

Sommario

| | |
|--|---|
| Definizione del problema | 3 |
| Specifica dell'ambiente | 3 |
| Soluzione | 4 |
| Il confronto con OPT | 4 |
| Soluzione ottima | 4 |
| Valutazione configurazione dell'algoritmo rispetto ad OPT | 4 |
| Parametri | 5 |
| Selezione degli individui | 5 |
| Crossover..... | 5 |
| Mutazione | 5 |
| Funzione di fitness..... | 5 |
| Aging | 7 |
| Stopping condition | 7 |
| Passi dell'algoritmo | 7 |
| Reiterazione | 8 |
| Testing dei parametri scelti | 8 |
| Numero di iterazioni | 8 |
| Durata delle iterazioni | 8 |
| Size della popolazione | 8 |
| Dimensione dell'individuo..... | 9 |
| Considerazioni finali | 9 |

Definizione del problema

Il progetto prevede un sito che permette all'utente di navigare tra le schede tecniche di vari cellulari e un tool che ne consiglia alcuni all'utente sulla base di un budget e di quanto gli importino determinate caratteristiche.

e.g. L'utente dovrà dire su una scala da 1 a 5 che importanza hanno per lui batteria, fotocamera, display e prestazioni

Il dataset è stato facilmente reperibile grazie all'utilizzo di un semplice tool di scraping.

Il metro di paragone per confrontare le caratteristiche dei cellulari è stato ottenuto attraverso un sistema di punteggi inseriti da recensori* parte integrante della scheda tecnica del prodotto.

**Recensori: nel nostro dominio applicativo equivale a un team di professionisti*

Specifiche dell'ambiente

- **Performance**

Le prestazioni dell'agente sono valutate attraverso le seguenti misure:

1. Lo scarto dei parametri *Battery, Performance, Display, Camera* corrispondenti ai punteggi delle diverse componenti
2. La data di uscita

e.g. A parità dei parametri (1), uno smartphone uscito nel 2019 sarà preferito rispetto a uno uscito nel 2016

- **Environment**

Completamente osservabile:

In ogni momento l'agente conosce tutte le specifiche dei diversi cellulari.

Agente singolo

Stocastico:

Lo stato successivo non è determinabile dato lo stato corrente e l'azione da applicare, data la presenza di un componente randomico.

Discreto:

Il numero di percezioni dell'agente è limitato in quanto ha un numero discreto di specifiche, azioni e percezioni possibili.

Statico:

Nonostante il sito permetta l'aggiunta di nuovi smartphone, l'agente elabora sull'insieme corrente delle schede tecniche presenti.

Episodico

- **Actuators**

Mostrare a video un insieme di N schede tecniche conformi

- **Sensors**

Input da parte dell'utente delle preferenze sui campi *Battery, Performance, Display, Camera*

Dataset accessibile attraverso il database già in uso dal sito

Soluzione

Formulato il problema, era chiara la necessità di un algoritmo di **ottimizzazione**.

Data l'assenza di relazioni evidenti tra gli elementi del nostro dataset è stata scartata l'ipotesi di rappresentare il problema come un grafo.

Inoltre, essendo il problema *offline*, anche gli algoritmi di ricerca *online* non soddisfavano le esigenze.

Non avendo un ambiente multi-agente le soluzioni mediante la teoria dei giochi non erano applicabili.

Infine l'opzione più promettente è sembrata essere quella di utilizzare un **algoritmo genetico** in quanto le sue caratteristiche sono simili a quelle del problema.

Il nostro obiettivo, infatti, è quello di restituire un insieme di cellulari che meglio approssimino le caratteristiche richieste in input dall'utente.

Pertanto un **individuo** non è altro che un insieme di N schede tecniche.

Il confronto con OPT

Durante la progettazione dell'algoritmo genetico si è cercato un modo per stabilire la bontà di una configurazione di parametri.

In particolare ci è possibile calcolare la **soluzione ottima** senza troppe difficoltà poiché il dataset si presta per un'operazione di ricerca lineare.

Soluzione ottima

La soluzione ottima è composta da un individuo che minimizzi lo scarto rispetto ai parametri presi in input. Sarà quindi composta dalle N schede tecniche che più si avvicinano a quei parametri.

