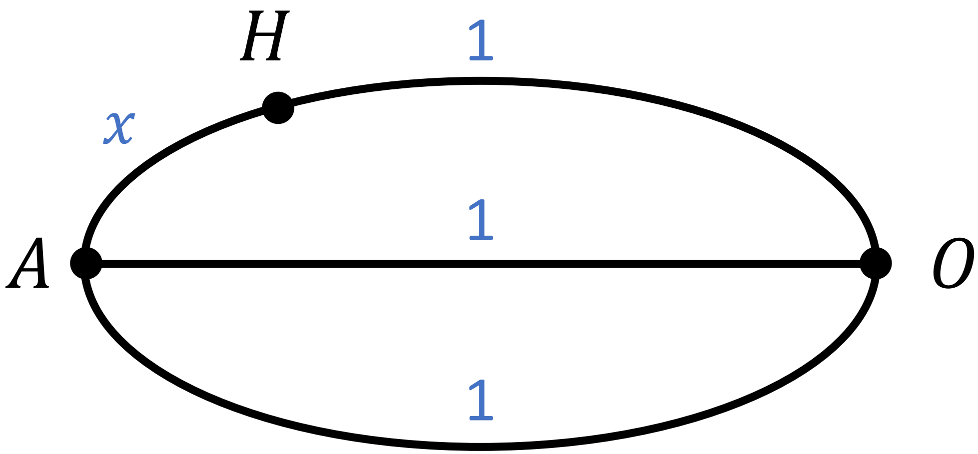
Theorem: The value of the search game on a network Q is $\bar{\mu}/2$ if and only if Q is weakly Eulerian.

Proof (⟸): $$\bar{\mu}/2 \geq V(Q) \geq V(Q^{*}) \geq V(Q^{**}) = \bar{\mu}/2$$

Q^{**}



The "Three arc" network



$\mu = 3, \bar{\mu} = 4$

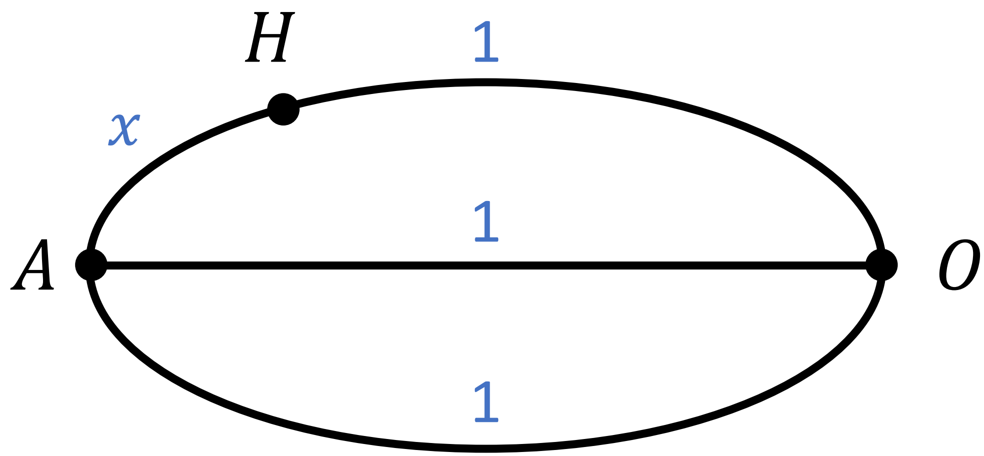
$$3/2 = \mu/2 \leq V(Q) \leq \bar{\mu}/2 = 2$$

If the Searcher successively chooses unsearched arcs at random, then

$$T(S,H) = \frac{1}{3}(1-x) + \frac{1}{3}(1+x) + \frac{1}{3}(3-x) = \frac{5-x}{3} \leq 5/3.$$

So $3/2 \leq V(Q) \leq 5/3$.

