Alma Mater Studiorum · Università di Bologna
Campus di Cesena

**DIPARTIMENTO DI INFORMATICA – SCIENZA E INGEGNERIA**
**Corso di Laurea Magistrale in Ingegneria e Scienze Informatiche**

# TOWARDS ARTIFICIAL CREATIVITY: EVOLUTIONARY METHODS FOR GENERATING ROBOT CHOREOGRAPHIES

**Tesi in:**
**SISTEMI INTELLIGENTI ROBOTICI**

**Relatore:**
**Prof.**
**ANDREA ROLI**
**Co-Relatore:**
**Dott.**
**MATTIA BARBARESI**

**Presentata da:**
**STEFANO BERNAGOZZI**

**Sessione II**
**Anno Accademico**
**2019/2020**

# Abstract

Today robotics is widely used in many fields, from simple houseworks like floor cleaning to more complex tasks like rescuing people in dangerous situations such as earthquakes. Recently it has been expanding to a more creative field: entertainment. For this reason we have thought of developing a genetic algorithm that allows the robot to dance, starting from the codification of movements in order to achieve the creation of true choreographies. We start by analysing Noh choreographies, and then we transpose them onto a humanoid robot, Nao. We then proceed by going through the implementation of an algorithm that allows the creation of choreographies. One of the hardest challenges that we will face is to create choreographies that are both faithful to Noh theater and new at the same time. We will conclude focusing on the evaluation criteria of the results and presenting some hypothesis for future developments in this field.

# Contents

# List of Figures

# List of Tables

# Introduction

Today's robots are becoming more and more present in everyday life and as the research continues they look more and more similar to human beings. Robots like Atlas by Boston Dynamics [Boston Dynamics, 2020a], Asimo by Honda [Honda, 2011] or Nao by Softbank Robotics [Softbank Robotics, 2011a] can reproduce human movements and sense the world around them with great precision and with graceful gestures. Alongside them a wide range of robots are being developed for both industries and consumers, from fish robots that explore the sea to quadruped robots like Spot by Boston Dynamics [Boston Dynamics, 2020b], passing through entertainment robots like Vector by Anki [Anki, 2020] and AIBO by Sony [Sony, 2018]. In recent years these robots are becoming available for private use and entertainment, and companies like Boston Dynamics or Tim have used them to create entertainment and make them dance [Boston Dynamics, 2018] [Tim, 2018].

Furthermore many researchers started to look closer at creativity in computer programs, from music composition [Marques et al., 2000] to choreography creation[Peng et al., 2016], passing through poetry composition [Manurung, 2004], and art creation [Vinhas et al., 2016].

The problem that we are facing is *how to create novel choreographies based on a real environment like the Noh and how a robot can execute them*.

This means we have to define a codification for a choreography that is computer readable and that can be elaborated. After this the codification we will subsequently find an algorithm that creates new choreographies given some basic restrictions, in our case Noh theatre, to be executed on the robot and judged by human experts.

Starting from the first chapter we will analyse the creation of choreographies in Noh theatre and subsequently we will transpose them on a humanoid robot. After this step, in chapter 5 we will explain how to generate new choreographies with a genetic algorithm, while in chapter 6 and 7 we will discuss results evaluation and parameters tuning. Then in chapter 8 and 9 we will discuss the obtained results and how the algorithm performs. Finally in chapter 10 we will discuss the future works related to this project.

# Chapter 1

# State of the art

## 1.1 Novelty search in evolutionary computation

Currently evolutionary computation is widely used and studied in computer science to solve many classes of problems, from optimisation to constraint satisfaction problems [Eiben and Smith, 2015] [Michalewicz and Schoenauer, 1996] [Yang, 2015]. Many fields of study have taken advantage of this type of computation due to its easy usability: in fact for most problems you only need to specify the fitness function and a simple individual representation. An annual conference regarding this topic is Evomusart and a relevant subcategory of this large topic is the novelty search in evolutionary computation. It uses this type of computation not only to maximise a fitness function, but also to explore the search space to search other local maximum or new individuals. The reference work for this type of computation is made by Stanley and Lehman, who implemented the minimal criteria novelty search [Lehman and Stanley, 2010] [Lehman and Stanley, 2011], where they define a minimal criteria that the individual must satisfy instead of a single fitness function. This is particularly important because it sets a dif-

ferent standard, one that is not based on maximising a fitness function but on a fitness threshold that individuals must pass and the exploration of all individuals, without having to maximise the fitness. This leads to a novelty search were all individuals are feasible, but novel. Based on this work there are other works, like a paper made by Gomes et al. [Gomes et al., 2012] where they implemented an algorithm with a progressively stricter fitness criterion to achieve better results. Other improvements have been made by Vinhas et al. [Vinhas et al., 2016] where they revised the algorithm with a slightly different dissimilarity function and, differently from the MNCS, they maximise both fitness and novelty using the Pareto front. They also limit the set for the calculus of novelty to a fixed size instead of considering the whole set composed by population and archive as in Lehman's work.

Another way to exploit novelty in evolutionary computation is the one that uses interactive genetic algorithms to take advantage of human evaluation during the evolution, like [Manfré et al., 2016] who have used it to create a humanoid dance.

Other works related to novelty can be found in [Gomes et al., 2015] although they are all based on some kind of neural network instead of an evolutionary algorithm.

Concerning the implementation of creative works and their evaluation two methods are the main reference:

- **Machine learning and deep learning**, used by [Datta et al., 2006]and [McCormack and Lomas, 2020] to implement a classifier for art aesthetic.

- **External evaluation approach**, as in [Takagi, 2001], [Sun et al., 2012], [Vircikova and Sincak, 2010] and [Manfré et al., 2016] that requires an

external evaluation in addition to the fitness value and judges individual's aesthetic.

- **Semi supervised learning**, as in [Sun et al., 2013] and [Peng et al., 2016] that combines human evaluation with machine learning to evaluate individuals.

## 1.2 Evaluation of creativity

The subject of creativity has been largely discussed. Regarding creativity in computer programs the first and main book is Boden's "The creative mind: myths and mechanism"[Boden et al., 2004]. Since its publication in 1990 it has been a reference for all the works related to creativity in general, lingering on the argument of computational creativity. In her work she defines creativity as "the ability to come up with ideas or artefacts that are new, surprising and valuable". She also brought the function from a discrete boolean set to a continue set. In fact the work looks at giving a percentage of novelty to each artefact instead of classifying each as novel or not. Deriving from Boden's work there are many other articles about the evaluation of creativity, but the ones that we have deemed more relevant to our work are:

- [Wiggins, 2006] that used her criteria to create a mechanism to formally apply the idea of creativity to algorithms and results.

- [Ritchie, 2007] that creates a set of criteria to evaluate the "ability" of a program to be creative, based on Boden's work.

# Chapter 2

# Noh theatre

## 2.1 A bit of history

Noh [Larsen, 2020] [japan guide, 2020] [the-NOH, 2020] [Fenollosa and Pound, 2004] is a japanese drama performance which requires many abilities of dancing and acting (in japanese Noh means skill or talent). It is a form of theatre composed of music, dance and drama and its origin can be found in Sangaku, a form of performance art similar to circus imported by Japanese people from China. Due to its old origin, the Noh is one of the oldest extant theatrical forms in the world.

The discipline, that can be seen today in Nipponic theatres, was born in the 14th century by the hand of Kan'ami Kiyotsugu and his son, who brought it to the Imperial Court. It is part of the UNESCO list of Intangible Cultural Heritage and is still largely practised in Japan. The main theatre is the Noh national theatre in Tokio. The performers act more as storytellers who narrate the play instead of making a true recital, indeed stories are taken from classical Japanese literature and they are an ensemble of miths and true historical events.

## 2.2 The performance

### 2.2.1 Characters

All characters present in Noh theatre can be divided into 5 categories:

- **Shite** - This is the leading character of the stage. He can assume different role depending on the performance, like old man, deity, spirit or a true living man.

- **Waki** - This is the supporting actor. Like the Shite he can cover different role like priest, monk or samurai. In contrast to the Shite, the Waki always portrays living people.

- **Hayashi** - The Hayashi is the group of musicians. Composed by four elements it is also known as the shibyōshi. They provide accompaniment for the performance with a flute (fue), shoulder drum (kotsuzumi), hip drum (otsuzumi) and stick drum (taiko).

- **Jiutai** - It's the chorus. Normally composed by six to eight people it sits to the right of the stage in the jiutai-za and assists the shite in the narration of the story.

- **Koken** - They are the stage attendants. Normally they are one to three people dressed in black, they assist the performers in various ways, such as handing them props. They also dress the actors and come up if something goes wrong in the performance.

All the actors and actresses perform both male and female roles, in fact the gender of each role is not clearly defined.

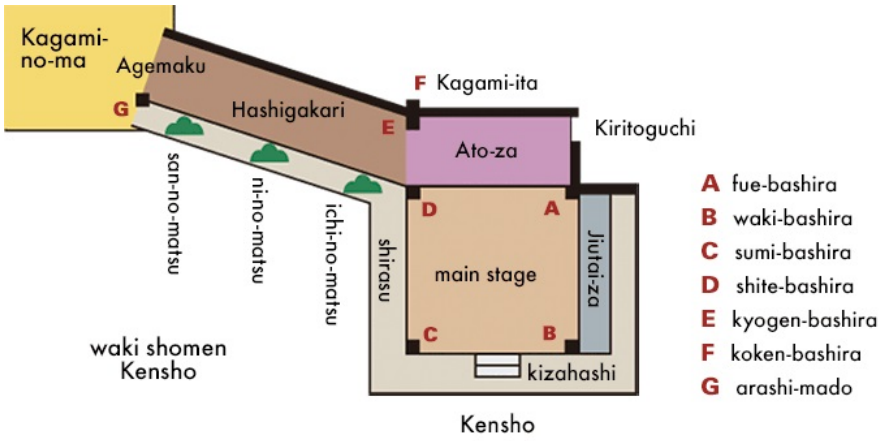



**Figure 2.1:** A plant of Noh stage(source https://www.the-noh.com/)

### 2.2.2 The Stage

The stage is constituted by the Hon-butai (main playing area) composed by a square of 5.4 meters per side, hashigakari (bridgeway), ato-za (seating section for musicians and stage attendants) and the jiutai-za (seating section for the chorus). A plant of the whole stage can be seen in figure 2.1

### 2.2.3 Costumes, props and masks

**Costumes:** in Noh costumes are called noh shōzoku. In the early life of noh, costumes for the performance were composed by everyday clothes, but as the theatre became more popular costumes began to be more fine and crafted. In recent years they have become a truly work of art and separated from the actor that wears it. They can be divided into 7 categories:

- Kahatsu: all the things related to hair, like kazura (wig).
- Kaburi-mono: costumes that a performer can wear on the head, like eboshi (hat) or tengan (headdress).
- Uwagi: various garments worn over kimono.

- Kitsuke: a short sleeved kimono worn under outer kimono.
- Uwagi / Kitsuke The outer kimono, the most beautiful piece of cloth.
- Hakama: a pair of japanese loose-fitting trousers.
- Other small items like sashes used for tying things on.

**Props:** Props in Noh are used to enhance the expressiveness of the gesture. Most of them employs the use of tsukurimono (made things) and they are not realistic representations but more symbolic: for example a folding fan, that is one of the main props, can be a lantern, a sword or a shield.

**Masks:** In Noh there are over 200 different types of masks. They are usually made with wood and sculpted by artisans because they need to fit exactly to the character played by the actor.

### 2.2.4 Our Noh choreographies

Since not all Noh choreographies can be reproduced on a robot due to quick movements or complex ones that might put the robot out of balance, we have chosen to make some basic choreographies inspired by simple stories.

Our Noh choreographies are based mainly on 2 characters: the warrior and the priest. Warrior choreographies contain fighting, such as defence and attack movements. The other ones portray the priest while praying, offering and asking for objects, as well as showing sadness.

