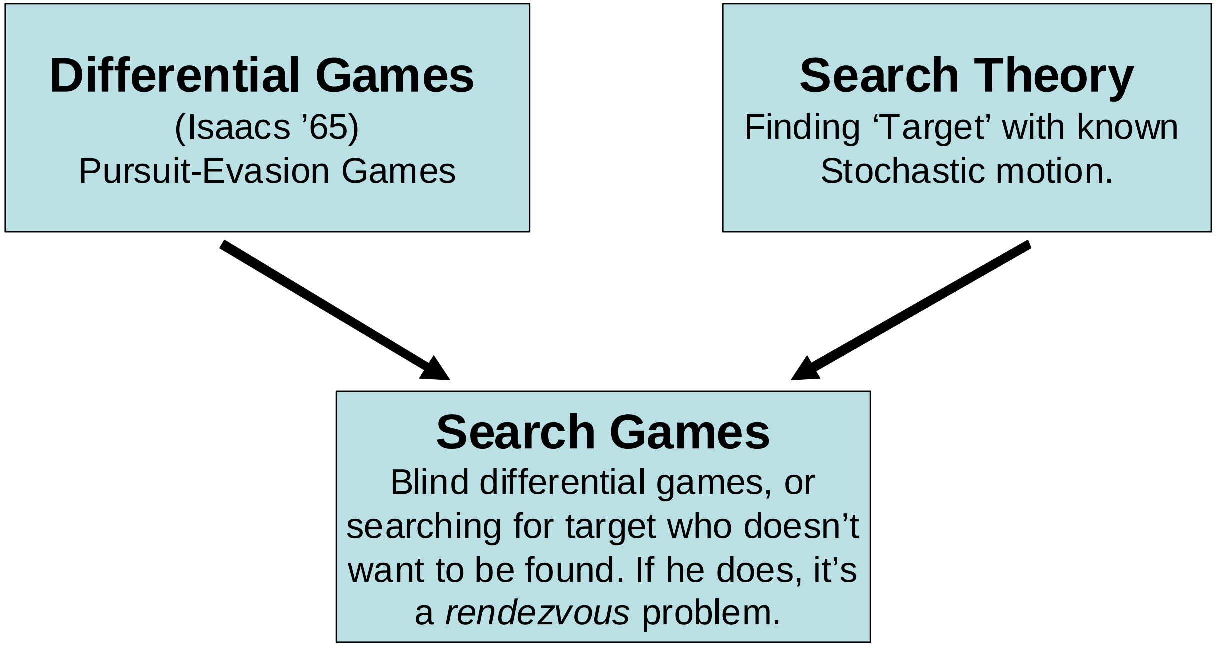
What are Search Games?



A Pursuit-Evasion Example

We consider the 'Lion and Man' Problem of Rado (1920's). A unit speed Lion L wants to catch a Man M (for dinner) in the plane. Man has speed $v \leq 1$.

1. How does Lion win if $v < 1$?

2. How does Man escape if $v = 1$?

3. How does Lion win if $v = 1$ inside circle?

Answer * to Lion-Man part 3

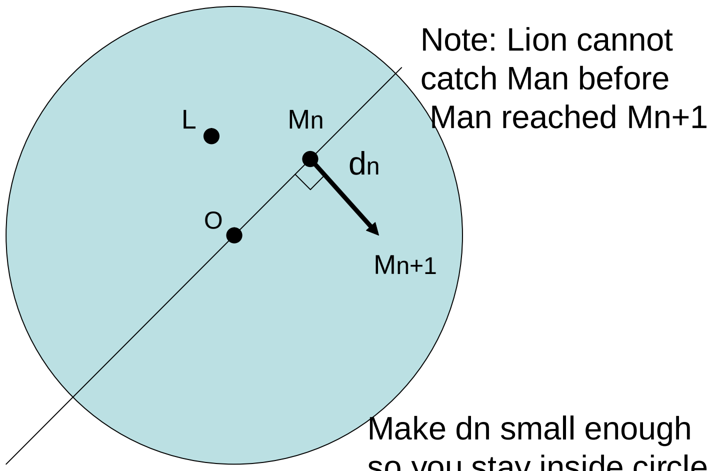
The Lion keeps on the radius OM, so he has same θ as Man. Since he has smaller r, he doesn't need all his speed to keep same θ, and can use rest to increase r, until he has same r as Man. Gobble.

Last Slide Was Wrong (Besicovitch)