The "Three arc" network



$\mu = 3, \bar{\mu} = 4$

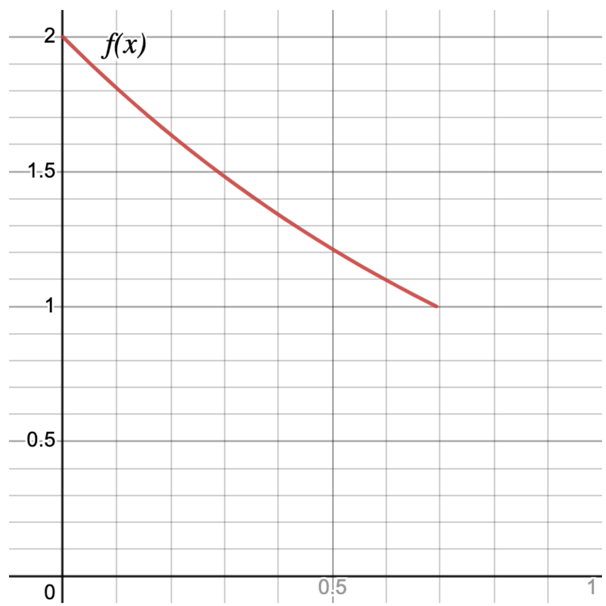
Theorem (L. Pavlovic): It is optimal for the Hider to choose x according to the p.d.f.

$$f(x) = 2e^{-x}, 0 < x < \ln 2 \approx 0.693.$$

It is optimal for the Searcher to go to A, go distance y towards O, back to A, to O on another arc, to A on the untraversed arc, where y is chosen according to the c.d.f.

