Università degli Studi di Bari "Aldo Moro"

Dipartimento di Informatica

Corso di laurea in Informatica Magistrale

Ingegneria della Conoscenza e Intelligenza nelle Macchine

Anno accademico 2016 / 2017

Sperimentazione di una guida genetica per l'algoritmo A*

Insegnamento: Intelligenza Computazionale (6 CFU)

Docente: Dott. Corrado Mencar

Studente: Vito Cuccovillo – matricola: 634488

Sommario

Introduzione

Nel processo di individuazione di una soluzione ad un problema si possono impiegare differenti tecniche, ognuna delle quali può portare o meno ad una soluzione ottimale. In questa sperimentazione si analizza l'algoritmo A*, che consente di individuare una soluzione ottima ad un determinato problema. Tale algoritmo, basandosi su una ricerca informata su un grafo, sceglie il percorso solutivo in base alla "bontà" degli stati in cui si trova. In altri termini preferisce esplorare per primi i percorsi più promettenti, che probabilmente consentiranno di giungere alla soluzione.

Un algoritmo genetico trae ispirazione dall'evoluzione e selezione biologica. Il loro principio consiste nel partire da alcune soluzioni di partenza, ricombinarle, perturbarle e selezionarne quelle più promettenti in modo da generarne sempre di migliori.

La sperimentazione corrente fonde questi due approcci in modo tale da fornire una linea guida all'algoritmo A* laddove il suo criterio di preferenza non riesca a discriminare uno stato migliore da espandere rispetto ad altri. Dunque la soluzione di un algoritmo genetico fungerà da filo conduttore per sostenere A* cercando di indirizzarlo verso un percorso solutivo laddove ci fossero situazioni di parità.

L'obiettivo finale di questo esperimento è quello di verificare se l'efficienza generale di A* può essere statisticamente migliorata con l'intervento di un algoritmo genetico.

1. Progettazione

In questo paragrafo si fornisce una descrizione concettuale degli elementi che compongono il sistema, unita ad una spiegazione di come l'algoritmo A* funziona a seguito dell'adozione di una coda con priorità con diversi livelli.

1.1. Descrizione del sistema

Da un punto di vista concettuale il sistema è organizzato in diversi settori, ognuno dei quali ingloba determinate caratteristiche. È necessario dunque fornire una descrizione del modello adottato, al fine di poterne comprendere i flussi.

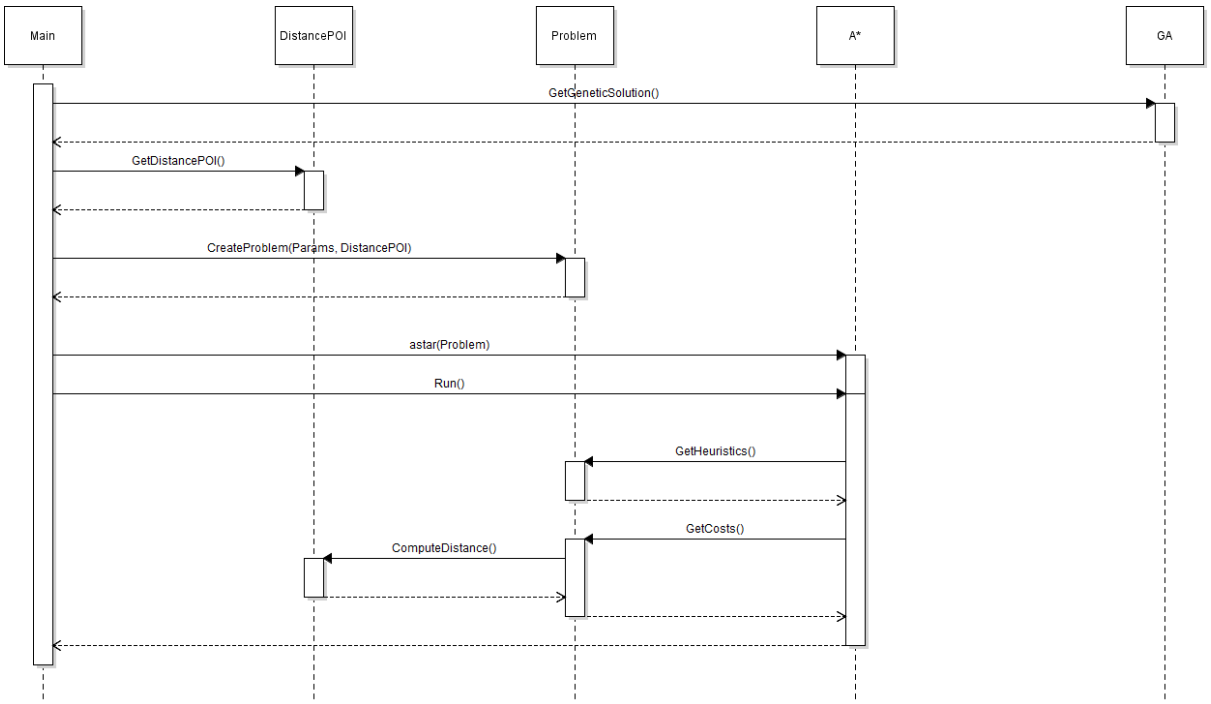


Figure 1 Diagramma di sequenza del flusso eseguito dal sistema

Nel precedente diagramma di sequenza, realizzato con il tool [4], viene fornita una visione complessiva dell'esecuzione tipica. In particolare il modulo "Main" invoca un algoritmo genetico il quale produce una soluzione molto probabilmente subottimale. Tale soluzione viene incapsulata in un modulo "DistancePOI", il cui compito è quello di misurare la distanza fra la soluzione genetica fornita e una qualsiasi altra soluzione. In questo caso il concetto di distanza è abbastanza generico, in quanto è possibile fornire a questo modulo una qualsiasi misura di distanza, che però possa riflettere l'effettiva distanza fra le due soluzioni. In seguito, il modulo "Problem" contiene la logica e le specifiche del particolare problema di cui si vuole trovare una soluzione. "Problem" possiede un riferimento al modulo "DistancePOI" che gli consentirà di fornire, oltre alle misure di costo tipiche del problema, anche un costo dato dalla distanza della soluzione corrente da quella genetica generata in precedenza.

