

KP_FA

April 26, 2022

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

df = pd.read_csv('./knapsack.csv')
df.sum()
```

```
[1]: Unnamed: 0      45
Weight      52
Profit     520
dtype: int64
```

```
[2]: df
```

```
[2]:   Unnamed: 0  Weight  Profit
0           0        3      10
1           1        3      90
2           2        6      30
3           3        9      90
4           4        5      10
5           5        1      40
6           6        7      80
7           7        8      60
8           8        9      40
9           9        1      70
```

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

def f_constrain(X,df,lim):
    return np.sum( X* df['Weight'].values ) <= lim

def f_profit(X,df):
    return np.sum(X * df['Profit'].values)

def f_obj(X,df,lim):
    return f_profit(X,df) if f_constrain(X,df,lim) else 0
```

```

def diskritisasi(kunangs):
    return np.round( 1/ ( 1 + np.exp(-1 * kunangs) ) )

def calculate_fitness(kunangs,df,p):
    d_kunangs = diskritisasi(kunangs)
    fitness = np.array( list(map( lambda x:f_obj(x,df,p['lim']) , 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()[::-1]]

def solusi(kunangs_w_f):
    df_kota = pd.DataFrame(diskritisasi(kunangs_w_f[:, :-1]))
    cols = [ 'Barang ' + str(i+1) for i in range( df_kota.shape[1]) ]
    df_kota.columns = cols
    df_kota['Profit'] = 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,params))
    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,params))
        temp = new_kunangs_w_f[:, :-1]

        # print(list(new_kunangs_w_f[:5][-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')

```

```

[4]: # Main Program
dfparams = pd.read_csv('./params_KP.csv')
dfparams['n_kota'] = 10
dfparams['lim'] = 30
dfparams['a'] = -8
dfparams['b'] = 8
dfparams['alpha'] = 3
dfparams['max_generasi'] = 2
# dfparams['']
dfparams

```

```

[4]:
   n_individu  n_kota  a  b  max_generasi  alpha  beta0  gamma  lim
0          100     10 -8  8              2      3      1  0.00001  30
1           50     10 -8  8              2      3      1  0.10000  30
2           30     10 -8  8              2      3      1  0.10000  30
3           20     10 -8  8              2      3      1  0.10000  30
4           10     10 -8  8              2      3      1  0.10000  30
5           10     10 -8  8              2      3      1  0.01000  30
6           10     10 -8  8              2      3      1  0.00100  30

```

```

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

```

```

[6]: # save_FA(hasils)

```

```

[7]: hasils[4]

```

```

[7]:
   Barang 1  Barang 2  Barang 3  Barang 4  Barang 5  Barang 6  Barang 7  \
0         1.0         0.0         0.0         0.0         0.0         1.0         1.0
1         1.0         1.0         1.0         1.0         1.0         1.0         1.0
2         1.0         1.0         1.0         1.0         1.0         1.0         1.0
3         1.0         1.0         1.0         1.0         1.0         1.0         1.0

```

4	1.0	1.0	1.0	1.0	1.0	1.0	1.0
5	1.0	1.0	1.0	1.0	1.0	1.0	1.0
6	1.0	1.0	1.0	1.0	1.0	1.0	1.0
7	1.0	1.0	1.0	1.0	1.0	1.0	1.0
8	1.0	1.0	1.0	1.0	1.0	1.0	1.0
9	1.0	1.0	1.0	0.0	1.0	0.0	1.0

	Barang 8	Barang 9	Barang 10	Profit
0	1.0	1.0	1.0	300.0
1	1.0	1.0	1.0	0.0
2	1.0	1.0	1.0	0.0
3	1.0	1.0	1.0	0.0
4	1.0	1.0	1.0	0.0
5	1.0	1.0	1.0	0.0
6	1.0	1.0	1.0	0.0
7	1.0	1.0	1.0	0.0
8	0.0	0.0	0.0	0.0
9	1.0	1.0	1.0	0.0

[]:

[]: