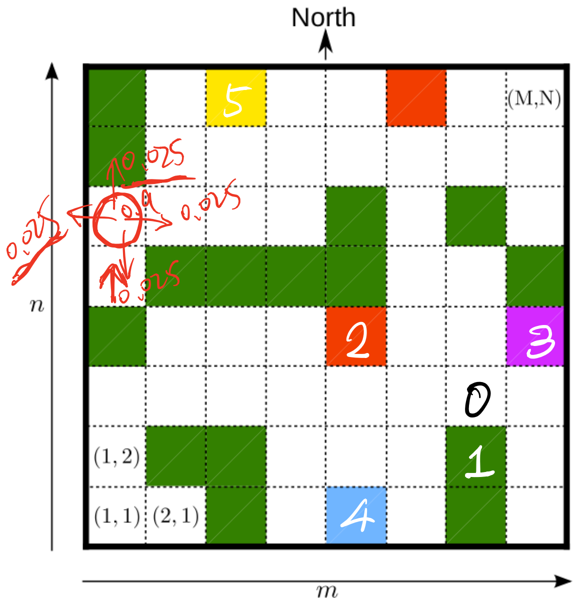
① Problem Setup

map = double m×n



mapSize = [,]

random world or not

plot : policy / cost = T/F

GAMMA R Nc P_WIND

FREE TREE SHOOTER PICK_UP

DROP_OFF BASE

input = { ↑ ↓ → ← ⊗ }

	NORTH	SOUTH	EAST	WEST	HOVER
input	↑	↓	→	←	⊗
	1	2	3	4	5

stateSpace : K×1 Matrix

index			
1	1	1	0
2	1	1	1
3	1	2	0
4	1	2	1
⋮	⋮		
	m	n	0
K	m	n	1

⚠ not continuous in N^*, because no "TREE"

⚠ Some important states:

① terminal state : $(m_{drop}\ n_{drop}\ 1)$

② base state after crash : $(m_b\ n_b\ 0)$

③ pick up state : $(m_p, n_p, 0)$

⚠ index ⟷ (m, n, p)

How to debug : switch xxx implemented from "false" to "true"

② Help functions

<1> bool = invalid_input (m, n, u, map)

→ when position is (m,n), input u is invalid (=1) or valid (=0) depends on boundary and trees

<2> idx = index (m, n, p, stateSpace)

→ find the index of the state(m, n, p) in stateSpace "for"

<3> pr = p_not_shot (m, n, map)

→ return the Pr. of not get shot by residents when position is (m, n)

→ for every residents, Pr = pr × (1 − getshot)

<4> tpm = tempM (m, n, u, map)

→ if u is vaid input for (m, n), return m after u. else return NaN with error displayed

<5> tpn = tempN (m, n, u, map)

→ if u is vaid input for (m, n), return n after u. else return NaN with error displayed

③ Compute Transition Probabilities

$P(i,j,u)$ = zeros(K,K,5): the Pr from state i to state j with input u.

for $i = 1:K$

if i = Terminal then $P(i,i,:) = 1$, $P(i,\neq i,:) = 0$

continue;

end

for $u = 1:5$

if invalid(m,n,u,map) == 1 $P(i,j,u) = NaN$.

continue;