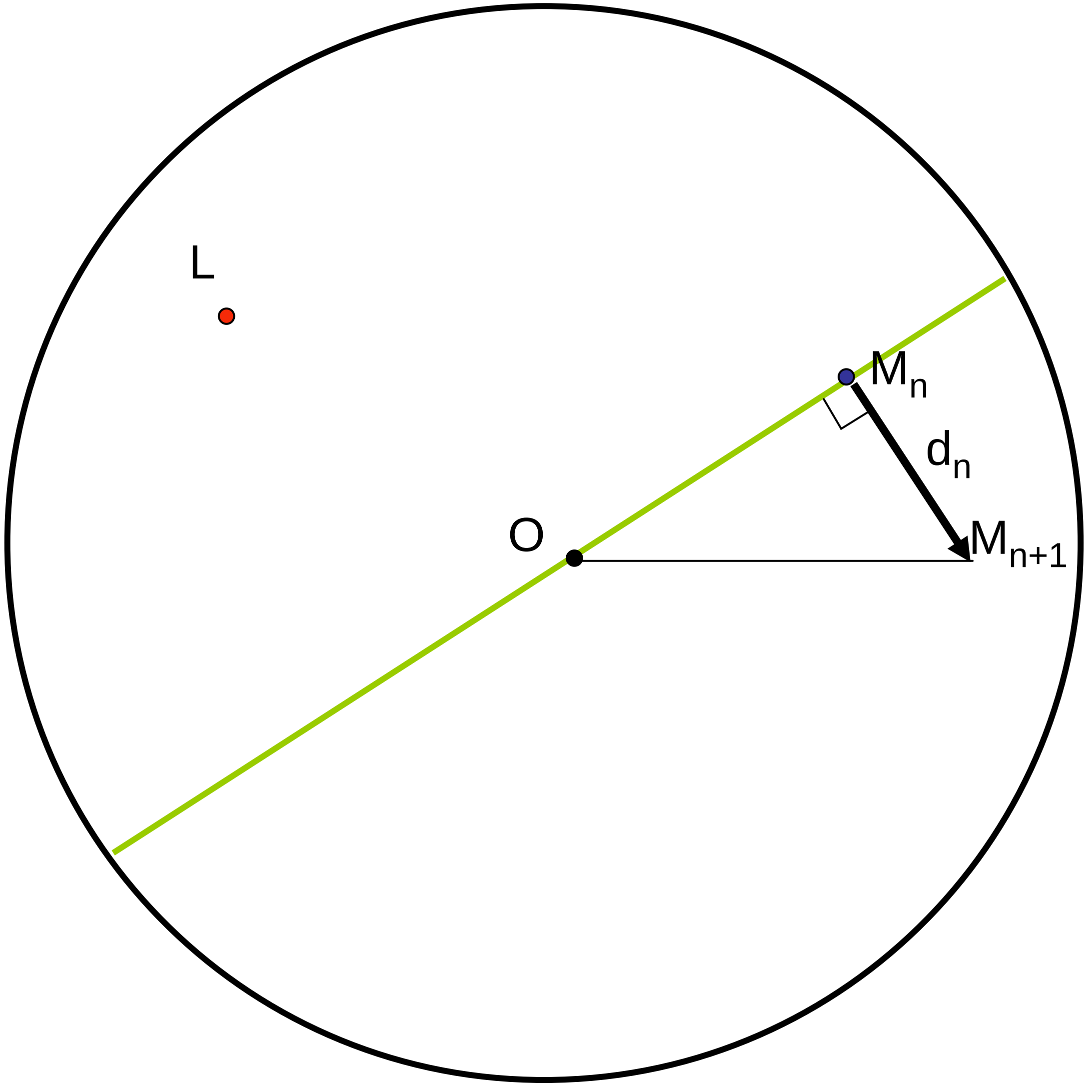
Although it was the ‘known answer for thirty years’, Besicovitch showed it is WRONG (1950). Littlewood published this in his Mathematician’s Miscellany (pp 153-4) (and suggested problem should have been dropped over Germany to waste scientist-hours, as the ’weighing pennies’ wasted them in England.

Lion-Avoiding (just barely) Strategy

Starting at some location M_0 at time t_0, recursively choose M_{n+1}, given M_n as follows. Draw line through 0 (center) and M_n, and move a distance d_n perpendicular to line, in region NOT containing $Lion$.



Man can not be captured during this straight run of length d_n.



$$(OM_{n+1})^2 = (OM_n)^2 + d_n^2, (OM_n)^2 \leq (OM_0)^2 + \Sigma_n d_n^2$$

Achilles and Tortoise Problem?

Not if we pick the d_n correctly. To stay inside the circle note

$$\begin{aligned} (0M_{n+1})^2 &= (0M_n)^2 + d_n^2, (0M_n)^2 \leq (0M_0)^2 + \Sigma_n d_n^2 \\ \Sigma_n d_n^2 &< c \text{ keeps } M_n \text{ insides circle.} \end{aligned}$$

To avoid 'Achilles' need $\Sigma_n d_n = \infty$. So need divergent but square-convergent distances, e.g., c/n.

Unfortunately Lion gets arbitrarily close to Man.

Non-game Search Theory Problem

Alpern and Howard, 'Alternating Search at Two Locations', *Dynamics and Control* (2000)

Problem: A searcher has to find an object which is hidden either at location 1 or 2. At location 1 it is hidden in box $i = 1, \ldots m$ with probability p_i and at location 2 in box $j = 1, \ldots, n$ with probability q_j. The probabilities add up to 1, that is

$$\sum_{i=1}^{m} p_i + \sum_{j=1}^{n} q_j = 1.$$

At each location the boxes must be searched in order, that is, you can't search box $i = 4$ until box $i = 3$ has been searched. How to find object in least expected time?