Procediamo quindi scorrendo tutta la lista calcolando il **valore di similitudine** (*fit*) rispetto ai parametri e ordiniamo la lista in maniera non decrescente basandoci su questo valore.

La soluzione ottima è formata dall'insieme dei primi N individui della lista ottenuta.

Valutazione configurazione dell'algoritmo rispetto ad OPT

Data una configurazione dell'algoritmo genetico, lo eseguiamo un numero K di volte. Da ogni iterazione otteniamo un valore minimo di *fit* e al termine ne effettuiamo la **media**.

Questa media verrà confrontata con il valore di *fit* di OPT ottenuto sommando tutti i valori di *fit* delle schede tecniche all'interno di OPT.

Per calcolare il grado di ottimalità di una determinata configurazione dell'algoritmo genetico, non facciamo altro che valutare la sua vicinanza al valore di OPT.

e.g. $OPT = 3.12$, $G1 = 4.7$

La vicinanza è $G1 - OPT = 4.7 - 3.12 = 1.58$

$OPT = 3.12$, $G2 = 5.3$

La vicinanza è $G2 - OPT = 5.3 - 3.12 = 2.18$

Viene preferita la configurazione $G1$, in quanto lo scarto è quello che più si avvicina a OPT

Nella prossima sezione del documento analizzeremo tutti i parametri utilizzati e tutti i test effettuati

Parametri

Selezione degli individui

L'unico metodo di selezione utilizzato è tramite *Truncation* dove vengono scelti i primi M individui con il valore di *fit* minore.

Elitismo

La selezione in base alla funzione di *fit* in maniera non decrescente racchiude implicitamente l'elitismo, solo i migliori* individui vengono mantenuti per le generazioni successive.

*Un individuo x è migliore di un individuo y se: $fit(x) < fit(y)$

Crossover

Per semplicità i primi tentativi sono stati effettuati con un crossover di tipo *Single Point* che ha soddisfatto le necessità con ottimi risultati ed è stato quindi scelto come definitivo.

Ad avvalorare ciò, alla singola scheda tecnica all'interno dell'individuo non corrisponde un peso, o comunque non è attribuito un valore particolare in base alla posizione e dato che le schede tecniche non sono legate tra loro da qualche parametro, la scelta di un algoritmo di crossover rispetto ad un altro risulta poco incisiva nel risultato finale.

Mutazione

La mutazione avviene tramite *Random Resetting*, all'interno dell'individuo viene iniettata una scheda tecnica, scelta in modo random dal dataset, in una posizione random dopo aver effettuato il crossover.

Funzione di fitness

Originariamente il problema era pensato come *multiobiettivo* in quanto l'obiettivo era minimizzare lo scarto dei diversi parametri. Il tutto è stato ricondotto ad un *singolo obiettivo* utilizzando un *whole approach* ottenuto tramite la fusione delle singole funzioni di *fitness* in una singola.

Sorge, a questo punto, il problema di confrontare diverse funzioni tra di loro, in quanto, ad esempio, un paragone tra il risultato di una funzione lineare e il risultato di una funzione esponenziale non risulta immediato. Per ovviare a questo problema al termine della computazione delle diverse funzioni, viene effettuato il confronto tra i risultati portandoli a fattor comune.

e.g. confronto tra funzione lineare e funzione esponenziale: si applica la funzione lineare sull'output della funzione esponenziale in modo tale da ottenere un valore confrontabile

1. Una semplice **somma lineare** di tutti i parametri:

$$F(n) = \text{scartoPerformance} + \text{scartoBattery} + \text{scartoCamera} + \text{scartoDisplay}$$

Risultata fallimentare in quanto non scarta abbastanza velocemente gli individui peggiori.

2. **Somma di tutti i parametri moltiplicati per una costante c:**

$$F(n) = \text{scartoPerformance} * c + \text{scartoBattery} * c + \text{scartoCamera} * c + \text{scartoDisplay} * c$$

Leggermente migliore perché rende più netta la differenza tra individui ma ancora non soddisfacente.

