

Contents

1	Definizione del progetto	1
1.1	L'idea del progetto di IS	1
1.2	L'idea del modulo intelligente di FIA	1
1.3	Analisi del problema	1
1.4	Analisi dell'algoritmo	2
2	Progettazione dell'algoritmo genetico	3
2.1	Spazio degli stati	3
2.2	Codifica degli individui	3
2.3	Passi dell'algoritmo genetico	4
2.4	Inizializzo la popolazione	4
2.5	Valuto la popolazione	6
2.6	Mi fermo? – Criteri di arresto	8

1 Definizione del progetto

1.1 L'idea del progetto di IS

L'idea del progetto di IS e' quello di permettere ad uno studente di trovare dei percorsi formativi da apprendere. Questi percorsi formativi sono insegnati da dei formatori, in particolare, ogni percorso formativo e' sostenuto da uno studente in una o piu' fasce orarie in uno o piu' giorni della settimana, ed ogni percorso formativo e' insegnato da un solo formatore quindi, avremo dei percorsi formativi individuali.

1.2 L'idea del modulo intelligente di FIA

L'idea del modulo intelligente di FIA e' il seguente:

Fornire ad uno studente un piano formativo personalizzato sulla base dei seguenti parametri:

1. Gli interessi dello studente.
2. lo storico dei percorsi formativi sostenuti dallo studente.
3. I giorni della settimana in cui lo studente e' libero.

1.3 Analisi del problema

Analisi del problema:

- Ambiente:
 - completamente osservabile: in ogni momento l'agente ha a disposizione tutti i percorsi formativi;
 - stocastico: ad ogni azione dell'agente non sappiamo quale sara' il prossimo stato;

- episodico;
- statico: l'ambiente e' sempre lo stesso;
- discreto: le percezioni dell'agente sono limitate poiche' avra' un numero limitato di specifiche;
- singolo: abbiamo un unico agente.
- Sensori:
 - gli interessi dello studente;
 - lo storico dei percorsi formativi sostenuti dallo studente.
 - i giorni della settimana in cui lo studente e' libero;
- Attuatori:
 - un vettore di massimo 7 percorsi formativi mostrati a video.

1.4 Analisi dell'algoritmo

Utilizziamo un algoritmo di ricerca.

Non mi e' sembrato opportuno utilizzare un grafo, poiche' non ho trovato dei collegamenti opportuni tra i vari nodi.

Escludiamo le seguenti ricerche:

- Ricerca non informata.
- Ricerca informata.

Non mi e' sembrato opportuno utilizzare un albero, poiche' non ho trovato una gerarchia tra i vari nodi e inoltre abbiamo soltanto un unico agente.

Escludiamo la seguente ricerca:

- Ricerca con avversari.

Utilizziamo un algoritmo genetico.

Poiche' la ricerca deve essere il piu' ottimale possibile, questo potrebbe comportare un tempo piu' lungo rispetto ad una normale ricerca per appunto avvantaggiare l'efficienza.

Nasce quindi la necessita' di avvisare lo studente che potrebbe essere necessario aspettare qualche secondo in piu'.

Quindi, nell'arco della progettazione dell'algoritmo genetico, Prenderemo in considerazione la seguente classifica numerata dal piu' importante al meno importante:

1. Completezza e ottimalita'.
2. Complessita' temporale.
3. Complessita' spaziale.

Ovviamente, l'utente non vorra' aspettare molto tempo quindi, all'algoritmo gli daremo del tempo non superiore ai 10 secondi, mentre la visualizzazione del piano formativo avverra' dopo massimo 12 secondi.

2 Progettazione dell'algoritmo genetico

2.1 Spazio degli stati

Lo spazio degli stati lo ricaviamo da una interrogazione al database ricavandoci i percorsi formativi insegnati nei giorni della settimana in cui lo studente e' libero.

Abbiamo quindi un vettore (Vettore stati) di N percorsi formativi in cui abbiamo l'id presente nel database:

Figure 1: Vettore stati

0	1	2	...	N-1
1	10	5	...	10

2.2 Codifica degli individui

Come codifica utilizzeremo un vettore di massimo 7 interi positivi (individuo).

2 Esempi di individui:

Figure 2: Esempio individui

Esempio 1:

0 1 2 3

N = 4 →

0 1 2 3

Esempio 2:

0 1 2 3 4 5 6

N = 20 →

2 19 8 0 15 4 3

Ogni intero rappresenta una posizione di un percorso formativo all'interno del vettore stati.