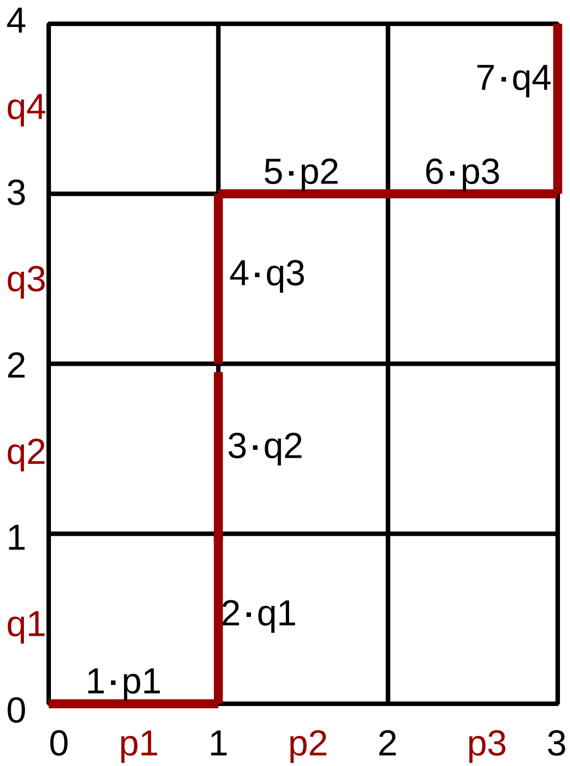
In each time period $t = 1, \ldots, m + n$ the next box at location 1 or 2 can be searched. Suppose $m = 3$ and $n = 4$. A search strategy is a sequence $a_t \in A = \{1, 2\}$ which indicates that location a_t is to be searched at time t. If $(a_1, \ldots, a_7)$ is $(1, 2, 2, 2, 1, 1, 2)$ then the expected time to find the object is

$$E = 1 \cdot p_1 + 2 \cdot q_1 + 3 \cdot q_2 + 4 \cdot q_3 + 5 \cdot p_2 + 6 \cdot p_3 + 7 \cdot q_4$$

What strategy $(a_1, \ldots, a_7)$ minimizes E? A *state* (i, j) indicates that i boxes at location 1 and j boxes at location 2 have been searched (at the beginning of period $t = i + j + 1$).

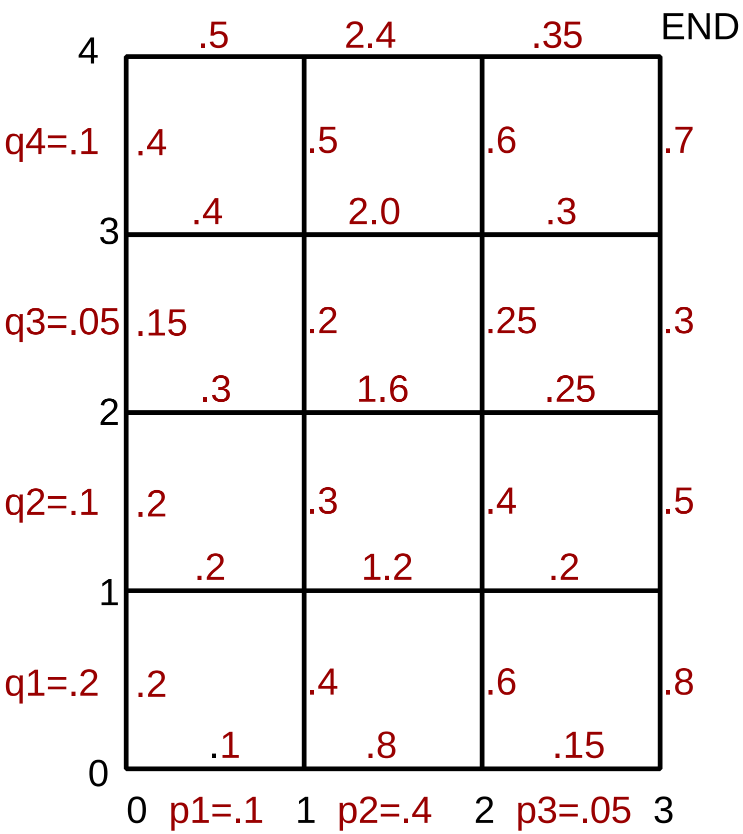
Dynamic Programming Formulation

$$
\begin{aligned}
S &= \{(i,j) : 0 \le i \le 3, 0 \le j \le 4\} \text{ state space, } (0,0) \text{ initial} \\
A &= \{1,2\} \text{ action space} \\
f\left((i,j), a\right) &= \begin{cases} (i+1, j) & \text{if } a = 1 \\ (i, j+1) & \text{if } a = 2 \end{cases} \quad \text{transition function} \\
r\left((i,j), 1\right) &= (i+j+1)\, p_{i+1} \text{ reward if 1} \\
r\left((i,j), 2\right) &= (i+j+1)\, q_{j+1} \text{ reward if 2}
\end{aligned}
$$