A questo punto è possibile avviare la ricerca vera e propria invocando il modulo "A*" il quale partendo da una situazione iniziale, esplorare il grafo degli stati del problema. Per far ciò esso necessita del calcolo della cosiddetta funzione di valutazione euristica $f^*(x) = g^*(x) + h^*(x)$ dove g rappresenta

il costo sostenuto per transitare dallo stato iniziale allo stato corrente *x*, *h* invece il costo stimato per raggiungere uno stato obiettivo partendo dallo stato corrente *x*.

Il modulo "A*" dunque per poter calcolare il valore effettivo della funzione di valutazione euristica richiede al modulo "Problem" gli effettivi costi. "Problem" fornisce ad "A*" una serie di valori euristici corrispondenti ad altrettante funzioni euristiche tipiche del problema, ed una tupla di valori che rappresentano costi, uno dei quali è calcolato dal modulo "DistancePOI". In sintesi, il modulo "A*" potrà servirsi di informazione non soltanto proveniente dal problema in sé, ma avere un'ulteriore stima del costo data dalla distanza fra la soluzione genetica e lo stato corrente dell'esplorazione.

1.2. A* con coda di priorità

La peculiarità di questo approccio, rispetto ad A* tradizionale, consiste nel fatto che tale algoritmo può godere di diversi livelli di priorità tramite i quali poter indirizzare il processo di visita degli stati del problema. Fornendo ad A* non più un singolo valore con cui scegliere la prossima configurazione da generare ma una tupla di valori, è possibile utilizzare ulteriori criteri in caso di situazioni di assoluta parità dei valori della funzione *f** al livello cosiddetto n-esimo, procedendo alla verifica del livello (n+1)-esimo. In sintesi, l'utilizzo di *n* livelli di priorità consente di ridurre al minimo la scelta arbitraria di A* nel caso di situazioni di parità. Nella tabella seguente è mostrato un esempio del criterio di preferenza di A* tradizionale, confrontato con A* dotato di coda con priorità.

OPEN A* tradizionale	**OPEN A* coda priorità**
A – [2]	A – [2,4]
B – [2]	B – [2,5]
AC – [4]	C – [3,6]
…	…

Nel caso tradizionale A* opererebbe una espansione arbitraria, data la parità della funzione *f** riguardo i nodi A e B. Nel caso si abbia una coda con priorità, sul primo livello A e B si trovano in parità, dunque A* esaminerebbe il secondo livello preferendo quindi il nodo A.

1.3. Funzione di distanza

Un aspetto di fondamentale importanza è appunto la funzione di distanza che verrà appunto utilizzata per fornire una discriminante nei livelli successivi. In questo progetto è stata introdotta una funzione che implementa la *distanza di Levenshtein*. Questa distanza misura la differenza fra due stringhe fornite in input in base al numero di:

- sostituzione di caratteri
- aggiunta di caratteri
- rimozione di caratteri

da effettuare sulla seconda stringa al fine di trasformarla nella prima. Nella fattispecie, questa funzione di distanza ha complessità quadratica, dunque è opportuno scegliere le funzioni sia in base alla loro complessità, sia in base al loro grado di rappresentare l'effettiva distanza che intercorre fra le soluzioni.

2. Implementazione

Il sistema descritto in progettazione è stato implementato nel linguaggio Python versione 3.6.2 ed è suddiviso in diversi package in base alle funzionalità.

2.1. Algoritmo genetico

Come già indicato, il sistema si serve di un algoritmo genetico come guida all'esplorazione. Fra le diverse librerie Python esistenti, è stata utilizzata la libreria open source "Inspyred" nella versione 1.0.1.

Al fine di poter avviare l'algoritmo genetico con suddetta libreria è necessario fornire esplicitamente le funzioni di generazione e valutazione degli individui. Queste funzioni, una volta avviato il tutto, verranno richiamate in automatico dal framework di inspyred.

Aspetto fondamentale è quello relativo alla rappresentazione degli individui, in quanto è necessaria una rappresentazione a lunghezza fissa affinché gli operatori genetici di crossover e mutation integrati nella libreria possano funzionare. Nel caso fosse impossibile avere una rappresentazione degli individui a lunghezza fissa, è possibile fornire delle funzioni "custom" di incrocio e mutazione che siano in grado di gestire questa variabilità.

Prima dell'avvio, è possibile impostare delle ulteriori caratteristiche:

- *observer:* consente di mostrare un plot, una lista dettagliata oppure statistiche relative alla computazione genetica in corso. L'utente può scegliere il tipo di output che preferisce
- *terminator:* indica quando l'evoluzione deve terminare, impostando ad esempio dei vincoli di tempo, di fitness media, di non ulteriore miglioramento etc.

La vera e propria computazione genetica si avvia invocando il metodo *evolve()* che accetta in input:

- *generator*: funzione di generazione degli individui
- *evaluator:* funzione di valutazione degli individui (funzione di fitness)
- *max_evaluations:* numero massimo di valutazioni che l'algoritmo deve effettuare
- *num_elites:* numero di migliori individui della popolazione che deve sopravvivere
- *mutation_rate:* la probabilità con la quale si verificano mutazioni, di default 0.1
- *pop_size:* il numero di individui della popolazione
- *crossover_rate:* tasso con il quale è effettuato l'incrocio, di default 1
- *num_crossover_points:* quanti punti di incrocio utilizzare, di default 1

Al termine della computazione, l'algoritmo restituirà un insieme di individui che corrispondono a soluzioni generate, ordinate in base al valore di *fitness* conseguito. Per gli scopi del progetto corrente si sceglie soltanto la prima soluzione, che sarebbe la migliore generata.

2.2. Implementazione del sistema

Al fine di garantire una buona modularizzazione e separazione delle responsabilità, è stato implementato un wrapper per gestire l'algoritmo genetico. Tale wrapper fornisce sostanzialmente un'interfaccia da implementare in base alla specifica libreria di algoritmi genetici utilizzata nel sistema.

