ROBOerTo Bolle

Team

Pietro Tellarini

Giorgia Bravi

HEURISTIC

The evaluation function, driven by heuristics, determines the node to expand. The algorithm produces the optimal sequence of moves after considering the intensity of the music, the move history, and the time. In this function, the time component carries a weight of 50%, while the intensity of music and the avoidance of move duplication each contribute 25%. All the parameters related to the function are normalized to 1.

Time

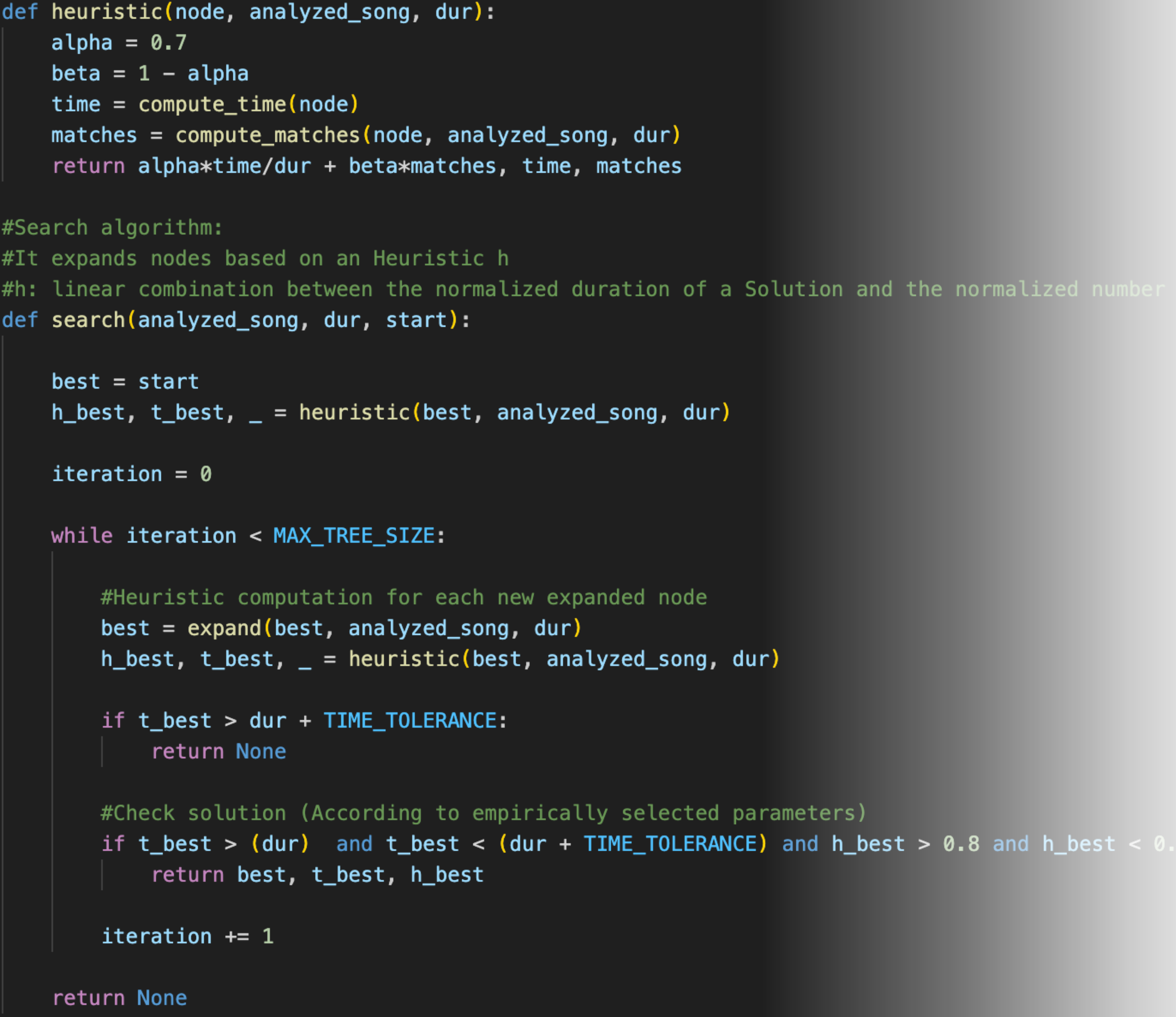
Assign the time to the single move to respect the maximum time of the choreography.

Moves History

The duplicated moves will give a lower score to the sequence, 1/number of repetitions.

