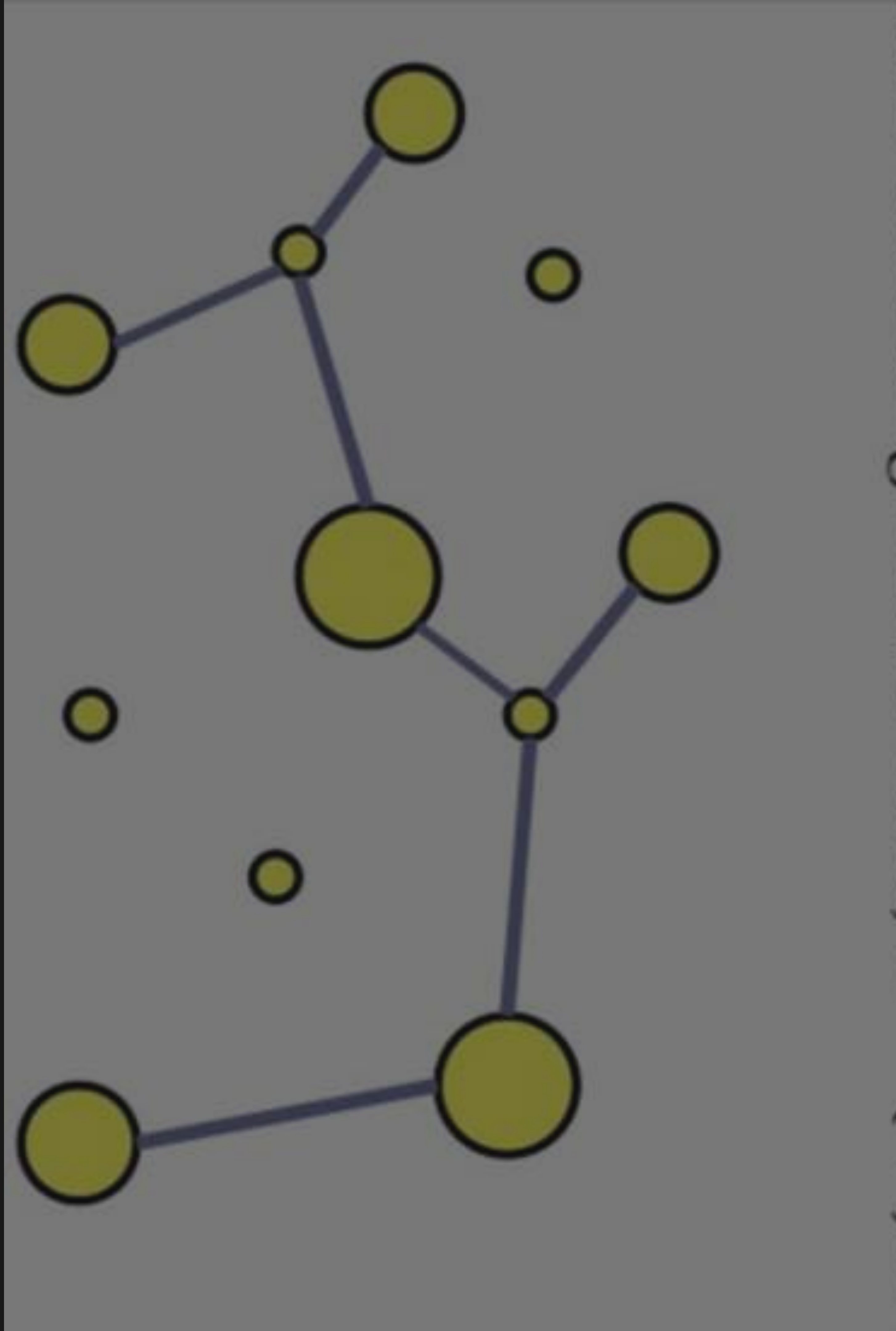
Steiner Tree:
A solution using a genetic algorithm

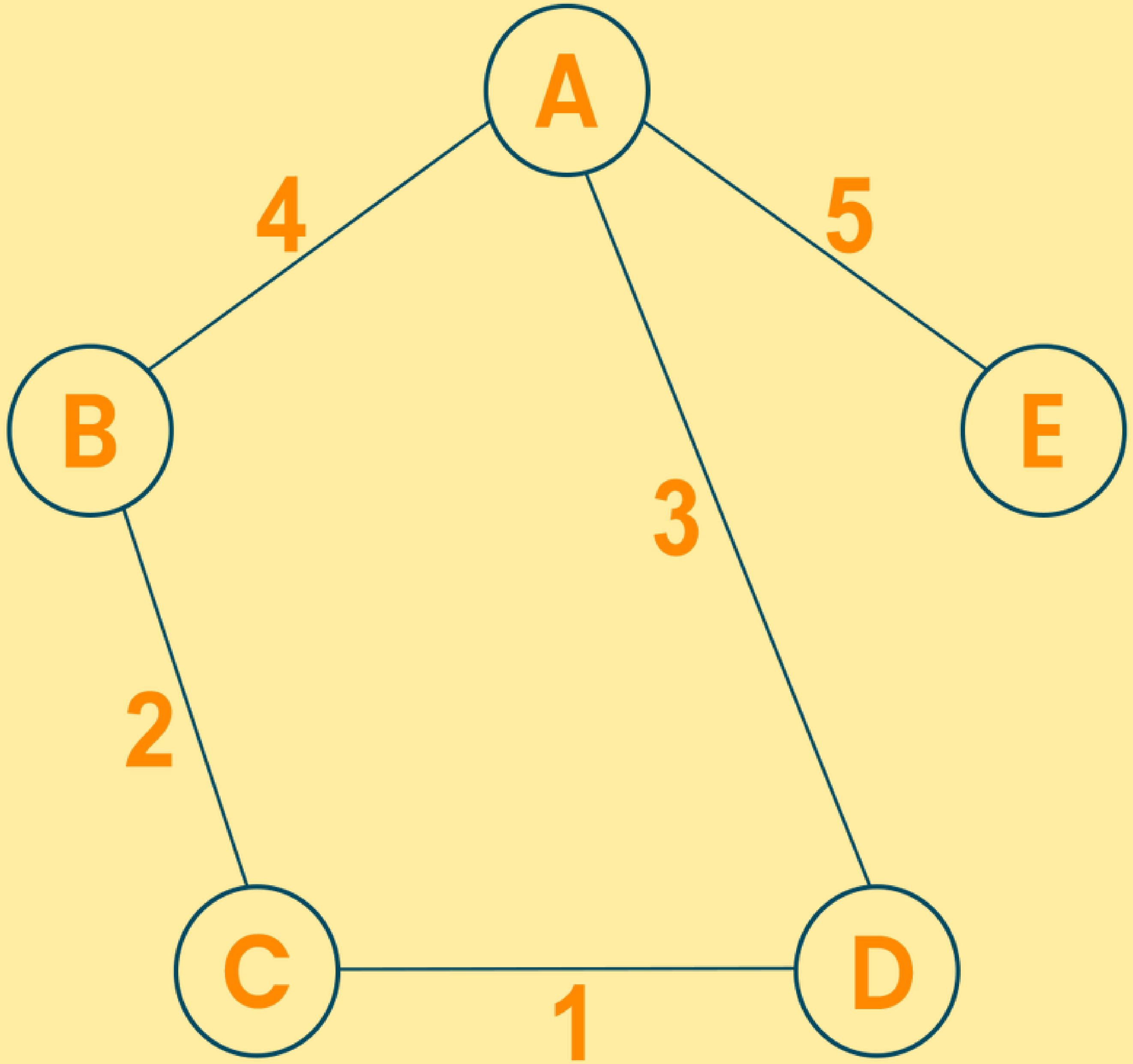
Daniele Petrillo Maria Zampella



STEINER PROBLEM

What is a tree?

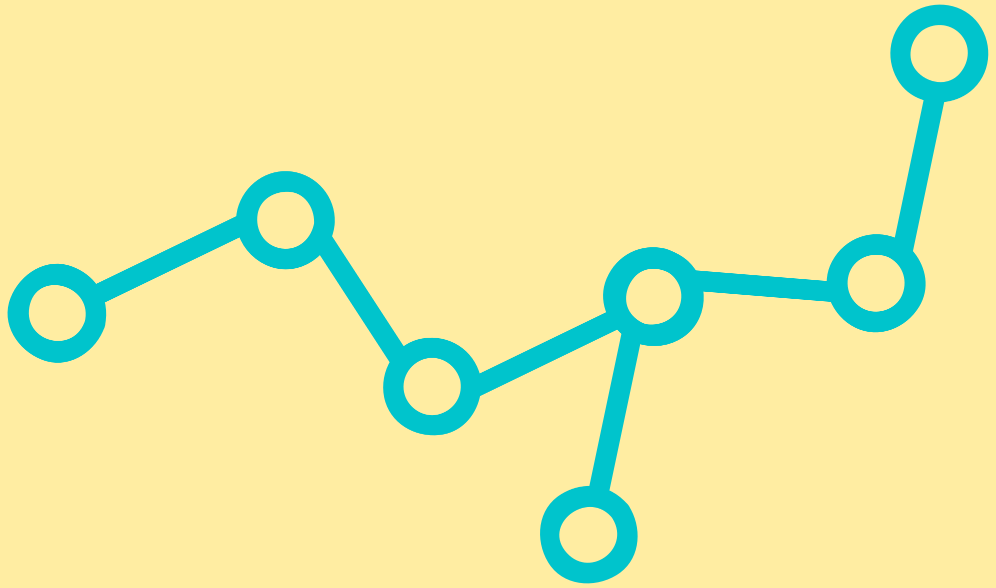
A **tree** is a non oriented graph, in which every couple of vertices are connected by one and only one path.

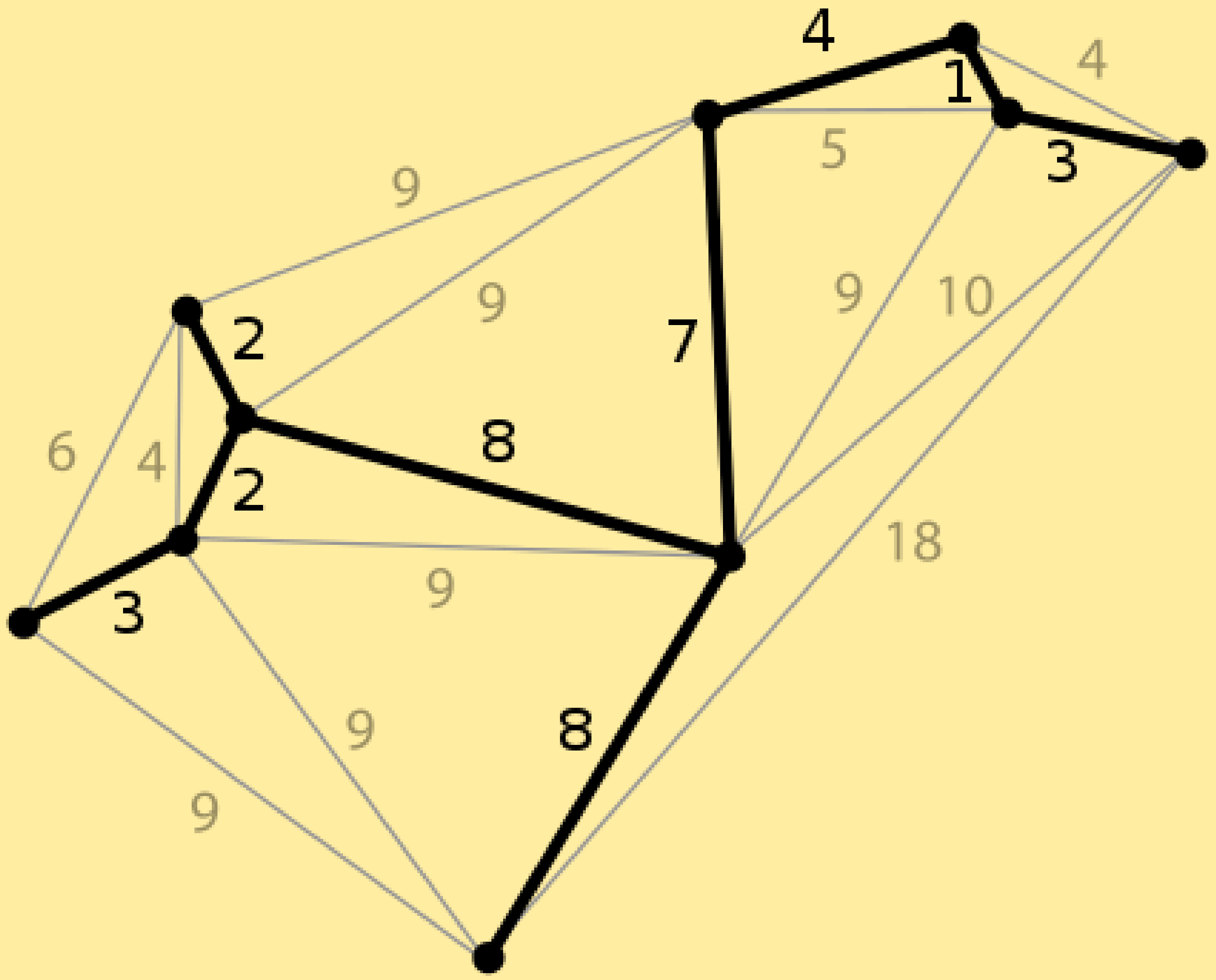


STEINER PROBLEM

Minimum Spanning Tree (MST)

It is the tree interconnecting N given points, such that it has the shortest length possible, according to some distance metric.

