.5¿

Formulation and Clustering of Solutions in Pick and Place Puzzle

Erfan Baradarantohidi
Supervisors:
Dr. Jinan Allan
Svetlana Levit

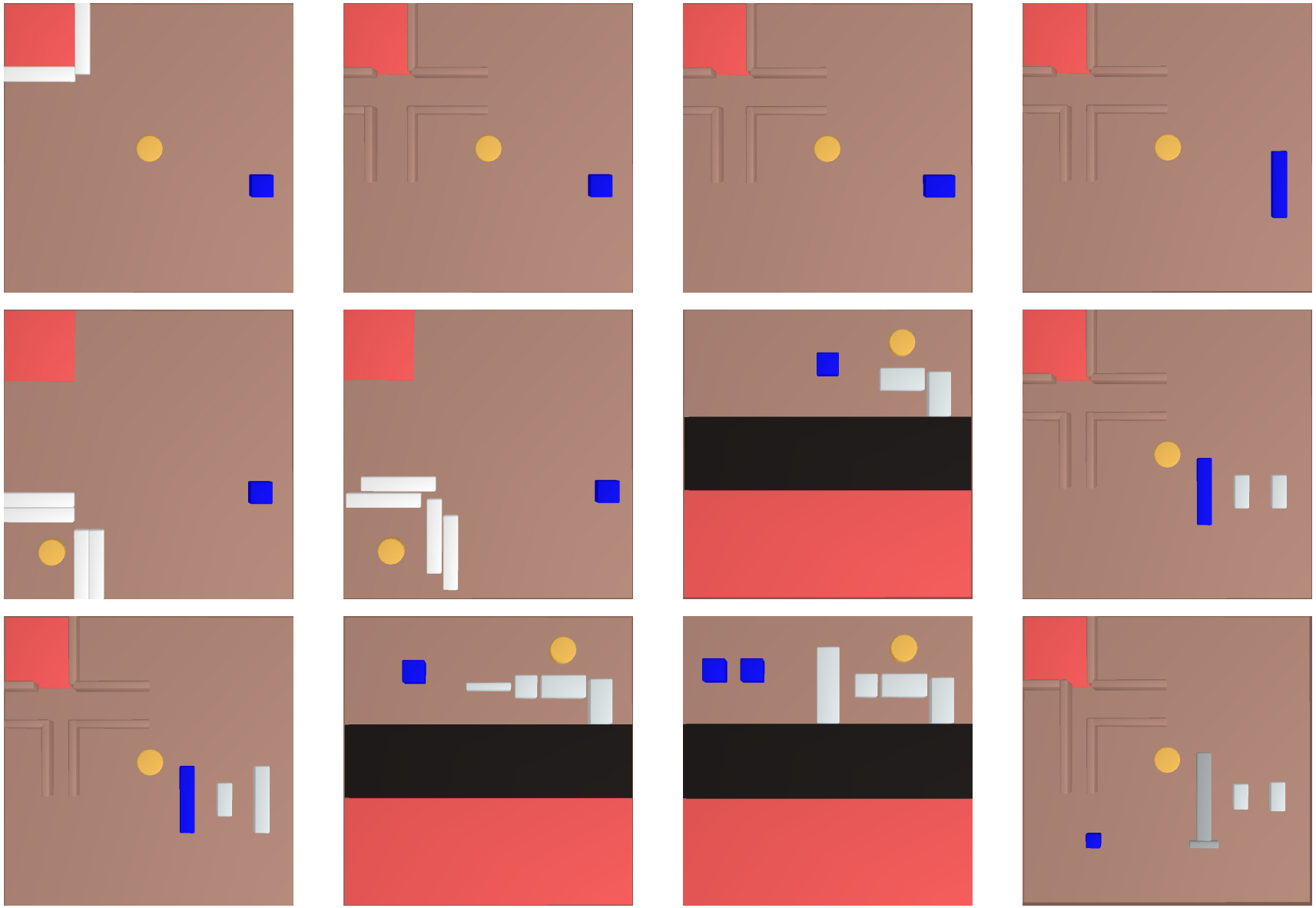
Science of Intelligence

3. July 2023

table of contents

1. Our Puzzles
   - Pick and place Puzzles
   - Solutions
2. Definition of a solution sequence
   - Possible solution type
   - Necessary Sets Definition
   - How each solution type is defined
3. Similarity or distance between Solutions
   - Dynamic Time Warping Method
4. Clustering results