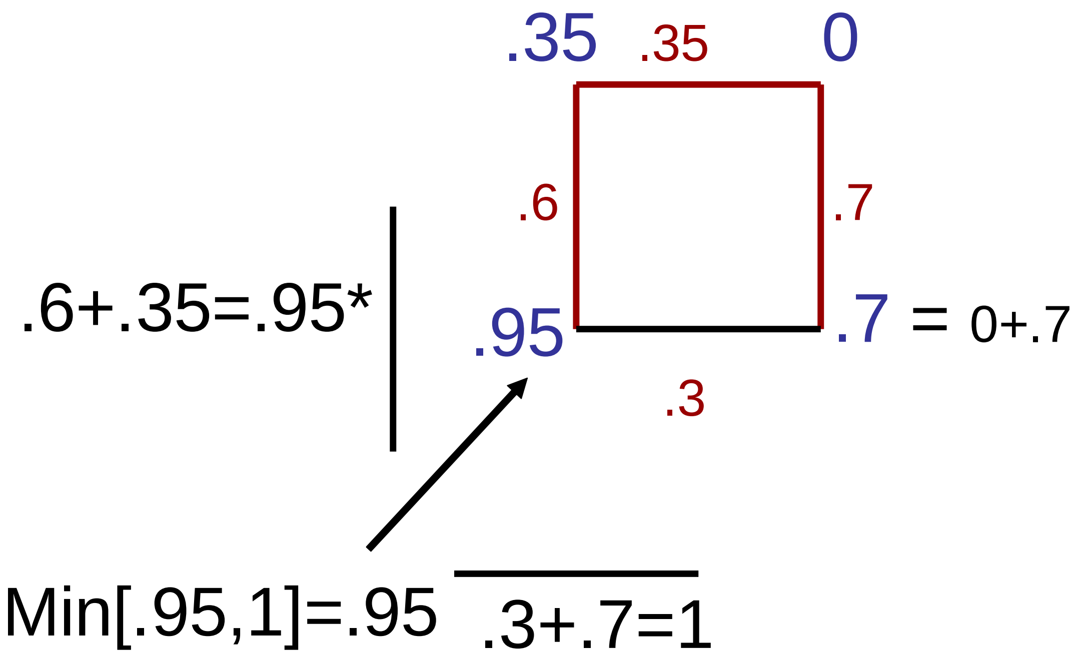


if $a=(1,2,2,2,1,1,2)$

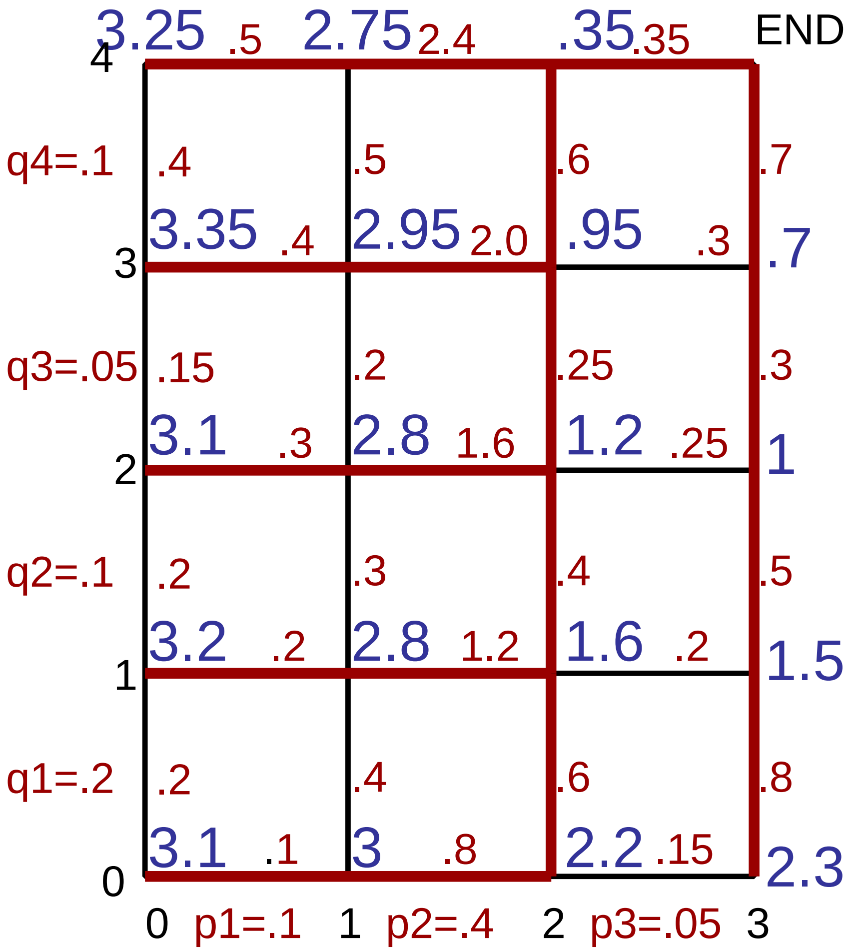
$$E = 1\cdot p1 + 2\cdot q1 + 3\cdot q2 + 4\cdot q3 + 5\cdot p2 + 6\cdot p3 + 7\cdot q4$$



