CS_KP

April 26, 2022

[2]:
```
import numpy as np
import pandas as pd
from scipy.stats import levy_stable as l
```

[3]:
```
df = pd.read_csv('./knapsack.csv')
df.sum()
df
```

[3]:
```
   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
```

[4]:
```
def movement( X , p):
    levy = l.rvs( p['lb'][0] , p['lb'][1] , size=X.shape)
    return  X + p['alpha'] * levy
```

[5]:
```
def selec(new_cuckoos,p):
    pa = np.round( np.random.uniform(0,1) * (new_cuckoos.shape[0] -1) ).
 ↪astype(int)
    sz = new_cuckoos[:-pa][:].shape
    new_cuckoos[:-pa,:] = np.random.uniform(p['a'],p['b'], size=(sz))
    return new_cuckoos
```

[6]:
```
def solusi(cuckoos_w_f):
    df_barang = pd.DataFrame(diskritisasi(cuckoos_w_f[:,:-1]))
    cols = [ 'Barang ' + str(i+1) for i in range( df_barang.shape[1]) ]
    df_barang.columns = cols
    df_barang['Profit'] =  cuckoos_w_f[:,-1].reshape(-1,1)
    return df_barang
```

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

     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(cuckoos):
         return np.round( 1/ ( 1 + np.exp(-1 * cuckoos) ) )

     def calculate_fitness(cuckoos,df,p):
         d_cuckoos = diskritisasi(cuckoos)
         fitness = np.array(  list(map( lambda x:f_obj(x,df,p['lim']) , d_cuckoos ))␣
      ↪)
         fitness = fitness.reshape( (-1,1) )
         return np.concatenate( ( cuckoos ,fitness ) ,axis=1)

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

     def solusi(cuckoos_w_f):
         df_barang = pd.DataFrame(diskritisasi(cuckoos_w_f[:,:-1]))
         cols = [ 'Barang ' + str(i+1) for i in range( df_barang.shape[1]) ]
         df_barang.columns = cols
         df_barang['Profit'] =  cuckoos_w_f[:,-1].reshape(-1,1)
         return df_barang

     def inisialisasi(params,df):
         cuckoos =␣
      ↪gen_individu(int(params['n_individu']),int(params['n_barang']),params['a'],params['b'])
         cuckoos_w_f = sort_individu(calculate_fitness(cuckoos,df,params))
         return cuckoos_w_f

     def CS(params,df):

         generasi = 0

         # new_cuckoos_w_f = inisialisasi(params,df)
         cuckoos =␣
      ↪gen_individu(int(params['n_individu']),int(params['n_barang']),params['a'],params['b'])
         cuckoos_w_f = sort_individu(calculate_fitness(cuckoos,df,params))
```

```
    temp = np.zeros_like(cuckoos_w_f[:,:-1])
    while generasi<params['max_generasi']:

        #bangkitkan cuckoo secara acak dengan levy flight
        cuckoos = movement(cuckoos,p)

        #evaluasi fitness cuckoo
        new_cuckoos_w_f = sort_individu(calculate_fitness(cuckoos,df,params))

        # seleksi
        cuckoos = selec(cuckoos,params)

        #next generasi
        generasi = generasi+1

    return solusi(cuckoos_w_f)

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

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

[24]:
```
# Main Program
dfparams = pd.read_csv('./FA_params_KP.csv')
dfparams['n_barang'] = 10
dfparams['lim'] = 30
dfparams['a'] = -4
dfparams['b'] = 4
dfparams['max_generasi'] = 2
# dfparams['']
dfparams

p = { 'alpha': 1, 'lb':[1.8,-0.5] }

hasils = run_CS(dfparams,df)
hasils[4]
```

[24]:
```
   Barang 1  Barang 2  Barang 3  Barang 4  Barang 5  Barang 6  Barang 7  \
0       1.0       1.0       0.0       1.0       1.0       1.0       1.0
1       1.0       1.0       1.0       1.0       0.0       1.0       1.0
2       1.0       1.0       0.0       0.0       0.0       1.0       1.0
3       0.0       1.0       0.0       1.0       0.0       1.0       0.0
4       0.0       1.0       0.0       1.0       0.0       1.0       0.0
5       1.0       0.0       0.0       1.0       1.0       1.0       0.0
```

```
6        1.0        0.0        1.0        0.0        1.0        0.0        1.0
7        1.0        0.0        0.0        0.0        1.0        1.0        0.0
8        1.0        1.0        1.0        1.0        0.0        1.0        1.0
9        1.0        1.0        1.0        1.0        1.0        1.0        1.0

   Barang 8  Barang 9  Barang 10  Profit
0       0.0       0.0        1.0   390.0
1       0.0       0.0        0.0   340.0
2       0.0       1.0        1.0   330.0
3       1.0       1.0        0.0   320.0
4       0.0       0.0        1.0   290.0
5       0.0       0.0        1.0   220.0
6       0.0       0.0        0.0   130.0
7       1.0       0.0        0.0   120.0
8       0.0       1.0        1.0     0.0
9       1.0       1.0        0.0     0.0
```