# Chapter 3

# First steps on the robot

## 3.1 Nao overview

Nao® is an humanoid robot made by Softbank robotics [Softbank Robotics, 2011a]. As can be seen in 3.1, it has 25 degrees of freedom, controlled by 23 servo motors:

- 2 in the head (pitch and yaw)
- 10 in the arms (5 for each arm: shoulder pitch and roll, elbow roll and yaw and wrist yaw)
- 1 in the hip yaw (one motor controls the yaw of both legs)
- 10 in the legs (5 for each leg: hip pitch and roll, knee pitch and ankle pitch and roll)

It has 7 touch sensors located in head, hands and feet, in addition to sonars and an inertial unit to sense the environment around him and locate himself in space. Moreover it has 2 two-dimensional cameras to recognise the world around and it is open and fully programmable. It is 58 cm tall and

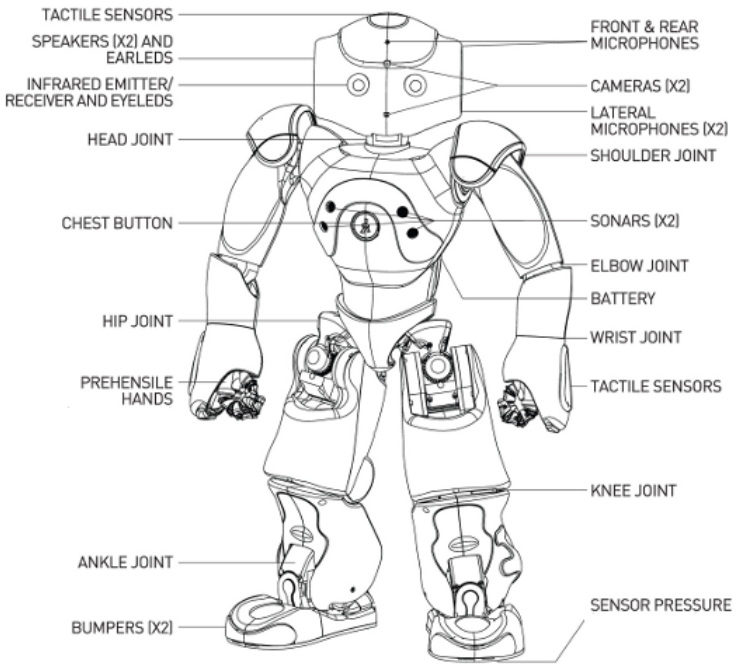



**Figure 3.1:** Nao H25 (image taken from `http://doc.aldebaran.com/2-1/family/nao_h25/index_h25.html`, copyright Softbank Robotics)

weights approx 5.4 kilos.

The main reason behind the choice of this robot is that it is fully programmable and it can be controlled remotely through the Naoqi framework [Softbank Robotics, 2011b]. In addition to that it has been used in Robocup ®[Robocup Federation, 2020] [Kitano et al., 1997], a worldwide soccer competition played by robots.

Nao is simulated in many environments and due to the closure of universities we have chosen to use CoppeliaSim [Coppelia Robotics, 2020] in order to have a simulation similar to reality. The scene used is the one made by Pierre Jacquot [Jacquot, 2015]. There is an official simulator by Softbank Robotics made in Webots [Cyberrobotics, 2020] and Choregraphe for the simulation but the first one is no longer supported and it doesn't work in

recent operating systems, while the second one has no physical motor for equilibrium, so the robot doesn't fall in case it loses balance.

## 3.2 Initial moves

Since the main purpose of this thesis is to create a true Noh choreography, we decided to use a robot to perform it. The chosen one was a Nao due to its easy manoeuvrability and the large software support.

At the beginning the obstacle proved to be creating a simple choreography with the robot. We tried to solve this issue in various ways:

- The first one involved using Microsoft Kinect® v1 [Microsoft, 2010] to capture the human skeleton and then to transpose each pose into a list of Nao's joint to reproduce it.
- The second one was based on the discretisation of the choreographies into fixed poses and the transition between them, with time as an important factor.
- The last option considered was to use Nao's API to make the robot walk and to set his arms to a given position.

### 3.2.1 First attempt: Microsoft Kinect

Our first attempt was made with kinect: we found a program [Pinson, 2018] to collect data from the sensor, elaborate them and send to the Nao. Unfortunately it was only for the upper limbs and it has a considerable error rate, so each pose had to be modified manually.

Despite Microsoft's sensor had a high performance and it was easy to create a large number of choreographies, it revealed some problems: The

noise rate of the human skeleton captured was quite high, preventing an accurate reproduction of the moves on the robot, mainly on the legs and to a lesser extent on the performer's arms. This required to validate and correct each position with direct and inverse kinematics, which was not our purpose. In addition the dancer must be right in front of the camera, which limits the movements and doesn't allow the performer to rotate or assume side poses. We attempted many kind of corrections, but since the result was not good enough we opted for other options.

We tried with another program made by Pourya Shahverdi [Shahverdi and Masouleh, 2016], who took a version of ROS [ROS community, 2020] with Kinect libraries to make Nao reproduce human movements with both legs and arms. The problems of this program were mainly:

1. The library has been discontinued and it is not usable with newer version, requiring too much code porting;

2. The library uses continuous monitoring of the equilibrium of the Nao, and to get stable poses each one has to be checked manually (as with the previous program).

Due to this problems the Kinect was discarded, since our main purpose was to focus on the algorithm.

### 3.2.2 Second attempt: making static poses with Nao

The only option left, with the purpose in mind to create a choreography, was to discretise the choreographies into basic poses and then codify a choreography as a list of poses with time of execution. At this stage the possibilities were either to take the physical robot and make the poses with it or to make each pose with Choregraphe and then transpose them to the

real Nao. Due to the impossibility of accessing universities because of the Corona virus, the method chosen was to manually create each pose with Choregraphe® [Softbank Robotics, 2015], a suite made by Softbank robotics that allows the user to create a virtual robot and to set the joint angles. This enables the visual creation of poses with low effort and allows retrieving the angles for later use.

Each pose was codified as a Json structure with two fields: a list of joint angles and the support leg (or legs), as can be seen below.

```
{
    "supportLeg": "Legs",
    "angles": {
        "LShoulderRoll": 29.5,
        "LShoulderPitch": 69.2,
        "LElbowRoll": -38.2,
        "LElbowYaw": -7.7,
        "LWristYaw": -58.9,
        "LHand": 1,

        "RShoulderRoll": -12.1,
        "RShoulderPitch": -5.9,
        "RElbowRoll": 43.4,
        "RElbowYaw": -12.5,
        "RWristYaw": 94.9 ,
        "RHand" : 1.0,

        "HeadPitch": 0,
        "HeadYaw": 0
```

```
20
21          "LHipYawPitch" : -12.4,
22          "LHipRoll": -0.6,
23          "LHipPitch": -22.6,
24          "LKneePitch": 54.0,
25          "LAnklePitch": -23.0,
26          "LAnkleRoll": 1.3,
27
28          "RHipRoll": -1.3,
29          "RHipPitch": -2.5,
30          "RKneePitch": -48.3,
31          "RAnklePitch": -37.0,
32          "RAnkleRoll": 0.7
33      }
34  }
```

The first poses only involved the upper limbs. The main reason behind this choice is that it allows us to not take into consideration the balance problem and to focus on the algorithm instead.

After the creation of some basic poses we have tried to add some foot movements, starting from a basic forward walk. For this movement there were two options: make it walk through Naoqi's API or calculate the foot positions with inverse kinematics. The second one was discarded since it requires deep kinematics knowledge and much more time.

# Chapter 4

# The creation of a choreography: bringing Noh on Nao

## 4.1 The first choreography on Nao

To create base poses consistent with Noh theatre we used many video tutorials created ad-hoc by a professional Noh performer, who has made for us many basic movement of her artistic discipline to grant a faithful reproduction. Those videos have been analysed to get the basic poses. Since many poses are very similar and they differs by only few movements, we decided to reduce the cluster into a single pose common for all.

The chosen poses are shown in figure 4.1

Each figure matches a particular position, but not all the poses have a meaning: warrior and priest poses are distinguished but a subset is common for both the characters, like the standard pose A. more deeply:

- **A** this is a pose with arms down, the standard pose of Noh theatre.
- **B** this is a pose where the robot keeps the folding fan closed in the right

arm to signify a straight sword (it is related to the warrior).

- **C** this pose is the composition of two poses: with the right hand the robot is holding a fan closed to signify a straight sword as the previous pose, while with the left hand it is making a request (the request is one of the standard poses in Noh)(it is related to the warrior).
- **G** this is a pose where the robot has the arms to his chest to signify extremely sadness (it is related to the priest).
- **H** this is the same pose as **C** but with arms inverted (it is related to the warrior).
- **J** this is a pose where the robot mean sadness with the right arm at chest (it is related to the priest).
- **K** this is a pose where the robot is requesting something with the right arm (it is related to both warrior and priest).
- **L** this is equivalent to **K** but the robot is requesting with both arms (it is related to both warrior and priest).
- **M** in this pose the robot is offering something with both arms (it is related to the priest).
- **O** here the robot is expressing sadness with his left arm in front of his face (it is related to the priest).
- **P** this pose is the same as **O** but with both arms (it is related to the priest).
- **Q** this pose is the same as **O** but with the right arm (it is related to the priest).

- **T** here the robot has the arms raised as a sign of prayer (it is related to the priest).
- **U** in this pose the robot has the left arm folded meaning it has a shield in his arm (it is related to the warrior).
- **V** this pose is the same as **U** but with the right arm (it is related to the warrior).
- **D,E,F,I,N,R,S** are passage poses without a particular meaning.
- **W** this pose is not shown but it is a 90 degree rotation to the right.
- **X** this pose is not shown but it is a 90 degree rotation to the left.
- **Y** this pose is not shown but it is a walk backward.
- **Z** this pose is not shown but it is a walk forward.

After the formalisation of the poses we took some Noh performances (see section 2.2.4) and then we decomposed them into a list of our poses. In order to have a standard length for our choreographies we have chosen a fixed length of 16 poses for each choreography.

At the beginning the robot was standing on his feet without making any step, as the poses can show. This led to a very static choreography, so lately we decided to add some basic steps as "poses": move forward, move backward, turn right and turn left (poses W,X,Y and Z).

This last passage allows us to make poses more dynamic, since in Noh theatre actors walk around the stage often.