L'intero sistema è rappresentato tramite il seguente diagramma delle classi UML, realizzato con il tool [3]:

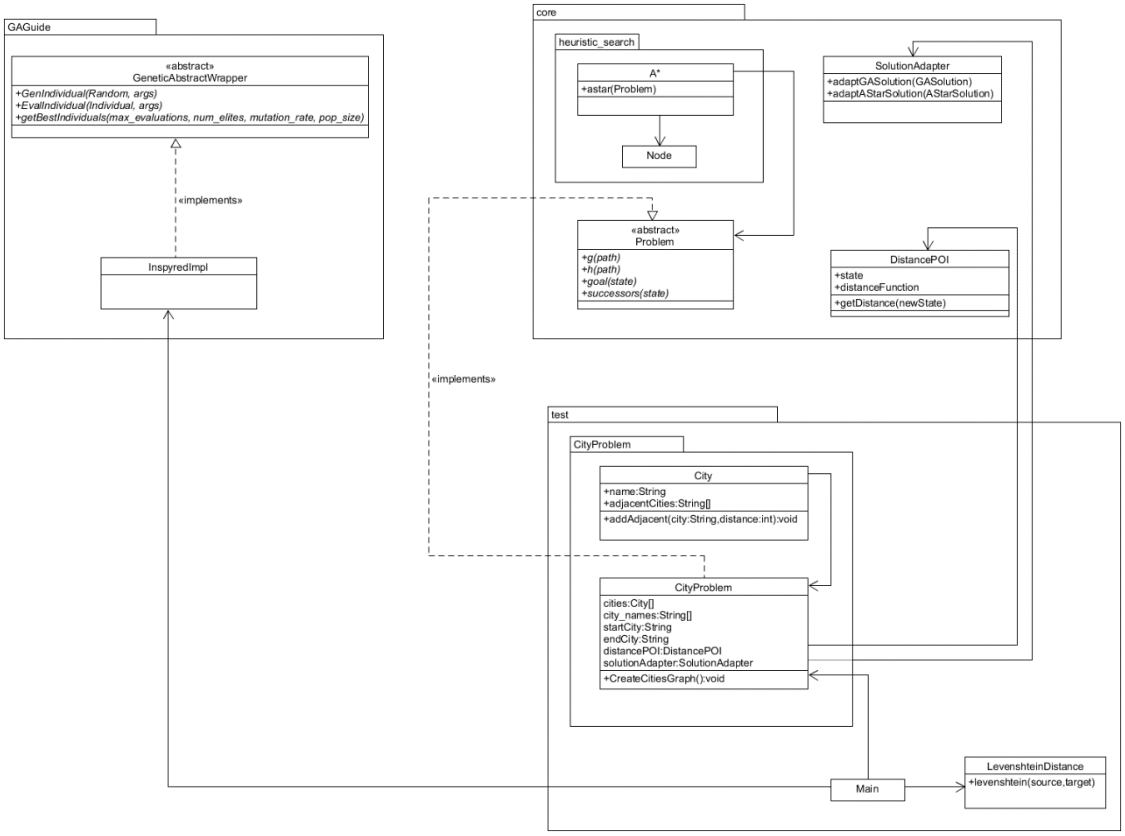


Figure 2 Diagramma delle classi impiegate nel sistema

Nel precedente diagramma è possibile individuare diversi package di cui ***GAGuide*** contenente un wrapper per algoritmi genetici e la relativa implementazione con la libreria *inspyred*.

Il package ***Core*** contiene la classe astratta *Problem* che dovrà essere implementata da un problema concreto. Si trovano anche *SolutionAdapter* che si occupa di uniformare le soluzioni di algoritmo genetico e di A* in modo da renderle confrontabili ed infine *DistancePOI*, inizializzato con la soluzione dell'algoritmo genetico, il quale calcola la distanza fra tale soluzione ed un'altra qualsiasi soluzione. Nella fattispecie la distanza è calcolata tramite una funzione che deve essere fornita dall'utente. Nel sotto-package *heuristic_search* si trova l'implementazione dell'algoritmo A* che accetta in input un oggetto di tipo *Problem*.

Il package **Test** contiene la realizzazione del problema utilizzato come esempio, CityProblem, un'implementazione della funzione di distanza di Levenshtein ed infine un modulo Main che funge da entry point per avviare il processo.

3. Sperimentazione

L'obiettivo di questa sperimentazione è verificare che l'utilizzo di una guida genetica possa in qualche modo rendere più efficiente l'algoritmo A*. Tale efficienza è misurata nel numero di nodi espansi prima di giungere ad uno stato obiettivo.

Per svolgere le relative prove del sistema è stato implementato, come già accennato, il cosiddetto **CityProblem**. Tale problema consiste nel trovare il percorso migliore, costituito da una sequenza di città da attraversare, per giungere ad una città obiettivo partendo da una città indicata. Si dispone a questo scopo di una struttura a grafo utile a rappresentare la topologia delle città, i relativi collegamenti e le distanze che intercorrono fra l'una e l'altra. La rappresentazione dei percorsi, al fine di essere compatibile con i requisiti dell'algoritmo genetico, è di questo tipo:

[0,1,0,3,2,0,0,5,0,4]

Questa sequenza di interi rappresenta un individuo generato da un algoritmo genetico. Ogni individuo ha una lunghezza fissa pari al numero delle città presenti nel grafo. Ad ogni posizione nell'individuo corrisponde una precisa città, di cui si indica il non attraversamento con lo zero oppure un intero che rappresenta l'ordine con le quali le città del percorso sono attraversate. Le città che costituiscono i nodi del grafo sono rappresentate in un'altra sequenza in cui è rilevante la posizione ai fini delle soluzioni genetiche:

[A,B,C,D,E,F,G,H,I,J]

Dunque l'individuo rappresentato precedentemente indica un percorso di questo tipo: **B->E->D->J->H**. Tali rappresentazioni verranno poi adattate al fine di essere confrontabili con le soluzioni prodotte dall'algoritmo A*.