3. **Esponenziale dei singoli parametri per una costante d:**

$$F(n) = \text{scartoPerformance}^d + \text{scartoBattery}^d + \text{scartoCamera}^d + \text{scartoDisplay}^d$$

Molto migliore delle precedenti introducendo però un comportamento indesiderato. Per valori di scarto minori di 1 il *quadrato risultava minore dello scarto stesso* rendendo gli individui di quell'intervallo troppo simili tra loro.

$$\text{e.g. Scarto} = 0.5, d = 2 \rightarrow 0.5^2 = 0.25 < \text{Scarto}$$

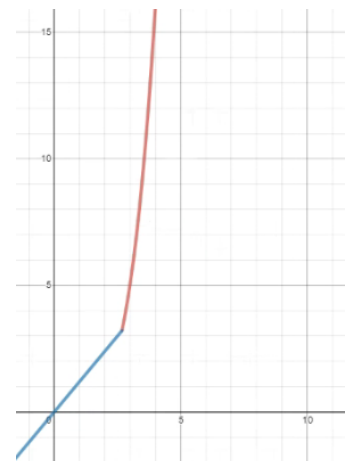
4. **Funzione composta:**

$$F(n) = \text{expand}(\text{scartoPerformance}) + \text{expand}(\text{scartoBattery}) + \text{expand}(\text{scartoCamera}) + \text{expand}(\text{scartoDisplay})$$

$$\text{expand}(x) = \begin{cases} \frac{5}{4}x, & x < \text{threshold} \\ \left(\frac{x}{2}\right)^4, & x \geq \text{threshold} \end{cases}$$

threshold: la soglia individuata equivalente al punto di intersezione delle due funzioni (~2.6)

Risolve il problema della precedente portando a risultati soddisfacenti.



Aging

In seguito alle considerazioni precedenti è stata introdotta la misura **data di uscita** introducendo un fattore di invecchiamento (*aging*) delle schede tecniche avendo come obiettivo quello di privilegiare le più recenti.

Nello specifico è a disposizione il mese e l'anno di uscita del cellulare. Nella prossima parte valuteremo questo campo come *quantità di mesi passati dalla data di uscita*.

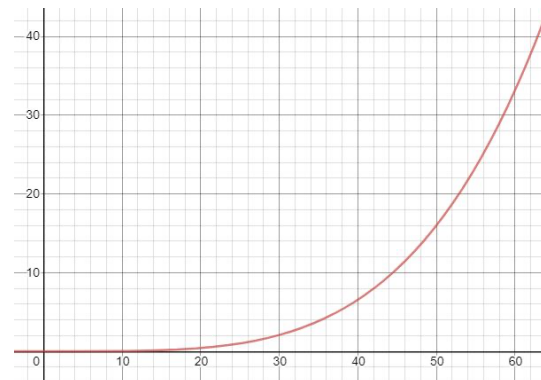
e.g. Data corrente = 12/2020, Data di uscita = 10/2020 → Value = 2

Inizialmente l'algoritmo vincolava in maniera troppo stretta i risultati alla data di uscita ed è stato necessario pesare questo valore in maniera efficace.

La scelta è subito ricaduta su una funzione esponenziale e dopo diversi test, con piccole modifiche ai fattori moltiplicativi ed esponenziali, la seguente ha rispecchiato le nostre necessità:

$$f(x) = \left(\frac{x}{25}\right)^4$$

Dove x rappresenta i mesi passati



Infine, nella valutazione di una scheda tecnica, si è ritenuto opportuno sostituire la somma dei singoli valori di *fit* dei parametri con la loro **media** in modo da slegare il valore dal numero di geni presenti nell'individuo. In questo modo in una futura implementazione gli individui potranno anche avere un **numero variabile di geni**.

Stopping condition

Basandosi su un *requisito non funzionale* all'interno del progetto di Ingegneria del Software per cui il tool deve restituire un valore entro massimo 4 secondi si è scelto di monitorare il tempo di esecuzione e stoppare l'algoritmo al termine di essi.

Inoltre, dato il numero molto limitato di entry nel dataset (*al più circa un migliaio di device*), l'algoritmo arriva molto velocemente a convergenza.

Passi dell'algoritmo

1. Costruire una popolazione iniziale contenente tutti gli individui del DataSet
2. Calcolare il loro valore di fitness
3. Selezionare un sottoinsieme di individui della popolazione iniziale per ammetterli nel *Mating Pool* tramite *Truncation*
4. Gli individui selezionati vengono fatti accoppiare tramite *Single Point Crossover*
5. Gli individui ottenuti vengono mutati tramite una mutazione *Random Resetting* e si ritorna al passo 3
6. Allo scadere del tempo massimo viene restituito l'individuo che minimizza il valore di fit

Reiterazione

Dopo diverse esecuzioni è risultato evidente che l'algoritmo performi meglio se eseguito in più istanze nel lasso di tempo. Si è scelto di eseguire l'algoritmo N volte e scegliere l'individuo che minimizza la **varianza**.

