



# Rielaborazione di algoritmi di generazione automatica di coreografie con Robot NAO

Chiara Cippitelli  
Benedetta Rogato

Corso di Fondamenti di Intelligenza Artificiale M - Anno  
Accademico 2022/2023

# 1 Introduzione

L'obiettivo dell'attività progettuale consiste nella rielaborazione di 2 algoritmi di generazione automatica di coreografie robotiche, dati dei vincoli riguardanti le mosse da eseguire e una traccia audio in input. Sono stati scelti, per questo, due progetti che avevano partecipato alla NAOPlanningChallenge [NAOPlanningChallenge] e sono stati rielaborati sotto 3 punti di vista:

- portabilità del codice sul sistema operativo Windows
- upgrade da Python2 a Python3 per l'esecuzione dei progetti
- modifica sostanziale del codice per il campionamento e l'analisi musicale
- aggiunta di features al fine di personalizzare dinamicamente le scelte coreografiche e musicali. In particolare è stata aggiunta la possibilità di :
  - scegliere una traccia musicale da analizzare
  - scegliere la durata della coreografia
  - scegliere quali mosse obbligatorie eseguire
  - scegliere se si vuole rimuovere delle mosse non obbligatorie

Si riportano qui le operazioni comuni ad entrambi i progetti, mentre verranno analizzate successivamente le modifiche specifiche per i singoli progetti.

Il punto di partenza è stato l'analisi del codice dei progetti scelti: ANaoRhythm [ANaoRhythm] e Silk-Sonic [Silk-Sonic]. Entrambi risultavano funzionanti esclusivamente su macchina virtuale, motivo per il quale erano scarsamente portabili e non era garantita una buona riuscita nei test con il robot simulato su Choregraphe. Di conseguenza, è stato modificato il codice rendendolo funzionante anche su Windows (10 e 11), in modo che all'avvio sia possibile riconoscere il sistema operativo su cui si esegue (Windows o Unix), invocando di conseguenza funzioni differenti. Il tutto risulta trasparente all'utente.

Successivamente è stato reingegnerizzato il main file strutturandolo come segue: un primo processo dedicato alla scelta e all'analisi della traccia musicale, un secondo per la riproduzione di questa ed un terzo per l'esecuzione della coreografia generata.

Per quanto riguarda la scelta della traccia musicale, viene chiesto all'utente di specificare quale tra quelle disponibili utilizzare. Nel caso in cui si volesse scegliere una traccia non presente nell'elenco, sarà sufficiente aggiungerla in formato .wav nella cartella Music. Uno dei punti cruciali dell'attività progettuale è stato infatti l'eliminazione della dipendenza dell'algoritmo da un'unica traccia musicale cablata nel codice. Viene poi invocato l'algoritmo di analisi di PartyNAO [PartyNAO] sulla traccia scelta. Questo permette di campionare in ogni istante di tempo l'intensità della musica, che sarà il parametro fondamentale su cui gli algoritmi di ricerca genereranno le coreografie. Nell'implementazione originale, vi era un vincolo stringente riguardo la durata della traccia, che era

cablata nel codice. Per evitare questa limitazione, è stata modificata la funzione di analisi in modo da ottenere la reale durata della traccia e basare l'analisi su questa.

Come scelta progettuale, si è deciso di adottare lo stesso pool di mosse in entrambi i progetti al fine di confrontare al meglio le coreografie. Queste sono state categorizzate in 3 sottoinsiemi in base alla loro velocità di esecuzione: mosse lente, normali e veloci. Insieme all'intensità musicale, questa categorizzazione risulta fondamentale per la generazione della coreografia.