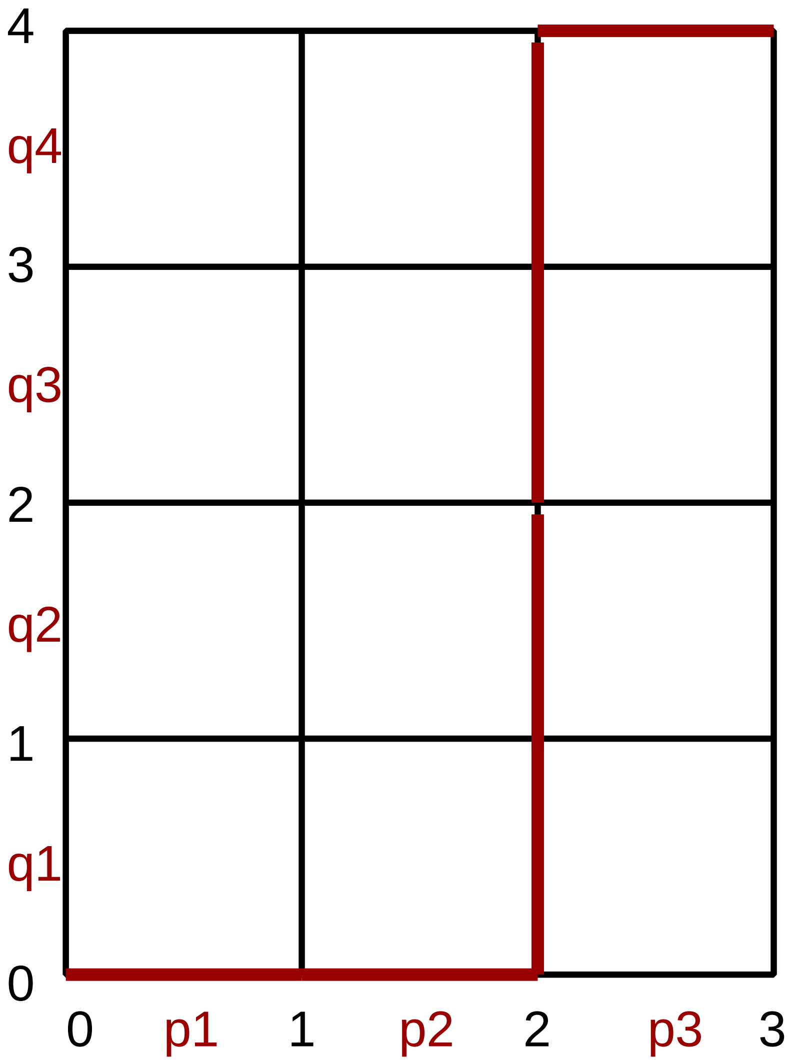
Find ‘shortest’ path from 0 to END, one with smallest sum of edge numbers.



Backwards recursion, or Bellman Eq



Find 'shortest' path from 0 to END, one with smallest sum of edge numbers.



The optimal
Strategy is
(1,1,2,2,2,2,1)

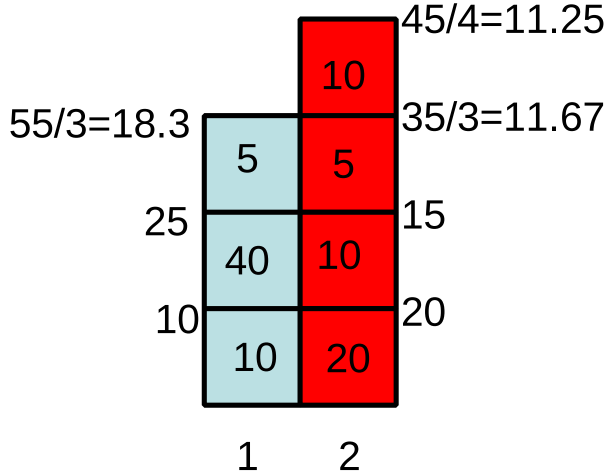
A Forward Looking Method

The dynamic programming method works fine, but to find the first place to search you have to work backwards from the end. A better rule is this:

Start with the initial block of cells at either location that has the highest density (total probability/number of cells). This is essentially a greedy algorithm, but with blocks rather than cells.

Then take successive blocks in order of decreasing density for each location.

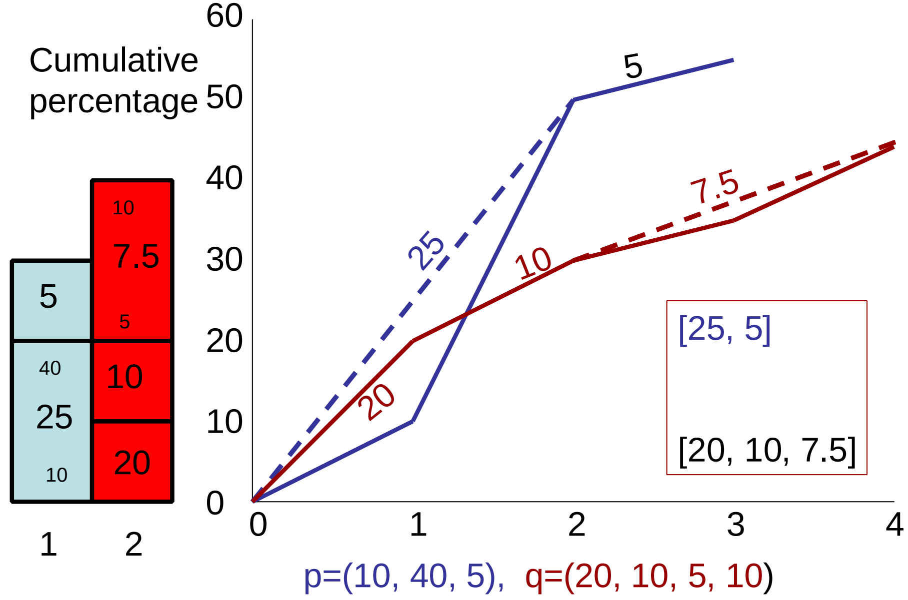
There are seven initial blocks to consider,