La sperimentazione è stata eseguita utilizzando come misura il numero di nodi (che rappresentano le configurazioni del problema) espansi dall'algoritmo A*. Il CityProblem è stato affrontato ponendo come città di inizio la **A** e come città di fine la **U** come indicato nella rappresentazione seguente del grafo in figura 3.

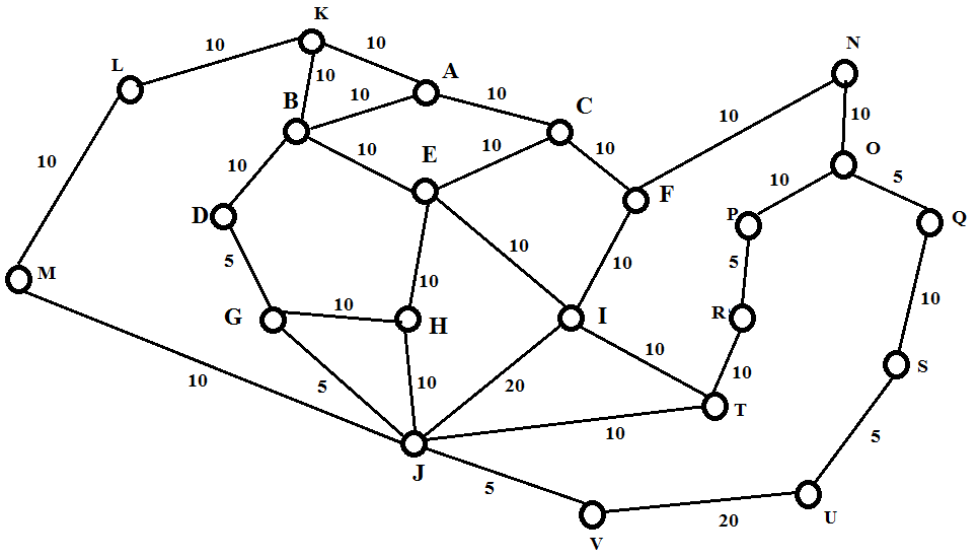


Figure 3 Grafo del CityProblem

Nella seguente tabella invece sono raccolti i parametri utilizzati dall'algoritmo genetico in questa sperimentazione:

Parametri algoritmo genetico	
N° valutazioni	100
N° individui per elitismo	1
Mutation rate	0.1
Popolazione iniziale	10

Il confronto sull'efficienza fra i due approcci è stato eseguito con un run del sistema con il solo algoritmo A* senza guida genetica ed una serie di 100 run dell'algoritmo A* con guida genetica. È stato necessario ripetere l'esecuzione dell'algoritmo con guida genetica in modo da ottenere un numero medio di espansioni, in quanto ad ogni run l'algoritmo genetico propone una soluzione differente.

L'esperimento è stato eseguito impostando il valore euristico di A*, in altre parole il valore della funzione *h(x)*, pari ad 1. Ciò consente sostanzialmente di ridurre il problema ad una BFS (*breadth first search*) al fine di enfatizzare il più possibile il contributo genetico alla soluzione finale.

Nella seguente tabella è indicato il numero di espansioni della lista OPEN di A* nel caso puro e con guida genetica:

	A* Puro	**A* con GA**
N° medio di espansioni della lista OPEN	*30*	*29,5*

I risultati sperimentali mostrano un guadagno di 0.5 espansioni rispetto all'esecuzione pura. Ovviamente tale guadagno è molto lieve per la natura del problema di esempio utilizzato. Se si dovesse impiegare un problema molto più complesso, tale discrepanza sarebbe più accentuata.
Al fine di validare statisticamente i risultati sono stati eseguiti due differenti test statistici, il *T-Test Unpaired* ed il *Test Wilcoxon*.

Il *Wilcoxon Rank-Sum Test* [5] è stato eseguito sul *sample A* costituito dal valore costante 30, che indica il numero di espansioni di A* puro e sul *sample B* costituito dal numero di espansioni eseguite con la guida genetica. Tale test è stato eseguito con il tool online [] configurato con questi parametri:

- *H0: median (difference) = 0*
- *Ha: median (difference) > 0*
- *Significance level (alfa): 0.005*

Il risultato dell'esperimento mostra una differenza significativa fra i due campioni, dunque secondo questo test statistico il guadagno dovuto all'utilizzo della guida genetica è significativo. Si rigetta quindi l'ipotesi H0 secondo la quale fra le due medie non sussiste alcuna differenza e dunque la differenza è maggiore di 0 ad un livello di significatività pari a 0.005.

Di seguito si riporta l'output del test eseguito:

(1) *Null and Alternative Hypotheses*

The following null and alternative hypotheses need to be tested:

H_0: Median (Difference) = 0

H_a: Median (Difference) > 0

Observe that both sample sizes are greater than 10, then we can use normal approximation. The following z-statistic will be used.

(2) *Rejection Region*

Based on the information provided, the significance level is $\alpha = 0.005$, and the critical value for a right-tailed test is $z_c = 2.58$.

The rejection region for this right-tailed test is $R = z : z > 2.58$

(3) *Test Statistics*

The z-statistic is computed as follows:

$$z = \frac{R - n_1(n_1 + n_2 + 1)/2}{\sqrt{n_1 n_2 (n_1 + n_2 + 1)/(12)}} = \frac{12550 - 100(100 + 100 + 1)/2}{\sqrt{100100(100 + 100 + 1)/(12)}} = 6.108$$

(4) *Decision about the null hypothesis*

Since it is observed that $z = 6.108 > z_c = 2.58$, it is then concluded that *the null hypothesis is rejected.*

Using the P-value approach: The p-value is $p = 0$, and since $p = 0 < 0.005$, it is concluded that the null hypothesis is rejected.

(5) *Conclusion*

It is concluded that the null hypothesis Ho is *rejected.* Therefore, there is enough evidence to claim that the population median of differences is greater than 0, at the 0.005 significance level.

