Parallel MCTS-Pseudo Code

Objects

General objects

Action space A : tuple with action space of a node.

Histogram $\boldsymbol{h}$: a histogram $\boldsymbol{h}$ with n_b bins is an array of int of size n_b and $\boldsymbol{h}[i]$ is the number of elements in the i^{th} bin.

Modification sequence $\boldsymbol{m}$: log generated whenever a WorkerTree expands a new node. It is an array composed of $[id, h, h_p, o, \Delta]$. With id the attribute of the WorkerTree that expanded the new node. h, h_p, o the attributes of the corresponding MasterNode. And Δ the return obtained from expanding this new node.

Action Dict $\boldsymbol{D}$: a dictionary giving for each action the corresponding index : $\{0° : 0, 45° : 1, ..., 315° : 7\}$

Worker objects

WorkerTree $\boldsymbol{\theta}$: a worker tree is a tree operating a MCTS-UCT search for a given weather scenario. It has the following attributes:

- **the root node** $\boldsymbol{\nu_0}$: a reference pointing towards the workernode type root node.
- **a simulatior** S: a simulator with initial state and given weather conditions.
- **the time horizon** T: final time of the simulator (and horizon of the search).
- **estimated time** T_{min}: time estimated at the initialisation of the search to reach the destination given the weather conditions.
- **worker index** id: an int that characterises the weather scenario on which the worker is searching.
- **a buffer** $\boldsymbol{B}$: a chronological list of the modification sequences that have not been transmitted to the master yet.

WorkerNode $\boldsymbol{\nu}$: How nodes are represented in the Workers domain. It has the following attributes:

- **a parent** p: a reference toward the parent node.
- **origins** $\boldsymbol{\omega}$: a list of the actions taken form the root node to get to this node. We call arm the last action taken o.
- **children** $\boldsymbol{c}$: a list of references towards the children nodes.
- **actions** $\boldsymbol{a}$: a list of the remaining available actions.
- **values** $\boldsymbol{Q}$: a array of size len(A) containing Histograms. $\boldsymbol{Q}$[i] is the histogram of the rewards provided by children which origin is A[i].

Master objects

MasterTree $\boldsymbol{\Theta}$: Master tree that manages n_s different WorkerTree working in parallel. Each WorkerTree is searching on a different weather scenario.

- **nodes** $\boldsymbol{N}$: a dictionary with key the hash of a node and value a reference toward the corresponding MasterNode.
- **proba** $\boldsymbol{P}$: a array with the probability of occurrence of each scenario.

MasterNode $\boldsymbol{\mu}$: How nodes are represented in the Master domain. It has the following attributes:

- **a hash** h: hash based on the origins $\boldsymbol{\omega}$ of the corresponding WorkerNode $\boldsymbol{\nu}$.
- **arm** o: action taken from the parent node to extend the present node.
- **parent hash** h_p: hash of the parent.
- **rewards** $\boldsymbol{R}$: an array of size (number of scenario, len(A)) containing Histograms of returns. Thus for each MasterNode $\boldsymbol{\mu}$ we have for each action taken from it and for each scenario a Histogram of all the obtained returns.

Algorithm 1: The UCT algorithm of a WorkerTree

function UCTSEARCH($\boldsymbol{s_0}$)**:**
- create root node ν_0 with state $\boldsymbol{s_0}$
- **while** within computational budget **do**
  - $\nu_l,\ \boldsymbol{s} \leftarrow$ TREEPOLICY($\nu_0,\ \boldsymbol{s_0}$)
  - $\Delta \leftarrow$ DEFAULTPOLICY($\nu_l,\ \boldsymbol{s}$)
  - $\nu.\boldsymbol{Q}[:] \leftarrow \Delta$
  - BACKUP(ν_l, Δ)
  - $h,\ h_p \leftarrow$ GETHASH(ν_l), GETHASH($p(\nu_l)$)
  - $\boldsymbol{m} \leftarrow [id, h, h_p, \boldsymbol{\omega}(\nu_l)[-1], \Delta]$
  - append $\boldsymbol{m}$ to $\boldsymbol{B}$
  - **if** *it is time to feed master* **then**
    - send $\boldsymbol{B}$ to master
    - $\boldsymbol{B} \leftarrow []$

function TREEPOLICY($\nu,\ \boldsymbol{s}$)**:**
- **while** $\boldsymbol{s}$ is nonterminal **do**
  - **if** ν not fully expanded **then**
    - **return** EXPAND($\nu,\ \boldsymbol{s}$)
  - **else**
    - $\nu \leftarrow$ BESTCHILD($\nu, Cp, \rho, \boldsymbol{\Theta}$)
    - $\boldsymbol{s} \leftarrow S(\boldsymbol{s}, a(\nu))$

return $\nu,\ \boldsymbol{s}$

function EXPAND($\nu,\ \boldsymbol{s}$)**:**
- choose random $a \in \boldsymbol{a}(\nu)$ the untried actions of ν
- add a new child ν' to ν with $a(\nu') = a$
- $\boldsymbol{s} \leftarrow S(\boldsymbol{s}, a)$

return ν', $\boldsymbol{s}$

function BESTCHILD(ν,C_p,ρ, U_m)**:**
- $N_\nu \leftarrow$ sum($\nu.p.\boldsymbol{Q}[\boldsymbol{D}[\nu.o]]$)
- $U \leftarrow []$
- **for** ν_c *in* $\nu.\boldsymbol{c}$ **:**
  - $i \leftarrow \boldsymbol{D}[\nu_c.o]$
  - $N_c \leftarrow$ sum($\nu.\boldsymbol{Q}[i]$)
  - $e \leftarrow C_p\sqrt{\frac{2\log(N_\nu)}{N_c}}$
  - $v \leftarrow$ GETVALUEW($\nu_c.\boldsymbol{Q}$)
  - append $\rho(v+e) + (1-\rho)U_m[\nu_c]$ to U
- $\nu_b \leftarrow \nu.\boldsymbol{c}$[argmax($U$)]

return ν_b

function DEFAULTPOLICY($\nu,\ \boldsymbol{s}$)**:**
- **while** $\boldsymbol{s}$ is nonterminal **do**
  - $D, \theta \leftarrow G_1(\boldsymbol{x}, \boldsymbol{x_d})$ with $\boldsymbol{x}$ the position of $\boldsymbol{s}$
  - $\boldsymbol{s} \leftarrow S(\boldsymbol{s}, \theta)$

return reward for state $\boldsymbol{s}$

function BACKUP(ν,Δ)**:**
- **while** $\nu.p$ is not null **do**
  - $\nu.p.\boldsymbol{Q}[\boldsymbol{D}[\nu.o]]$.add($\Delta$)
  - $\nu \leftarrow \nu.p$