PNP



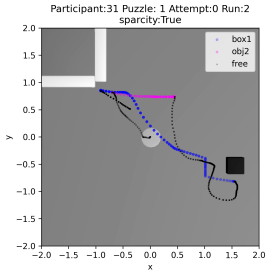
Solution

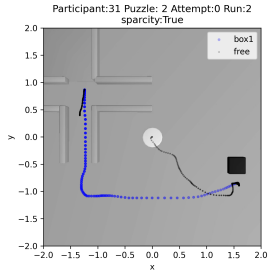
```
36_2_5_0
['free 0.52 -0.86 2.23s', 'obj2 0.01 1.0 0.67s', 'free 0.0 -1.0 2.4s',
'Glue obj3 to obj2', 'obj2 1.0 0.05 0.7s', 'free 0.87 0.49 1.75s', 'box1 -0.87 0.49 0.77s']

39_2_5_0
['free -0.06 1.0 1.81s', 'obj4 1.0 -0.01 2.0s', 'free 0.0 -1.0 0.81s',
 'obj2 -0.07 1.0 0.25s', 'free -0.99 0.15 0.86s', 'obj4 1.0 0.01 0.7s',
  'free -0.95 0.31 2.53s', 'obj1 0.75 0.66 1.57s', 'free 0.74 -0.67 1.45s', 'box1 -0.81 0.58 1.35s']
```

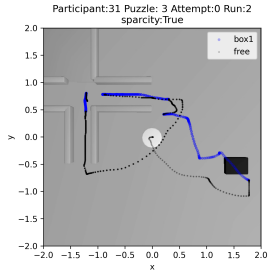
Figure 1: description of one's solution

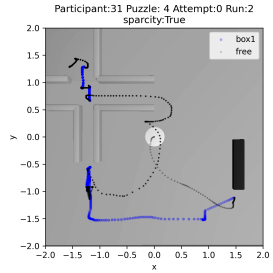
```
Participant:31 for the Puzzle:21 in the Attempt:3 and run:1 took follwing movements:
['obj1 W 1.12s', 'obj1 W 1.2s', 'obj2 W 2.42s', 'Glue obj2 to box1', 'Unglue obj2', 'Glue obj2 to box1',
'box1 S 1.32s', 'obj1 E 3.5s', 'Glue obj1 to obj2', 'box1 S 1.54s']
Solved: True
```