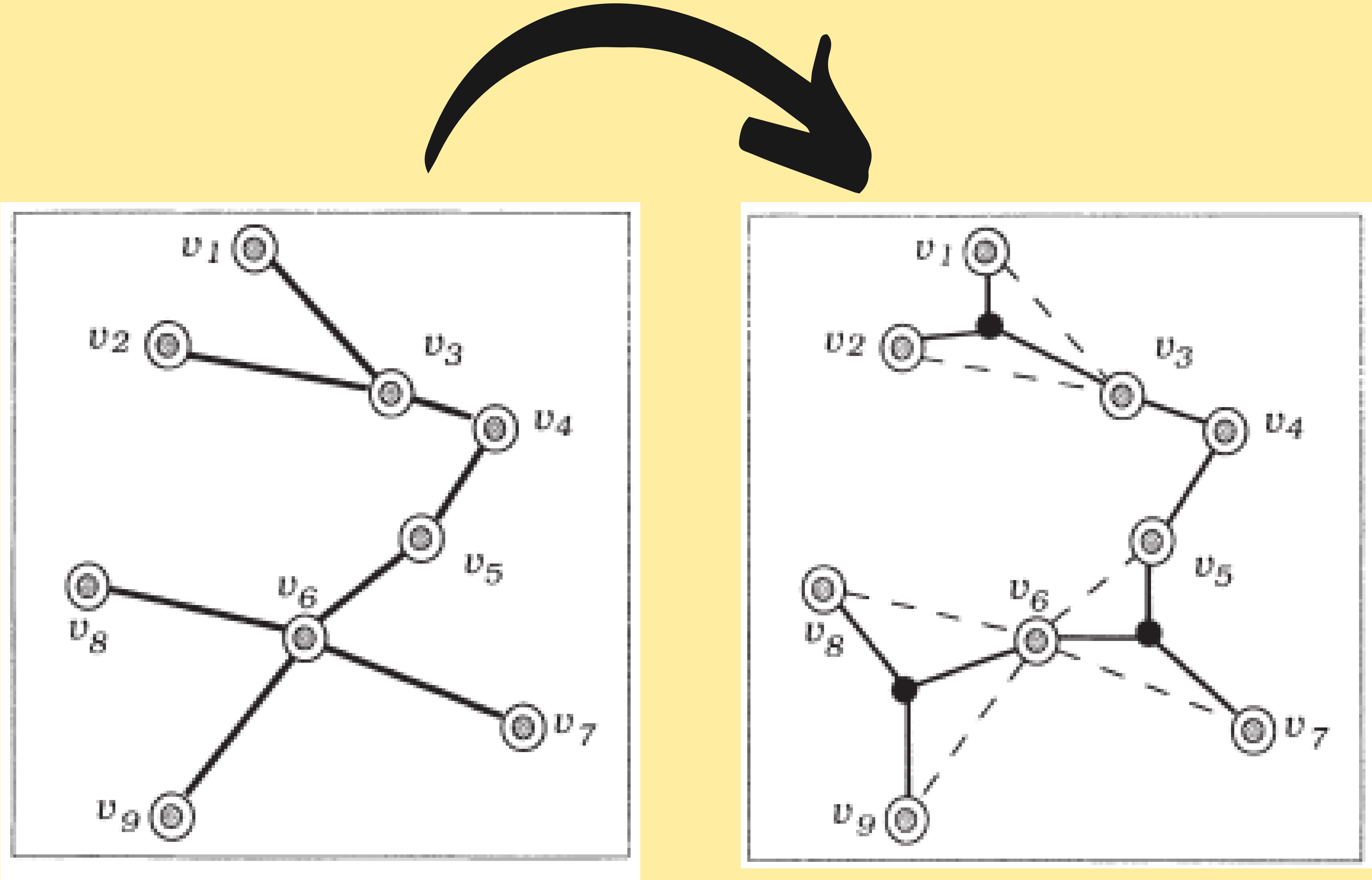




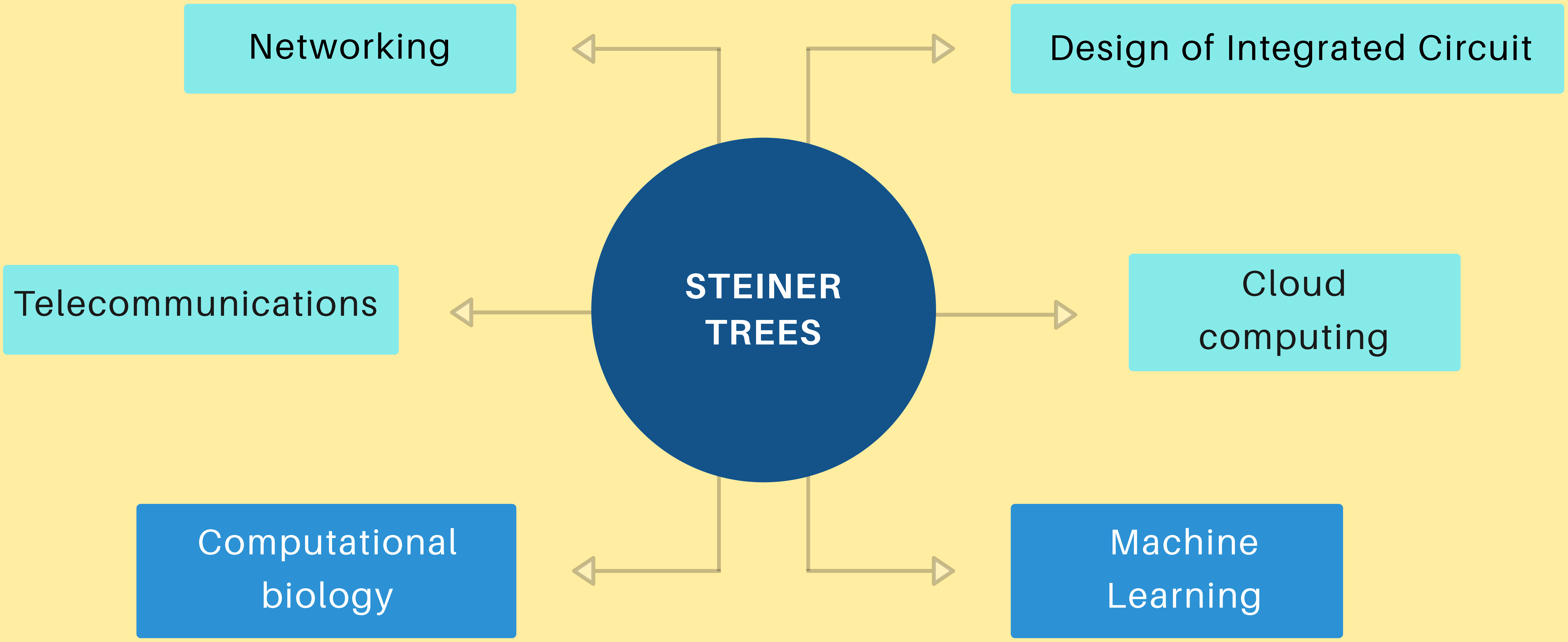
STEINER PROBLEM

Minimum euclidean Streiner tree (MEStT)

Given a set of N coplanar points, called site nodes, the Minimum Euclidean Steiner tree problem (**MEStT**) is to find the shortest tree connecting all N points, where the tree may contain nodes in the plane other than the site nodes. These **extra vertices** are called the **Steiner Points**.



MAIN APPLICATIONS



GENETIC ALGORITHM
X
STEINER PROBLEM

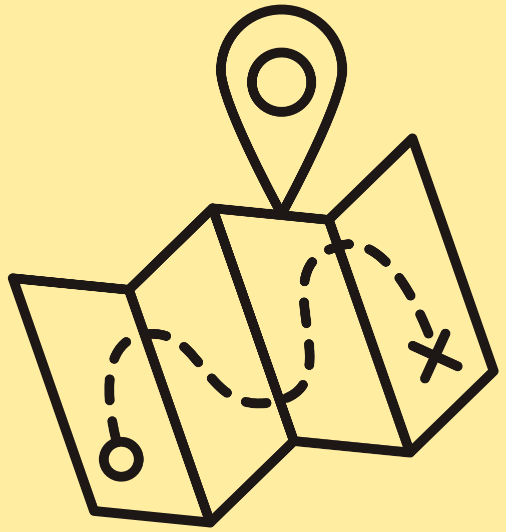
GENETIC ALGORITHM

Why using a genetic algorithm?

- MEStT problem is **NP-complete**;
- The problem grows **exponentially** with increase in N (number of fixed points);
- Even the best traditional algorithms need heavy optimizations and a lot of heuristic knowledge.

COMPLETE ALGORITHM

Workflow



Initialization

The fixed points are given as input and their MST is calculates.

Genetic Algorithm

The genetic algorithm is executed and it returns as output the best configuration of the Steiner points.

Optimization

Delete all the useless Steiner point according to some criteria such as the number of connections with the fixed vertices.

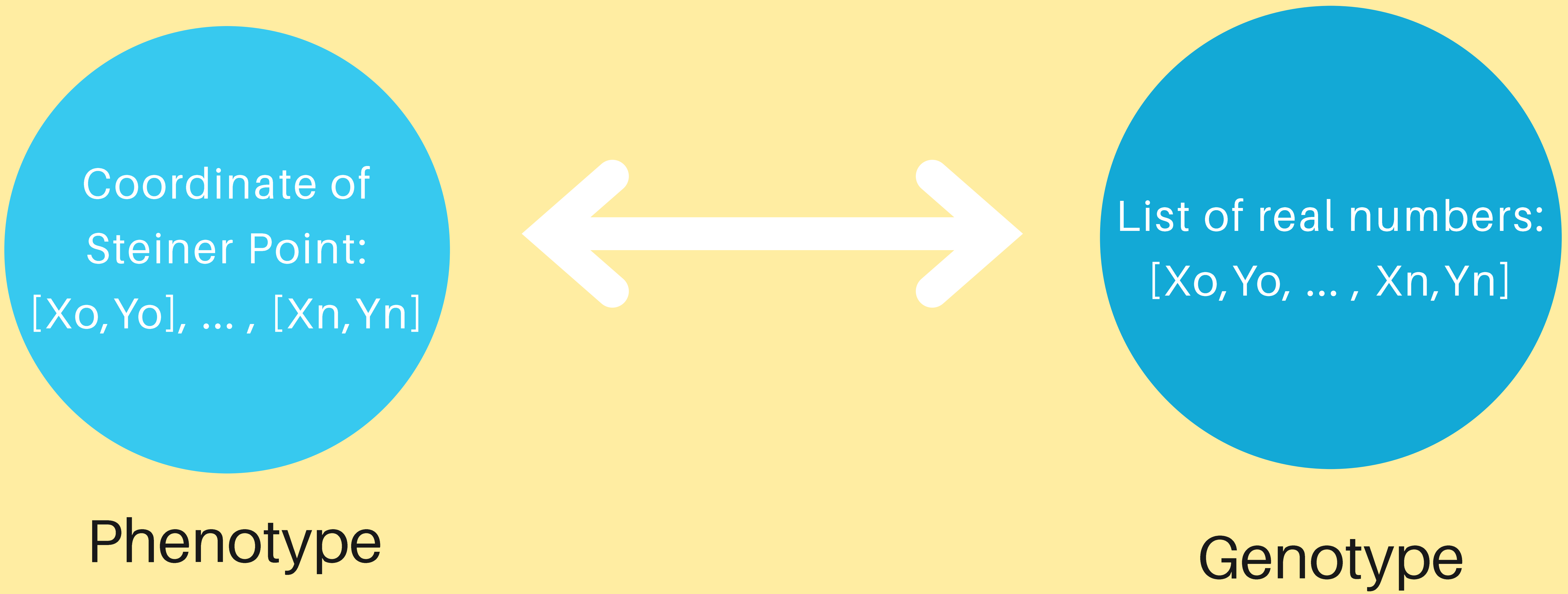
Visualization

Finally, the optimal steiner tree is plotted and compared with the starting MST.

GENETIC ALGORITHM

Genetic encoding: Individual

The genetic algorithm only works on the points that must be added (**Steiner points**) in order to obtain the shortest path which links all the fixed vertices.



GENETIC ALGORTIHM

Genetic encoding: Fitness

What is it?

It is the total length of the minimum spanning tree (containing both fixed and steiner vertices).

How to find it?

By analizing the possible paths which link all the points.

Prim's Algorithm

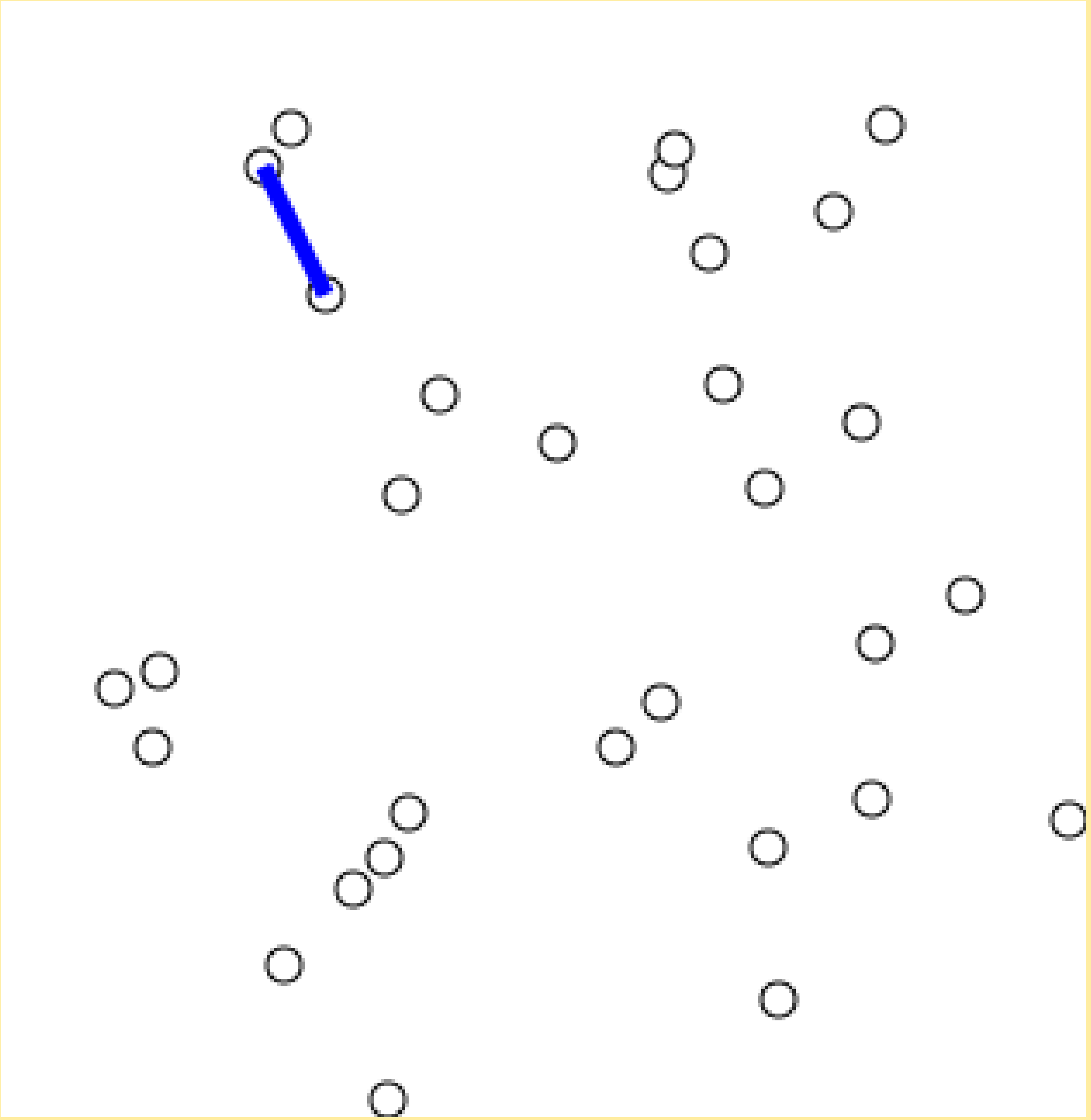
Starting from a set with only one point, it iteratively includes the closest point not yet in the set, saving its connection.

```python
def prim_alg(vertices):
    remaining_points = vertices.copy()
    mst = []
    connections = []
    source = random.choice(remaining_points)
    remaining_points.remove(source)
    mst.append(source)

    while remaining_points:
        outer_min = 1000
        for selected in mst:
            min_dist = 1000
            for possible in remaining_points:
                d = math.dist(selected, possible)
                if d < min_dist:
                    min_dist = d
                    new_point = possible
            if min_dist < outer_min:
                outer_min = min_dist
                final_b = new_point
                final_a = selected

        mst.append(final_b)
        connections.append([final_a, final_b, outer_min])
        remaining_points.remove(final_b)

    return(connections)
```



GENETIC ALGORITHM

Algortihm instance

Parent selection

Tournament Slection

N inidividuals are randomly chosen from the population and the best one is selected. This process is repeted population size time.

Parameters

N = 10

Recombination

Blend Crossover

Selected two parentes x and y, a new individual is randomly selected in [x-a(y-x), y+a(y-x)] .

Parameters

a = 0.5
CXPB = 0.9

Mutation

Gaussian Mutation

It adds a random value from a Gaussian distribution to each element of an individual's vector to create a new offspring.

Parameters

MUTPB = 0.1
INDPB = 0.5
mu = 0; sigma = 0.3

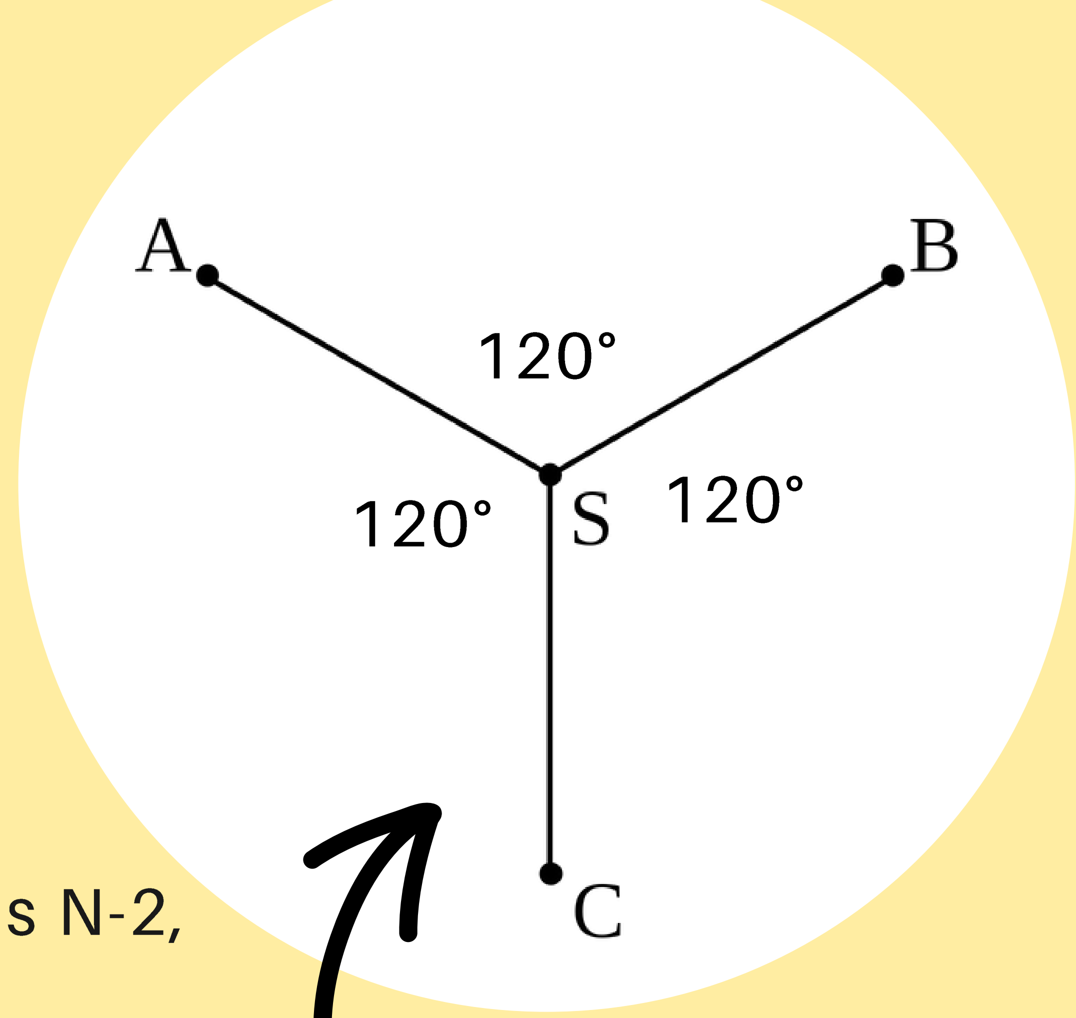
Survival Selection

Generational with Elitism

At each generation, all parents are replaced by the offsprings; the worst offspring is replaced by the best individual yet.

OPTIMIZATION

Heuristics



1

Number of Steiner points

The maximum possible number of Steiner Point for N fixed point is N-2, so this number was selected.

2

Number of links of Steiner points

Each optimal Steiner point always has 3 connections with the other points.

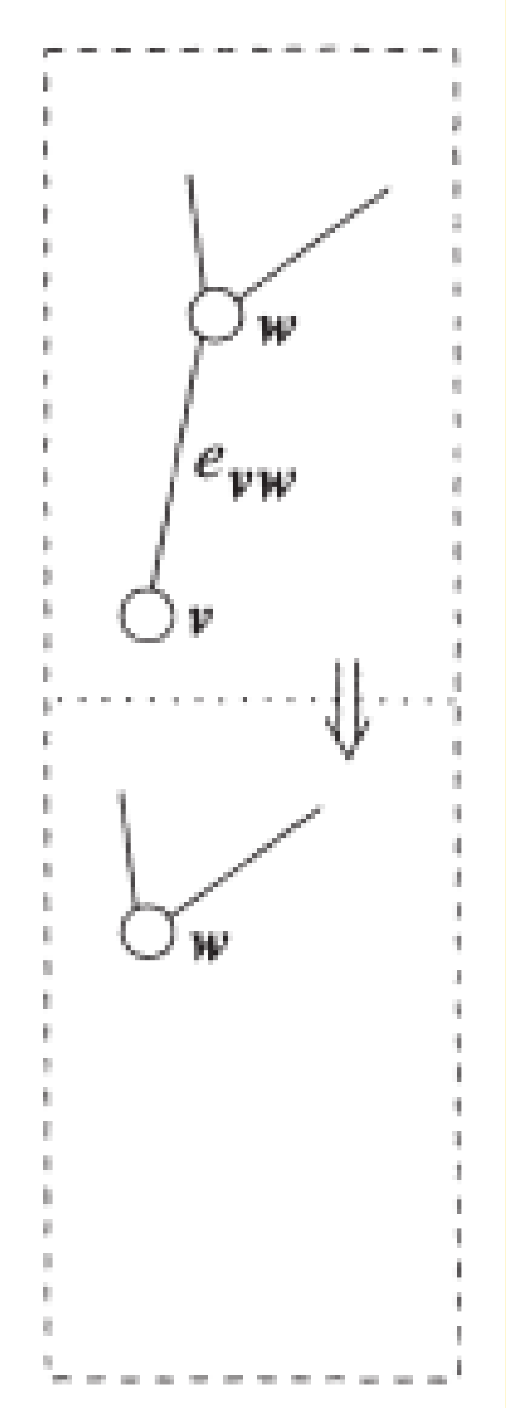
3

Degree of links of Steiner points

Each optimal Steiner point's connections form angles of 120° degrees.