$$F(y) = \frac{1}{2} + \frac{e^{y}}{4}, 0 \leq y \leq \ln 2\,.$$

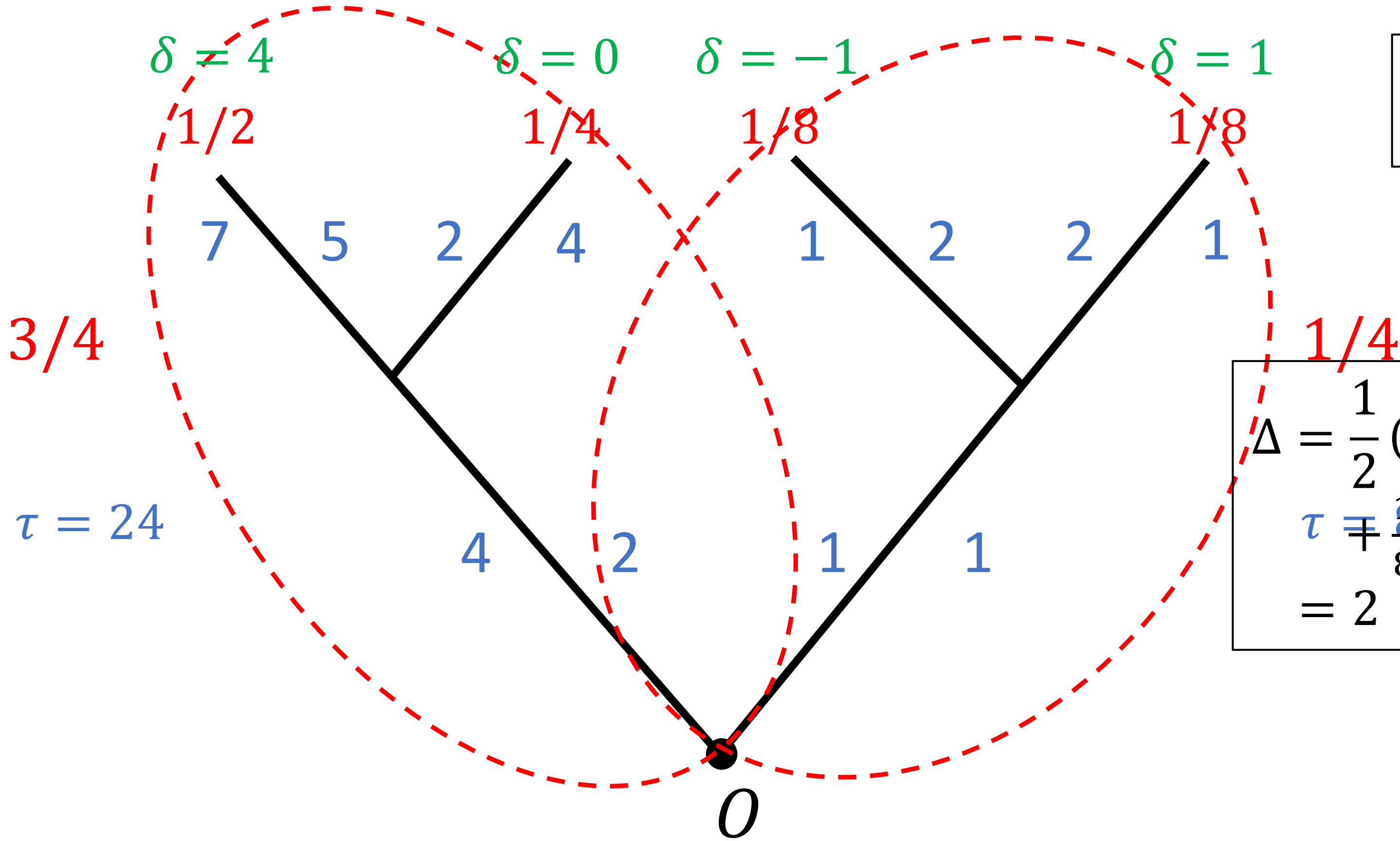
$V = (4 + \ln 2)/3 \approx 1.56$



Part II: Variations to the model

- Steve Alpern (2010) Search games on trees with asymmetric travel times
- Steve Alpern & Thomas Lidbetter (2014) Searching a variable speed network
- Steve Alpern & Thomas Lidbetter (2013) Mining coal or finding terrorists: the expanding search paradigm
- Steve Alpern (2011) Find-and-fetch search on a tree (2011)

Variable speed network (tree)

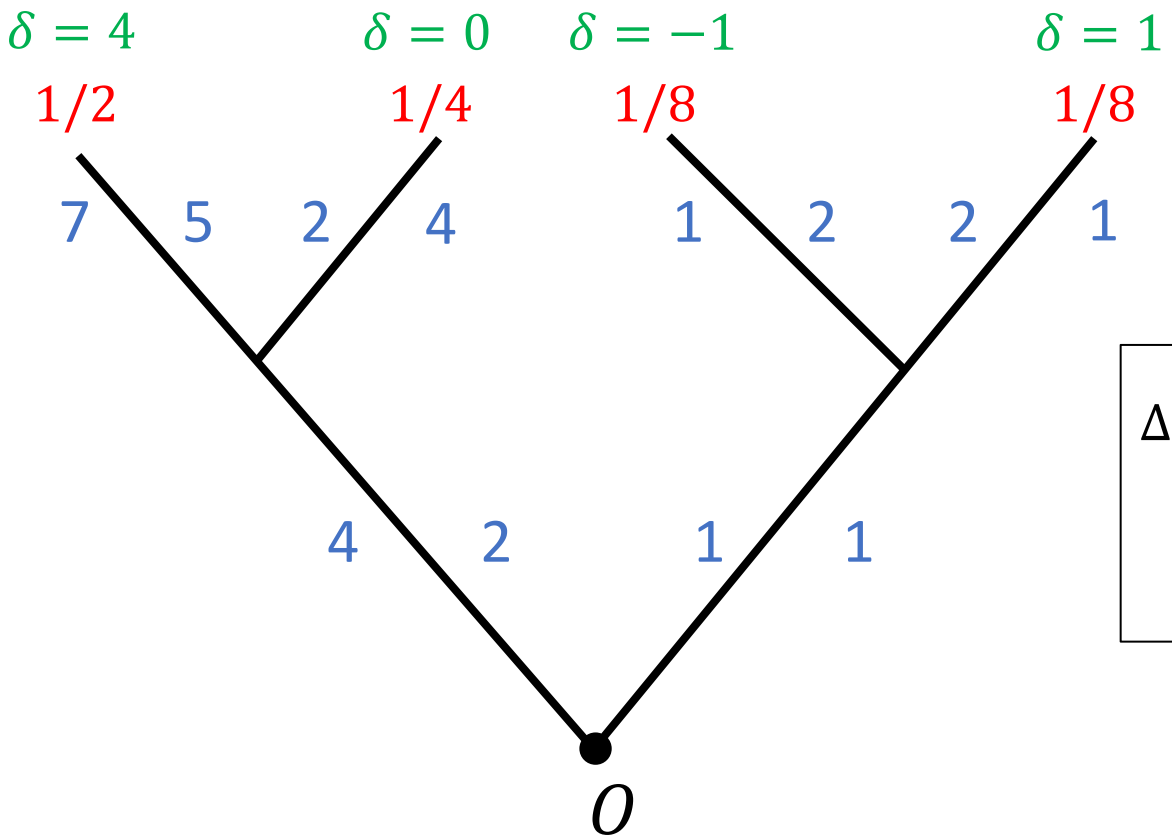


Out-time on left, in-time on right

$$\Delta = \frac{1}{2}(4) + \frac{1}{4}(0) + \frac{1}{8}(-1) + \frac{1}{8}(1) = 2$$

- Define EBD using tour times τ instead of lengths
- Define *height* $\delta(v)$ of a leaf node v as the difference between the time from O to v and the time from v to O.
- Define the *incline* Δ as the average height of a leaf node, weighted according to EBD.