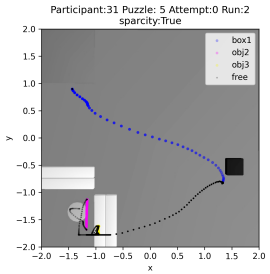
Solution Trajectory

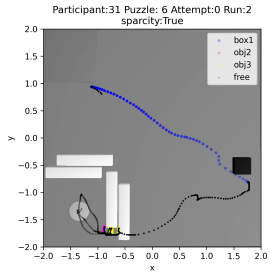


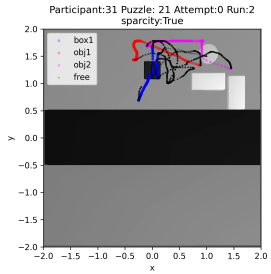


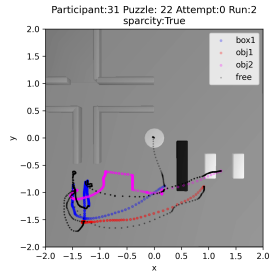


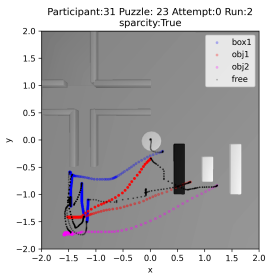


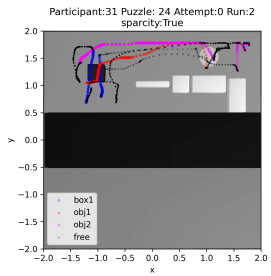


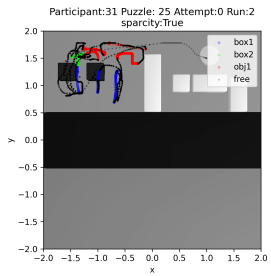


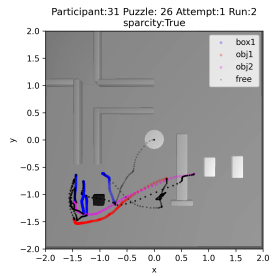




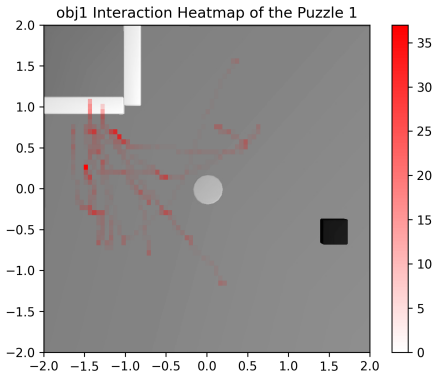


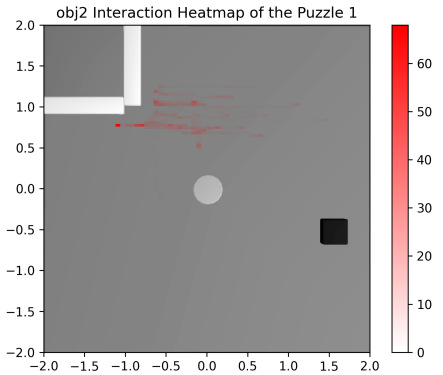


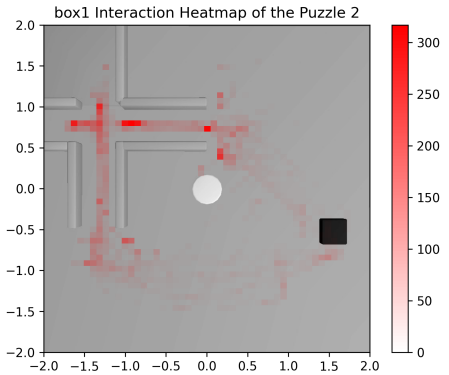


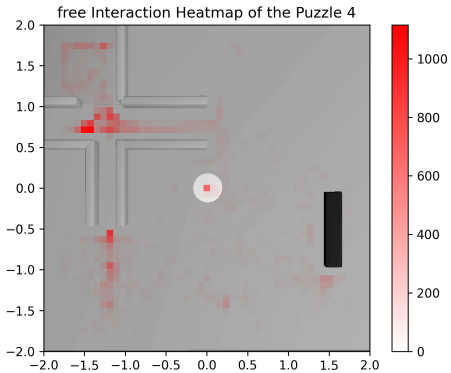


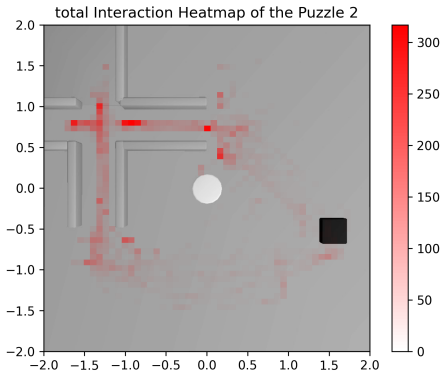
Heat Map of Interaction

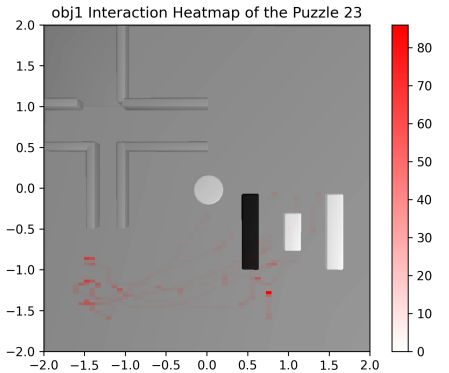












Possible solution type

How can a recorded solution to a puzzle looks like?