## 2 ANaoRhythm

La struttura originale del progetto non ha subito variazioni, ad eccezione del main, come anticipato. Le vere modifiche riguardano l'euristica utilizzata dall'algoritmo di ricerca. Inizialmente infatti l'algoritmo, che è un'implementazione della Iterative Deepening Search, si basava su un'euristica definita come la combinazione lineare tra la durata normalizzata della soluzione e il numero normalizzato di mosse compatibili con il beat (beat matching).

$$h = \alpha \cdot ST + \beta \cdot BM$$

Dove:

- $\beta = 1 - \alpha$ ,
- $ST = \text{Total Time} / 180$  (Durata normalizzata della soluzione)
- $BM = \text{Numero di mosse che partono su un beat} / \text{Numero totale di mosse}$  (Beat Matching)

In questo modo però vi era una stretta dipendenza tra la traccia musicale, da cui dipendeva il parametro BM, e l'euristica. Inoltre qualora si volesse generare una coreografia con durata diversa da 3 minuti (180 secondi), l'algoritmo non avrebbe generato correttamente la coreografia.

A questo proposito è stata definita una funzione *value\_BC* che permette di associare ad ogni mossa un valore compreso tra 0 e 1, dove 1 rappresenta una perfetta corrispondenza tra l'intensità relativa alla traccia musicale in un dato istante e la velocità della mossa. In particolare:

- Match buono (0.9 - 1)
- Match accettabile (0.3)
- Match errato (0 - 0.1)

Quindi il valore BM viene sostituito con un nuovo parametro AV (Average Value) ottenuto come la somma del punteggio di ciascuna mossa il numero totale di mosse.

Inoltre, per rimuovere il vincolo sulla durata della coreografia, è stato aggiunto un nuovo parametro, personalizzabile dall'utente all'avvio, che permette di definire la durata del ballo.

Dunque è stato modificato il codice in modo da garantire all'utente un maggior grado di libertà anche relativamente alle mosse da eseguire. In particolare all'avvio l'utente può scegliere quali mosse obbligatorie devono essere incluse nella coreografia e quali, tra quelle non obbligatorie, può rimuovere. Si riporta un esempio di esecuzione.

```
Insert Robot IP: 127.0.0.1
Insert Robot Port: 62313

AVAILABLE SONGS:
1 : Arctic_Monkeys_Do_I_Wanna_Know_3MIN.wav
2 : Muse_Feeling_Good_3MIN.wav

Which song would you like to play? Choose the number: 1

Set the duration of the choreography (in seconds): 30

You chose the song || Arctic_Monkeys_Do_I_Wanna_Know_3MIN.wav || NAO will dance for: 30 seconds!

AVAILABLE MANDATORY POSITIONS:
0 : M_Hello
1 : M_Stand
2 : M_StandZero
3 : M_Sit
4 : M_WipeForehead
5 : M_SitRelax
Choose the mandatory position you want (integer): 0 1 2

AVAILABLE OPTIONAL POSITIONS:
0 : StandUp
1 : AirGuitar
2 : ArmDance
3 : BlowKisses
4 : Bow
5 : DiagonalRight
6 : DanceMove
7 : SprinklerL
8 : SprinklerR
9 : TheRobot
10 : ComeOn
11 : StayingAlive
12 : Rhyth
13 : PulpFiction
14 : Wave
15 : Glory
16 : Clap
17 : Joy
Choose the optional position you want to remove (integer): 0

Starting moves: ▶ I_StandInit ▶ M_Hello ▶ M_Stand ▶ M_StandZero ▶ F_Crouch
Computing Sequence: 8
```

```
Starting moves: ▶ I_StandInit ▶ M_Hello ▶ M_Stand ▶ M_StandZero ▶ F_Crouch
Computing Sequence: 8

Computing Time: 0.012689352035522461 seconds
Number of epochs: 7

STATS:
Best Move Sequence:
I_StandInit ▶ StandUp ▶ TheRobot ▶ M_Hello ▶ M_Stand ▶ BlowKisses ▶ M_StandZero ▶ F_Crouch

Total Time: 30.009999999999998s
Nodes in the last Tree: 13
Dance!
Move: I_StandInit
Move: StandUp
Move: TheRobot
Move: M_Hello
Move: M_Stand
Move: BlowKisses
Move: M_StandZero
Move: F_Crouch
```