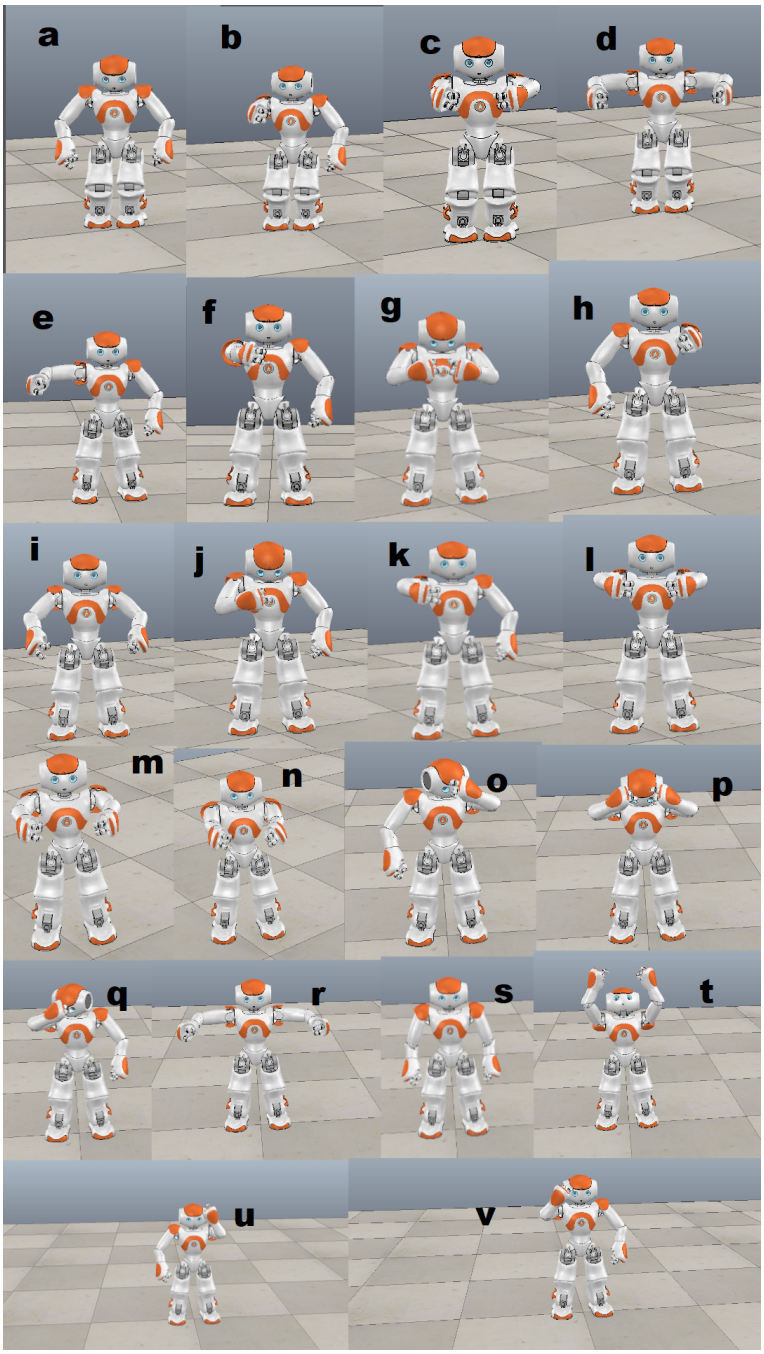



**Figure 4.1:** In the image are shown the poses made for the algorithm. They are made on Coppelia simulator.

## 4.2 Base choreographies

We made 6 base Noh choreographies (a video example can be found here [Bernagozzi, 2020b]) to later use them as models for our new choreographies, as described in section 5.2.2. These choreographies were divided into two characters, the priest and the warrior, to have more than one category. At the moment they were unused since we have not implemented the algorithm to distinguish between them, but in the future it can be a good improvement. The choreographies made are:

- for the priest:
  - azrtxwwgxyntzmay
  - aljgzrtmydiaklra
  - azpxowwqxmnldylm
- For the warrior:
  - anbzhiuayvbzubay
  - avnevbhzxbwwwzcr
  - abcewhvxfnubzrha

# Chapter 5

# How to generate novelty with a genetic algorithm

## 5.1 Evolutionary computation

Evolutionary computation is the application of Darwin's evolutionary theory to problem solving. It was born as different fields such as evolutionary programming, genetic algorithm and evolution strategies but since 1990 they're viewed as part of the same field: evolutionary computing. The relation between problem solving and evolutionary computation can be seen in table 5.1 An evolutionary algorithm is composed by six main parts, as described in [Eiben and Smith, 2015]:

- Representation
- Evaluation function
- Population
- Parent selection mechanism

| Evolution | Problem Solving |
|---|---|
| Environment | Problem |
| Individual | Candidate Solution |
| Fitness | Quality |

**Table 5.1:** The evolutionary metaphor (source [Eiben and Smith, 2015])

- Variation operators, recombination and mutation
- Survivor selection mechanism (replacement)

**Representation:** The representation is the link between the real world and the algorithm: the original problem context and the problem solving space must be related and each individual must match a possible solution in the problem space.

**Evaluation function:** The evaluation function corresponds to a good quality of the solution proposed or rather a requirement that the population should meet. The function is applied to each individual at each generation and it is used for the parent selection mechanism

**Population:** A population is a set of individuals that represents part of the possible solutions. It is the changing part of the algorithm, since individuals are static and they do not change, new individuals are made instead.

**Parent selection mechanism:** The purpose of the parent selection is to choose the best individuals (or most of them) to allow them to become parents of the next generation and to create with variation operators the offspring

**Variation operators** The variation operator are used to create new individuals from the parents selected by the mechanism above. They are divided in two types: a unary operator (mutation) and an n-ary operator (recombination).

- *Mutation:* is applied to an individual and returns a slightly different individual based on a statistical process.
- *Recombination:* is an n-ary operator that takes two or more individuals and merges their information into one or more new individuals.

The variation operators are representation dependent.

**Survivor selection mechanism** The last important part is the survivor selection mechanism, or replacement. This is similar to the parent selection mechanism but, differently from the previous one, it selects, from a number of individuals larger than the size of the population, n individuals (where n is the size of the population) to create the population for the next generation.

## 5.2 The novelty algorithm

Because our algorithm needs to create choreographies both novel and faithful to Noh we have chosen to implement a multi-objective algorithm with both novelty and fitness inside. It is based on the work "Fitness and novelty in evolutionary art" [Vinhas et al., 2016], where the authors have made an evolutionary computation to create "images that are both suitable and diverse". In particular the focus of the research was on novelty search in evolutionary algorithms. Vinhas and the others compare 2 types of evolution: the first one has the population composed by figurative images [Correia

| ***Element*** | ***Our implementation*** |
|---|---|
| Representation | List of characters |
| Evaluation function | Fitness function as 5.2.2 and novelty function as 5.2.3. The choice of the evaluation function is based on the number of individuals as 5.1 |
| Population | set of list of strings |
| Parent selection mechanism | Depending on the number of feasible individuals, SPEA2 when the evaluation method is hybrid or tournament selection when the evaluation method is only fitness |
| Variation operators | *Mutation:* |
| Replacement | This is not implemented since the parent selection mechanism alongside to variation operators returns a set of 100 elements that is the size of our population. |

**Table 5.2:** The elements of our evolutionary computation

et al., 2013], while the second one has Context Free Design Grammars as individuals [Horigan and Lentczner, 2014] [Horigan and Lentczner, 2015].

The main difference from a standard genetic algorithm is that it can switch the evaluation method according to the number of individuals with a fitness higher than a threshold. This allows the evolution to converge to better individual when the evaluation is made only with fitness, while when the evolution is hybrid it evaluates also individuals with slightly lower fitness but different from the ones already seen.

Another important fact is that the results are not the one obtained from the whole evolution, but feasible individuals are picked and saved into an archive that subsequently will be the set of our results.

Our evolutionary computation parts, as explained in 5.1, are defined in table 5.2.

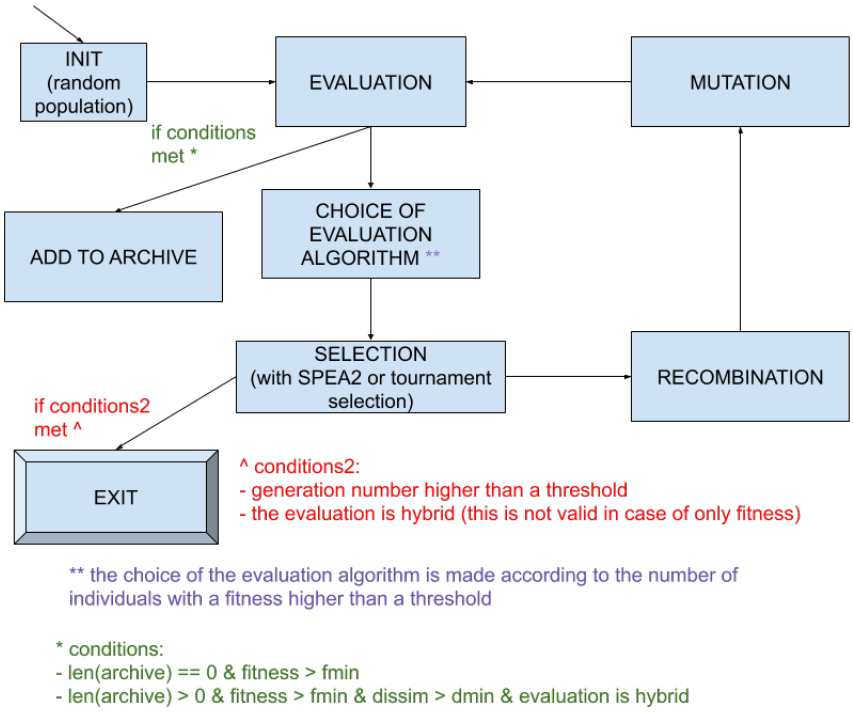



**Figure 5.1:** A diagram of the genetic algorithm used in our program.

### 5.2.1 The archive

The archive is the most important part of the algorithm since it contains the future results. It also functions as database for the evaluation of future individuals with a high fitness value, because they must be different from the one already present in the archive. more deeply the individuals that will be inserted in the archive must meet some requirements:

- if the archive is empty the first individual with its fitness value above *fitness_threshold* is added
- if the archive has one or more element then the individual is added only if its fitness is value above *fitness_threshold* and its dissimilarity is above *dissim_threshold*.

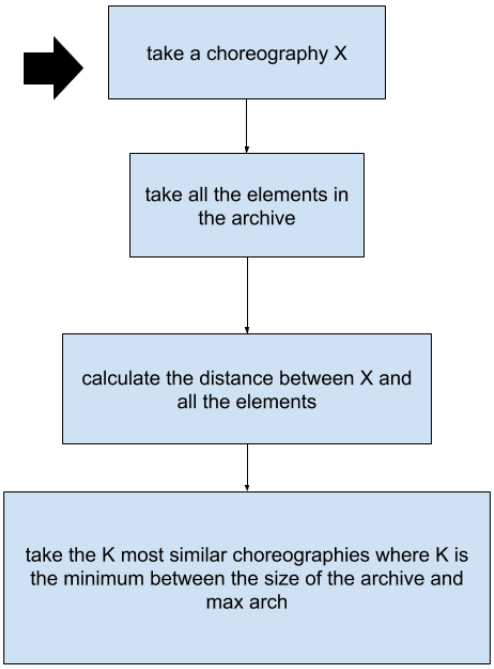



**Figure 5.2:** A diagram of the calculus of dissimilarity

The dissimilarity of an individual is calculated as figure 5.2. In particular each element of the archive is evaluated with its distance between the selected individual with equation 5.4, then $k$ is calculated as equation 5.1 where $max_{arch}$ is a pre-defined parameter.

$$k = \min(len(archive), max_{arch}) \tag{5.1}$$

### 5.2.2 Fitness function

At the beginning the first alteration of the algorithm has been to fitness function because our representation was different from the one mentioned

in the paper. Due to this factor we have decided to implement the fitness function as a string similarity between the individual and a given repertoire. The repertoire is a set of predetermined choreographies (strings) that are based on Noh theatre and each choreography has a particular main role, chosen between priest and warrior. As can be seen in 4.1 also the poses are different and related to one or both characters, this will be useful for further explorations on the algorithm.

The chosen fitness is represented in 5.2

$$fitness(x) = \sum_{n=1}^{len(repertoire)} similarity(rep[n], x) \tag{5.2}$$

The similarity mentioned in 5.2 is defined as function 5.3 [1]:

$$similarity(x, y) = 1 - \frac{JaroWinker(x, y) + Jaccard(x, y)}{2} \tag{5.3}$$

### 5.2.3 Novelty function

As mentioned above, our algorithm has both fitness and novelty so two different evaluation function must be chosen. The second one is the novelty, that in our case is representing how a choreography is different from the ones already seen or present in the archive. This function is quite interesting since it considers both the archive and the current individuals in the population and takes the most similar to the chosen individuals. The individuals are chosen in 2 steps:

- The first step is composed by a tournament selection between all the other individuals in the population: each individual is evaluated with

[1] for Jaro-Winkler and Jaccard see [Cohen et al., 2003]. Here both functions are considered as string distances, so both are 1 when the string are completely different

5.3 where the two parameters are the individual to evaluate and each individual from the population (excluded the one to evaluate) after that a tournament selection with $k = 4$ and $tournament\_size = 5$ is made to select the most similar individuals.