Perche' massimo 7? perche' vogliamo ritornare un sottoinsieme di 7 percorsi formativi (piano formativo) ordinati in base al giorno e all'orario.

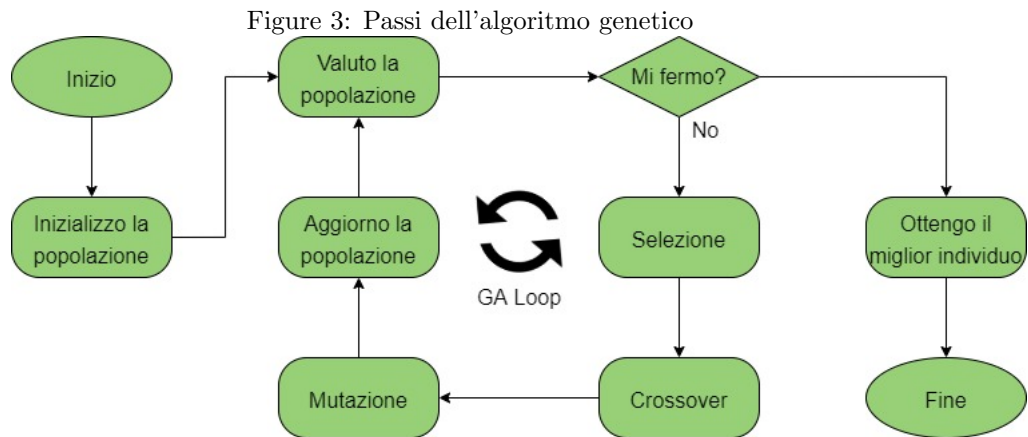
Questo sara' il DNA della soluzione composta da massimo 7 geni. Quindi, ogni individuo non e' altro che un piano formativo.

Ora, considerando il vettore stati e N il numero di percorsi formativi presente nel vettore stati:

- se $N = 0$ allora ritorno: non ho trovato nessun percorso formativo.
- se $0 < N \leq 7$ allora ritorno tutti i percorsi formativi senza eseguire l'algoritmo.
- se $N > 7$ allora eseguo l'algoritmo genetico e ritorno l'individuo trovato.

2.3 Passi dell'algoritmo genetico

Questi saranno i passi dell'algoritmo genetico:



2.4 Inizializzo la popolazione

Sia N il numero di percorsi formativi presenti nel vettore stati, la nostra popolazione sarà composta da un numero K di individui tale che:

$$K = \lfloor \frac{N}{3} \rfloor.$$

All'inizio, tutti i K individui saranno generati in maniera random.

Esempi:

Figure 4: Esempi popolazioni

$N = 4$

Popolazione:

	0	1	2	3
Individuo 1:	0	1	2	3

$N = 4$

$$K = \left\lfloor \frac{N}{3} \right\rfloor = \left\lfloor \frac{7}{3} \right\rfloor = \left\lfloor 2,3 \right\rfloor = 2$$

	0	1	2	3	4	5	6
Individuo 1:	0	2	4	1	6	5	3
	0	1	2	3	4	5	6
Individuo 2:	5	3	4	2	1	3	6

$N = 12$

$$K = \left\lfloor \frac{N}{3} \right\rfloor = \left\lfloor \frac{12}{3} \right\rfloor = \left\lfloor 4 \right\rfloor = 4$$

	0	1	2	3	4	5	6
Individuo 1:	0	10	2	8	4	6	1
	0	1	2	3	4	5	6
Individuo 2:	1	11	3	9	5	7	2
	0	1	2	3	4	5	6
Individuo 3:	3	5	7	2	4	6	11
	0	1	2	3	4	5	6
Individuo 4:	10	6	5	7	5	8	3

2.5 Valuto la popolazione

Come possiamo valutare la popolazione?

Abbiamo bisogno di una funzione obiettivo (`funzioneObiettivo(codifica)`).

Dalla funzione obiettivo ci ricaviamo una funzione di valutazione (`funzioneValutazione(codifica)`) che sarà:

- uguale alla funzione obiettivo nel caso in cui quest'ultima non richiede molte risorse, sia accessibile, non è complessa e non è impossibile da realizzare.
- un approssimazione della funzione obiettivo nel caso in cui il punto precedente non è vero.