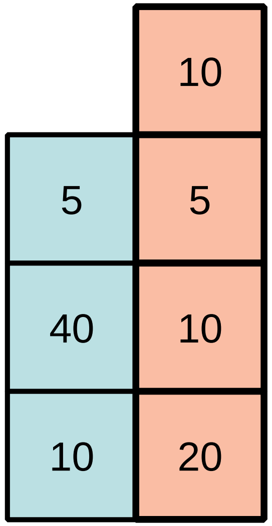


Of these, the first two cells of location 1 has highest density, 25

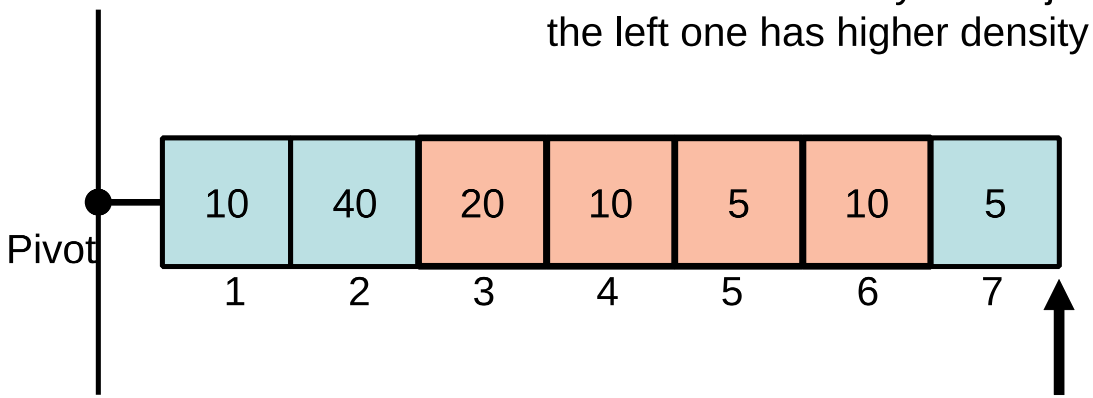
Graphical Method Gives Full Optimal Strategy

Rule: Begin with initial block with highest density, first two cells of 1



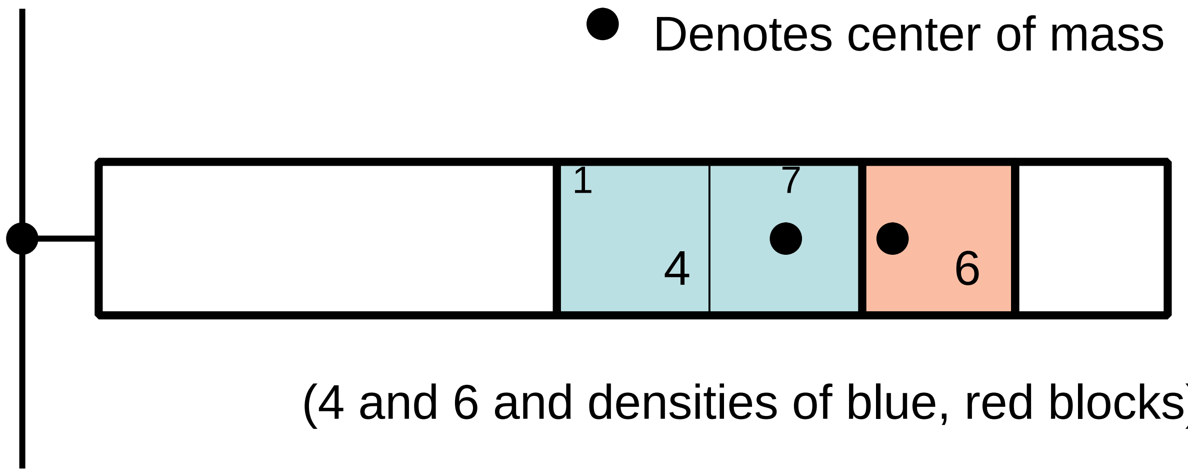


Observe that for any two adjacent ‘blocks’ of different color the left one has higher density (total weight / # boxes)