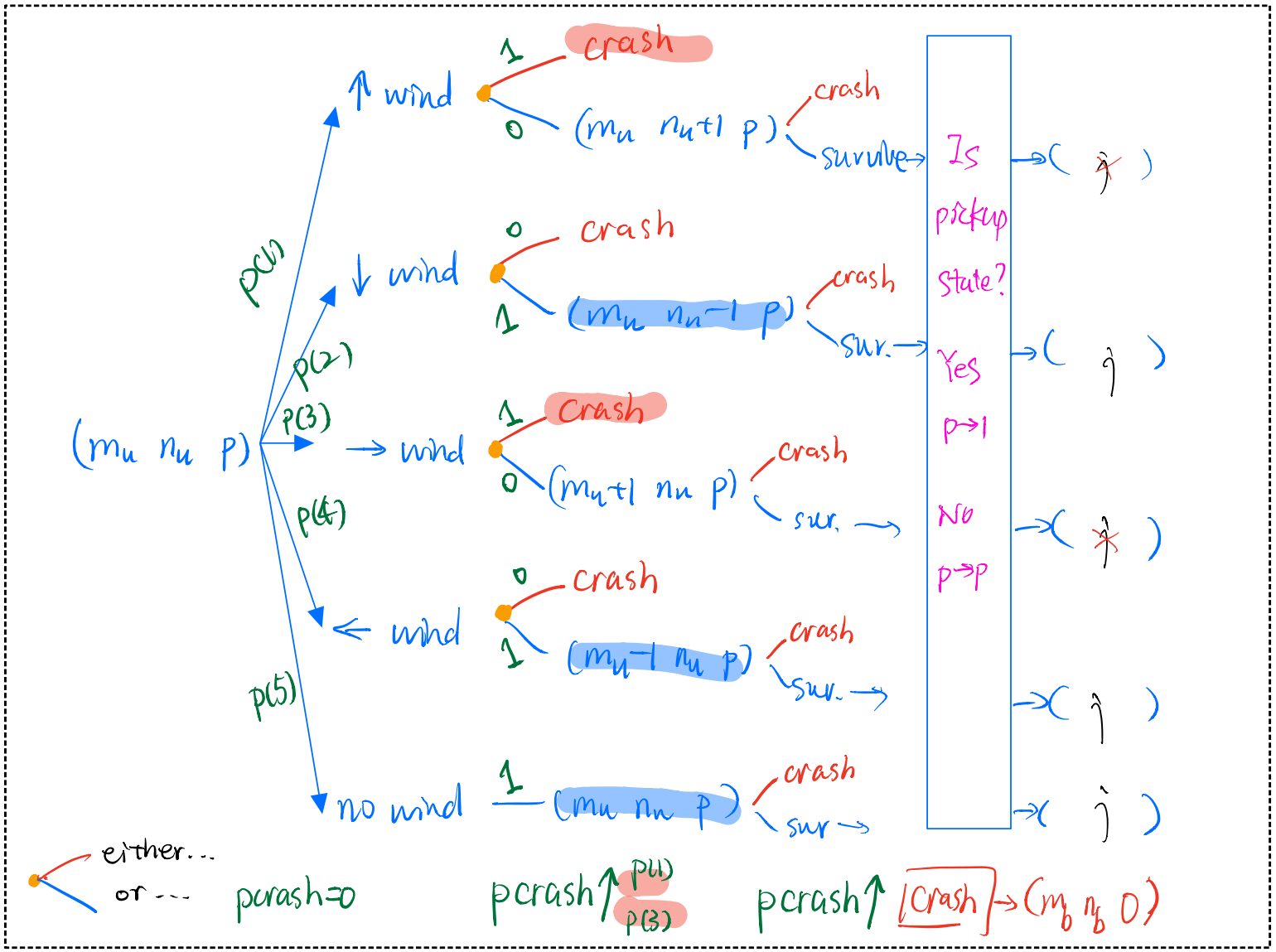
end

$(m,n,p) \xrightarrow{u} (m_u, n_u, p)$

NaN + NaN = NaN

NaN × NaN = NaN

number + NaN = NaN



Check out: $P(i, \cdot, u)$: with State Space / Map

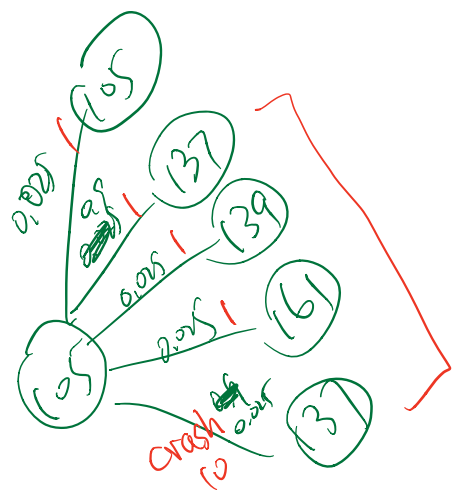
① if u is invalid, $P(i, \cdot, u) = NaN$

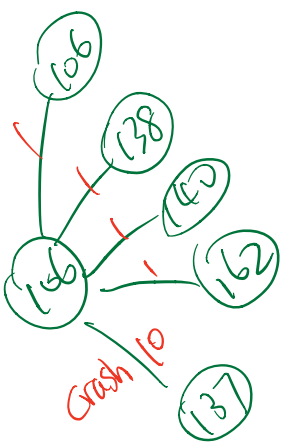
② if no shooter, $P(i, \cdot, u) =$ some 0.025

