Note sull’euristica.

Le due euristiche che verranno implementate si dividono in:

- Intelligenti (ovverosia “con memoria”)
- Stupide (“senza memoria”)

Ulteriormente, possono essere:

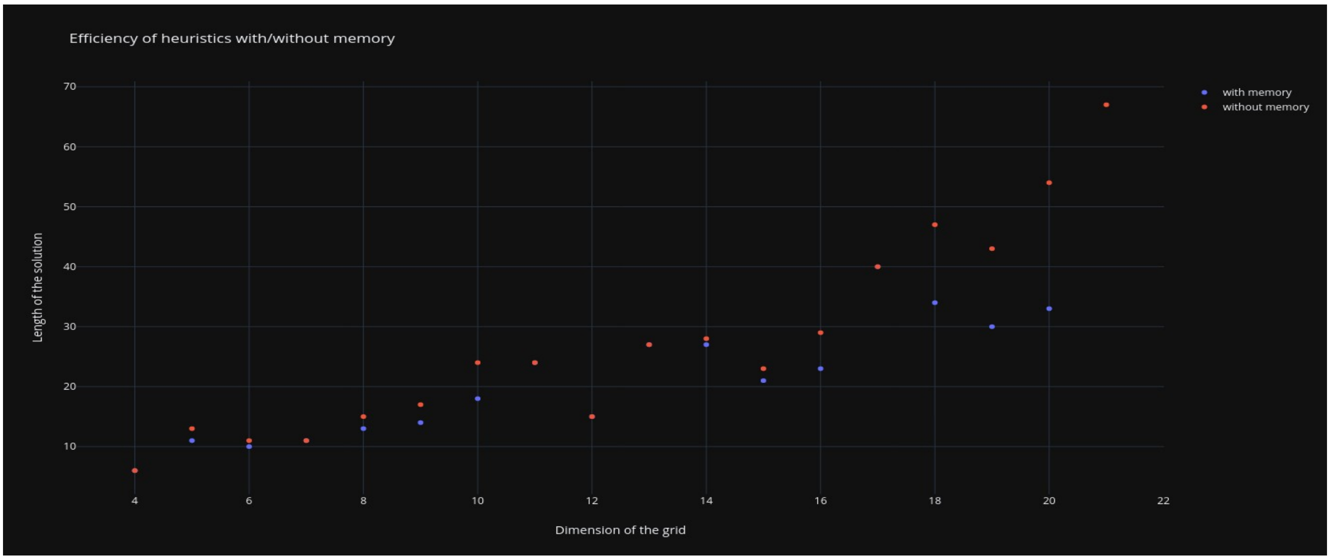
- Non pesate (non tengono conto dei costi del colorare)
- Pesate (tengono conto dei costi del colorare)

Vedremo come funzionano e seguiranno delle note sui bias di ognuna.

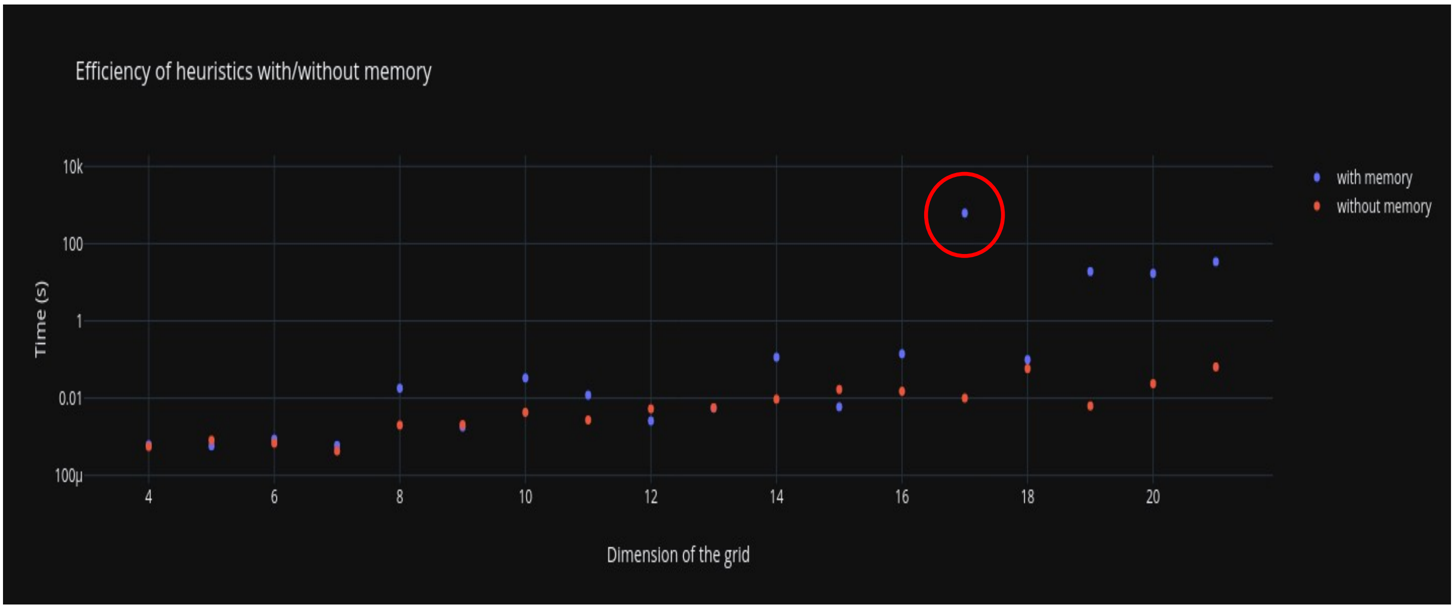
Memoria = con “memoria” intendiamo un qualsiasi tipo d’informazione (di qualsiasi dettaglio o complessità) riguardante le azioni precedenti compiute da un agente nella risoluzione di un problema.
Nel caso del problema *Uniform Coloring*, si intende il costo di un qualsiasi percorso di azioni finora compiute, che denomineremo *path_cost*.