Figure 4 Risultati del test statistico Wilcoxon

L' Unpaired T-Test è stato eseguito con il tool online [6] e di seguito si riporta l'output ottenuto:

Unpaired *t* test results

P value and statistical significance:
The two-tailed P value is less than 0.0001
By conventional criteria, this difference is considered to be extremely statistically significant.

Confidence interval:
The mean of A* Puro minus A* + GA equals 0.50
95% confidence interval of this difference: From 0.40 to 0.60

Intermediate values used in calculations:
t = 9.9499
df = 198
standard error of difference = 0.050

Learn more:
GraphPad's web site includes portions of the manual for GraphPad Prism that can help you learn statistics. First, review the meaning of P values and confidence intervals . Then learn how to interpret results from an unpaired or paired *t* test. These links include GraphPad's popular *analysis checklists* .

Review your data:

Group	A* Puro	A* + GA
Mean	30.00	29.50
SD	0.00	0.50
SEM	0.00	0.05
N	100	100

Figure 5 Risultati del test statistico Unpaired T-Test

Anche in questo caso si è scelto come *sample A* il valore costante 30, come *sample B* i valori ottenuti da ognuna delle 100 ripetizioni dell'esperimento. Il risultato ottenuto, come si evince dall'output, mostra che la differenza fra i due campioni è statisticamente significativa, il che vuol dire che la guida genetica offre un supporto concreto all'espansione dell'algoritmo A*.

Conclusioni

L'esecuzione di questa sperimentazione ha mostrato, seppur in maniera ridotta, che l'efficienza di un algoritmo di graph search come A* può trovare giovamento con una guida genetica. Il miglioramento nell'efficienza di A* potrebbe essere percepito in maniera più consistente con un problema la cui complessità necessiti di una guida genetica al fine di evitare espansioni *random* nei casi di parità.

L'esperimento con la guida genetica è stato eseguito 100 volte in modo da coglierne l'andamento rispetto al numero medio di espansioni. Infatti ad ogni iterazione l'algoritmo genetico propone una soluzione differente, la quale in molti casi fornisce un supporto ad A*.

Riferimenti

[1] Libreria Python Inspyred per algoritmi genetici v.1.0.1: https://pypi.python.org/pypi/inspyred

[2] Libreria Python Inspyred documentazione: http://pythonhosted.org/inspyred/

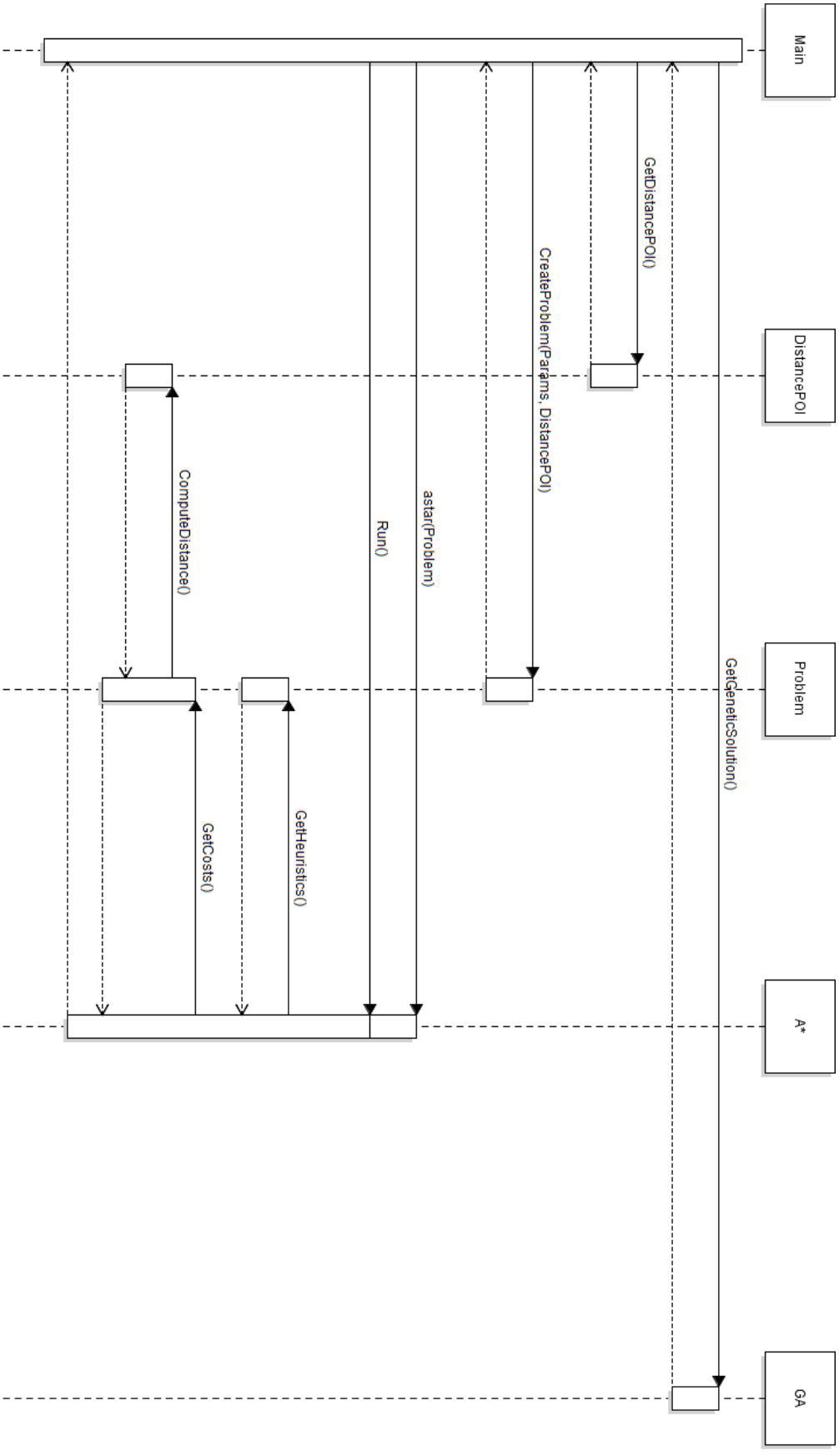
[3] UMLet, tool open source per diagrammi UML: http://www.umlet.com/