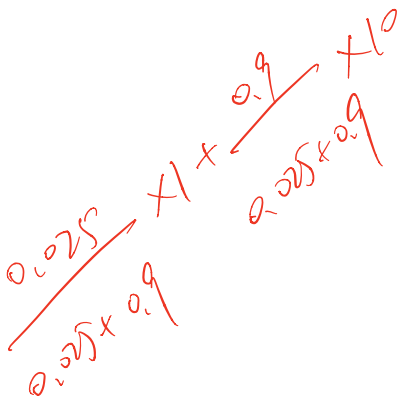
Base 137 ↑

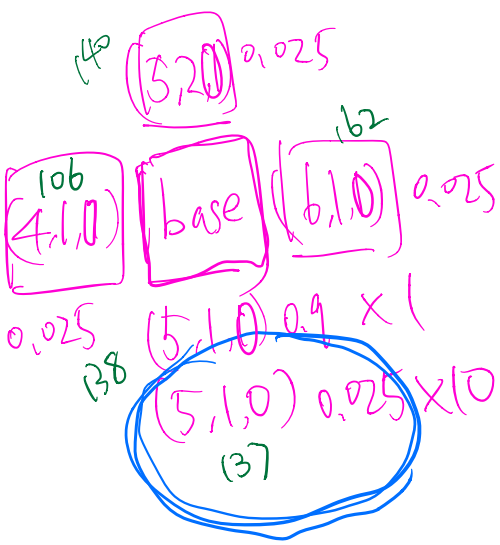
③ if shooter, $P(i, \cdot, u) =$

Debugging...









$n = 4$

$n = 1$

$\frac{0.9+1}{0.025}$

$n_b = 5$

tpn = 5

$n_b = 1$

tpn = 1

④ Compute Stage Costs.

$$G(i,u) = g(i,u) = \underset{(w|x_k=i,u_k=u)}{E}\left[g(x_k,u_k,w)\right]$$

$$= \sum_j p(i,j,u) \cdot \text{cost}(i,j,u)$$

1) *Given any initial conditions* $V_0(1),\ldots,V_0(n)$, *the sequence* $V_l(i)$ *generated by the iteration*

$$V_{l+1}(i) = \min_{u\in\mathcal{U}(i)}\left(q(i,u) + \sum_{j=1}^{n} P_{ij}(u)V_l(j)\right), \quad \forall i \in \mathcal{S}^+ \tag{4.7}$$

where $\mathcal{S}^+ := \mathcal{S}\backslash\{0\}$ *and*

$$q(i,u) := \underset{(w|x=i,u=u)}{E}\left[g(x,u,w)\right],$$

converges to the optimal cost $J^*(i)$ *for all* $i \in \mathcal{S}^+$;

cost(i, j, u, stateSpace, map):

average cost when state (i) $\xrightarrow{u}$ (j) with input u.

⚠ crash ⇒ "j = basestate" but "j = base state" ⇏ crash

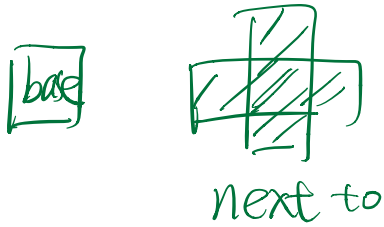
so apply averge cost

Solution:

① cost(i, base, u) = Nc × (1 − p(not crash | i, base, u)) + 1 × p(not crash | i, base, u)

② $p(\text{not crash} \mid i \xrightarrow{u} \text{base}) = \dfrac{p(\text{not crash} \;\&\; i \xrightarrow{u} \text{base})}{p(i \xrightarrow{u} \text{base})}$

$$= \begin{cases} p(\text{good wind}) \times p(\text{not get shot}) \;/\; p(i,\text{base},u) & \text{if "next to"} \\ 0 & \text{if not "next to"} \end{cases}$$



next to base.

⑤ Value Iteration

1) *Given any initial conditions* $V_0(1), \ldots, V_0(n)$, *the sequence* $V_l(i)$ *generated by the iteration*

$$V_{l+1}(i) = \min_{\mathrm{u} \in \mathcal{U}(i)} \left(q(i, \mathrm{u}) + \sum_{j=1}^{n} P_{ij}(\mathrm{u}) V_l(j) \right), \quad \forall\, i \in \mathcal{S}^+ \tag{4.7}$$

where $\mathcal{S}^+ := \mathcal{S} \backslash \{0\}$ *and*

$$q(i, \mathrm{u}) := \mathop{\mathrm{E}}_{(w | x=i, u=\mathrm{u})} [g(x, u, w)],$$

converges to the optimal cost $J^*(i)$ *for all* $i \in \mathcal{S}^+$;

2) *The optimal costs satisfy the Bellman Equation* (BE)*:*

$$J^*(i) = \min_{\mathrm{u} \in \mathcal{U}(i)} \left(q(i, \mathrm{u}) + \sum_{j=1}^{n} P_{ij}(\mathrm{u}) J^*(j) \right) \quad \forall\, i \in \mathcal{S}^+;$$

3) *The solution to the BE is unique;*

4) *The minimizing* u *for each* $i \in \mathcal{S}^+$ *of the BE gives an optimal policy, which is proper.*

stateSpace : K×1

index

1	1	1	0
2	1	1	1
3	1	2	0
4	1	2	1
⋮	⋮	⋮	⋮
T			terminal state {0}
⋮	⋮	⋮	⋮
m	n	0	
K	m	n	1

$i \in \mathcal{S}^+ = \mathcal{S} / \{0\}$.

for iterator = 1 : K

if itor = T , continue

V(itor) = min{ _ _ _ _ }

M = [_ Inf _ _ _]

if not invalid u

Question ?