OPTIMIZATION

Algorithm




```
while 1 in counter or 2 in counter:

    for index in range(len(steiner_point)):

        if counter[index] == 1:
            deleted_connection=[]

            for connection in connections:

                if steiner_point[index] in connection:
                    deleted_connection.append(connection)

            connections.remove(deleted_connection[0])
            counter = n_conn(steiner_point, connections)
            break
```

OPTIMIZATION

Algorithm

```python
elif counter[index] == 2:
    new_connection = []
    deleted_connection=[]
    new_points = []
    for connection in connections:

        if steiner_point[index] == connection[0]:
            new_points.append(connection[1])
            deleted_connection.append(connection)
        elif steiner_point[index] == connection[1]:
            new_points.append(connection[0])
            deleted_connection.append(connection)

    if [new_points[0], new_points[1], math.dist(new_points[0], new_points[1])] not in connections:
        new_connection.append([new_points[0], new_points[1], math.dist(new_points[0], new_points[1])])

    for deletion in deleted_connection:
        connections.remove(deletion)

    connections = connections + new_connection
    counter = n_conn(steiner_point, connections)
    break
```

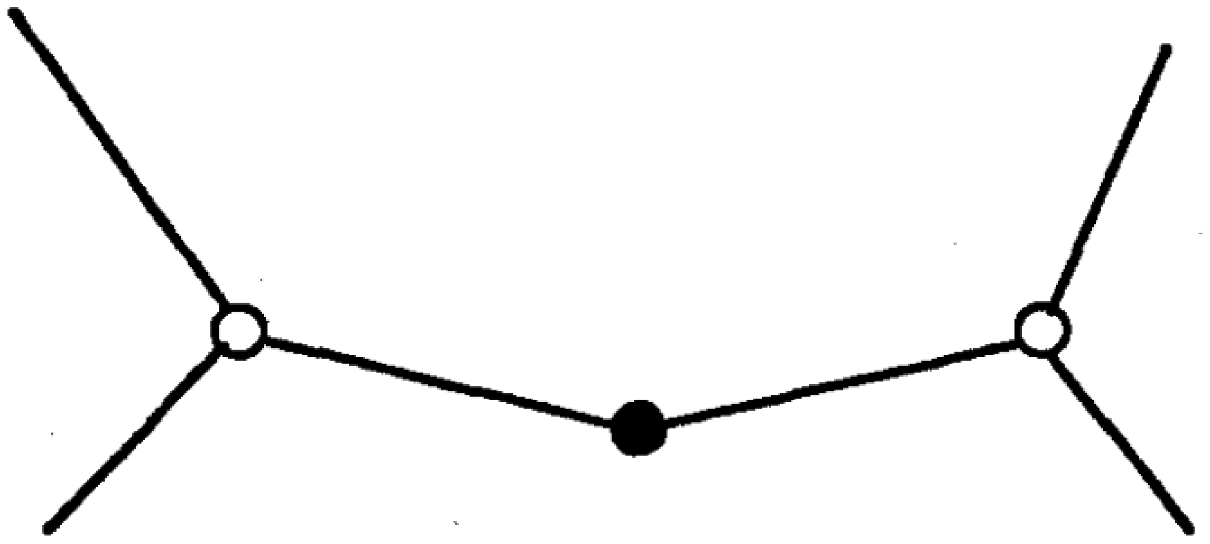


Figure 5: A Steiner point of degree 2.

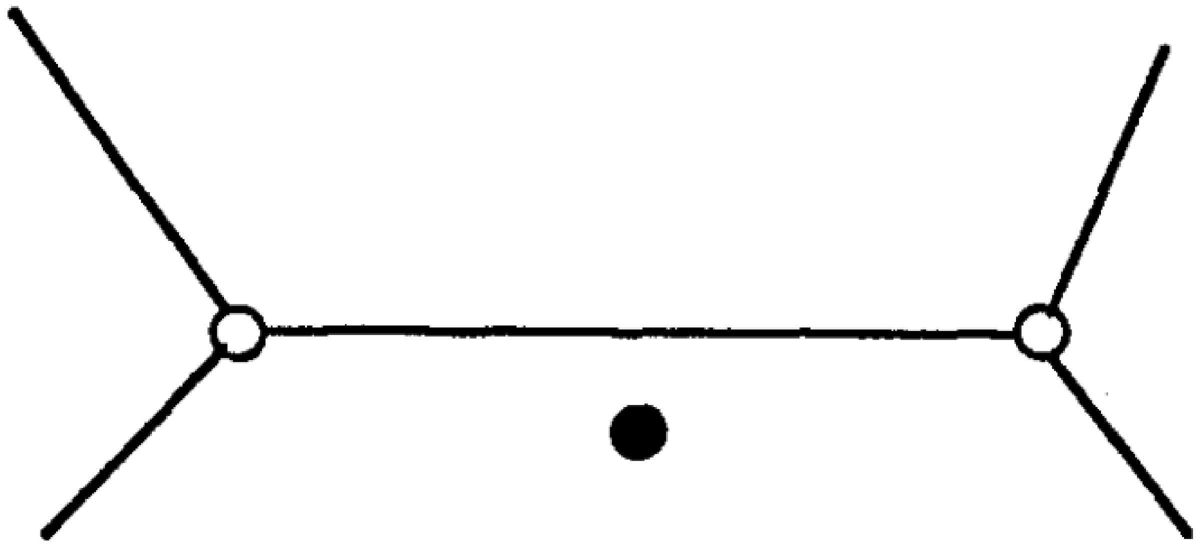
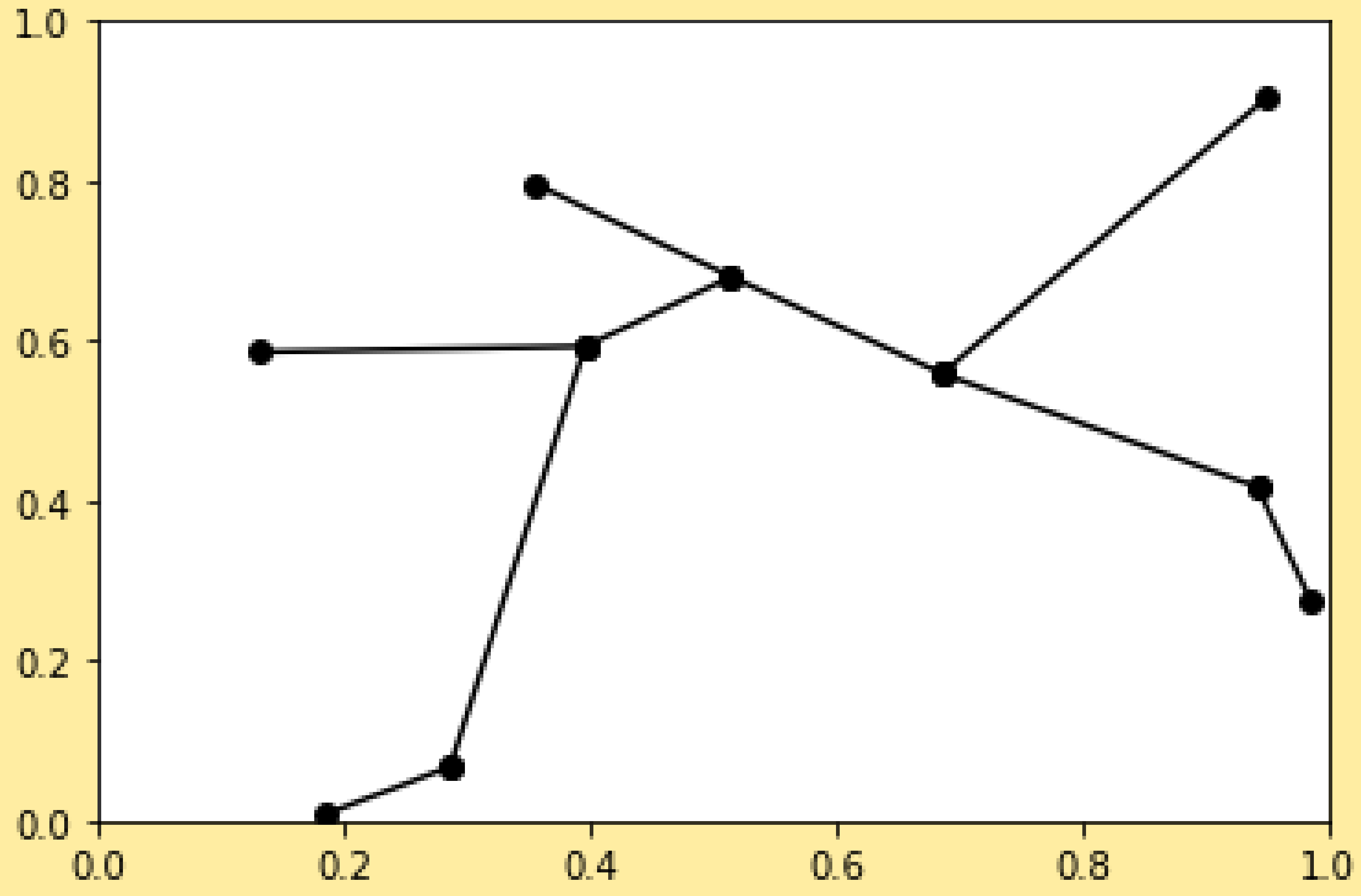


Figure 6: Removal of a Steiner point of degree 2.

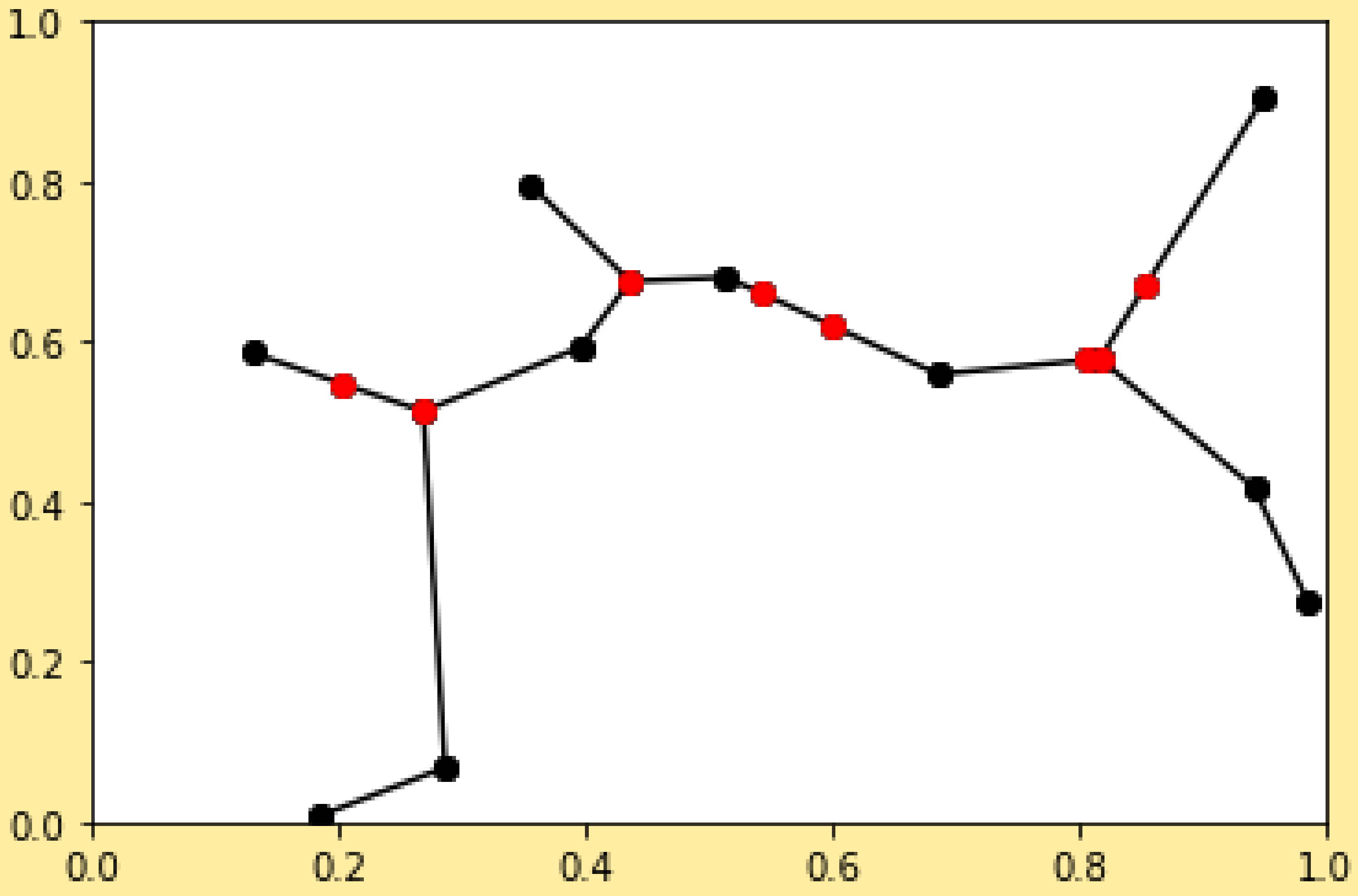
OPTIMIZATION

Example

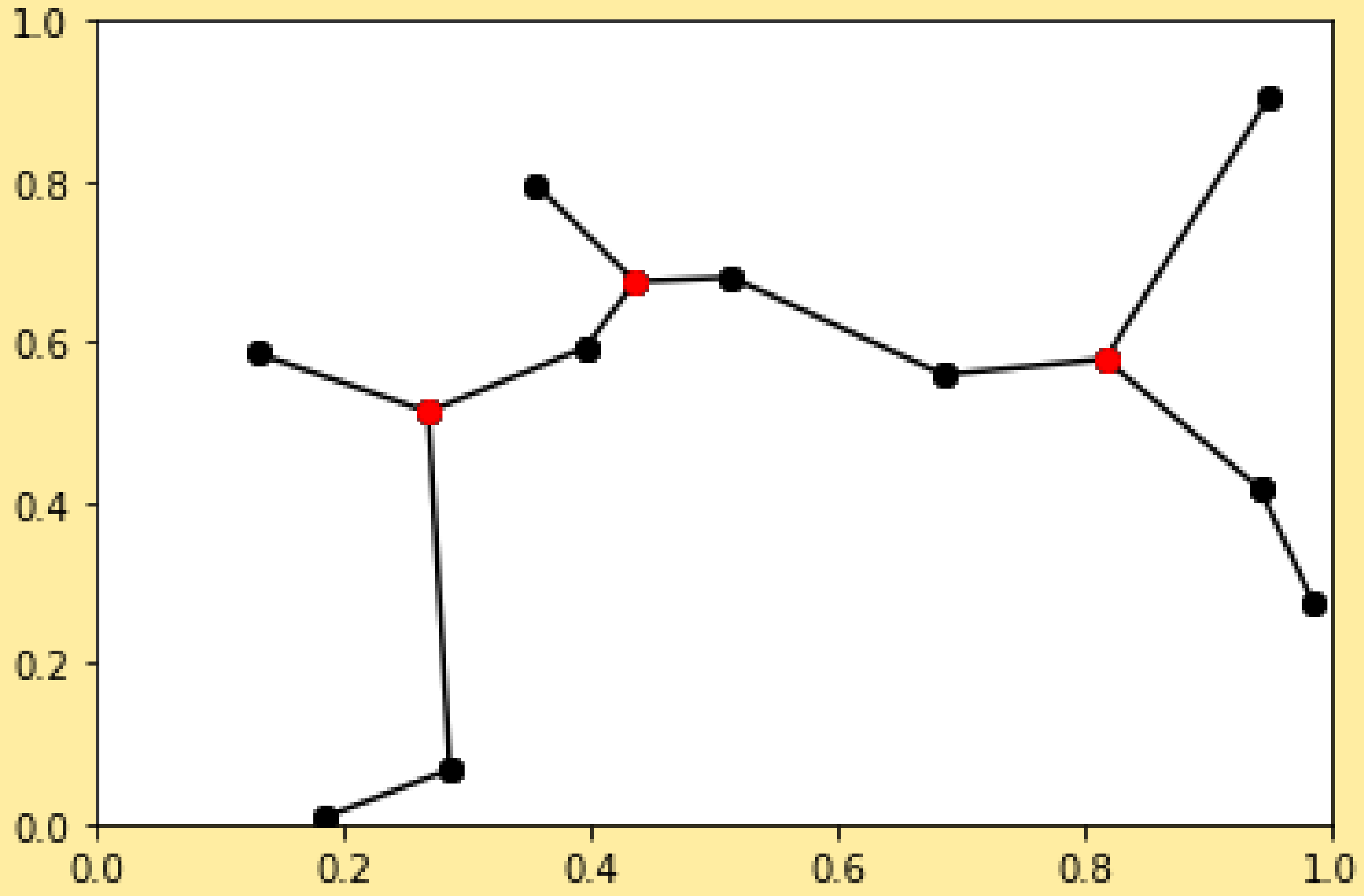
1



2



3

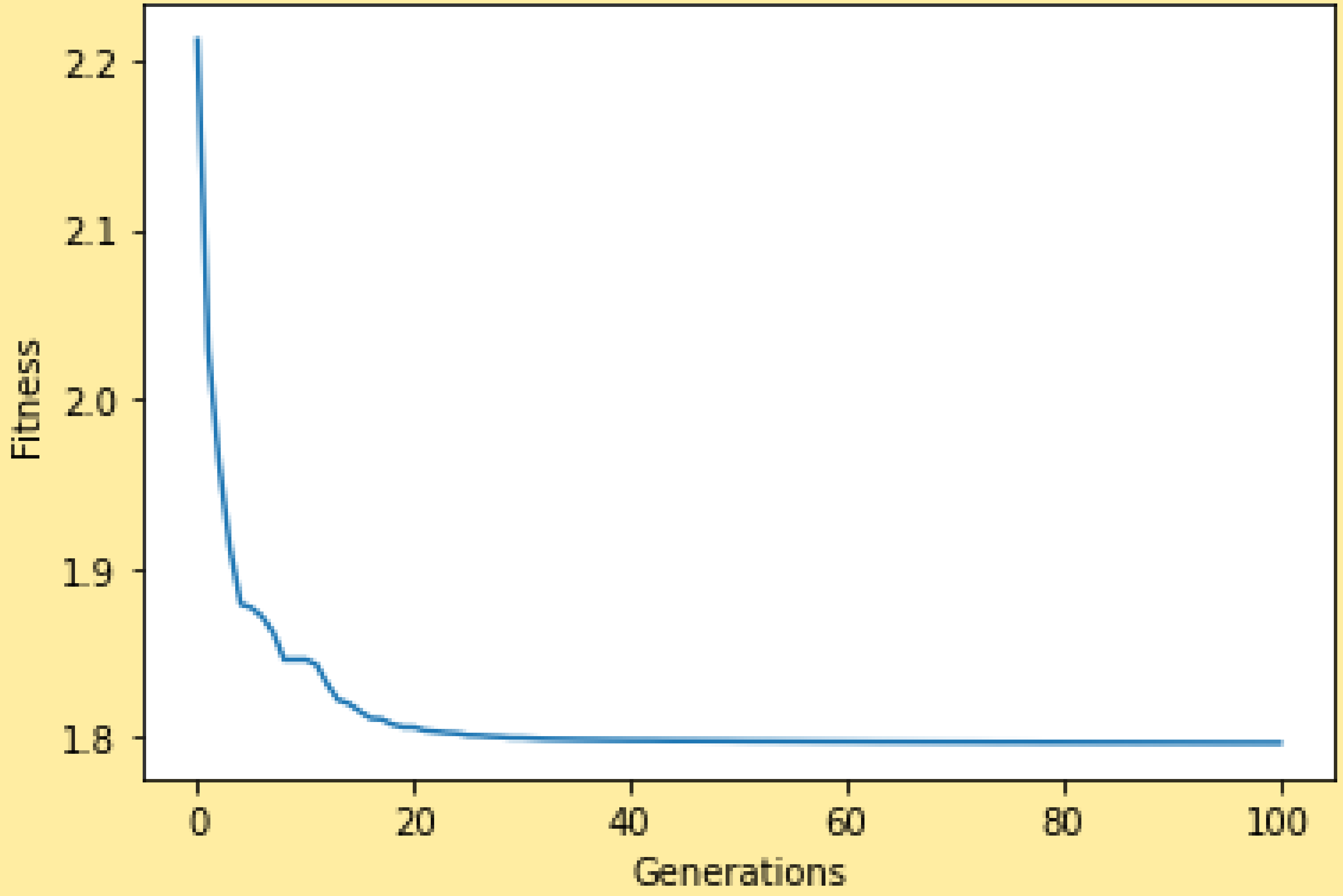


PERFORMANCES

Fitness VS Generations

Population size = 400

Number of Generation = 100



After several run we noticed that the curve was reaching a plateau after around (at maximum) 50 generations.

For this reason, we set the number of generations of all following runs to 50.

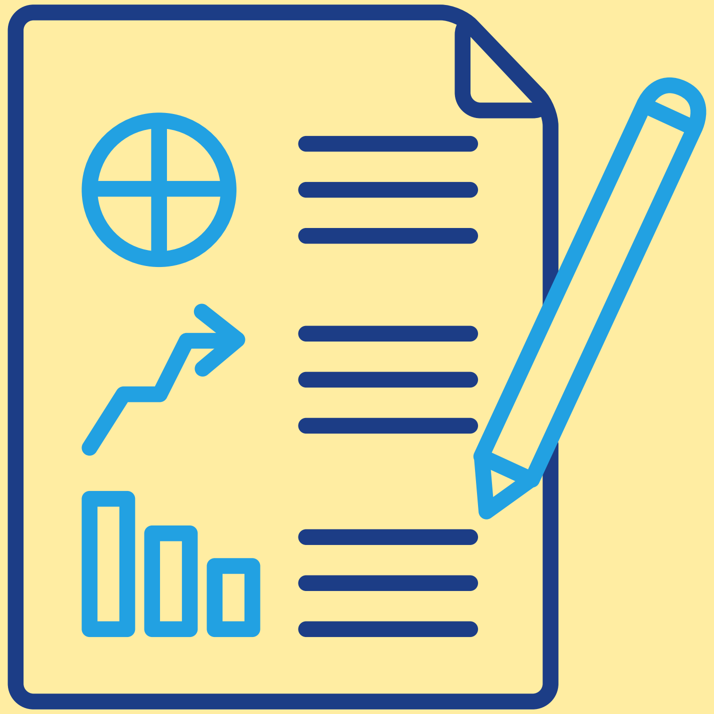
PERFORMANCES

Metrics

MST = length of minimum spanning tree (only including the fixed points).
MEStT = length of optimized minimum spanning tree (including both fixed and Steiner points).

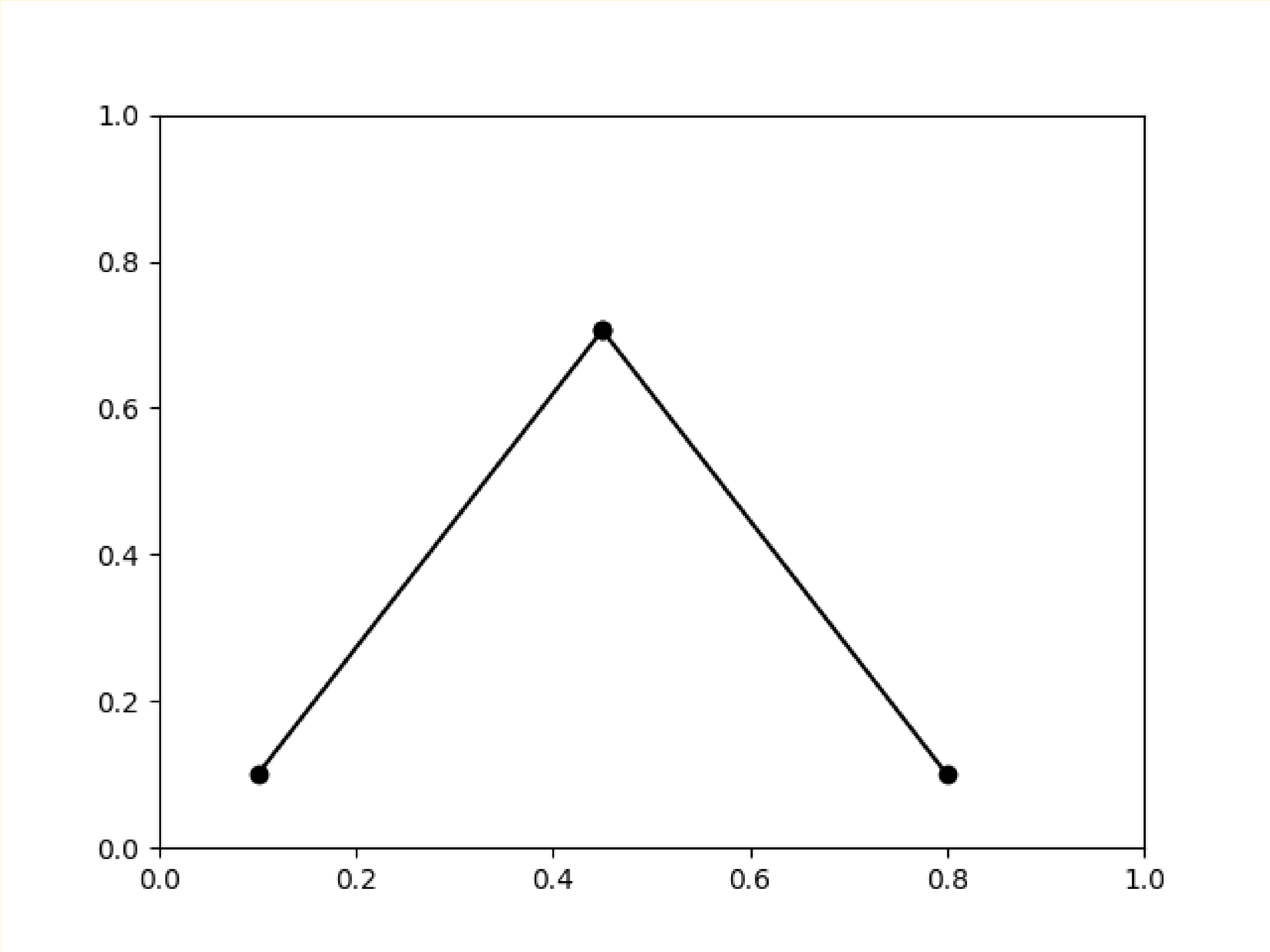
The metrics used for the evaluation of performances are:

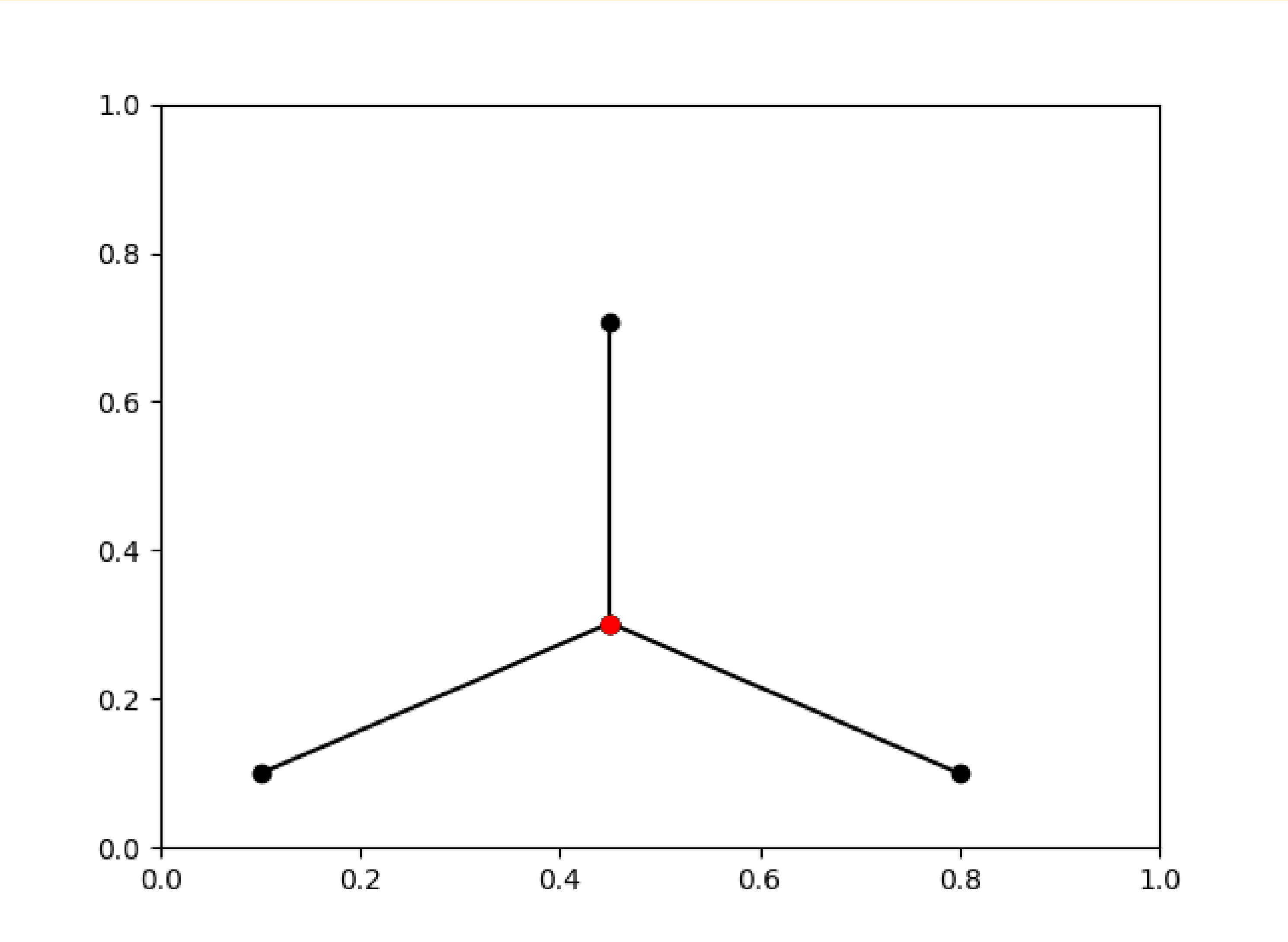
- **Difference = MST - MEStT**
- **Ratio = MEStT / MST** (a lower ratio represents a better approximization)



PERFORMANCES

Example (n = 3)





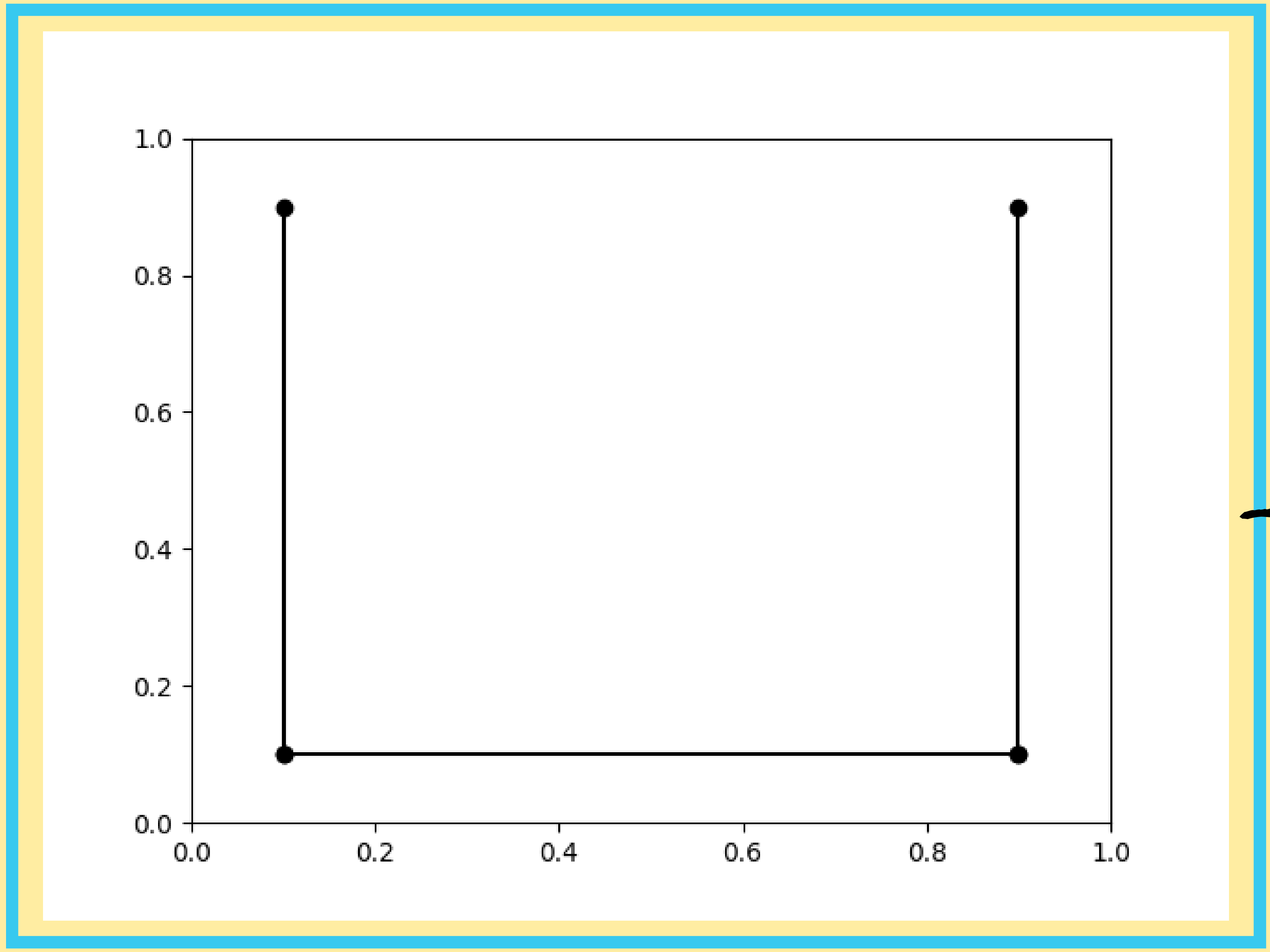
MST

```
MST length =  1.3996228063303342
MEStT length =  1.2122177826491067
Difference =  0.18740502368122747
Ratio =  0.8661031937793412
```

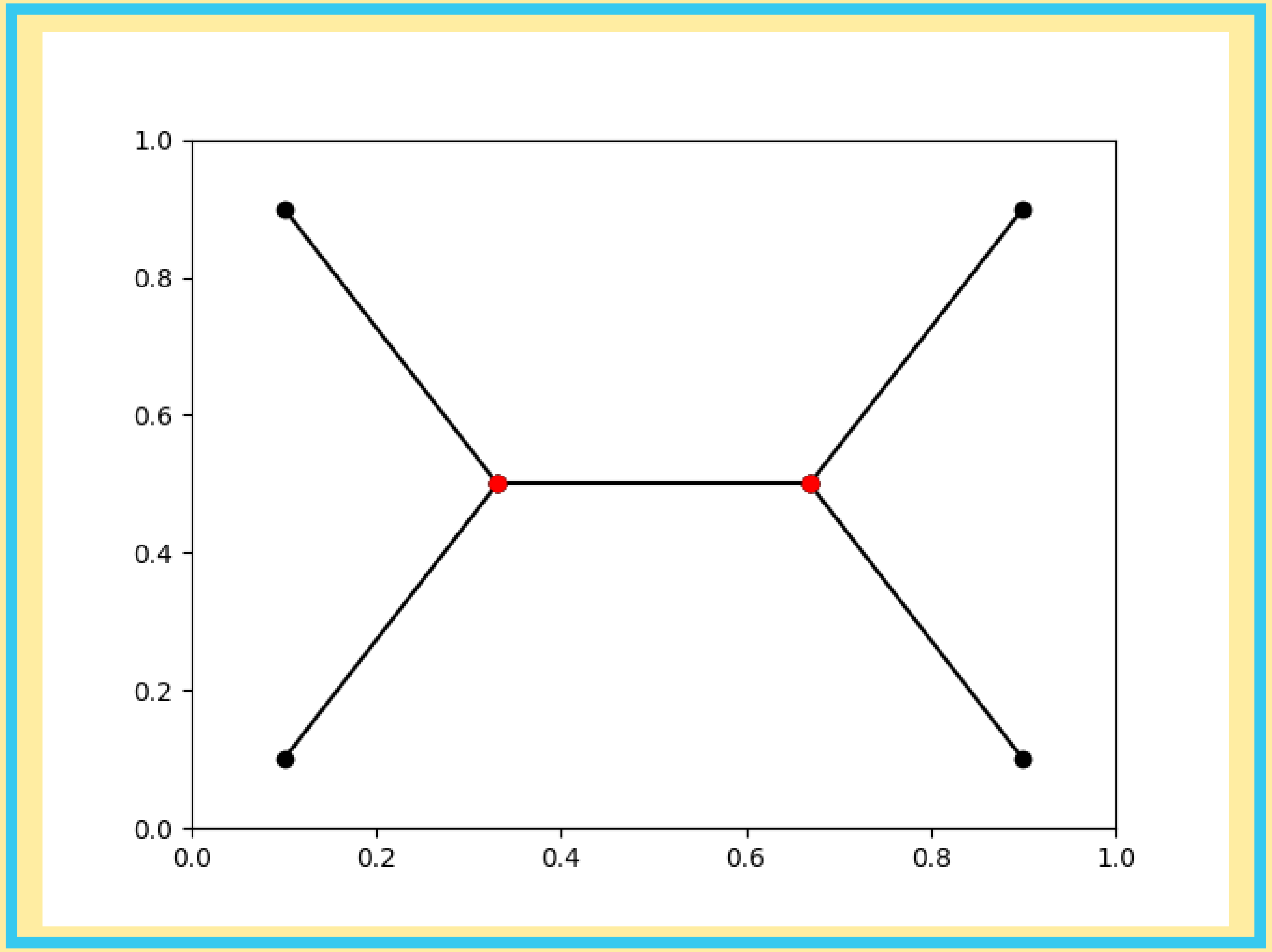
MEStT

PERFORMANCES

Example (n = 4)