- The second step is to add the individuals from the archive with the same evaluation as above to the ones selected from the tournament selection and then select the best 4 individuals from that pool. This returns the 4 most similar individuals to the individual to evaluate.

After that it considers the distance between the individual to evaluate and the ones selected. The similarity function used to choose the most similar individuals is the same as 5.3, while the one used to calculate the distance is equation 5.4.

$$dissimilarity(x, y) = \frac{JaroWinker(x, y) + Jaccard(x, y)}{2} \tag{5.4}$$

A diagram of novelty evaluation can be seen in figure 5.3

### 5.2.4 Evaluation method and parent selection mechanism

As mentioned in table 6.2 our evolution has two evaluation methods and two parent selection mechanism: a fitness evaluation used together with tournament selection and a hybrid evaluation used together with SPEA2 selection. Both evaluation and selection switch according to 2 parameters, $T_{max}$ and $T_{min}$, following equation 5.5

$$evaluation\_algorithm = \begin{cases} switch\_to\_fitness, & \text{if } feasible_{inds} < T_{min} \\ switch\_to\_hybrid, & \text{if } feasible_{inds} > T_{max} \end{cases} \tag{5.5}$$

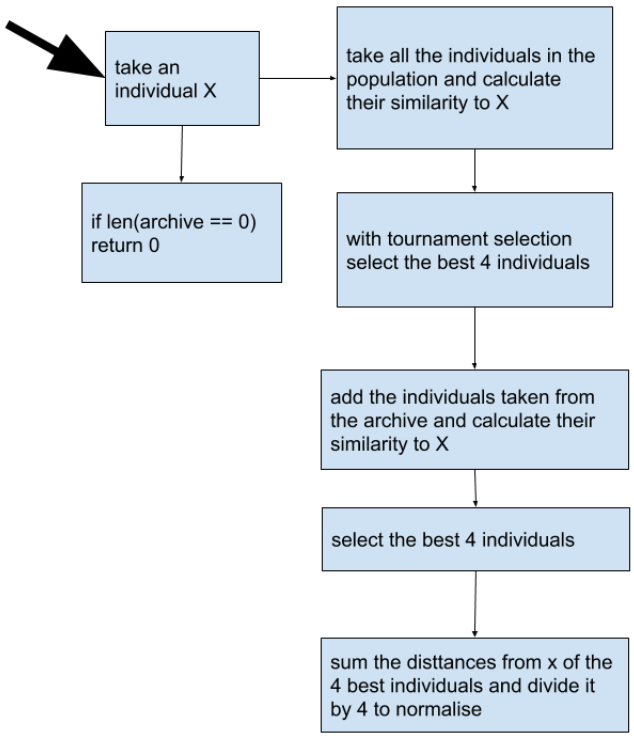



**Figure 5.3:** A diagram of the novelty evaluation used in our program

The importance of the dual evaluation method is that it allows a convergence to better individuals when few individuals have a good fitness value, while it explores the search space with novel individuals when many individuals have a good fitness value, so new individuals will have a better novelty value and a good fitness.

### 5.2.5 Variation operators

The variation operators used in our program were the 2 most common operators: *mutation* and *recombination*:

- *Mutation* as mentioned above is the change of a part of individual in order to introduce novelty. In our case the mutation has a probability to

occur given by the parameter *MUTPB* in GeneticAlgorithm.Constants, and when it happens it can change from 1 to 4 poses (letters in our case) with a new random pose taken from the list of all poses.

- *Recombination* is the method through which two individuals generate one or more individuals by mixing genes from both parents. In our case the method used is the two point crossover, which taken 2 individuals a and b split them into two parts and put together the first part of a with the second of b and vice versa.

### 5.2.6 All the important parameters

Summing up the algorithm the important parameters can be synthesised in table 6.3

N.B. some parameters are placed in the Parameters class for a simpler launch of the program and for a simpler parameters passing

| ***Parameter*** | ***Usage*** | ***Location*** |
|---|---|---|
| number_of_moves | The number of moves in each individual | GeneticAlgorithm. Constants |
| MUTPB | The probability for each individual to be mutated | GeneticAlgorithm. Constants |
| t_min | Parameter for switch to fitness evaluation | GeneticAlgorithm. Constants |
| t_max | Parameter for switch to hybrid evaluation | GeneticAlgorithm. Constants |
| max_arch | The maximum number of individuals to be compared with the selected in the calculus of dissimilarity | GeneticAlgorithm. Constants |
| max_number_ of_mutations | the maximum number of genes that can mute in an individual | GeneticAlgorithm. Constants |
| population_size | the size of the population | GeneticAlgorithm. Constants |
| number_of_ generations | the number of generations in the evolution | GeneticAlgorithm. Parameters |
| repertoire_index | the index of the repertoire path (all the paths are in GeneticAlgorithm. Constants | GeneticAlgorithm. Parameters |
| evaluation_method_ index | the evaluation method chosen (0 is for only fitness, 1 is for only novelty and 2 for both of them) | GeneticAlgorithm. Parameters |

| | | |
|---|---|---|
| random_seed | the random seed used to make computation completely deterministic | GeneticAlgorithm. Parameters |
| multi_objective_ selection | The algorithm used for multi objective selection (spea2 or nsga2) | GeneticAlgorithm. Parameters |
| dissim_threshold | The threshold used for add individuals to the archive | GeneticAlgorithm. Parameters |
| fitness_threshold | The threshold used for add individuals to the archive and for calculate feasible individuals | GeneticAlgorithm. Parameters |

**Table 5.3:** All the parameters used in the algorithm

# Chapter 6

# Algorithm evaluation method

## 6.1 Evaluation methodology

One of the most difficult problems in artificial creativity is the evaluation of what is produced, as its characteristics depend upon several factors involving also cultural background, aesthetics and individual expertise. Some recent works may help addressing this issue. The first evaluation that came into our mind was the comparison between a set of predefined choreographies, as can be seen in paragraph 5.2. At the time of writing there are some evaluation algorithms to judge if a program is creative or not, but the reference article for this problem is Wiggins research [Wiggins, 2006]. Since our evolution results don't have any fixed scheme and Wiggins parameters are too strict we decided to discard them and to focus on the other main article: Ritchie's paper [Ritchie, 2007] who defines a set of criteria which aim to evaluate both creativity and quality of the outputs of a program. Since this work is more lax than Wiggins article, it allows us to evaluate our program in a better way. Because we don't have a quality evaluation function (the idea could be to have an external Noh expert to judge our choreographies as

explained in section 6.3) and our resulting set is completely different from the inspiring set, we cannot use many of the criteria explained in that article, and we will use only the first criterion, together with two other criterion defined by us in section 6.2.

## 6.2 Typicality evaluation

The functions used to evaluate the typicality are

$$min\_typicality(C) = \min_{val} \mid val = \forall_{X\ in\ repertoire} string\_similarity(X, C) \quad (6.1)$$

$$NCD(a, b) = \frac{comp(a + b) - min(comp(a), comp(b))}{max(comp(a), comp(b))} \quad (6.2)$$

Where comp is equal to:

$$comp(x) = bz2.compress(x) \quad (6.3)$$

Bz2 compression is an algorithm used in computer science to create archives [Seward, 1996], but in this case we calculate the normalised compression distance (NCD) [Ming Li et al., 2004] This was used both for compute the compression of all the results and for compute the compression of each result. The 3 measures that include NCD are:

- **full_NCD** where the first parameter is a concatenation between all the strings resulted from the algorithm while the second is a concatenation between all the strings present in the repertoire. This is useful to evaluate how much the strings obtained are similar to the repertoire, because the higher the bz2 compression is the higher the results have

substrings similar to the repertoire.

- **Ritchie's criterion 1** that uses the NCD as equation 6.4, given rep_string as the concatenation of all strings in the repertoire. This told us the average typicality of the results.

$$criterion\_1(results) = avg(\forall_{x\ in\ results} NCD(x, rep\_string)) \tag{6.4}$$

- **Average min typicality** that uses equation 6.1 to calculate the minimal typicality of each individual present in results.

## 6.3 Quality evaluation

The quality of a choreography is a problem that has not been deeply studied, most of the works refers to it as a personal consideration instead of a common definition of good or bad aesthetics. Philosophers like Baumgarten and Leibniz have discussed this problem in many works but as result they obtained that aesthetic "is in fact only a pseudo-science or pseudo-philosophy, a study that no self-respecting member of an academic faculty can safely devote himself to exclusively, or even mainly" as Prall says or "that the work artists dislike lacks a je ne sais quoi" [Ogden, 1933]. In addition to it aesthetic is a strictly sectorial parameter, so an implementation that is good for paintings is not good for music and vice versa. Because of this deep subjectivity, the quality and the aesthetics of a choreography cannot be fully judged by a computer. Some works have been made hitherward trying to discretise and algorithmically define aesthetics:

- [Datta et al., 2006]and recently [McCormack and Lomas, 2020] have

used a machine learning approach, with neural network and deep learning, to implement a classifier for art aesthetic.

- [Takagi, 2001], [Sun et al., 2012], [Vircikova and Sincak, 2010], [Manfré et al., 2016] have used a "man in the middle" approach to interact with the evolutionary algorithm and judge individual's aesthetic. Unfortunately this approach has the downside that it requires an interaction during the run of the algorithm and a human can evaluate less individuals than a computer.

- [Krasnow and Chatfield, 2009] have tried to define a standard for dance aesthetic but it requires a visual recognition.

- [Sun et al., 2013] have made a program that uses both semi supervised learning and human evaluation as a fitness for the algorithm. A similar scheme has been used by [Peng et al., 2016] to create a robotic choreography.

Due to this problems we have chosen to not evaluate quality at the moment, but in the future can be done by some external Noh-specific critics.

## 6.4 Trend evaluation

The other interesting evaluation that we have performed is the trend of the algorithm, that means four parameters are monitored during the execution and their values analysed at the end of it. The values used are:

- Archive size
- NCD full
- Ritchie’s criterion 1
- Fitness and novelty

# Chapter 7

# Choreographies evolution

## 7.1 Parameter tuning

Since the launches of the program are parameter dependent, the first runs have helped us to tune them. Some parameters like *number_of_moves* will not change while others like *fitness_threshold* can change every run.

This tuning is important because otherwise our algorithm will not switch between hybrid and fitness evaluation, but instead it will focus on a single evaluation method for all the generations.

**fitness_threshold:** this parameter is particularly important because we have noticed that it must be tuned according to the size of the repertoire, otherwise the program will not switch between hybrid and fitness evaluation correctly. After some tries we have decided to tune it with the following formulas:

$$fitness\_threshold = \min_{X\ in\ repertoire} rep\_similarity(X) \tag{7.1}$$

Where *rep_similarity* is calculated as:

$$rep\_similarity(X) = \frac{\sum_{Y\ in\ repertoire} similarity(Y, X)}{len(repertoire) - 1}\ Y \neq X \tag{7.2}$$

*similarity* is the same as equation 5.3.
Depending on the repertoire, fitness threshold can be calculated alternatively as the maximum:

$$fitness\_threshold = \max_{X\ in\ repertoire} rep\_similarity(X) \tag{7.3}$$

**T_min and T_max** Two other important parameters related to each other are $T_{min}$ and $T_{max}$. At the beginning we used the values from the article by [Correia et al., 2013] but we found that the evolution was performing only hybrid evaluation (except for the first 10 generations), so we decided to raise both of them and set them as $T_{min} = 65$ and $T_{max} = 80$. A test was also made with $T_{min} = 40$ but as results the algorithm focuses only on hybrid evaluation.