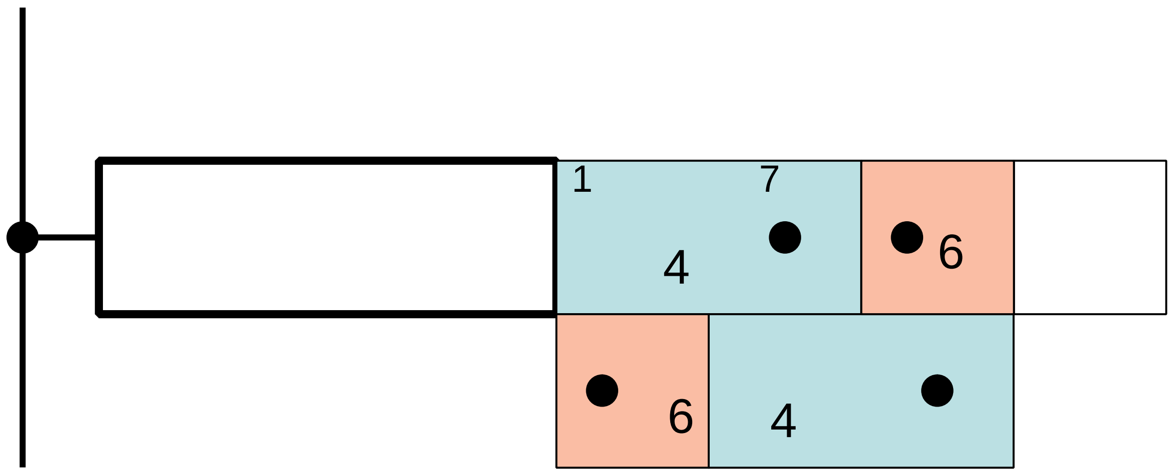


Problem equivalent to minimizing this force, called the ‘moment’

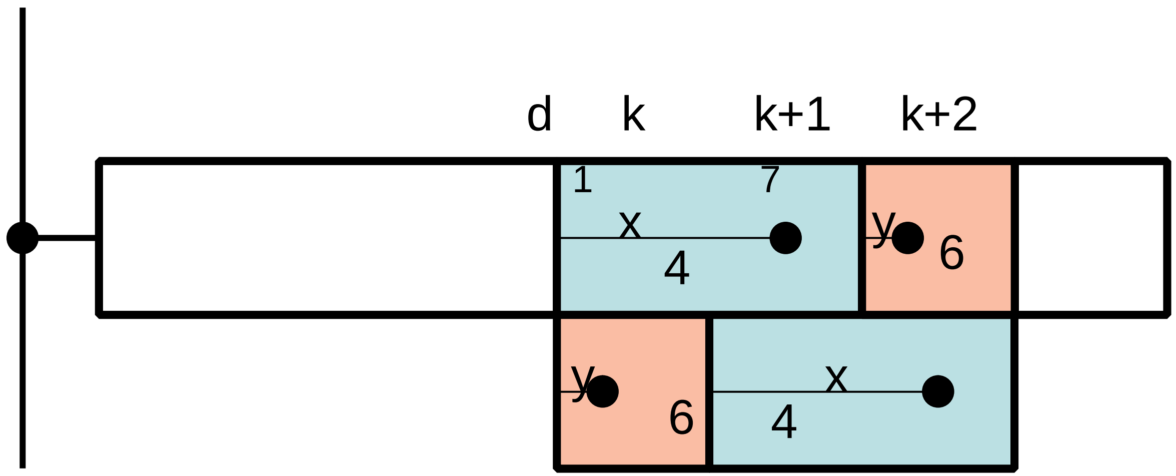
Suppose different color blocks in 'wrong' order



(4 and 6 and densities of blue, red blocks)



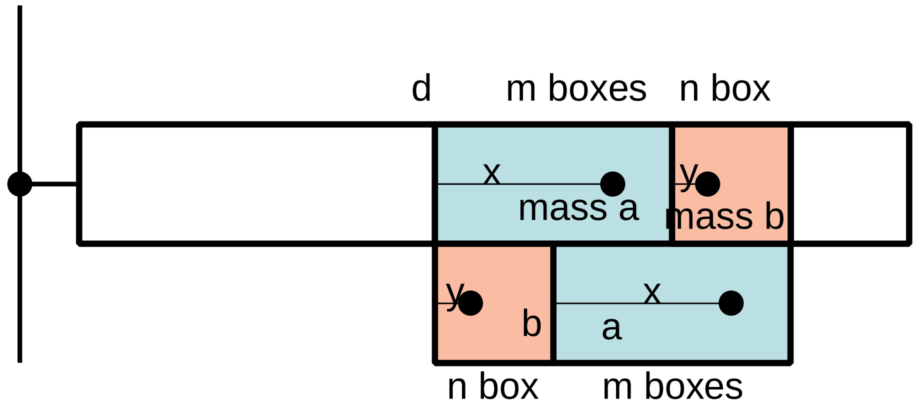
Claim: lower scheme makes smaller contribution to sum



Claim: lower scheme makes smaller contribution to sum

Suppose box centers are a distance k, $k+1$, $k+2$. Let masses be $a = 8$ (left), $b = 6$ (right) sizes $m = 2$ (left), $n = 1$ (right)

top $k\,(1){+}(k+1)\,7{+}(k+2)\,6 = 14k{+}19$, bottom$k\,(6){+}(k+1)\,1{+}(k+2)\,7 = 14k{+}15$



$(d+x)\,a + (d+m+y)\,b$ top; bottom $(d+y)\,b + (d+n+x)\,a$

top-bottom$= (d+x)\,a + (d+m+y)\,b - ((d+y)\,b + (d+n+x)\,a) = bm - an$

$sign\,(bm-an) = sign\left(\frac{bm-an}{mn}\right) = sign\left(\frac{b}{n} - \frac{a}{n}\right) = sign\,(\delta$ red -δ blue)

Bottom smaller if δ =density of red$>$ density blue - put higher δ first!