Variable speed network (tree)



Out-time on left, in-time on right

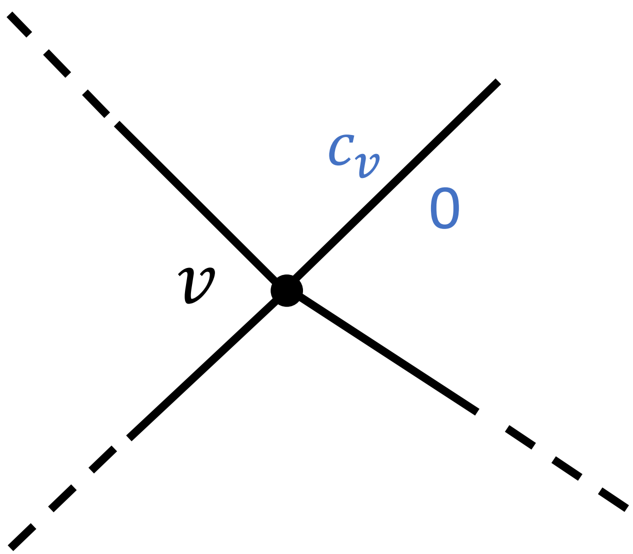
$$\Delta = \frac{1}{2}(4) + \frac{1}{4}(0) + \frac{1}{8}(-1) + \frac{1}{8}(1) = 2$$

$$V = \frac{32+2}{2} = 17$$

Theorem: The value of the variable speed search game is $\frac{\tau+\Delta}{2}$. The EBD is optimal for the Hider and it is optimal for the Searcher to use a probabilitistic "branching strategy".

Applications of variable speed: 1. Kikuta-Ruckle game

- Like the original Isaacs-Gal game, but the Hider can only hide at nodes and each node v has a search cost c_v.
- Searcher can either continue without searching a node or pay the search cost to search it.
- Replace search cost of c_v with a "variable speed" arc with outward travel time c_v and inward travel time 0.



Applications of variable speed: 2. Find-and-fetch

- Another variation on the classic model, where the Searcher has to return the Hider to the root (eg. search and rescue, foraging)
- Add a variable speed arc to each leaf node v with outward travel time equal $d(O,v)$ and inward travel time equal – $d(O,v)$.

Applications of variable speed: 3. Expanding search

- Searcher picks a sequence of arcs $a_1, a_2, \dots$ such that a_1 is incident to the root and each a_1 is incident to a node already reached.
- Suitable in cases where the cost to retrace your steps is negligible, eg. mining coal, searching for landmines.
- Can also model search with many searchers.
- For trees, this can be modeled by variable speed search: an arc of length a can be replaced by a variable speed arc with outward travel time a and inward travel time 0.

Part III: Search games with multiple hidden objects

- Hider hides k balls in n boxes
- Cost of searching box j is c_j
- Searcher looks in boxes one by one till finding all the balls
- Payoff is cost of finding all the balls.

Lemma: The Hider can make the Searcher indifferent between all her strategies by choosing a subset H of k boxes with probability $p^*(H) = \frac{\prod_{i \in H} c_i}{S_k}$, where $S_k = \sum_{|A|=k} \sum_{i \in A} c_i$.

All orderings have expected cost

$$C - \frac{S_{k+1}}{S_k},$$

Where $C = \sum_{j=1}^{n} c_j$.

Eg. ($k = 3$) This choice of H is picked with probability proportional to $3 \times 3 \times 2 = 18$.