**novelty_threshold** The $novelty\_threshold$ was empirically determined at 0.55 since with a high parameter the archive can be empty, while with a low parameter the archive can be too big, and because these will also be our final results we have chosen to select individuals with a novelty not so strong.

## 7.2 Runs

The algorithm was tested on many runs and with different parameters, as can be seen below, but some variables are kept constants in each run, in particular they are:

- number_of_moves = 16
- max_arch = 5
- MUTPB = 0.35
- t_min = 40
- t_max = 70
- max_number_of_mutations = 4
- population_size = 100

Each combination was tested with 2 methods:

- **only hybrid** in this case only multi-objective evaluation with both fitness and novelty is executed (the selection algorithm is SPEA2). The termination condition is the number of generations.
- **full** in this case the full algorithm explained in 5 is executed, with both evaluation methods (fitness and hybrid) and the shift between them as equation 5.5. The termination condition is given by the number of generations.

All results of the evolution can be found in [Bernagozzi, 2020a] in the folder *json/archive/risultati genetico* and in the excel file *json/archive/evaluation.xslx*.

The variable parameters used are described in table 7.1:

| | |
|---|---|
| random seed | 100 |
| number of generations | 500 |
| t_min | 40,65 |
| novelty_threshold | 0.55 |
| repertoire size and<br>fitness threshold | 1 - 0.62<br>3 - 0.53<br>6 - 0.41 |

**Table 7.1:** variable parameters for the runs

# Chapter 8

# Analysis during the execution

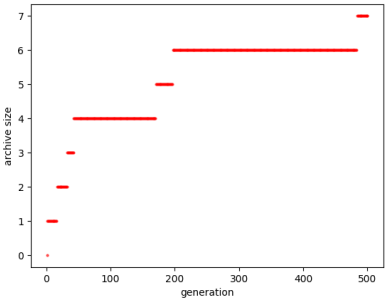
During the execution five parameters were monitored to better understand how the algorithm works. These parameters are:

- archive size
- fitness and novelty,
- full_NCD
- Ritchie's criterion 1

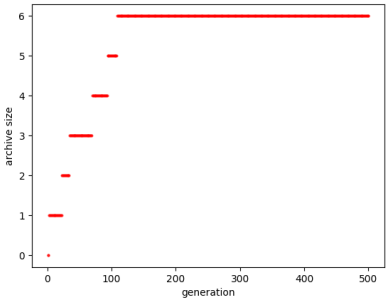
## 8.1 Archive/Results size

The first important parameter monitored during the execution of the algorithm is the size of the archive, or rather the number of choreographies found by our algorithm.

It can be seen from graphs 8.1 and 8.2 that in both cases the archive size has a logarithmic growth, with the full algorithm that has a better convergence. We can also notice that with higher repertoire size the algorithm has

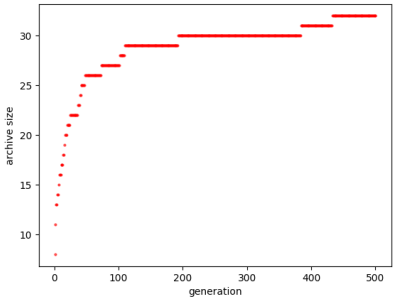



**(a)** hybrid

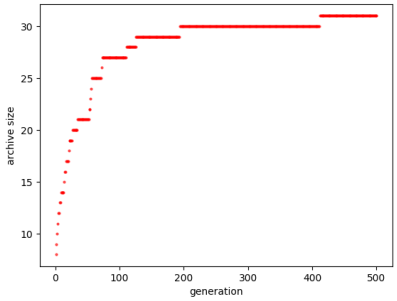



**(b)** full

**Figure 8.1:** Size of the archive during the execution of the algorithm with 500 generations, repertoire size = 1 and random_seed = 100




**(a)** hybrid




**(b)** full

**Figure 8.2:** Size of the archive during the execution of the algorithm with 500 generations, repertoire size = 6 and random_seed = 100

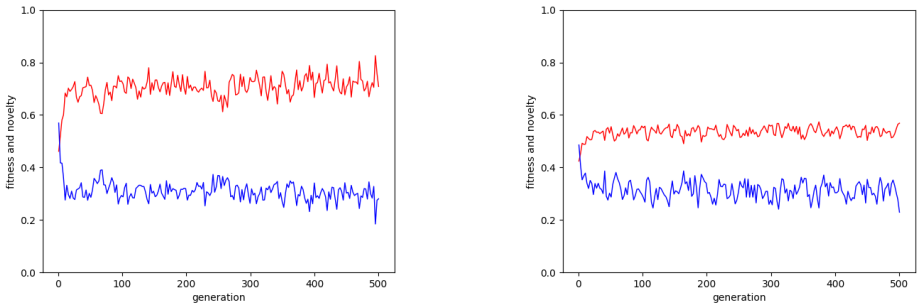
less difficulties to find individuals that suits the requirements necessary to the addition to the archive. The logarithmic shape can be explained because when there are already some individuals in the archive, new individuals that meets fitness requirements are discarded because too similar to the one in the repertoire.

## 8.2 Fitness and novelty comparison

The analysis of fitness and novelty during the algorithm shows us how the algorithm works and its trend, both to discover new individuals and to refine them to obtain a better fitness.

### 8.2.1 Only hybrid

Moving on to only hybrid algorithm it can be seen from figure 8.3 that both fitness and novelty value tend to converge to a single value. Particularly with the repertoire size equal to 1 (although there are some peaks, mostly for novelty) and with repertoire size equal to 6, that means the algorithm has other individuals but the best are converging to the average.




**(a)** repertoire_size = 1 **(b)** repertoire_size = 6

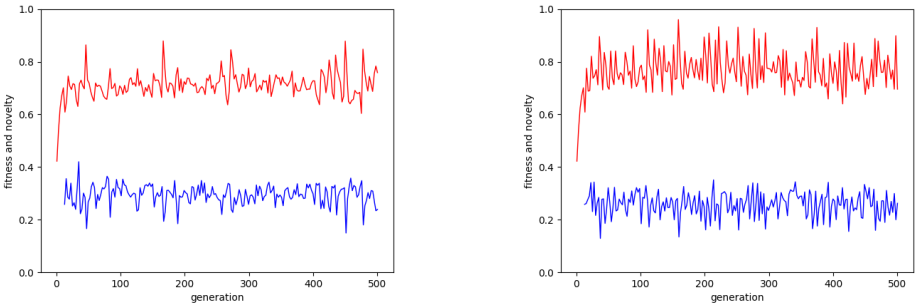
**Figure 8.3:** Average values of the results of each generation of the only hybrid algorithm with random_seed = 100 and 500 generations. Red dots are for fitness and blue dots are for novelty

### 8.2.2 Full algorithm

For what concerns the execution of the full algorithm we can see from graphs 8.4, 8.5 and 8.6 that with low repertoire size (fitness are higher)

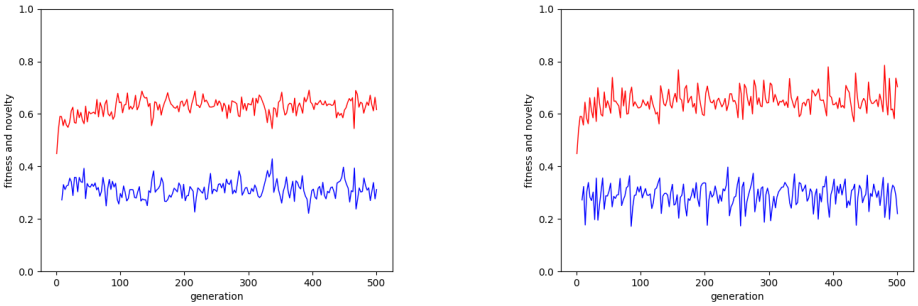
the difference in t_min between 40 and 65 is consistent, because with t_min equal to 65 the only fitness evaluation happens more frequently, with the other instead is quite similar to hybrid alone. As we expected the average value of fitness and novelty is more or less constant because individuals with higher fitness and novelty are took out from the population and added to the archive to be part of the final result. This allows the development of a population novel and with a high fitness that enables the creation of better individuals.

The graphs comparison is useful to see that t_min parameter is important in order to have more changes between evaluation functions, but also fitness_threshold is important because as repertoire_size increases there are less changes of evaluation function (less peaks). This is because more individuals have a fitness higher than the threshold, which leads to have only hybrid evaluation.



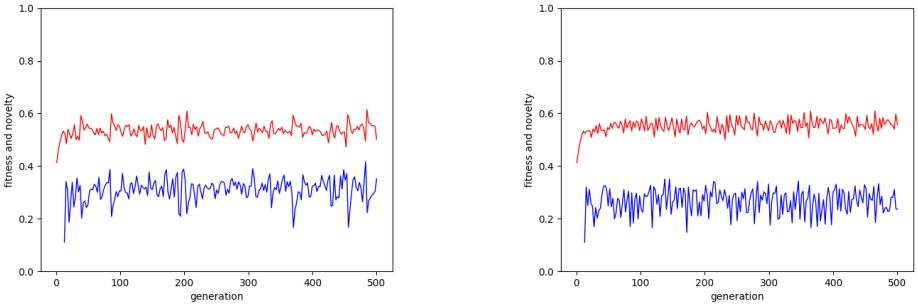

**(a)** t_min = 40 **(b)** t_min = 65

**Figure 8.4:** Average values of the results of each generation of the full algorithm with random_seed = 100, repertoire_size = 1 and 500 generations. Red line is for fitness and blue line is for novelty




**(a)** t_min = 40 **(b)** t_min = 65

**Figure 8.5:** Average values of the results of each generation of the full algorithm with random_seed = 100, repertoire_size = 3 and 500 generations. Red line is for fitness and blue line is for novelty



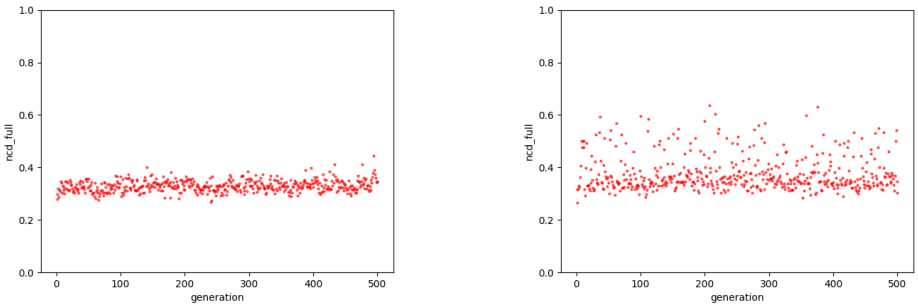

**(a)** t_min = 40 **(b)** t_min = 65

**Figure 8.6:** Average values of the results of each generation of the full algorithm with random_seed = 100, repertoire_size = 6 and 500 generations. Red line is for fitness and blue line is for novelty

## 8.3 Full NCD Comparison

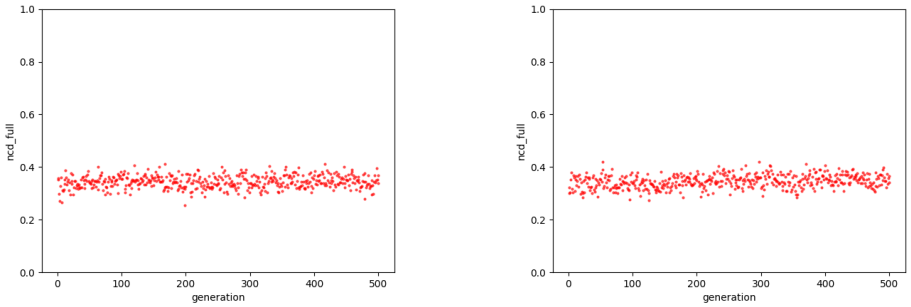
The analysis of the full NCD during the execution of the algorithm gives us an estimation of how much the intermediate results are similar to our repertoire.