[4] VioletUML, tool open source per diagrammi UML: http://alexdp.free.fr/violetumleditor/page.php

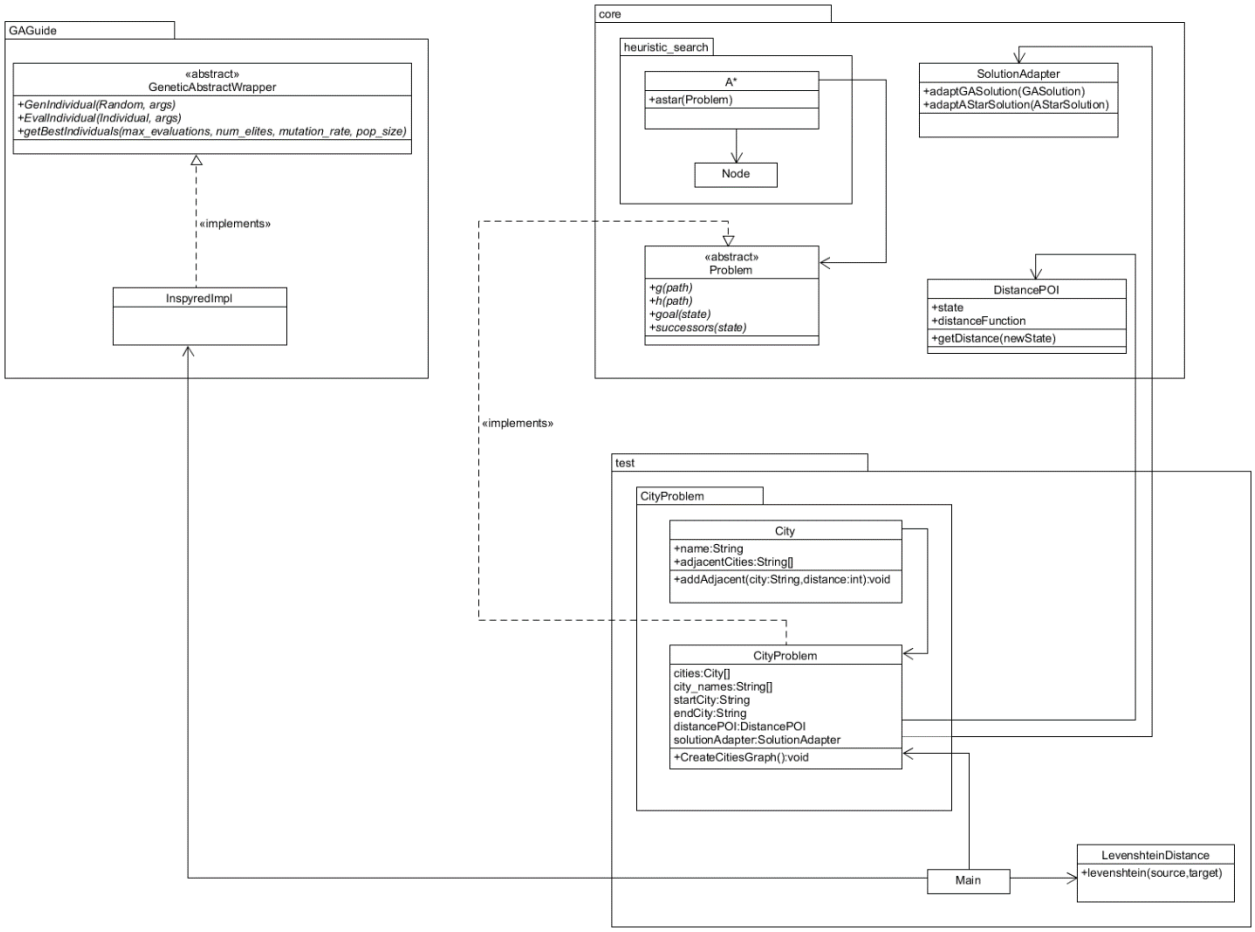
[5] Wilcoxon Rank-Sum Test Calculator: http://mathcracker.com/wilcoxon-rank-sum.php

[6] Unpaired T-Test Calculator: https://www.graphpad.com/quickcalcs/ttest1/?Format=C