MST

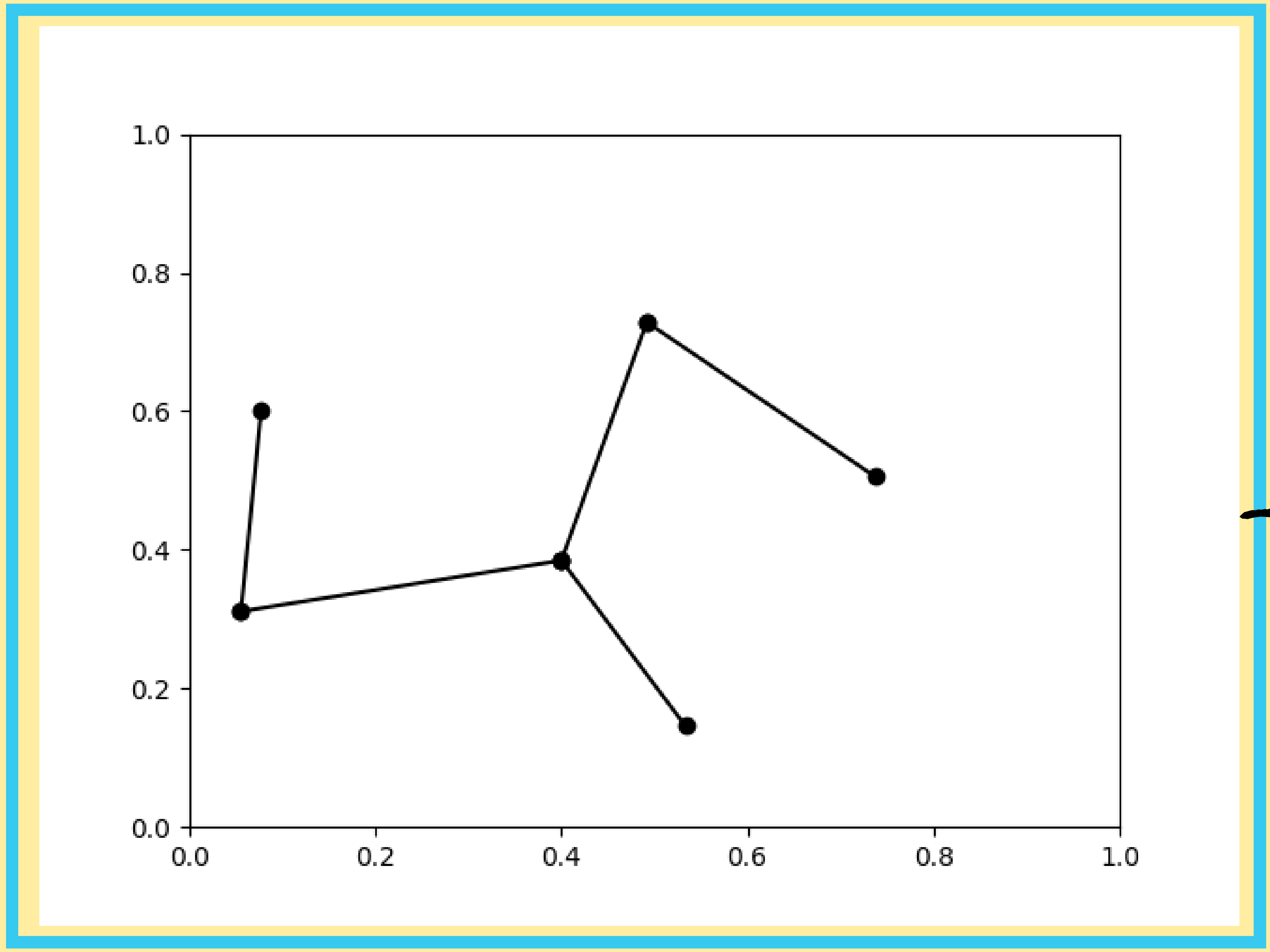


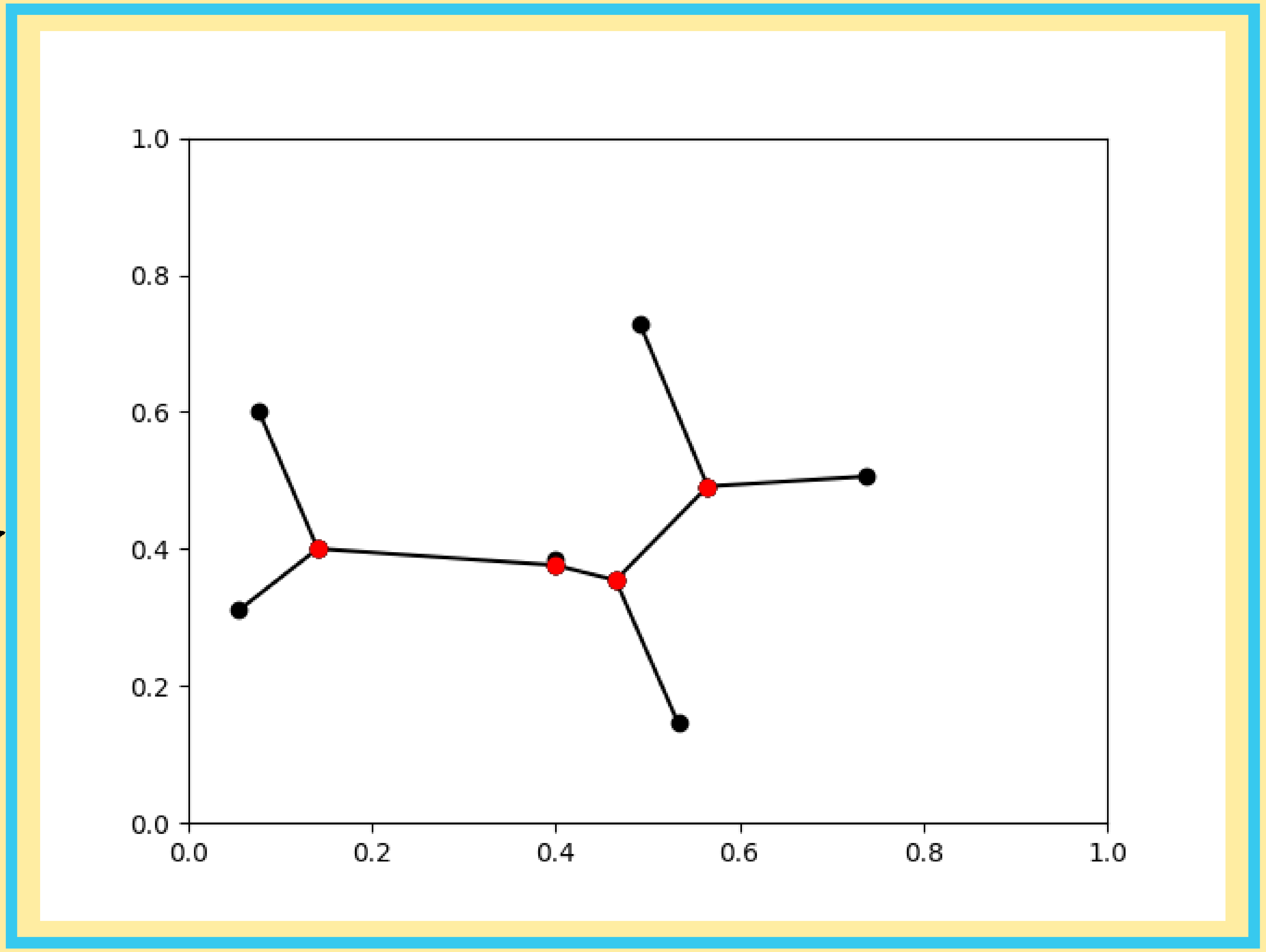
MEStT

```
MST length =  2.4000000000000004
MEStT length =  2.185640646055102
Difference =  0.21435935394489825
Ratio =  0.9106836025229591
```

PERFORMANCES

Example (n = 6)





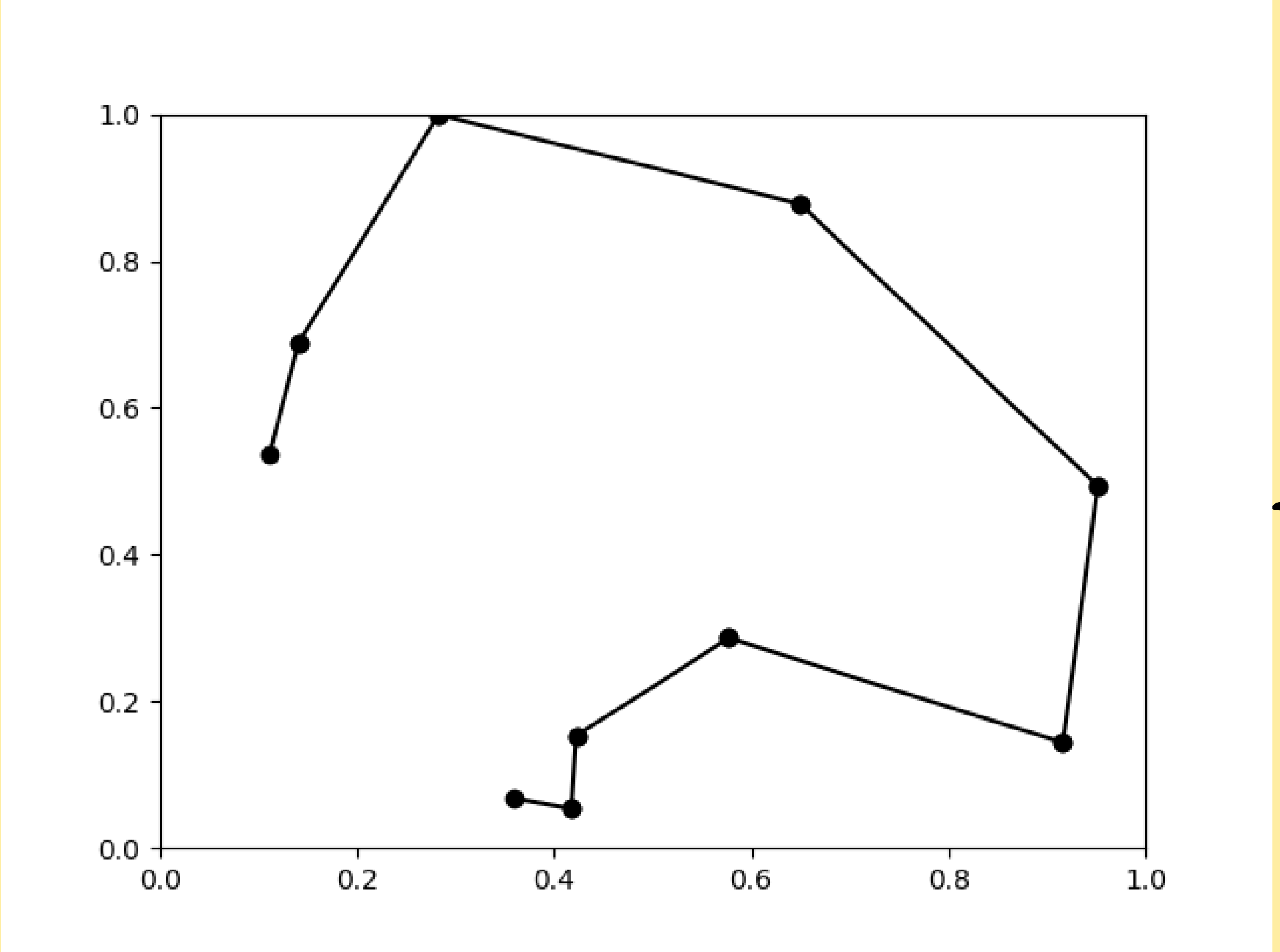
MST

```
MST length =  1.6038844389241862
MEStT length =  1.48114990552657
Difference =  0.1227345333976162
Ratio =  0.9234766979347084
```