From its comparison (graphs 8.7 and 8.8) we can see that with a lower repertoire size the full algorithm tend to have more standard individuals than the hybrid algorithm, this is due to the action of the only fitness evaluation in the full algorithm that aims to find better individuals.




**(a)** hybrid **(b)** full

**Figure 8.7:** Average values of full_ncd value applied to results of each generation of hybrid and full algorithms with random_seed = 100, t_min = 65 and 500 generations and repertoire size = 1.



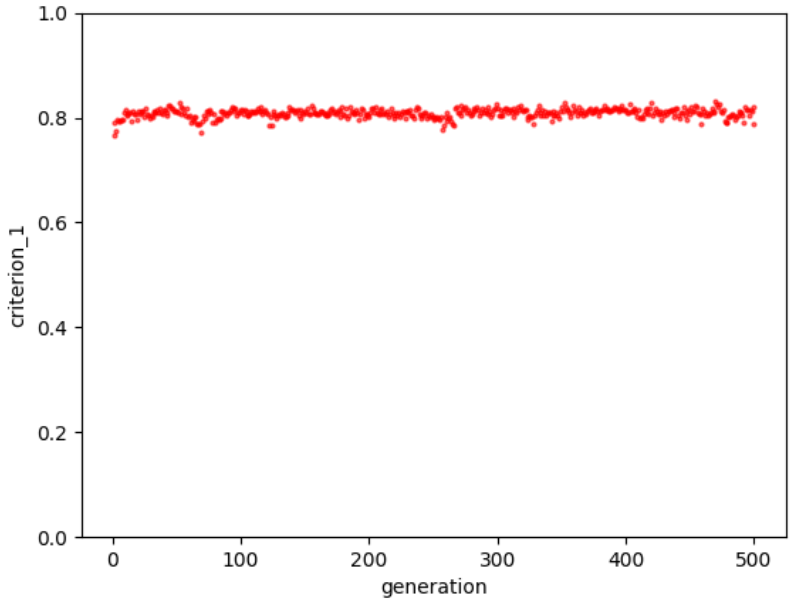

**(a)** hybrid **(b)** full

**Figure 8.8:** Average values of full_ncd value applied to results of each generation of hybrid and full algorithms with random_seed = 100, t_min = 65, 500 generations and repertoire size = 6.

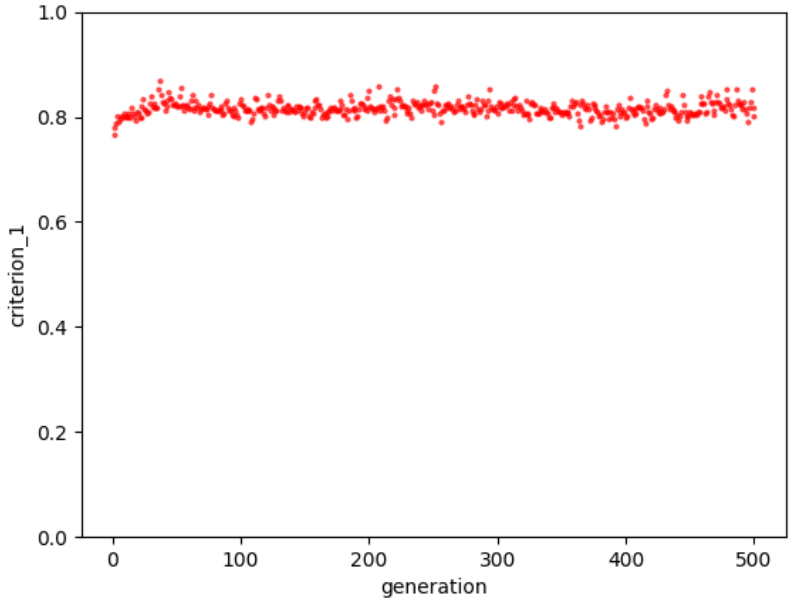
## 8.4 Ritchie’s criterion 1 comparison

The analysis of Ritchie’s criterion 1 during the execution explains, similarly to NCD_full, how much the intermediate results are similar to our repertoire but this is the average between each result.

As can be seen from graphs 8.9 and 8.10 the individuals in the population have more or less a fixed typicality value and there is not much difference between the two algorithms, despite the one with full algorithm is slightly more scattered as before. This is probably because the value is averaged, differently from the previous one, that avoids outliers and tends to favour more common individuals.

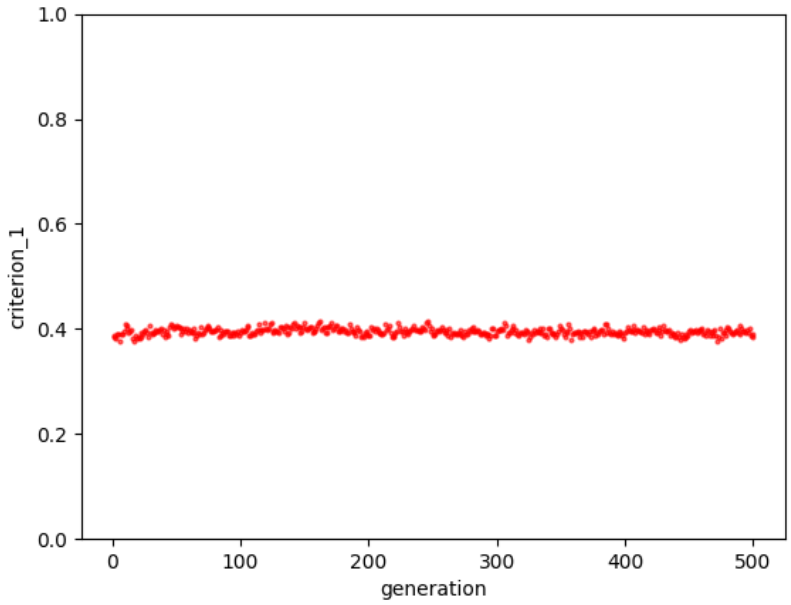



**(a)** hybrid

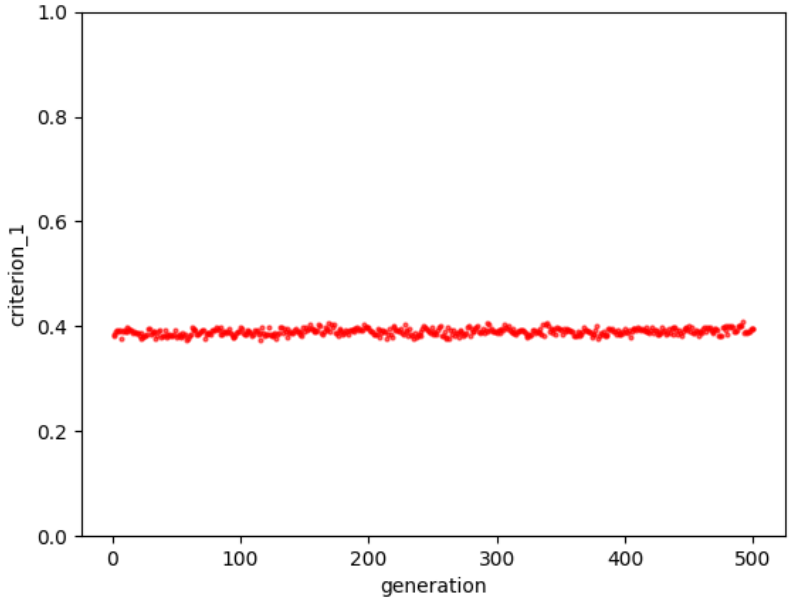



**(b)** full

**Figure 8.9:** Average values of Ritchie's criterion 1 value applied to the results of each generation of hybrid and full algorithms with random_seed = 100, t_min = 65, 500 generations and repertoire size = 1.




**(a)** hybrid




**(b)** full

**Figure 8.10:** Average values of Ritchie's criterion 1 value applied to the results of each generation of hybrid and full algorithms with random_seed = 100, t_min = 65, 500 generations and repertoire size = 6.

# Chapter 9

# Ex post analysis

For the ex post analysis we have taken into consideration 5 parameters:

- fitness,
- archive size,
- full_NCD,
- Ritchie's criterion 1,
- average min typicality.

## 9.1 Relations Between indexes

As expected, from an ex post analysis we can see that all the 3 indexes analysed (full_NCD, average min typicality and Ritchie's criterion 1) are co-related, showed by Pearson index in the table 9.1 below. This means all indexes have more or less the same value for what is concerning the final result. For this reason we have chosen to use full NCD index to be compared with other values.

| | full_NCD | Ritchie's criterion 1 |
|---|---|---|
| **average min typicality** | 0.96 | 0.99 |
| **Ritchie's criterion 1** | 0.96 | |

**Table 9.1:** Pearson value between the indexes

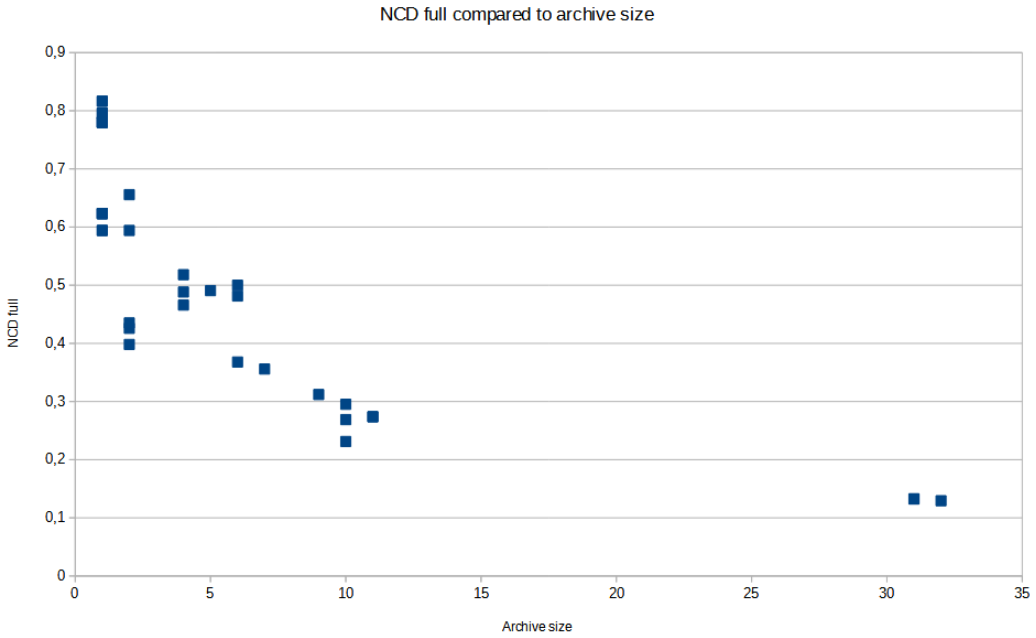



**Figure 9.1:** Full NCD compared to archive size

From the analysis we have discovered a strong positive relation (Pearson index equal to 0.95) between repertoire and archive sizes, this can be because with a higher repertoire size more individuals can have a high fitness.

Another interesting relation is the one between fitness threshold and archive size (figure 9.2), that tells us the lower the threshold the higher the archive size. This together with the relation between the archive size and the full NCD value (figure 9.1) shows us that the with a proper fitness threshold we can find many novel individuals with a high fitness value, but with a threshold too high the archive size is very small and the individuals are very similar to the archive.