Appendice A – Diagramma di Sequenza



Appendice B –Diagramma delle Classi



Appendice C – Risultati delle 100 ripetizioni

SOLUZIONE ALGORITMO GENETICO: ['B', 'K', 'D', 'E', 'J', 'I', 'M', 'L', 'F', 'T', 'N', 'O', 'Q', 'S', 'P', 'R']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['K', 'B', 'E', 'C', 'L', 'F', 'J', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['B', 'D', 'E', 'I', 'F', 'M', 'C', 'O', 'P', 'T', 'Q', 'S', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['B', 'E', 'H', 'J', 'I', 'D', 'C', 'F', 'O', 'P', 'K', 'L', 'S', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['C', 'B', 'D', 'L', 'M', 'J', 'I', 'T', 'E', 'H', 'G', 'V', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['B', 'D', 'E', 'C', 'G', 'F', 'V', 'I', 'R', 'L', 'P', 'O', 'Q', 'S', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['C', 'F', 'E', 'B', 'I', 'R', 'P', 'D', 'N', 'Q', 'J', 'L', 'M', 'S', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['C', 'F', 'I', 'K', 'E', 'H', 'D', 'B', 'M', 'G', 'N', 'O', 'J', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['B', 'E', 'C', 'F', 'L', 'M', 'T', 'R', 'P', 'O', 'Q', 'N', 'J', 'U']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['B', 'C', 'F', 'I', 'M', 'J', 'P', 'V', 'N', 'O', 'Q', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['C', 'F', 'I', 'B', 'K', 'N', 'J', 'P', 'E', 'H', 'D', 'G', 'M', 'V', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['C', 'E', 'I', 'J', 'H', 'G', 'D', 'R', 'P', 'O', 'Q', 'N', 'V']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['C', 'E', 'B', 'J', 'I', 'V', 'G', 'T', 'U']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['K', 'C', 'F', 'I', 'E', 'J', 'L', 'G', 'D', 'B', 'H', 'U']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['C', 'B', 'D', 'E', 'I', 'F', 'H', 'G', 'L', 'J', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['B', 'D', 'E', 'I', 'H', 'J', 'L', 'N', 'T', 'R', 'P', 'O', 'Q', 'S', 'U']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['C', 'K', 'L', 'J', 'I', 'R', 'E', 'P', 'O', 'N', 'T', 'M', 'V', 'S', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['C', 'E', 'B', 'F', 'I', 'R', 'P', 'H', 'G', 'S', 'N', 'D', 'V']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['C', 'F', 'I', 'J', 'E', 'H', 'G', 'D', 'B', 'K', 'L', 'P', 'U']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['B', 'K', 'L', 'D', 'G', 'J', 'R', 'P', 'I', 'M', 'E', 'H', 'U']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['K', 'B', 'L', 'C', 'F', 'E', 'I', 'H', 'J', 'T', 'N', 'O', 'P', 'R', 'Q', 'S', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['K', 'B', 'C', 'E', 'F', 'I', 'J', 'T', 'R', 'P', 'D', 'G', 'L', 'N', 'Q', 'S', 'U']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['B', 'E', 'H', 'G', 'J', 'I', 'F', 'K', 'D', 'V', 'T']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['K', 'B', 'C', 'J', 'D', 'I', 'F', 'E', 'T', 'N', 'G', 'P', 'R', 'Q', 'S']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['C', 'K', 'B', 'I', 'J', 'D', 'T', 'M', 'U']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['C', 'F', 'E', 'G', 'D', 'B', 'I', 'J', 'T', 'R', 'N', 'O', 'P', 'Q', 'S', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['C', 'E', 'B', 'D', 'G', 'J', 'F', 'T', 'V', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['B', 'E', 'C', 'F', 'D', 'L', 'G', 'T', 'H', 'V', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['C', 'B', 'E', 'H', 'I', 'J', 'M', 'L', 'V', 'D', 'N', 'K', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['K', 'B', 'M', 'I', 'G', 'D', 'C', 'F', 'E', 'H', 'S', 'N', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['B', 'K', 'C', 'E', 'H', 'G', 'D', 'I', 'M', 'J', 'V', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['K', 'L', 'B', 'D', 'E', 'C', 'I', 'H', 'J', 'O', 'G', 'V', 'R', 'P', 'Q', 'F', 'M']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['C', 'B', 'E', 'F', 'T', 'D', 'P', 'J', 'M', 'Q', 'S', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['C', 'K', 'G', 'D', 'B', 'J', 'T', 'I', 'R', 'M', 'L', 'U']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['B', 'E', 'D', 'G', 'K', 'C', 'J', 'M', 'N', 'T', 'V', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['C', 'B', 'M', 'D', 'G', 'E', 'N', 'O', 'P', 'Q', 'T', 'V', 'R', 'U']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['K', 'C', 'F', 'I', 'E', 'J', 'H', 'G', 'T', 'L', 'M', 'N', 'O', 'Q', 'S', 'P', 'V']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['B', 'D', 'E', 'C', 'F', 'I', 'J', 'L', 'K', 'G', 'V', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['B', 'K', 'E', 'C', 'L', 'J', 'G', 'P', 'O', 'Q', 'S', 'V', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['K', 'L', 'B', 'J', 'I', 'F', 'C', 'E', 'D', 'G', 'N', 'H', 'R', 'P', 'O', 'Q', 'S', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['K', 'B', 'D', 'G', 'J', 'E', 'C', 'F', 'I', 'L', 'H', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['K', 'B', 'L', 'M', 'J', 'I', 'F', 'T', 'R', 'P', 'N', 'O', 'Q', 'S']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['C', 'B', 'D', 'K', 'L', 'M', 'J', 'G', 'F', 'I', 'V', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['B', 'K', 'L', 'D', 'G', 'J', 'I', 'F', 'C', 'E', 'H', 'T']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['C', 'B', 'D', 'I', 'G', 'J', 'T', 'L', 'P', 'O', 'N', 'F', 'M', 'U']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['C', 'B', 'G', 'L', 'I', 'R', 'P', 'F', 'N', 'O', 'Q', 'S', 'V']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['C', 'F', 'E', 'H', 'P', 'G', 'D', 'B', 'K', 'L', 'R', 'V', 'U']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['K', 'C', 'E', 'F', 'B', 'I', 'T', 'J', 'N', 'O', 'P', 'R', 'Q', 'V', 'U']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['K', 'B', 'D', 'C', 'G', 'J', 'T', 'R', 'I', 'F', 'V', 'S', 'U']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['C', 'B', 'D', 'E', 'H', 'G', 'J', 'T', 'R', 'P', 'O', 'U']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['B', 'E', 'I', 'J', 'F', 'N', 'T', 'R', 'H', 'V', 'M', 'L', 'U']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['B', 'K', 'C', 'E', 'I', 'J', 'T', 'M', 'U']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['B', 'E', 'C', 'F', 'K', 'D', 'G', 'J', 'I', 'N', 'O', 'P', 'R', 'T', 'V', 'M', 'U']
Numero espansioni A*: 30

SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['C', 'K', 'B', 'M', 'J', 'V', 'G', 'D', 'E', 'H', 'R', 'P', 'N', 'U']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['C', 'K', 'B', 'I', 'D', 'F', 'E', 'N', 'O', 'Q', 'P', 'R', 'V', 'U']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['B', 'K', 'D', 'G', 'J', 'M', 'I', 'F', 'C', 'R', 'V', 'N', 'O', 'Q', 'S', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['K', 'L', 'B', 'E', 'C', 'D', 'G', 'F', 'V', 'J', 'I', 'T', 'R', 'P', 'O', 'Q', 'S', 'U']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['K', 'L', 'B', 'C', 'F', 'I', 'E', 'J', 'T', 'G', 'D', 'N', 'M', 'P', 'R', 'Q', 'U']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['B', 'K', 'C', 'E', 'H', 'J', 'I', 'L', 'F', 'M', 'G', 'D', 'P', 'Q', 'S']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['B', 'D', 'E', 'G', 'J', 'V', 'T', 'R', 'U']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['C', 'E', 'H', 'K', 'F', 'B', 'G', 'T', 'M', 'R', 'P', 'O', 'Q', 'N', 'S', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['B', 'D', 'G', 'E', 'H', 'L', 'M', 'I', 'F', 'R', 'J', 'V', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['B', 'E', 'C', 'F', 'H', 'J', 'T', 'I', 'V', 'U']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['C', 'E', 'F', 'I', 'K', 'L', 'B', 'H', 'J', 'T', 'R', 'P', 'G', 'D', 'N', 'O', 'Q', 'S']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['B', 'E', 'L', 'D', 'G', 'H', 'T', 'I', 'J', 'O', 'F', 'N', 'M', 'V']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['B', 'C', 'L', 'M', 'D', 'E', 'T', 'G', 'F', 'N', 'O', 'Q', 'P', 'R', 'S', 'H', 'V']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['K', 'B', 'M', 'E', 'J', 'O', 'I', 'Q', 'S', 'G', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['B', 'E', 'H', 'J', 'I', 'G', 'D', 'K', 'U']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['C', 'B', 'D', 'E', 'F', 'T', 'V', 'I', 'H', 'M', 'O', 'Q', 'S', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['K', 'B', 'I', 'E', 'H', 'J', 'G', 'D', 'L', 'T', 'R', 'V', 'U']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['B', 'L', 'C', 'F', 'I', 'J', 'T', 'U']

Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['C', 'F', 'I', 'E', 'B', 'H', 'G', 'D', 'N', 'M', 'S', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['C', 'K', 'G', 'E', 'B', 'D', 'J', 'R', 'N', 'P', 'O', 'Q', 'S', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['K', 'B', 'C', 'E', 'D', 'G', 'L', 'I', 'H', 'F', 'R', 'P', 'U']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['C', 'B', 'E', 'H', 'T', 'R', 'J', 'P', 'O', 'Q', 'S', 'U']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['B', 'C', 'L', 'M', 'J', 'G', 'E', 'I', 'O', 'P', 'R', 'D', 'V', 'U']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['B', 'L', 'M', 'C', 'E', 'D', 'F', 'N', 'R', 'G', 'J', 'H', 'V', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['B', 'D', 'L', 'G', 'C', 'N', 'O', 'Q', 'P', 'R', 'T', 'V', 'U']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['C', 'E', 'F', 'B', 'D', 'J', 'N', 'G', 'M', 'V', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['K', 'B', 'M', 'D', 'J', 'V', 'T', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['B', 'D', 'E', 'H', 'G', 'N', 'O', 'P', 'R', 'Q', 'S', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['C', 'B', 'D', 'G', 'K', 'E', 'H', 'J', 'N', 'O', 'P', 'Q', 'S', 'I', 'T', 'M']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['K', 'C', 'E', 'B', 'D', 'F', 'T', 'I', 'H', 'G', 'J', 'L', 'R', 'P', 'O', 'Q', 'S', 'U']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['C', 'F', 'I', 'J', 'L', 'S', 'B', 'D', 'E', 'H', 'T', 'R', 'V', 'U']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['B', 'C', 'E', 'H', 'J', 'V', 'T', 'R', 'P', 'Q', 'S', 'U']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['B', 'E', 'H', 'I', 'G', 'D', 'T', 'R', 'P', 'O', 'Q', 'N', 'V', 'S']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['C', 'E', 'K', 'T', 'D', 'B', 'R', 'J', 'G', 'F', 'H', 'L', 'M', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['K', 'L', 'B', 'C', 'F', 'N', 'I', 'J', 'M', 'T', 'V', 'U']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['K', 'C', 'E', 'H', 'J', 'T', 'R', 'V', 'U']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['K', 'C', 'E', 'B', 'D', 'J', 'G', 'H', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['B', 'D', 'E', 'H', 'T', 'G', 'N', 'O', 'P', 'Q', 'R', 'J', 'I', 'L', 'M', 'V', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['B', 'E', 'C', 'H', 'I', 'F', 'D', 'J', 'T', 'V', 'M', 'G', 'R', 'P', 'O', 'N', 'Q', 'S', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['C', 'F', 'B', 'E', 'I', 'J', 'K', 'N', 'O', 'D', 'G', 'H', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['C', 'K', 'B', 'D', 'G', 'J', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['B', 'L', 'C', 'F', 'I', 'O', 'D', 'G', 'J', 'T', 'R', 'V', 'U']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['C', 'F', 'I', 'E', 'B', 'R', 'H', 'J', 'N', 'V', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['K', 'B', 'L', 'M', 'J', 'G', 'I', 'E', 'T', 'V', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['B', 'K', 'C', 'E', 'H', 'F', 'G', 'D', 'I', 'V', 'N', 'O', 'U']
Numero espansioni A*: 30
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

SOLUZIONE ALGORITMO GENETICO: ['B', 'D', 'G', 'E', 'H', 'J', 'I', 'F', 'C', 'L', 'M', 'V', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('B', 10), ('D', 10), ('G', 5), ('J', 5), ('V', 5), ('U', 20)]

SOLUZIONE ALGORITMO GENETICO: ['B', 'E', 'C', 'J', 'M', 'L', 'F', 'K', 'G', 'H', 'Q', 'S', 'U']
Numero espansioni A*: 29
SOLUZIONE A*:[('A', 0), ('C', 10), ('F', 10), ('N', 10), ('O', 5), ('Q', 5), ('S', 10), ('U', 5)]

MEDIA ESPANSIONI: 29.5