- solution $S_{(id)}$ where $id = Partcipant_id - run - Puzzle - attempt$
- sequence of n states $S_{(id)} = (a_k)_{k=1}^{n}$

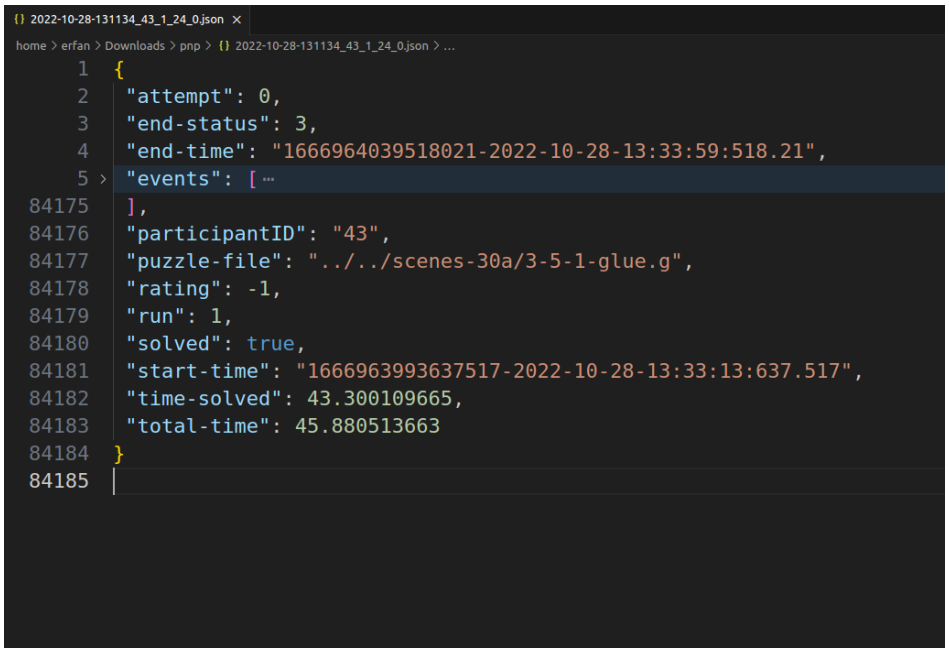


Figure 2: Events column contain the raw recording of the participant solution

Possible solution type

Possible ways of capturing states

- String - Time
- Color - Trajectory
- Color - Time
- Color
- Color - Displacement (?)

Necessary Sets Definition

- I_i : set of all **possible interaction** type for puzzle i. e.g.
  $I_2 = \{"free","box1"\}$
  $I_5 = \{"free","obj1",\ldots,"obj4","box1"\}$

  as for Gluing action an ordered subset can be used
  "Glue obj1 to box1" : {"obj1","box1"}

Necessary Sets Definition

- C_i : Mapping of I_i by **One Hot Encoding**. e.g.

$$C_2 = \{\begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}1\\0\end{bmatrix}\} \qquad C_5 = \{\begin{bmatrix}0\\0\\0\\0\\0\\1\end{bmatrix}, \begin{bmatrix}0\\0\\0\\0\\1\\0\end{bmatrix}, \ldots, \begin{bmatrix}1\\0\\0\\0\\0\\0\end{bmatrix}\}$$