How to speed up VI ?

⑥ Policy Iteration

Policy Iteration

Initialization: Initialize with a proper policy $\mu^0 \in \Pi$.

Stage 1 (*Policy Evaluation*): Given a policy μ^h, solve for the corresponding cost J_{μ^h} by solving the linear system of equations

$$J_{\mu^h}(i) = q\Big(i, \mu^h(i)\Big) + \sum_{j=1}^{n} P_{ij}(\mu^h(i)) J_{\mu^h}(j)\,, \ \forall i \in \mathcal{S}^+. \tag{5.3}$$

$J = Q + P \cdot J$

Stage 2 (*Policy Improvement*): Obtain a new stationary policy μ^{h+1} as follows

$$\mu^{h+1}(i) = \arg\min_{u \in \mathcal{U}(i)} \left(q(i,u) + \sum_{j=1}^{n} P_{ij}(u) J_{\mu^h}(j) \right), \ \forall i \in \mathcal{S}^+. \tag{5.4}$$

Iterate between Stage 1 and 2 until $J_{\mu^{h+1}}(i) = J_{\mu^h}(i)$ for all $i \in \mathcal{S}^+$.

$\mu_0 = 5 \times \text{ones}(K,1)$ (Assumption: μ is valid)

$$J_{\mu^h} = \begin{bmatrix} J_{\mu^h}(1) \\ J_{\mu^h}(2) \\ \vdots \\ J_{\mu^h}(T) = 0 \\ \vdots \\ J_{\mu^h}(K) \end{bmatrix} \qquad Q_{\mu^h} = \begin{bmatrix} q(1, \mu^h(1)) \\ q(2, \mu^h(2)) \\ \vdots \\ q(T, \mu^h(T)) = 0 \\ \vdots \\ q(K, \mu^h(K)) \end{bmatrix}$$

stateSpace : $K \times 1$

index			
1	1	1	0
2	1	1	1
3	1	2	0
4	1	2	1
⋮	⋮	⋮	⋮
T			
⋮	⋮	⋮	⋮
	m	n	0
K	m	n	1

terminal state $\{0\}$

$$P = \begin{bmatrix} \vdots \\ \cdots \ P(i,j,\mu^h(i)) \ \cdots \\ \vdots \end{bmatrix}$$

$i \in \mathcal{S}^+ = \mathcal{S} / \{0\}$.

⑦ Linear Programming.

Theorem 5.2. *The solution to the optimization problem*

$$\underset{V}{\text{maximize}} \quad \sum_{i \in \mathcal{S}^+} V(i)$$

$$\text{subject to} \quad V(i) \le \left(q(i,u) + \sum_{j=1}^{n} P_{ij}(u) V(j) \right), \quad \forall u \in \mathcal{U}(i),\ \forall i \in \mathcal{S}^+. \tag{5.12}$$

also solves the Bellman Equation to yield the optimal cost J^* for the SSP problem.

maximize $V_1 + V_2 + \cdots + V_{T-1} + V_{T+1} + \cdots + V_K$

$\Leftrightarrow$ minimize $-V_1 - V_2 - \cdots - V_{T-1} - V_{T+1} - \cdots - V_K$

$\Leftrightarrow$ minimize $f^T V$, $f = \begin{bmatrix} -1 \\ -1 \\ \vdots \\ -1 \\ 0 \\ -1 \\ \vdots \\ -1 \end{bmatrix}$ $V = \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_T \\ \vdots \\ V_K \end{bmatrix}$

---

for $i = 1:K$

  if $i = T$ continue.

  for $u = 1:5$

    if isnan(G(i,u)) continue.

$V(T) = 0$.

$\rightarrow$ $V(i) \le q(i,u) + P(i,1,u)V(1) + P(i,2,u)V(2) + \cdots + P(i,i,u)V(i) + \cdots + P(i,T,u)V(T) + \cdots + P(i,K,u)V(K)$

$$\begin{bmatrix} -P(i,1,u) & -P(i,2,u) & \cdots & 1-P(i,i,u) & \cdots & -P(i,K,u) \end{bmatrix} \begin{bmatrix} V(1) \\ V(2) \\ \vdots \\ V(T)=0 \\ V(K) \end{bmatrix} \le q(i,u)$$

476×1

2124×476

optimal policy ? $\mu^*(\cdot)$

2) *The optimal costs satisfy the Bellman Equation* (BE):

$$J^*(i) = \min_{u \in \mathcal{U}(i)} \left(q(i,u) + \sum_{j=1}^{n} P_{ij}(u) J^*(j) \right) \quad \forall i \in \mathcal{S}^+;$$