Utilizzando la varianza come parametro di valutazione si predilige un individuo che al suo interno contenga risultati omogenei.

Una soluzione sub-ottima può risultare migliore rispetto ad un'altra soluzione con media minore ma varianza maggiore.

Testing dei parametri scelti

Scelte le preferenze sui parametri (*immagine a destra*) sono stati effettuati dei test per determinare il numero di iterazioni. Si è preso in considerazione la modifica del vincolo sul tempo massimo (*4 secondi*) ed è, quindi, stato necessario valutare anche la durata delle iterazioni.

I risultati equivalgono alla media di più esecuzioni dell'algoritmo

| | KEY | VALUE |
|-------------------------------------|-------------|-------|
| <input checked="" type="checkbox"/> | battery | 4.5 |
| <input checked="" type="checkbox"/> | performance | 4.0 |
| <input checked="" type="checkbox"/> | camera | 4.0 |
| <input checked="" type="checkbox"/> | display | 5.0 |
| <input checked="" type="checkbox"/> | maxBudget | 3000 |

Numero di iterazioni

1. **Run = 5** → Valore medio di fit = 120.86;
2. **Run = 10** → Valore medio di fit = 113.05, Miglior individuo = 99.09;
3. **Run = 20** → Valore medio di fit = **109.90**, Miglior individuo = **98.96**.

Oltre le 20 iterazioni il miglioramento si annulla e al crescere delle iterazioni il risultato è inaccettabile data la slice di tempo infinitesimale.

Durata delle iterazioni

1. **Durata = 200ms** → Valore medio di fit = **112.181**;
2. **Durata = 400ms** → Valore medio di fit = 109.35;
3. **Durata = 600ms** → Valore medio di fit = 111.92;
4. **Durata = 100ms** → Valore medio di fit = 114;
5. **Durata = 50ms** → Valore medio di fit = 116.51;

Seppure la durata di 400ms sembri essere migliore di quella di **200ms** quest'ultima garantisce un buon equilibrio in quanto consente di effettuare le 20 iterazioni, di cui sopra, nel tempo stabilito dal vincolo iniziale.

Size della popolazione

Data la grandezza molto ridotta del dataset sono state testate 3 dimensioni di popolazione: con al più **40**, **100** e **140** individui con Truncation effettuata sulla metà degli individui.

1. La popolazione da 40 individui ha avuto risultati molto scarsi.
2. La popolazione da max 140 individui invece non rendeva possibile trovare un buon risultato in tempi utili.

3. La scelta della popolazione da 100 individui è risultata un buon compromesso dando risultati soddisfacenti.

Dimensione dell'individuo

Per offrire all'utente una scelta variegata ma non confusionaria la scelta è ricaduta su una dimensione di **6 individui**. Infatti, come detto precedentemente, l'algoritmo è poco vincolato alla dimensione dell'individuo grazie all'utilizzo di *media* e *varianza*.

Probabilità di mutazione

La probabilità di mutazione di un individuo generato durante un'iterazione è 1.

Probabilità di accoppiamento

La probabilità che scelti due genitori venga generato un figlio è 1.

Considerazioni finali

Consideriamo il nostro algoritmo soddisfacente per i risultati ottenuti nonostante l'applicazione di alcune semplificazioni. Abbiamo deciso di non valutare il segno dello scarto, anche se quest'ultimo avrebbe migliorato leggermente i risultati in quanto avremmo preferito, a parità di scarto, un individuo con scarto positivo e quindi caratteristiche migliori.

Il rivalutare i parametri e le configurazioni utilizzate anche al termine della stesura di questo documento è una conferma del fatto che l'addentrarci nell'argomento degli algoritmi genetici, e più in generale dell'intelligenza artificiale, non è solo motivato da un interesse dei singoli membri del gruppo ma da un desiderio collettivo di migliorare qualitativamente il risultato. Il tutto reso possibile dalla padronanza acquisita nel processo.