Necessary Sets Definition

- P: sequence of ego **Position** in an interaction
  $P = \{(x_j, y_j)\}_{j=1}^{m} \qquad x_j, y_j \in [0, 2] \times [0, 2]$
- D: **Displacement** of ego during an interaction
  $D = \{(\Delta x, \Delta y)\}$
- T: **Duration** of an interaction in Seconds
  $T \in \mathbb{R}^{+}$

String - Time

$$S_{(id)} = (a_k)_{k=1}^{n}$$

- $a_k \in I_i \times T$

```
String-Time Sequence of the id 31_1_22_0 is:
 [['free' '1.600687']
 ['box1' '2.500098']
 ['free' '1.817846']
 ['box1' '1.80355']
 ['free' '1.279151']
 ['obj1' '4.158443']
 ['free' '1.165264']
 ['free' '0.992879']
 ['box1' '1.249687']
 ['free' '2.09325']
 ['obj2' '3.320016']
 ['free' '0.546941']
 ['free' '1.227313']
 ['box1' '0.766483']]
```

Figure 3: description of interaction and the duration in solution

Color

$$S_{(id)} = (a_k)_{k=1}^{n}$$

- $a_k \in C_i$
- Basically just ordered list of the types of interaction in a solution

```
color Sequence of the id 31_1_22_0 is:
 [[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0.]
 [0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
 [1. 0. 1. 0. 1. 0. 1. 1. 0. 1. 0. 1. 1. 0.]
 [0. 1. 0. 1. 0. 0. 0. 0. 1. 0. 0. 0. 0. 1.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]
```

Figure 4: One Hot Encoding of the solution
Column wise,last two elements for GLUE and UNGLUE, not yet implemented

Color - Time

$$S_{(id)} = (a_k)_{k=1}^{n}$$

- $a_k \in C_i \times T$

```
color-time Sequence of the id 31_1_2_0 is:
 [['1.0' '0.0']
 ['0.0' '1.0']
 ['2.578797' '4.650346']]
```

Figure 5: Color Time
free or box1 interaction and duration of each

Color - Trajectory

$S_{(id)} = (a_k)_{k=1}^{n}$

- $a_k \in C_i \times P$
- each state a_k here is a trajectory of ego in a interaction mode
- higher dimension compared to the rest of the types
- color-trajectory Sequence for 31-1-4-0 has the dimension:(1729, 4)

Displacement

- One of the first attempts that lead to the idea of transforming the solutions in different types was extracting following information from a puzzle:

```
36_2_5_0
['free 0.52 -0.86 2.23s', 'obj2 0.01 1.0 0.67s', 'free 0.0 -1.0 2.4s',
'Glue obj3 to obj2', 'obj2 1.0 0.05 0.7s', 'free 0.87 0.49 1.75s', 'box1 -0.87 0.49 0.77s']

39_2_5_0
['free -0.06 1.0 1.81s', 'obj4 1.0 -0.01 2.0s', 'free 0.0 -1.0 0.81s',
 'obj2 -0.07 1.0 0.25s', 'free -0.99 0.15 0.86s', 'obj4 1.0 0.01 0.7s',
  'free -0.95 0.31 2.53s', 'obj1 0.75 0.66 1.57s', 'free 0.74 -0.67 1.45s', 'box1 -0.81 0.58 1.35s']
```

Figure 6: Generated Text files
string,$\Delta x, \Delta y$, time

why do we need to different representation of Solution

To explore and identify strategy, depend on the setup of specific puzzle, different strategies can be observed.

- in puzzles such as puzzle 2, the only freedom for a different strategy is the **path**, one can take
- while in puzzle 5, moving two vertical **objects** (1,4) or the horizontal ones (2,3) can be interpreted as distinct strategy
- similar argument can be said for the displacement of the objects, changes of the position determines strategy and not exact path

Dynamic Time Warping

Since each solution sequence regardless of their type have **different length and speed**, in order to compare, DTW was chosen as the distance metric.

