

MINI REPORT

Nell'ultima realese di “progetto.nlogo” avevo all'interno del mio codice l'agente componente, il quale dopo l'ultima chiacchera è stato eliminato. Così come il bottone nella GUI che sceglieva il numero di componenti.

Nella mia GUI ora è presente un bottone per la scelta del numero dei Pod da istanziare. Tale numero nel codice viene ripreso come "number".



Quindi richiedi 5 pod, number sarà uguale a 5. Ad ogni pod ho aggiunto i propri attributi (flexM, flex_m, costoM, costo_m).

```
    pods-own [  
      flexM  
      flex_m  
      costoM  
      costo_m  
    ]
```

Fatto questo ho dovuto creare due tipologie di input e quindi come concordato due tipologie di dataset.

Np.array

```
Out[1]: array([[ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15,
 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31,
 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47,
 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63,
 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79,
 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95],
 [ 1, 83, 94, 33, 82, 95,  6, 46, 31, 83, 66, 88, 25, 49, 23, 68,
 73, 40, 65, 89, 40, 65, 96, 77, 39, 90, 11, 76, 46, 30,  2, 60,
 85, 29, 76, 36,  1, 90, 68, 68, 54, 41, 17, 18, 92, 83, 15, 89,
 29, 40, 64, 22, 66, 14, 72, 96, 37, 94,  0, 66, 60, 39, 49, 98,
 25, 43, 51, 86, 22, 10, 37, 62,  0, 42, 45, 90, 23, 68, 22, 28,
 54, 81, 75, 95, 21, 80, 35, 60, 82, 21, 43, 20, 56, 96,  7, 67],
 [ 1, 20, 77, 28, 53, 54, 28, 84, 76, 78, 69, 61, 21, 45, 11, 19,
  9, 39, 77, 37, 99, 57, 11, 35, 34,  6, 87, 41,  0, 90, 68, 65,
 60, 97,  8, 55, 32, 45, 82, 68, 92, 55, 35,  9, 61,  6, 17, 51,
 79, 79, 50, 52, 25, 47, 67, 73, 64, 42, 24, 72, 69, 90, 62, 21,
 91, 69, 63, 18, 26, 28, 60, 42, 61, 13, 34, 36,  1, 67, 29, 87,
  2, 42, 61, 55, 95, 99, 88,  7, 93, 66, 73, 86, 45, 22,  0, 27],
 [ 1, 59, 38, 58, 33, 79, 38, 18, 78, 85, 75, 64, 73, 39, 62, 36,
 36, 37, 60, 41, 17, 92, 92, 68, 81, 46, 50, 17, 73, 23, 55,  1,
 89, 40, 98, 50,  8, 93, 49, 38, 59, 51, 78,  5, 89,  3, 81, 89,
 40, 79, 49, 73, 17, 32, 71, 66, 67, 88, 21, 64, 68, 94, 56, 90,
  8, 43, 18, 23,  1,  0, 99, 74, 32, 16, 32, 98, 35, 10, 71, 67,
  7, 70, 31, 17, 73, 29, 21, 11, 80, 29, 93, 72, 86, 36, 20, 40],
 [ 1, 74, 63, 59, 97,  6, 12, 47, 65, 24, 66, 72, 51, 40, 97,  9,
 15, 13, 65, 66, 12, 65, 46, 65, 44,  0, 23, 44, 65, 26, 12, 23,
  9, 73, 74, 37, 19, 63, 69, 53, 56, 32,  0, 46, 20, 84, 63, 29,
  1, 96, 70, 19, 73, 35,  2, 46, 16,  8, 49, 14, 84, 99, 32, 89,
 32, 50, 17, 73, 29, 39, 58, 33, 69, 63, 27, 37, 45, 92, 71, 50,
 60, 87, 90, 35, 97,  5, 81, 63, 82, 25, 71, 58,  7, 12, 85, 12],
 [ 1, 51, 17, 16, 85, 26, 16, 92, 71, 33, 47, 29, 38, 62, 75, 62,
 64, 91, 84, 89, 87, 76, 97, 89, 81, 77, 60, 46, 43, 39, 63, 57,
 22, 79, 12, 47, 21, 72, 58,  5, 19,  6, 49, 74, 51, 55, 22, 19,
 74,  2, 34, 23, 54, 81, 35, 76, 25, 63, 24, 67, 34, 52, 38, 19,
 81,  8,  9, 90, 97, 49, 66, 29, 37, 17, 59, 73, 84, 33, 69, 36,
 78, 94, 73, 67, 72,  9, 47, 95, 30, 86, 49, 43, 61, 67, 46, 37],
 [ 1,  2, 39, 15, 67, 82, 47, 90, 92, 84, 38, 80, 84, 44, 68, 37,
 13, 81, 24, 77,  8, 25, 74, 68, 54, 75, 99,  1, 43, 36, 81, 72,
 87,  7, 81, 47,  0,  2, 51, 55, 43, 39, 76, 72, 66, 30, 33,  8,
 66, 63, 57, 39, 61, 76, 96, 67, 43,  8, 16,  3, 74, 65, 72, 88,
 46, 99, 17, 57, 33,  3, 53, 60, 38, 96, 31, 67,  0, 38, 60, 63,
 48, 60, 57, 16, 34, 75, 51,  3, 42, 10, 78, 24, 98, 58, 93, 55],
 [ 1, 13,  2, 83, 78, 67, 27, 38, 43, 89, 72, 22, 50, 38, 71, 57,
 28, 87, 86, 78,  7, 90, 10, 96, 55, 78, 39, 98, 10, 11, 61, 46,
 12, 65, 30, 98, 56, 48, 19, 24, 25, 57, 73, 83, 38,  2, 74, 15,
  0, 55, 33, 33, 30, 89, 24, 14, 66, 63, 92, 69,  7, 62, 75, 99,
  4, 70, 11, 51,  4, 36, 94, 44,  5, 35, 53,  2, 72,  8, 40, 95,
 53, 99, 29, 83, 48, 29, 37, 39, 70, 64, 71, 97, 49, 51, 49, 18],
 [ 1, 45, 47, 93, 81, 33, 96, 92, 92, 51, 42, 54, 10, 93, 64, 52,
  1, 67, 87, 33, 87, 37, 13, 70, 16,  8, 62, 20, 55, 64, 29, 92,
 23, 54,  0,  9,  0, 57, 37, 38, 21,  3, 16, 71, 14, 66, 17, 44,
 92, 95, 26, 92, 29, 94,  0, 98, 49, 30,  7, 19, 38, 73, 53, 42,
 34, 30, 22, 19, 49,  5, 69, 78, 85, 23, 97, 81, 62, 72, 23, 34,
 48, 90, 39, 62, 68, 48, 79,  4,  2, 54, 76, 25,  5, 53, 22, 11]])
```