Music

Analise the intensity of the music and assign to the different fragment of intensity fast, normal or slow moves



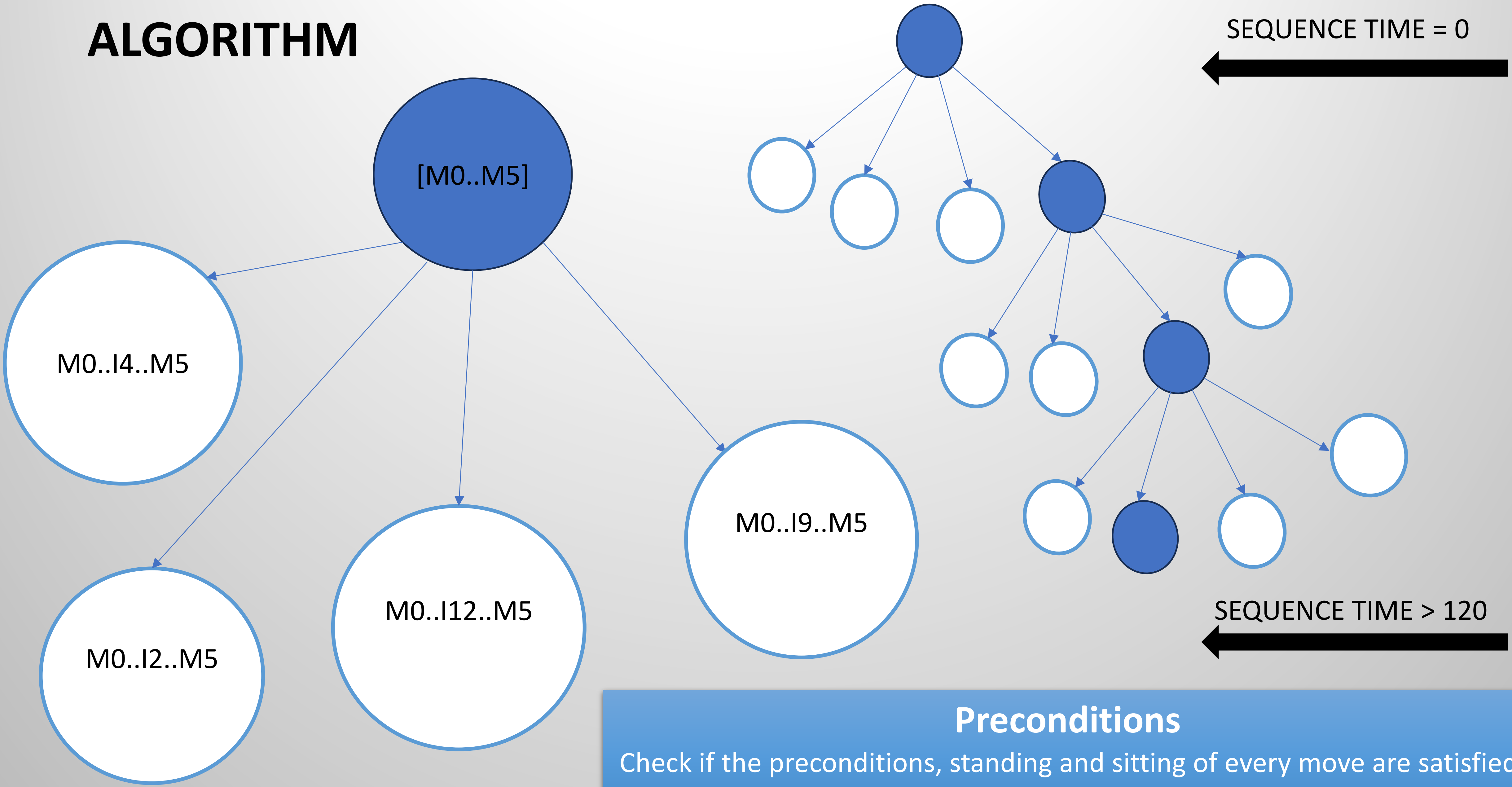
```
def heuristic(node, analyzed_song, dur):
    alpha = 0.7
    beta = 1 - alpha
    time = compute_time(node)
    matches = compute_matches(node, analyzed_song, dur)
    return alpha*time/dur + beta*matches, time, matches

#Search algorithm:
#It expands nodes based on an Heuristic h
#h: linear combination between the normalized duration of a Solution and the normalized number of ma
def search(analyzed_song, dur, start):

    best = start
    h_best, t_best, _ = heuristic(best, analyzed_song, dur)

    iteration = 0

    while iteration < MAX_TREE_SIZE:

        #Heuristic computation for each new expanded node
        best = expand(best, analyzed_song, dur)
        h_best, t_best, _ = heuristic(best, analyzed_song, dur)

        if t_best > dur + TIME_TOLERANCE:
            return None

        #Check solution (According to empirically selected parameters)
        if t_best > (dur)  and t_best < (dur + TIME_TOLERANCE) and h_best > 0.8 and h_best < 0.
            return best, t_best, h_best

        iteration += 1

    return None
```

Planning

The Algorithm return the best sequence of the moves based on a heuristic parameter.

The evaluation function is used to decide which node is going to be expanded, according to the Greedy Best Search Strategy.

ALGORITHM



[M0..M5]

M0..I4..M5

M0..I2..M5

M0..I12..M5

M0..I9..M5

SEQUENCE TIME = 0

SEQUENCE TIME > 120

Preconditions

Check if the preconditions, standing and sitting of every move are satisfied

THANKS for your ATTENTION