Peso = il peso viene dalla definizione del costo delle azioni, che seguono la definizione del problema.

Prestazioni. Qual'è "migliore"? Questo termine, qualitativo e quindi in buona parte soggettivo, non aiuta. Se ci si riferisce alla sola velocità di trovare una soluzione qualsiasi, allora l'euristica "stupida" vince. Se invece s'intende il costo della soluzione, indipendentemente dal tempo che ci mettiamo a trovarla, l'euristica intelligente ci permette di ottenere la soluzione più ottimizzata, evitando di sprecare mosse non necessarie.



[1]**.Dimension/Length of Solution plot.**



[2]**.Dimension/Time to find the solution plot.** Si nota che questo grafico ha l'asse Y in scala logaritmica. Il risultato evidenziato in rosso ha impiegato l'algoritmo per **600** secondi!

Nota sulle prestazioni Stupido/Intelligente. Il motivo di tale differenza va ricercato nell'influenza che ha l'inclusione dell'informazione aggiuntiva.

Sia l'euristica di un qualsiasi stato s, $\mathbf{h}(s) = h_s$.

Ora sia h_s rispettivamente stupida e intelligente,
$h_s = h_{(s)}$; valore interamente basato solo sulla configurazione della griglia
$h_s = \mathit{path\_cost}_s + h_{(s)}$. Valore limitato dal basso con rispetto alle azioni prese precedentemente.

L'espansione del dato stato *s* in k stati con k = #{azioni}, risulterà nel calcolo di **h**(s_i) per s_i = result(s, {azioni}). Dato che ogni azione può essere benefica, controproducente o inutile avremmo che **h**(s_i) stupida può essere:

inferiore a **h**(*s*)
superiore a **h**(*s*) o
uguale a **h**(*s*).

in questi tre casi, data la natura della frontiera nella ricerca con euristica, la quale dà priorità alla soluzione con euristica minore, l'azione che abbassa l'euristica viene naturalmente prima, seguita poi da quelle che la tengono uguale. Il tutto però avviene senza che l'agente sia informato sull'effettiva qualità dell'azione nel contesto delle precedenti. Dunque, la ricerca è estremamente veloce perché appena un'azione "sembra" buona viene espansa. Ciò porterà inevitabilmente ad una grande presenza di azioni sprecate. Data la buona definizione dei costi e della ricerca, le uniche azioni eseguite in maniera non ottimale sono quelle di movimento.
Più semplicemente, l'agente tenderà a fare decisioni "sicure" che non esplorino in profondità una scelta solo fino a quando non hanno altre azioni "sicure" da fare.

Nel caso dell'euristica intelligente, **h**(s_i) può essere:

superiore o uguale a **h**(*s*).

Dunque, un'azione apparentemente buona non avrà una priorità automatica nella frontiera, ciò porterà ad un espansione molto più ampia delle scelte possibili, le quali verranno esplorate in profondità. L'agente, tenuto a conoscenza tramite *path_cost* delle sue scelte precedenti, potrà farne un'altra solamente s'è certo che sia la più utile per ogni data configurazione, ovverosia se espandendo più e più volte tutte le azioni legali che possa compiere, sia quella che minimizza la somma tra l'euristica della tabella e la somma delle mosse per ogni dato momento. E dunque solo in quel caso venendo effettivamente a trovarsi all'inizio della frontiera.

In breve, l’agente farà le scelte migliori sempre, ma solo dopo averle controllate in profondità.

Nota sui risultati pesata/non pesata.
Sia il problema generalizzato per *c* colori, in una griglia di *k* dimensioni colorata seguendo una probabilità uniforme.
Una qualsiasi cella M_i ha 1/*c* probabilità di essere un dato colore.
Il costo di una soluzione che colori tutta la griglia è data dalla somma:
#{celle colorate <u>non</u> di un colore *c*}*costo(c) + percorso che passi per queste celle.
Data la probabilità 1/*c* e *k* celle, il numero di celle che non sono colorate di $c = k - k/c = (c-1)*k/c$.
Il percorso più breve che passi per $(c-1)*k/c$ celle è lungo al minimo $(c-1)*k/c$.
Dunque, **PATH**(*k*, *c*) = $(c-1)*k/c + (c-1)*k/c + a$. con a, numero di mosse della testina per tornare alla posizione di partenza.
Per *k* abbastanza grandi, **PATH**(*k*, *c*) $\sim= 2*(c-1)*k/c = 2*A$
Quindi, il costo della soluzione è: A*costo(c) + A, con A costante.
Nel nostro caso di c = {1, 2, 3}, colorare, anche una grande porzione della nostra griglia, di **blu** sembra in media essere la nostra scelta più economica.
Ciò conferma i dati sperimentali, per i quali l’euristica non pesata tende a distribuire una scelta di colore equamente tra il blu, il giallo ed il verde mentre quella pesata favorisce ampiamente la soluzione del blu.
Il limite oltre il quale queste due soluzioni differiscono, e quella pesata risulta quella giusta, è se un colore *c*, che non sia il meno costoso, fosse **costo(c)** volte più presente rispetto agli altri.
Nel nostro caso, se il colore verde sia 3 volte più presente rispetto al blu o al giallo, o il giallo se sia 2 volte più presente rispetto al blu o al verde.