Output.csv

[illegible]

Il primo dataset viene creato con uno script python ("DatasetNumpy.py"). Questo script python è richiamato dalla funzione run-py1 in netlogo attraverso l'estensione "py".

```
extensions [ py
              csv
            ]
```

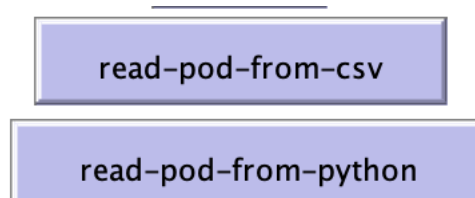
Ciò che sostanzialmente farà questo script è creare una lista da passare a netlogo con all'interno un tot di array moltiplicati per il numero di pod *4 che chiameremo "n" . Il numero di pod viene passato come "nu" (number) da netlogo a python. Infine dopo aver richiamato lo script su netlogo, la funzione "run-py1" ha una lista con all'interno n array che verranno inseriti tramite un while nei pod settando ogni attributo. Sostanzialmente il while itera il numero dei pod e ad ogni iterazione crea un pod e setta gli attributi, iterando anche la lista attraverso delle variabili. Ad ogni iterazione stampo i vari attributi e infine stampo tutta la lista.

Il secondo dataset invece viene creato attraverso la creazione di un file csv con lo script ("provaNumpyCsv.py"). Lo script python all'interno ha una funzione ("creaDataSet()") nel quale creo e apro il file csv, nel mio caso "output.csv", itero il numero dei pod richiesti e ad ogni iterazione vado a creare i 4 attributi del pod come dei numpy array.

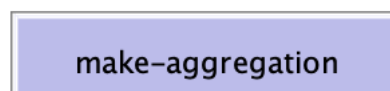
Dopo aver creato questi numpy array li unisco in una numpy detto "tot2" e scrivo sul file csv attraverso la funzione savetxt(...., tot2,....) .

Una volta creato il file csv si passa su netlogo e tramite la funzione read-csv ,con extension csv, leggo tutta la lista la inserisco in "fileList" e allo stesso modo del primo dataset creo Pod e gli attributi del pod.

La scelta dei due dataset può avvenire nella GUI attraverso due bottoni "read-pod-from-python" e "read-pod-from-csv" che rispecchiano rispettivamente i due metodi



Una volta creato il dataset con un qualsiasi metodo visto precedentemente vado a richiamare l'aggregatore in netlogo tramite il bottone "make-aggregation" presente nella GUI,



il quale mi restituisce un agente aggregatore con all'interno 4 attributi "bound_flexM, bound_flex_m , bound_costoM, bound_costo_m".

```
aggregatore-own [
  flex_R      ;; Flessibilità che darò in risposta al mercato
  costo_R     ;; Costo verso il mercato
  bound_flexM  ;; somma delle flex Max degli agenti pod usato poi per il confronto nel simulatore questa deriva dall'aggregatore
  bound_flex_m ;; somma delle flex min degli agenti pod usato poi per il confronto nel simulatore questa deriva dall'aggregatore
  bound_costoM ;; somma dei costi degli Max agenti pod usato poi per il confronto nel simulatore questa deriva dall'aggregatore
  bound_costo_m ;; somma dei costi min degli agenti pod usato poi per il confronto nel simulatore questa deriva dall'aggregatore
]
```

I 4 attributi sono settati tramite la somma
rispettivamente dell'attributo flexM per tutti i Pod, flexm di tutti i Pod, costoM di tutti i pod e
costom di tutti i Pod .

Nella funzione di netlogo "make-aggregation" vado a creare 4 liste

```
let lista_flexM []  
let lista_flex_m []  
let lista_costoM []  
let lista_costo_m []
```

per gli attributi una variabile "temp" in cui vado a sommare tutti gli "n_i" gli elementi
dell'attributo

una variabile "elemento" che va a prendere lo specifico elemento nell'attributo da sommare a
temp.

una variabile "n_i" che itera i 96 elementi dell'attributo.

```
let lista_flexM []  
let lista_flex_m []  
let lista_costoM []  
let lista_costo_m []  
let iterator 0  
let temp 0  
let elemento 0  
let n_i 0
```

La funzione sostanzialmente ha due cicli while innestati nella quale la prima itera i 96 elementi e il
secondo itera tutti i pod.

Questo per ogni attributo. Infine creo l'agente "aggregatore" che avrà all'interno la somma di tutti
gli attributi dei pod al giusto posto.

Infine ho una funzione setup collegata ad un bottone che cancella tutti i vari pod creati.