Dopo di che, una volta applicata la funzione di valutazione, ad ogni individuo gli sarà dato un valore di partenza che con la funzione di fitness (`fit(codifica)`) avremo il grado di sopravvivenza di ciascun individuo. La funzione di fitness dipende dall'algoritmo di selezione utilizzato.

Funzione obiettivo (`funzioneObiettivo(codifica)`) considerazioni:

poiché possono esistere 2 o più geni con lo stesso valore allora il punteggio sarà dato solo ai geni con valore diverso, quindi, ci servirà un modo per calcolare il punteggio solo ai nodi diversi (utilizzeremo una funzione chiamata `contains`, java la fornisce).

Le seguenti funzioni possono cambiare (non sono ancora quelle definitive):

Funzione obiettivo:

Figure 5: funzioneObiettivo(codifica) – pseudocodice:

```
funzioneObiettivo (codifica) {  
    b ← new array()  
    punteggio ← 0  
    size ← codifica.size()  
    FOR (i ← 0; i < size; i ← i + 1) {  
        gene ← codifica.get(i)  
  
        IF (i == 0) {  
            duplicato ← false  
        }  
        ELSE {  
            duplicato ← b.contains(gene)  
        }  
  
        b.add(gene)  
  
        IF (duplicato == false) {  
            punteggio ← punteggio + getPunteggio(gene)  
        }  
    }  
    return punteggio  
}
```

getPunteggio(gene):
con la funzione getPunteggio(gene):

1. Mi ricavo il percorso formativo.
2. Con il percorso formativo ricavato, mi ricavo:
 - (a) Il valoreInteresse.
 - (b) Il valoreStorico.
 - (c) Il valoreCosto
3. Utilizzero' una delle seguenti funzioni matematiche:
 - Somma lineare di tutti i parametri:
 $SP = \text{valoreInteresse} + \text{valoreStorico} + \text{valoreCosto}$.
Soluzione migliore: la soluzione che ha il punteggio:
 $SMP = (15 + 10 + 5) * 7 = 210$.
 - Somma lineare di tutti i parametri moltiplicati per una costante c:
 $SP = (\text{valoreInteresse} * c) + (\text{valoreStorico} * c) + (\text{valoreCosto} * c)$.

Soluzione migliore: la soluzione che ha il punteggio:

$$SMP = [(15 * c) + (10 * c) + (5 * c)] * 7 = [(15 + 10 + 5) * c] * 7 \\ = [30 * c] * 7 = 30 * 7 * c = 210 * c.$$

- Somma lineare di tutti i parametri elevati per una costante d:
 $SP = (valoreInteresse^d) + (valoreStorico^d) + (valoreCosto^d).$

Soluzione migliore: la soluzione che ha il punteggio:

$$SMP = [(15^d) + (10^d) + (5^d)] * 7$$

- Il quadrato della somma lineare di tutti i parametri:
 $SP = (valoreInteresse + valoreStorico + valoreCosto)^2.$

Soluzione migliore: la soluzione che ha il punteggio:

$$SMP = [(15 + 10 + 5)^2] * 7 = 900 * 7 = 6300$$

Li analizzeremo successivamente scegliendo la funzione che darà risultati migliori.

Figure 6: getPunteggio(gene) – pseudocodice:

```
getPunteggio (gene) {  
    // Mi ricavo il percorso formativo con indice uguale a gene  
    percorsoFormativo ← getPercorsoFormativo(gene)  
    materia ← percorsoFormativo.getMateria()  
    settore ← percorsoFormativo.getSettore()  
    valoreInteresse ← percorsoFormativo.getSettore()  
    valoreStorico ← percorsoFormativo.getValoreStorico()  
    valoreCosto ← percorsoFormativo.getValoreCosto()  
    // Con la funzione:  
    // getValoreFunzione(valoreInteresse, valoreStorico, valoreCosto)  
    // userò una delle funzioni mostrate sopra.  
    punteggio ← getValoreFunzione(valoreInteresse, valoreStorico, valoreCosto)  
    return punteggio  
}
```

2.6 Mi fermo? – Criteri di arresto

Criteri di arresto utilizzati:

1. Test Ottimalità: se raggiungiamo l'ottimo allora ci fermiamo.
2. Tempo di Esecuzione: dopo 10 secondi l'algoritmo termina.