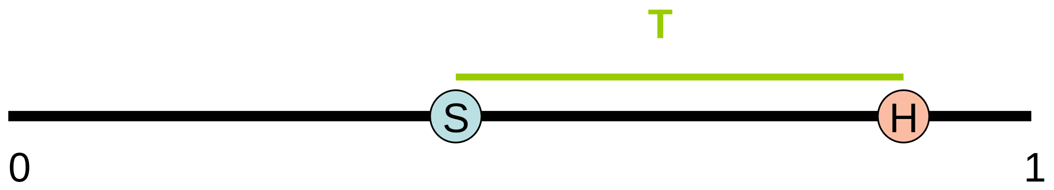
‘Continuous’ Zero-Sum Games

Let $\mathcal{S}$ and $\mathcal{H}$ be two compact sets (e.g. closed intervals) and $T : \mathcal{S} \times \mathcal{H} \to R$ be continuous. $\mathcal{S}$ and $\mathcal{H}$ are the pure strategy sets for minimizer (I, Searcher) and maximizer (II, Hider) and $T(S, H)$ is the Payoff function, capture time if $S(t)$ is pure search strategy and H is pure hider strategy. There is a number V called the Value of the Game, a mixed strategy s (probability distribution over $\mathcal{S}$) for the Searcher and a mixed strategy h (probability distribution over $\mathcal{H}$) for the Hider, such that

$$\begin{aligned} E[T(s, H)] &\leq V \text{ for any Hider pure strategy } H \\ E[T(S, h)] &\geq V \text{ for any Searcher pure strategy } S \end{aligned}$$

These s and h are called optimal - they guarantee at worst the Value. We use upper case letters for pure strategies, lower case for mixed strategies.

Example: $S = H = [0,1]\,,\; T\,(S,H) = |S - H|\,.$



Example: $\mathcal{S} = \mathcal{H} = [0, 1]$, $T(S, H) = |S - H|$.

Suppose Searcher (minimizer) adopts pure strategy $S_1 = 1/2$. Then

$$T(S_1, H) = |1/2 - H| \leq 1/2 \text{ for all } H \in \mathcal{H}.$$

Suppose Hider (maximizer) adopts mixed strategy h_1 of 0 or 1 equiprobably. Treat T as an expected value if one or both of its arguments are mixed.

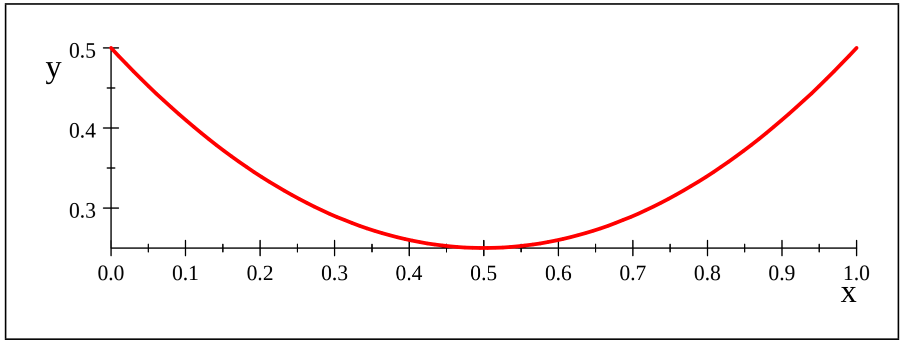
$$T(S, h) = \frac{1}{2}(S - 0) + \frac{1}{2}(1 - S) = \frac{1}{2} \text{ for all } S \in \mathcal{S}.$$

Thus the Value is $V = 1/2$. The Searcher has an optimal strategy which is pure. The Hider has an optimal strategy which guarantees the payoff is exactly the Value. Can you think of other optimal strategies?

Example: $\mathcal{S} = \mathcal{H} = [0,1]$, $T(S,H) = |S-H|$.

What if the Searcher adopts the uniform distribution on $[0,1]$ called s_1. This is also optimal:

$$\begin{aligned} T(s_1,H) &= \int_0^1 |x-H| \ dx = \int_0^H (H-x) \ dx + \int_H^1 (x-H) \ dx \\ &= \frac{1}{2}(1-H)^2 + \frac{1}{2}H^2 = H^2 - H + \frac{1}{2} \leq 1/2 \text{ for all } H \in \mathcal{H}. \end{aligned}$$



What is the set of all optimal Searcher strategies?

Let Q be an interval of length $\mu = a_1 + a_2$ as shown below

Only the terminal nodes v_1, v_2 are undominated by for the Hider. So assume his mixed strategy has $h(v_1) = p_1$ and $h(v_2) = p_2$. The Searcher has two 'searches' which can be done in either order, $A_1 = Ov_1O$ and $A_2 = Ov_2O$. We showed earlier that he should adopt the higher density (probability found/time) one first. That is, A_1A_2 if $p_1/(2a_1) > \ p_2(2a_2)$. If the hider makes both densities equal (so $p_i = a_i/(a_1 + a_2)$, then A_1A_2 and A_2A_1 are equally good and the expected payoff is (using A_1A_2)

$$T = \frac{a_1}{a_1 + a_2}(a_1) + \frac{a_2}{a_1 + a_2}(2a_1 + a_2) = a_1 + a_2 = \mu = V$$