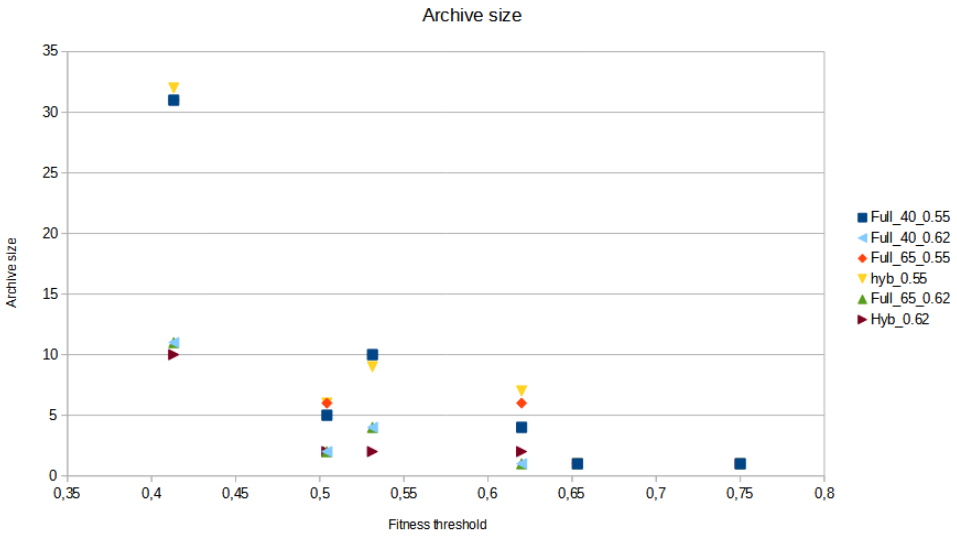



**Figure 9.2:** Archive size compared to fitness threshold

The other relation, that was also expected, is that as the fitness value increases the more individuals are typical (figure 9.3), this due to our fitness function as a similarity with a repertoire.

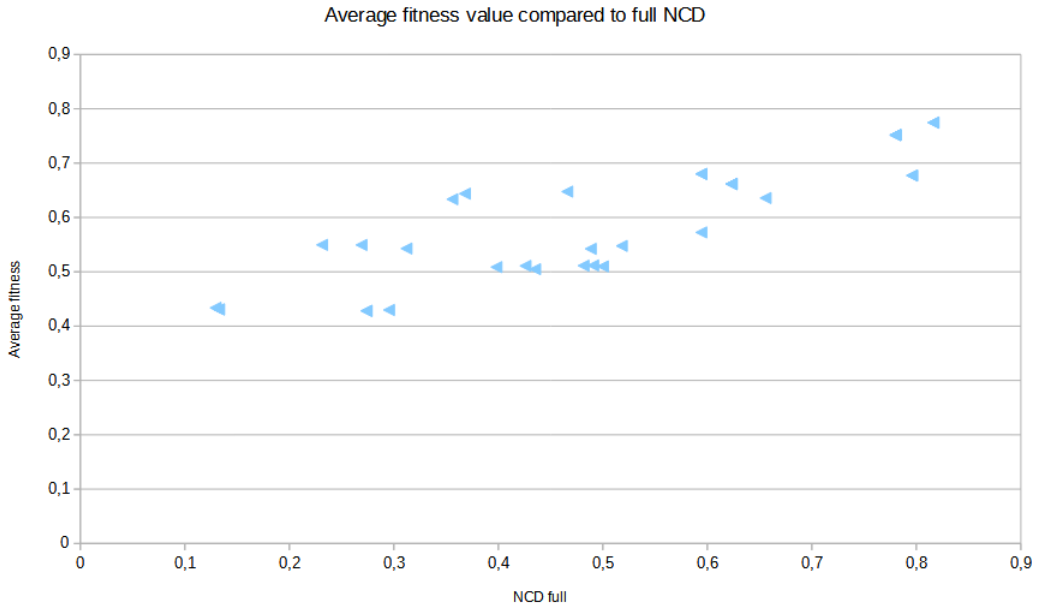



**Figure 9.3:** Average fitness value compared to full NCD values

## 9.2 Singular index analysis

**Results analysis** For what is concerning the resulting choreographies (an example in table 9.2) they are completely different, so the two algorithm have a different behaviour in the disclosure of the individuals but with the same resulting fitness.

| full | hybrid |
|---|---|
| wtgappsrsxmnpywf | tngwyfmpyzazqrto |
| xtnzeegznxzneymy | rwxtznxawmwcwjze |
| favamrgfzxfrwrta | awytbsxavgrtgnxt |
| mthadkwxwxmnayty | axygzaxzdmwdiaxw |
| ztnmcwbtgxzimatl | azhwwkwxulrtgzxt |
| aroxyrwxcxzumaag | zurwwyxnymrryxxi |
| | vztxcnxkymktgnvy |

**Table 9.2:** Results for repertoire size equal to 1, repertoire = azrtxwwgxyntzmay

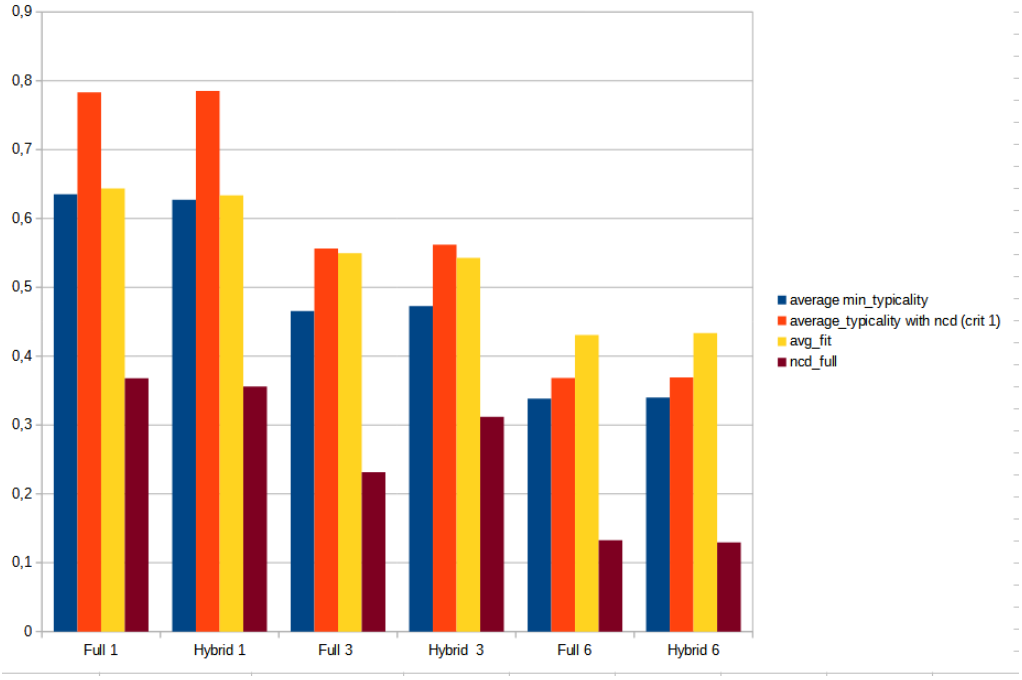



**Figure 9.4:** Comparison of the four indexes with repertoire size and algorithm used on the x axis

**Average fitness analysis** From graph 9.4 we can see how the average fitness, in yellow in the graph, is the same for both algorithms, and also fitness decreases as repertoire size increases (this is due to our choice with fitness threshold). This is an important result because shows that there are few differences between the choreographies found by the hybrid algorithm and the ones found by the full algorithm. An interesting fact related to fitness and criterion 1 is that as repertoire size increases they are inversely related, so we still have good results but less typical.

**Average full NCD analysis** Passing to full NCD analysis we can see from graph 9.4 that both algorithm have the same performance and as repertoire size increases it is easier for the algorithm find new individuals. This is enforced by the Pearson index of -0.97. An interesting fact to be noticed is that

the full algorithm seems to have a better method to find novelty individuals respect to the repertoire size, since with a repertoire size equal to 3 its value is lower compared to the hybrid algorithm. This means it has less pieces of choreography in common with the repertoire.

**Ritchie's criterion 1 and average min typicality** Ritchie's criterion 1 and average min typicality have the same behaviour as full NCD value, this is not surprising as their Pearson relation is very high and as they're constructed.

# Chapter 10

# Future Works

In order to create a true Noh choreography another thesis would be necessary, since there are several parameters that need to be finely tuned, and many things to be refined (such as robot's posture and walk).
The most important improvement of this work is the proper tuning of fitness threshold, dissim threshold, t_min and t_max, together with the analysis of the relations between them.

As far as the algorithm is concerned, the main work to be done is to expand the fitness function with a module that enables the distinction between a priest and a warrior in the generated choreographies employing the relation between the pose and the character already described in chapter 4. Furthermore we can improve the fitness function also with an evaluation not exclusively based on the repertoire. This might improve the generation of new choreographies far more different from the ones in the repertoire. Moreover another upgrade can be the addition of human evaluation into the algorithm, as in [Takagi, 2001], to have a fitness value completely independent from the repertoire.

Finally, another way to bring this work forward is to take advantage of the

paper from Augello et al. [Augello et al., 2017] to create new poses through a deep learning algorithm and use both old and new poses to produce innovative choreographies.

# Chapter 11

# Conclusions

We can say that the algorithm proposed by Vinhas et al. [Vinhas et al., 2016] is a good starting point to explore novelty. Unlike what we expected, the difference between their algorithm and a standard multi-objective genetic algorithm with both fitness and novelty is not so evident. This may be due to the fact that our fitness function evaluates the similarity match of an individual to a repertoire.

One perk of their algorithm is that it has a slightly better convergence than the only hybrid algorithm. In addition it seems to perform better than only hybrid algorithm if we consider the novelty of the individuals, because the typicality of the individuals evaluated with full NCD seems to be lower in the full algorithm. This means that it has less excerpts of choreographies in common with the repertoire.

We also think that the codification of poses with letters and choreographies with strings is a good choice, but the absence of a fitness function that includes an aesthetic evaluation together with a string evaluation is quite restrictive. Another advantage of this codification is that the letters have a unique correspondence to Nao poses, so that the fitness function can easily

consider the poses both as letters and as angles to enable a future evaluation of the balance or more characteristic related to angles.

Furthermore we can say that the evaluation of the choreographies cannot be codified into a single function because it needs to consider all the aspects that an evaluator can see, such as visual aesthetics. For this reason it requires an external human judgement or the codification of it, as in [Takagi, 2001] or other related work.

# Bibliography

[Anki, 2020] Anki (2020). Vector robot. https://www.digitaldreamlabs.com/pages/new-with-vector.

[Augello et al., 2017] Augello, A., Cipolla, E., Infantino, I., Manfre, A., Pilato, G., and Vella, F. (2017). Creative robot dance with variational encoder. *arXiv preprint arXiv:1707.01489*.

[Bernagozzi, 2020a] Bernagozzi, S. (2020a). Naomove. https://github.com/ste93/naoMove.

[Bernagozzi, 2020b] Bernagozzi, S. (2020b). Video example of choreographies in the repertoire. https://www.youtube.com/playlist?list=PLVRQBcMUJ9VOq_MkLxK6V6w_KGixDoRmO.

[Boden et al., 2004] Boden, M. A. et al. (2004). *The creative mind: Myths and mechanisms*. Psychology Press.

[Boston Dynamics, 2018] Boston Dynamics (2018). Spot robot dancing. https://www.youtube.com/watch?v=kHBcVlqpvZ8.

[Boston Dynamics, 2020a] Boston Dynamics (2020a). Atlas robot. https://www.bostondynamics.com/atlas.

[Boston Dynamics, 2020b] Boston Dynamics (2020b). Spot robot. https://www.bostondynamics.com/spot.

[Cohen et al., 2003] Cohen, W. W., Ravikumar, P., Fienberg, S. E., et al. (2003). A comparison of string distance metrics for name-matching tasks. In *IIWeb*, volume 2003, pages 73–78.