Proof: For the ordering $1,2,\dots,n$, the expected cost of boxes *not searched* is

$$\sum_{j=k+1}^{n} c_j \sum_{H\in[j-1]^{(k)}} p^*(H) = \sum_{j=k+1}^{n} c_j \sum_{H\in[j-1]^{(k)}} \frac{\prod_{i\in H} c_i}{S_k} = \frac{S_{k+1}}{S_k}.$$

Theorem: The value of the game is $V = C - \frac{S_{k+1}}{S_k}$. It is optimal for the Searcher to start by opening a subset H of k boxes with probability $p^*(H)$ and to open the remaining boxes in a (uniformly) random order. An optimal strategy for the Hider is p^*.

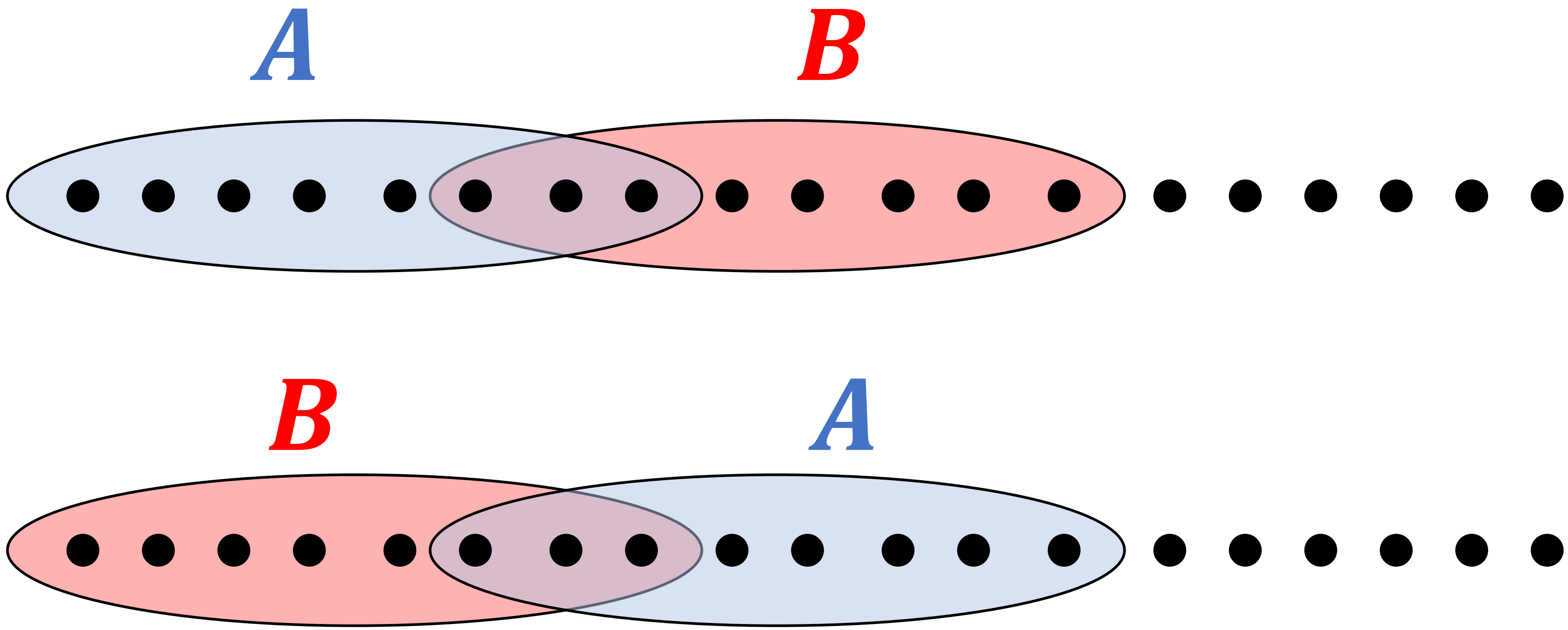
Proof:

- Restrict the Searcher to strategies of the form s_A = "search all boxes in A then search the remaining boxes in a random order", where $|A| = k$.
- Then payoff of s_A against B is same as payoff of s_B against A for $|A| = |B| = k$.

Expected search cost $= (7 + 2 + 6) + 3 + ½(3)$

Expected search cost $= (7 + 3 + 2) + 6 + ½(3)$

In general



- All boxes in A and B must be searched. Remaining boxes are all searched with the same probability.
- So payoff matrix is symmetric
- Thus Searcher can use strategy p^* to make Hider indifferent between all his strategies. Both players indifferent $\Longrightarrow$ equilibrium.