- best alignment between sequences and minimizing the effects of distortion in time and shifting.
- Multivariate version

DTW

```
def dtw_distance(s1, s2):
    n = len(s1)
    m = len(s2)
    dtw = np.zeros((n+1, m+1))

    for i in range(1, n+1):
        dtw[i][0] = float('inf')
    for j in range(1, m+1):
        dtw[0][j] = float('inf')
    dtw[0][0] = 0

    for i in range(1, n+1):
        for j in range(1, m+1):
            cost = abs(s1[i-1] - s2[j-1])
            #distance in diffrent timestep
            dtw[i][j] = cost + min(
                                dtw[i-1][j], #insertion
                                dtw[i][j-1],      #deletion
                                dtw[i-1][j-1])    #match

    return dtw[n][m]
```

Multi- Dimensional DTW

- Independent warping (DTWI):

$$DTW_I(V, W) = \sum_{p=1}^{d} DTW(V_p, T_p)$$

- Dependent (DTWD):

$$C_{i,j}(V, W) = \sum_{p=1}^{d} (V_{i,k} - W_{j,k})^2$$

  local cost matrix formed different

- Adaptive Warping (DTWA):
  Score function (ratio between the distance of the nearest neighbor in both modes), training a threshold

Results

- Performing Hierarchical Clustering with **ward** method
- dependent DTW
- Sequence type : **Color** and **Color- Trajectory**
- Visualization within a cluster
  - GIF of path frames
  - Ordered stacked bar plot of interaction in time

Results

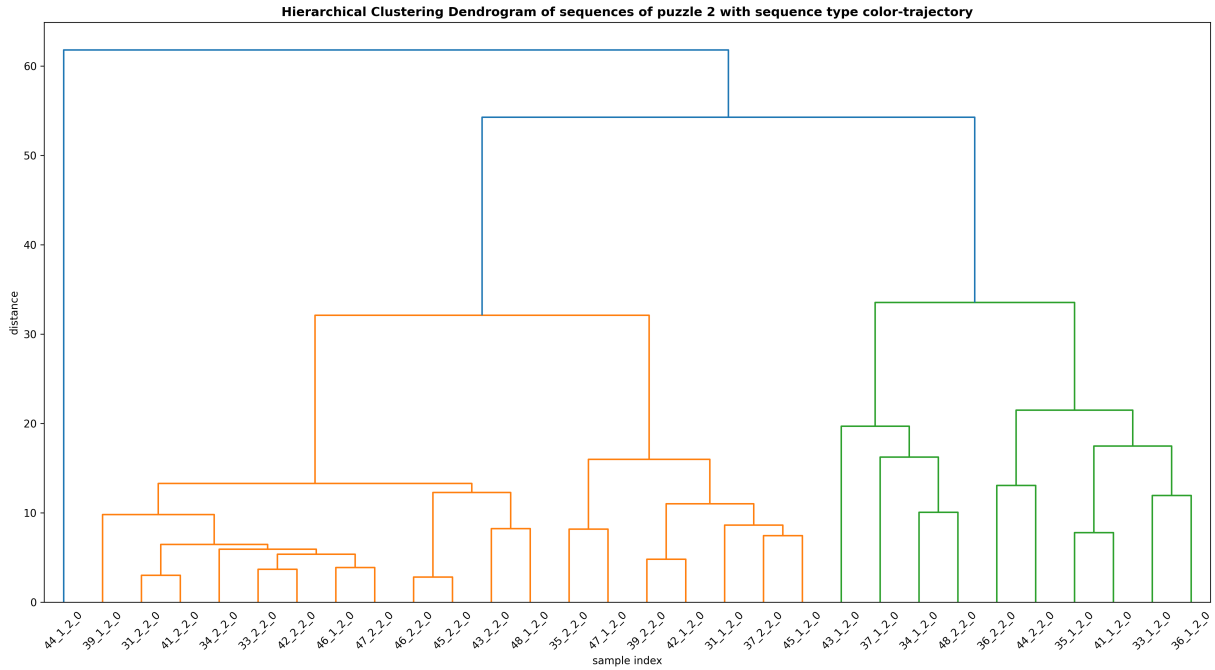


Figure 7: Dendrogram of puzzle2 color trajectory