

# TSP\_FA

April 26, 2022

```
[6]: import numpy as np
import pandas as pd

df = pd.read_csv('./adj_mat_kota.csv')

[7]: gen_individu = lambda n_individu,n_kota,a,b: np.random.uniform(
    ↪a,b,(n_individu,n_kota))

def calc_dist(X,adj_mat):
    return sum( map( lambda x,y: adj_mat[x,y] ,X,np.roll(X,-1) ))

def diskritisasi(kunangs):
    return np.argsort(kunangs)

def calculate_fitness(kunangs,df):
    d_kunangs = diskritisasi(kunangs)
    fitness = np.array( list(map( lambda x: calc_dist( x ,df.values) ,
    ↪d_kunangs )) )
    fitness = fitness.reshape( (-1,1) )
    return np.concatenate( ( kunangs ,fitness ) ,axis=1)

def sort_individu(kunangs_with_f):
    return kunangs_with_f[kunangs_with_f[:,-1].argsort()]

def solusi(kunangs_w_f):
    df_kota = pd.DataFrame(diskritisasi(kunangs_w_f[:,-1]))
    cols = [ 'Urutan ' + str(i+1) for i in range( df_kota.shape[1]) ]
    df_kota.columns = cols
    df_kota['Jarak'] = kunangs_w_f[:,-1].reshape(-1,1)
    return df_kota

def movement( X, i,j ,p):
    r = np.linalg.norm(X[i,:] - X[j,:])
    term1 = p['beta0'] * np.exp(-1 * p['gamma'] * r**2 )
    term2 = X[j,:] - X[i,:]
    term3 = p['alpha'] * np.random.uniform(0,1)
    return X[i,:] + ( term1 * term2 ) + term3
```

```

def inisialisasi(params,df):
    kunangs = []
    ↪gen_individu(int(params['n_individu']),int(params['n_kota']),params['a'],params['b'])
    kunangs_w_f = sort_individu(calculate_fitness(kunangs,df))
    return kunangs_w_f

def FA(params,df):

    generasi = 0
    new_kunangs_w_f = inisialisasi(params,df)
    temp = np.zeros_like(new_kunangs_w_f[:, :-1])

    while generasi<params['max_generasi']:

        for i in range(int(params['n_individu'])):
            for j in range(int(params['n_individu'])):
                if ( new_kunangs_w_f[i,-1] > new_kunangs_w_f[j,-1] ) and (i!=j):
                    ↪:
                        temp[i,:] = movement(new_kunangs_w_f[:, :-1],i,j,params)

        new_kunangs_w_f = sort_individu(calculate_fitness(temp,df))
        temp = new_kunangs_w_f[:, :-1]

        generasi = generasi+1

    return solusi(new_kunangs_w_f)

def run_FA(dfparams,df):
    return [ FA( dfparams.loc[i].to_dict() ,df) for i in range( dfparams.
    ↪shape[0]) ]

def save_FA(hasils):
    for h in enumerate(hasils):
        pd.DataFrame(h[1]).to_csv('hasil/hasil_' + str(h[0]) + '.csv')

```

```

[8]: # Main Program
dfparams = pd.read_csv('./params/FA_params.csv')
dfparams['n_kota'] = 10

```

```

[9]: hasils = run_FA(dfparams,df)

```

```

[14]: # hasils[1]

```

```

[15]: save_FA(hasils)

```