### 3 Silk-Sonic

Per poter confrontare al meglio le coreografie scelte dai due algoritmi di ricerca, si è cercato di mantenere una continuità tra i due progetti in modo che differissero solo nello scegliere le mosse da eseguire mentre fossero pressochè analoghi

nella struttura generale del codice. Per questo la struttura del main e le mosse utilizzate sono equivalenti a quelle di ANaoRhythm. Originariamente il progetto era compatibile con le versioni precedenti di Python, nello specifico Python2. Per garantire maggiore portabilità è stato modificato il codice in modo da poter eseguirlo con Python3, mantenendo Python2 per la sola esecuzione delle mosse dal momento che le librerie utilizzate da Coreographie sono compatibili solo con Python2. Per quanto riguarda l'algoritmo di ricerca è stato mantenuto quello del progetto iniziale aggiungendo però la stessa funzione di valutazione, usata anche in ANaoRhythm, (*value\_BC*).

Anche in questo progetto l'utente può scegliere quali mosse obbligatorie fare eseguire e quali, tra quelle non obbligatorie, può rimuovere.

Si riporta un esempio di esecuzione.

```

IP: 127.0.0.1
Port: 54411

AVAILABLE SONGS:
1 : Arctic_Monkeys_Do_I_Wanna_Know_3MIN.wav
2 : Muse_Feeling_Good_3MIN.wav

Which song would you like to play? Choose the number: 1

You chose the song || Arctic_Monkeys_Do_I_Wanna_Know_3MIN.wav || let's dance nao!

AVAILABLE MANDATORY POSITIONS:
1 : I_StandInit
2 : M_WipeForehead
3 : M_Stand
4 : M_Hello
5 : M_Sit
6 : M_SitRelax
7 : M_StandZero
8 : F_Crouch

Choose the mandatory position you want to remove (integer): 6

AVAILABLE OPTIONAL POSITIONS:
0 : StandUp
1 : AirGuitar
2 : ArmDance
3 : SlowKisses
4 : Bow
5 : DiagonalRight
6 : DanceMove
7 : SprinklerL
8 : SprinklerR
9 : TheRobot
10 : ComeOn
11 : StayingAlive
12 : Rhyth
13 : PulpFiction
14 : Wave
15 : Glory
16 : Clap
17 : Joy

Choose the optional position you want to remove (integer): 15

```

```
Move: I_StandInit
Move: ArmDance
Move: BlowKisses
Move: SprinklerL
Move: Glory
Move: Rhythm
Move: M_WipeForehead
Move: StayingAlive
Move: ComeOn
Move: Bow
Move: Wave
Move: ArmDance
Move: M_Stand
Move: Pulpfiction
Move: TheRobot
Move: Glory
Move: StayingAlive
Move: DanceMove
Move: M_Hello
Move: Wave
Move: Clap
Move: Pulpfiction
Move: Rhythm
Move: BlowKisses
Move: M_Sit
Move: StandUp
Move: M_StandZero
Move: Rhythm
Move: DanceMove
Move: AirGuitar
Move: SprinklerR
Move: Wave
Move: F_Crouch
```

## 4 Bibliografia

- [NAOPlanningChallenge] - <https://github.com/ProjectsAI/NAOPlanningChallenge/tree/main>
- [PartyNAO] - <https://github.com/helemanc/PartyNAO/tree/main>
- [Silk-Sonic] - <https://github.com/ProjectsAI/NAOPlanningChallenge/tree/main/2021-2022/Silk-Sonic>
- [ANaoRhythm] - <https://github.com/ProjectsAI/NAOPlanningChallenge/tree/main/2021-2022/anaorhythm>

Il codice di entrambi i progetti è reperibile su GitHub: <https://github.com/BenedettaRogato/NAO>