[Coppelia Robotics, 2020] Coppelia Robotics (2020). Coppelia simulator official website. https://www.coppeliarobotics.com/.

[Correia et al., 2013] Correia, J., Machado, P., Romero, J., and Carballal, A. (2013). Evolving figurative images using expression-based evolutionary art. In *Iccc*, pages 24–31.

[Cyberrobotics, 2020] Cyberrobotics (2020). Nao robot in webots simulator. https://www.cyberbotics.com/doc/guide/nao.

[Datta et al., 2006] Datta, R., Joshi, D., Li, J., and Wang, J. Z. (2006). Studying aesthetics in photographic images using a computational approach. In Leonardis, A., Bischof, H., and Pinz, A., editors, *Computer Vision – ECCV 2006*, pages 288–301, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Eiben and Smith, 2015] Eiben, A. E. and Smith, J. E. (2015). *Introduction to evolutionary computing*. Springer.

[Fenollosa and Pound, 2004] Fenollosa, E. and Pound, E. (2004). *The Noh theatre of Japan: with complete texts of 15 classic plays*. Courier Corporation.

[Gomes et al., 2015] Gomes, J., Mariano, P., and Christensen, A. L. (2015). Devising effective novelty search algorithms: A comprehensive empirical

study. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pages 943–950.

[Gomes et al., 2012] Gomes, J., Urbano, P., and Christensen, A. L. (2012). Progressive minimal criteria novelty search. In Pavón, J., Duque-Méndez, N. D., and Fuentes-Fernández, R., editors, *Advances in Artificial Intelligence – IBERAMIA 2012*, pages 281–290, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Honda, 2011] Honda (2011). Asimo robot. `https://global.honda/innovation/robotics/ASIMO.html`.

[Horigan and Lentczner, 2014] Horigan, J. and Lentczner, M. (2014). Context free art. `http://www.contextfreeart.org/`.

[Horigan and Lentczner, 2015] Horigan, J. and Lentczner, M. (2015). Context free design grammar version 2 syntax. `http://www.contextfreeart.org/mediawiki/index.php/Version_2_Syntax`.

[Jacquot, 2015] Jacquot, P. (2015). V-rep scene nao. `https://github.com/PierreJac/Project-NAO-Control`.

[japan guide, 2020] japan guide (2020). Noh theatre in japan guide. `https://www.japan-guide.com/e/e2091.html`.

[Kitano et al., 1997] Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., and Osawa, E. (1997). Robocup: The robot world cup initiative. In *Proceedings of the first international conference on Autonomous agents*, pages 340–347.

[Krasnow and Chatfield, 2009] Krasnow, D. and Chatfield, S. J. (2009). Development of the “performance competence evaluation measure”: assess-

ing qualitative aspects of dance performance. *Journal of Dance Medicine & Science*, 13(4):101–107.

[Larsen, 2020] Larsen, B. (2020). Complete guide to noh theatre. `https://japanobjects.com/features/noh/#what-is-noh`.

[Lehman and Stanley, 2010] Lehman, J. and Stanley, K. O. (2010). Revising the evolutionary computation abstraction: Minimal criteria novelty search. GECCO '10, New York, NY, USA. Association for Computing Machinery.

[Lehman and Stanley, 2011] Lehman, J. and Stanley, K. O. (2011). Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation*, 19(2):189–223. PMID: 20868264.

[Manfré et al., 2016] Manfré, A., Augello, A., Pilato, G., Vella, F., and Infantino, I. (2016). Exploiting interactive genetic algorithms for creative humanoid dancing. *Biologically Inspired Cognitive Architectures*, 17:12–21.

[Manurung, 2004] Manurung, H. (2004). An evolutionary algorithm approach to poetry generation.

[Marques et al., 2000] Marques, M., Oliveira, V., Vieira, S., and Rosa, A. C. (2000). Music composition using genetic evolutionary algorithms. In *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512)*, volume 1, pages 714–719 vol.1.

[McCormack and Lomas, 2020] McCormack, J. and Lomas, A. (2020). Understanding aesthetic evaluation using deep learning. In *International Conference on Computational Intelligence in Music, Sound, Art and Design (Part of EvoStar)*, pages 118–133. Springer.

[Michalewicz and Schoenauer, 1996] Michalewicz, Z. and Schoenauer, M. (1996). Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary computation*, 4(1):1–32.

[Microsoft, 2010] Microsoft (2010). Kinect v1. `https://developer.microsoft.com/en-us/windows/kinect/`.

[Ming Li et al., 2004] Ming Li, Xin Chen, Xin Li, Bin Ma, and Vitanyi, P. M. B. (2004). The similarity metric. *IEEE Transactions on Information Theory*, 50(12):3250–3264.

[Ogden, 1933] Ogden, R. M. (1933). A definition of aesthetics. *The Philosophical Review*, 42(5):500–510.

[Peng et al., 2016] Peng, H., Hu, H., Chao, F., Zhou, C., and Li, J. (2016). Autonomous robotic choreography creation via semi-interactive evolutionary computation. *International Journal of Social Robotics*, 8(5):649–661.

[Pinson, 2018] Pinson, J. (2018). Project sandro. `https://github.com/PinsonJonas/Project-2-sandro`.

[Ritchie, 2007] Ritchie, G. (2007). Some empirical criteria for attributing creativity to a computer program. *Minds Mach.*, 17(1):67–99.

[Robocup Federation, 2020] Robocup Federation (2020). Robocup web site. `https://www.robocup.org/`.

[ROS community, 2020] ROS community (2020). Ros website. `https://www.ros.org/about-ros/`.

[Seward, 1996] Seward, J. (1996). bzip2 and libbzip2. *avaliable at http://www. bzip. org*.

[Shahverdi and Masouleh, 2016] Shahverdi, P. and Masouleh, M. T. (2016). A simple and fast geometric kinematic solution for imitation of human arms by a nao humanoid robot. In *2016 4th International Conference on Robotics and Mechatronics (ICROM)*, pages 572–577.

[Softbank Robotics, 2011a] Softbank Robotics (2011a). Nao. http://doc.aldebaran.com/2-1/family/nao_h25/index_h25.html.

[Softbank Robotics, 2011b] Softbank Robotics (2011b). Naoqi. http://doc.aldebaran.com/2-5/index_dev_guide.html.

[Softbank Robotics, 2015] Softbank Robotics (2015). Choregraphe. http://doc.aldebaran.com/2-4/software/choregraphe/index.html.

[Sony, 2018] Sony (2018). Aibo robot. https://us.aibo.com/.

[Sun et al., 2013] Sun, X., Gong, D., Jin, Y., and Chen, S. (2013). A new surrogate-assisted interactive genetic algorithm with weighted semisupervised learning. *IEEE Transactions on Cybernetics*, 43(2):685–698.

[Sun et al., 2012] Sun, X., Gong, D., and Zhang, W. (2012). Interactive genetic algorithms with large population and semi-supervised learning. *Applied Soft Computing*, 12(9):3004–3013.

[Takagi, 2001] Takagi, H. (2001). Interactive evolutionary computation: fusion of the capabilities of ec optimization and human evaluation. *Proceedings of the IEEE*, 89(9):1275–1296.

[the-NOH, 2020] the-NOH (2020). Noh theatre in the-noh. https://www.the-noh.com/.

[Tim, 2018] Tim (2018). Tim robot dancing. https://www.gruppotim.it/it/archivio-stampa/mercato/2018/

Nota-stampa-TIM-uno-spot-da-Guinness-World-Record-7-febbraio.html.

[Vinhas et al., 2016] Vinhas, A., Assunção, F., Correia, J., Ekárt, A., and Machado, P. (2016). Fitness and novelty in evolutionary art. In *International Conference on Computational Intelligence in Music, Sound, Art and Design*, pages 225–240. Springer.

[Vircikova and Sincak, 2010] Vircikova, M. and Sincak, P. (2010). Dance choreography design of humanoid robots using interactive evolutionary computation.

[Wiggins, 2006] Wiggins, G. A. (2006). A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems*, 19(7):449–458.

[Yang, 2015] Yang, S. (2015). Evolutionary computation for dynamic optimization problems. GECCO Companion '15, New York, NY, USA. Association for Computing Machinery.

# Ringraziamenti

È difficile scrivere una lettera di ringraziamento a tutti quelli che mi hanno aiutato durante la tesi perché sono troppi ma proverò comunque a citarvi tutti.

Per iniziare vorrei ringraziare mio babbo e mia mamma, che nonostante tutto hanno sopportato e supportato i miei momenti bui e nei quali avrei voluto mollare tutto, a loro va il mio più grande grazie per tutto quello che hanno fatto e che stanno facendo, se sono qua gran parte del merito è loro.

Assieme a loro voglio dire grazie a Marco, che nonostante la grossa distanza mi è sempre stato vicino nel portare a termine gli studi e mi è stato di supporto in tutto.

Un terzo enorme grazie va ad Andrea e a Mattia, coi quali ho condiviso le gioie e i dolori di questo progetto ma assieme siamo riusciti a superarlo. La cosa più bella che devo a loro è l'essermi sentito pari, senza alcuna superiorità da parte di nessuno e l'essere aperti a tutte le condivisioni di idee.

Un enorme grazie anche a Dani, che mi ha aiutato all'ultimo nella correzione della tesi, senza di lui probabilmente non sarebbe venuta così, e per

le partite a snooker utili per staccare la mente.

Grazie a Miki e Chiara, per il loro supporto costante e per tutte le serate passate assieme, in particolare ringrazio Miki per i pomeriggi passati a suonare come svago per la tesi e per le discussioni fruttuose che abbiamo avuto.

Grazie a Matte e a Lucy, per le grigliate e le serate assieme, e per l'amicizia che si è venuta a creare. P.s. Matte e Miki per il Risiko ci sono sempre!

Grazie a Ema per il supporto durante la via degli dei, e per le birre assieme.

Grazie a Bera per tutte le discussioni informatiche, a volte completamente distanti dalla tesi, ma che sono state comunque sempre piacevoli.

Grazie a Monica per il supporto costante e per avermi aiutato nei momenti peggiori a portare a termine questo traguardo.

Grazie a Dalmi e Ele, per le grigliate e per le serate passate in taverna.

Grazie a tutti i miei zii che in tutti i momenti mi sono stati sempre vicino, in particolare grazie a Carlo per tutte le chiacchierate e a Giovanni per le partite a biliardo.

Grazie a Nonna Anna e nonno Giorgio, che mi hanno sempre accolto e aiutato a staccare con le loro cene.

Grazie anche a tutti i miei amici del settimo piano di matematica, con i quali abbiamo passato momenti di studio intensi e partite a miseria indimenticabili.

Vorrei ringraziare anche tutte le mie amiche di danza, che mi hanno accolto nel loro gruppo e tra cene e allenamenti mi hanno aiutato a portare a termine questo compito, ma soprattutto mi hanno fatto scoprire un lato di me diverso da quello ingegneristico.

Grazie al gruppo di pallavolo del martedì, che mi ha aiutato a staccare durante i periodi più intensi e mi è sempre stato vicino in tutto.

Grazie a tutti i miei amici dell'università, con i quali ho condiviso gioie e dolori di progetti e esami ma anche buoni crescioni dai piadinari.

Grazie a Laura e al gruppo dei 2002, perché nonostante le difficoltà siamo sempre rimasti uniti e siamo cresciuti assieme, non solo io ho provato a dare la mia conoscenza a voi, ma anche voi avete dato tantissimo a me.

Grazie anche a Costanza che mi ha aiutato a correggere tutti i refusi del mio inglese.

Infine vorrei ringraziare tutti coloro che mi sono dimenticato di citare, anche chi si è fermato solo per un ciao, perché comunque è riuscito a entrare nella mia vita e a darmi una mano per questo traguardo.