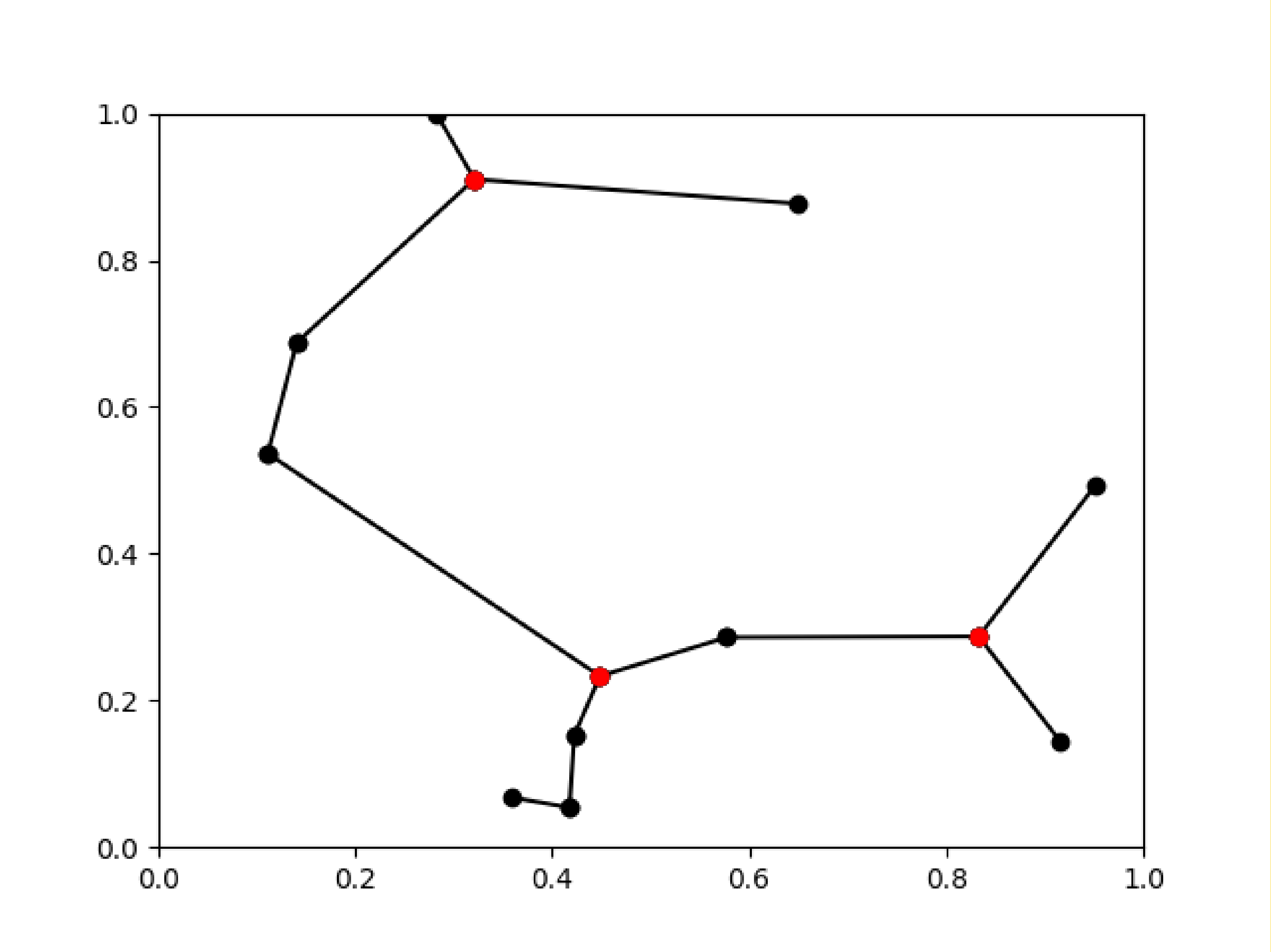
MEStT

PERFORMANCES

Example (n = 10)



MST

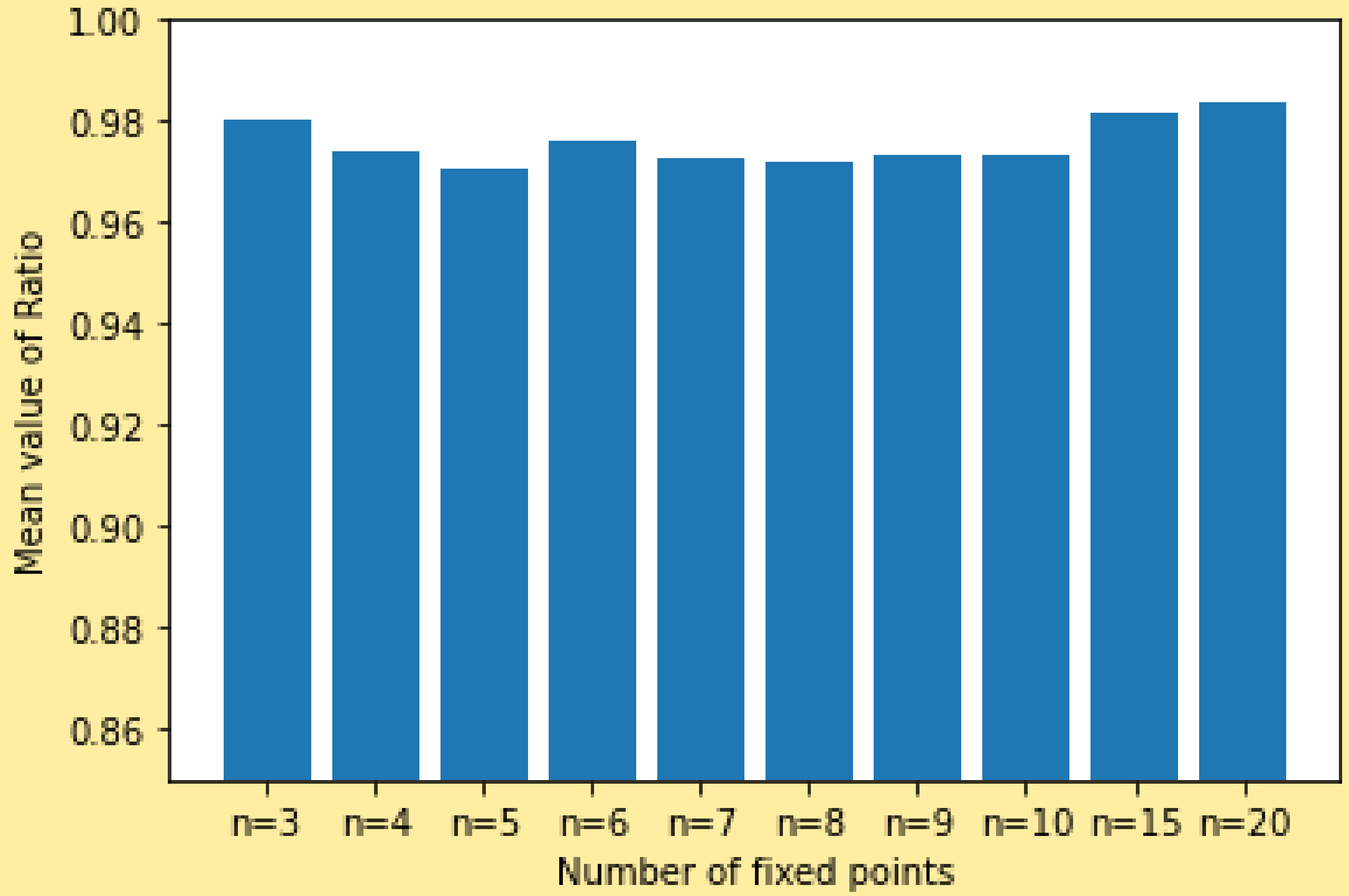


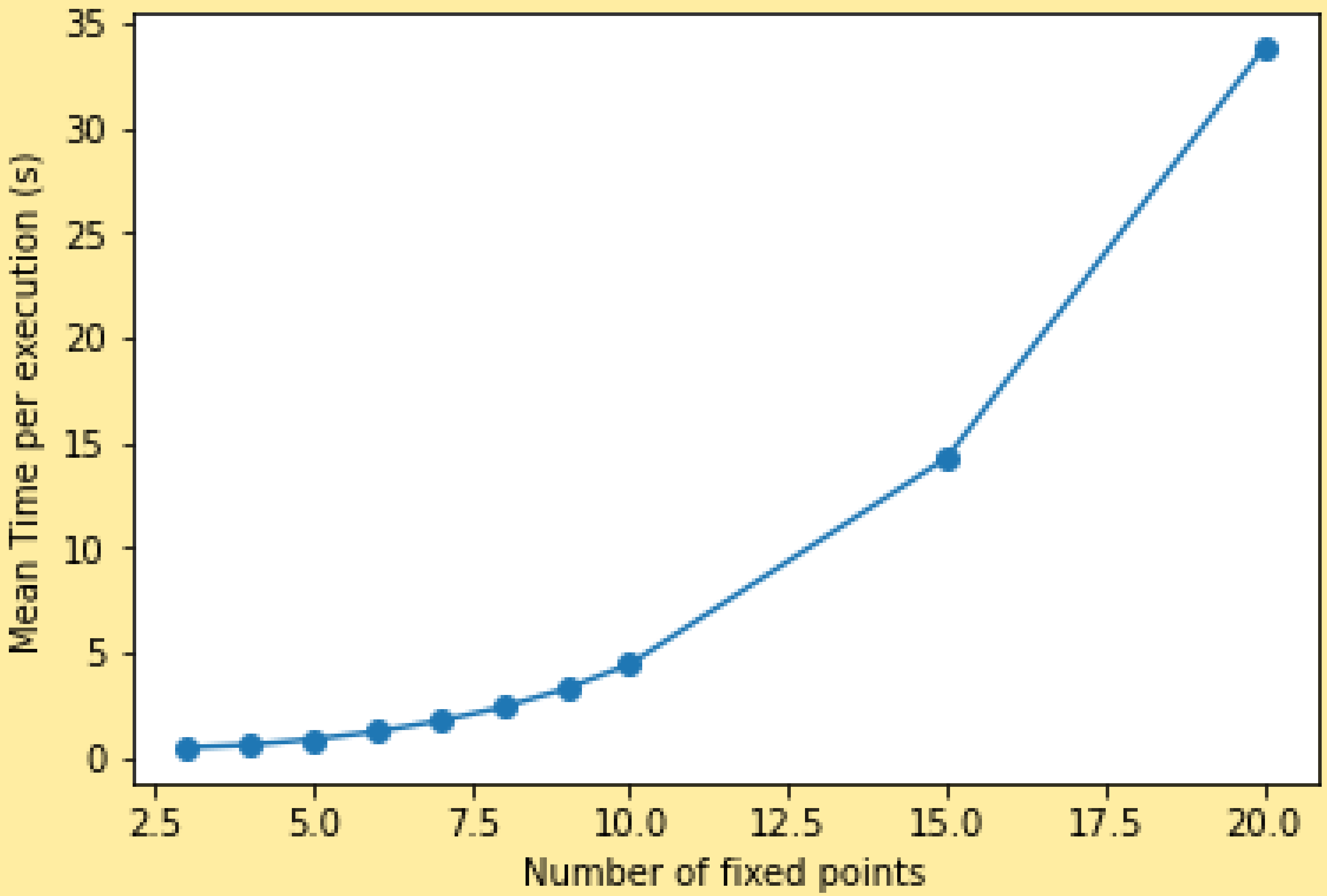
MEStT

```
MST length =  2.453447154785826
MEStT length =  2.36311782067855
Difference =  0.09032933410727617
Ratio =  0.9631826860704642
```

PERFORMANCES

Statistics





	n = 3	n = 4	n = 5	n = 6	n = 7	n = 8	n = 9	n = 10	n = 15	n = 20
Ratio	0.9799	0.9739	0.9706	0.9758	0.9724	0.9716	0.9734	0.9734	0.9816	0.9838
Time (s)	0.46	0.59	0.85	1.21	1.73	2.40	3.28	4.43	14.26	33.77

OBSERVATIONS

The problem can be trasformed in a **maximization** one, consequently, it is possible to use different types of crossover operators.

It is possible to use the 3rd heuristic about the **angle** of the links.

the **number of Steiner points** may be occasionally set to less than: (fixed points - 2).

Increasing the **number of fixed point** still considerably increase the computational time.

GENETIC ALGORITHM

References

[1] = "Combinatorial Optimizatioin, Steiner Trees in industry", Xiuzhen Cheng and Ding - Zhu Du
DOI : 10.1007/978-1-4613-0255-1

[2] = "On steiner trees and genetic algorithms", J. Hesser, R. Männer and O. Stucky
DOI: 10.1007/3-540-55027-5_30

Thank you!