ncaa_region_optimizer

April 6, 2020

1 Genetic Algorithms for Region Paritioning

We will be using some Python modules installed by pip rather than Anaconda, so I must adjust the import path.

[1]:
```
import sys
sys.path.insert( 1, '/usr/local/lib/python3.7/site-packages' )
```

Import other packages.

[2]:
```
import pandas
from geopy.distance import great_circle
import random
import statistics
from plotly_for_usa_points import usa_map # this works only if you've done␣
↪conda install plotly
# also one needs all the extensions mentioned here:
# https://github.com/plotly/plotly.py#jupyterlab-support-python-35
from tqdm.notebook import tqdm
from ga_for_partitions import optimize_partition, set_seed
import math
from matplotlib import pyplot as plt
```

[3]:
```
%matplotlib inline
```

1.1 Import the data

[4]:
```
data_filename = 'wrestling-schools-data.csv'
df = pandas.read_csv( data_filename )
len( df )
```

[4]: 106

[5]:
```
df.head()
```

```
[5]:    UniqueID          College/University Name                  Street  \
0          1                   Adrian College         10 S Madison St
1          2  Alfred State College  (add 2018)  10 Upper College Drive
2          3                     Alma College        614 W Superior St
3          4                         Augsburg      2211 Riverside Ave
4          5                   Augustana (IL)             639 38th St

          City  State   Latitude  Longitude  Power-1  Power-2  NCAA Asgt  \
0       Adrian    MI  41.899337 -84.044547   2.4514    2.927          3
1       Alfred    NY  42.254334 -77.789646   0.0000    0.000          0
2         Alma    MI  43.380011 -84.655654   5.1091    5.941          3
3  Minneapolis    MN  44.963541 -93.267835   9.6340    8.890          2
4  Rock Island    IL  41.470591 -90.583733   0.0000    0.301          1

   ND Asgt  ND Asgt2  ND Asgt3
0      2.0       1.0       5.0
1      NaN       NaN       NaN
2      6.0       1.0       5.0
3      6.0       5.0       3.0
4      2.0       4.0       3.0
```

1.1.1 Drop schools we don't want in this analysis

Some schools were dropped for various domain-specific reasons. See paper.

```
[6]: df = df.drop( [ 31, 61, 85 ] )
     num_schools = len( df )
     num_schools
```

```
[6]: 103
```

1.1.2 Make it easy to fetch desired rows/columns

```
[7]: def school ( key ):
         column = 'UniqueID' if type( key ) == int else 'College/University Name'
         return df[df[column] == key].iloc[0]
     ( SCH_ID, SCH_NAME, SCH_ADDR, SCH_CITY, SCH_STATE, SCH_LAT, SCH_LNG, SCH_POW1,␣
      ↪SCH_POW2,
       SCH_NCAA, SCH_ND1, SCH_ND2, SCH_ND3 ) = list( df.columns.values )
     def all_ids ():
         return list( df['UniqueID'] )
     def index_to_id ( index ):
         return all_ids()[index]
     # print( school( 2 )[SCH_NAME] )
     # print( school( 'Augsburg' )[SCH_ID] )
     # print( school( 50 )[SCH_LAT], get_school( 50 )[SCH_LNG] )
```

1.2 Map distance tools

Define measure for computing distance on the (curved) surface of the earth.

[8]:
```
def school_latlng ( school ):
    return ( school[SCH_LAT], school[SCH_LNG] )
```

Now pre-compute the distance between any two pair of schools and cache it in a matrix, because we'll be asking these distance questions a million times below, and this cache will speed it up a lot.

[9]:
```
school_locations = [ school_latlng( school( index_to_id( i ) ) )
                     for i in range( num_schools ) ]
distance_matrix = [ [ great_circle( school_locations[i], school_locations[j] ).
 ↪miles
                      for i in range( num_schools ) ] for j in range(␣
 ↪num_schools ) ]
def distance_lookup ( school_index1, school_index2 ):
    return distance_matrix[school_index1][school_index2]
```

1.3 Utilities for partitions

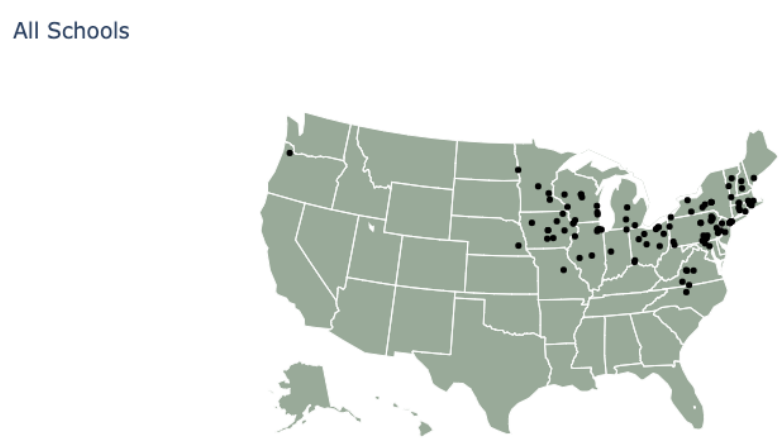
[10]:
```
num_parts_in_partition = 6
def indices_for_part_in_partition ( part_index, partition ):
    return [ i for i in range( len( partition ) ) if partition[i] == part_index␣
 ↪]
def schools_in_part_in_partition ( part_index, partition ):
    return [ school( index_to_id( i ) )
             for i in indices_for_part_in_partition( part_index, partition ) ]
def size_of_part_in_partition ( part_index, partition ):
    return len( indices_for_part_in_partition( part_index, partition ) )
def random_partition ():
    return [ random.randint( 0, num_parts_in_partition ) for i in range(␣
 ↪num_schools ) ]
```

[11]:
```
def print_partition ( partition ):
    for part_index in range( num_parts_in_partition ):
        schools = schools_in_part_in_partition( part_index, partition )
        powers = [ school[SCH_POW2] for school in schools ]
        print( 'Region {:1d}, {:2d} schools, mean power {:7.5f} (stdev {:7.5f}):
 ↪'.format(
            part_index + 1, size_of_part_in_partition( part_index, partition ),
            statistics.mean( powers ), statistics.stdev( powers ) ) )
        print( '----------------------------------------------------------' )
        centroid = (
```

```
            statistics.mean( [ school[SCH_LAT] for school in schools ] ),
            statistics.mean( [ school[SCH_LNG] for school in schools ] ),
        )
        print( '    Centroid: {:7.3f} lat, {:7.3f} lon'.format(
            centroid[0], centroid[1] ) )
        latlngs = [ school_latlng( school ) for school in schools ]
        print( '    Mean distance to centroid: {:8.3f} miles'.format(
            statistics.mean( [ great_circle( centroid, latlng ).miles for␣
↪latlng in latlngs ] )
        ) )
        for s in schools:
            print( '        {:30.30s} {:30.30s} {:>7.1f} miles'.format(
                s[SCH_NAME],
                '{}, {}, {}'.format( s[SCH_ADDR], s[SCH_CITY], s[SCH_STATE] ),
                great_circle( school_latlng( s ), centroid ).miles
            ) )
        print()
# print_partition( random_partition() )
```

1.4 Test map-drawing tools

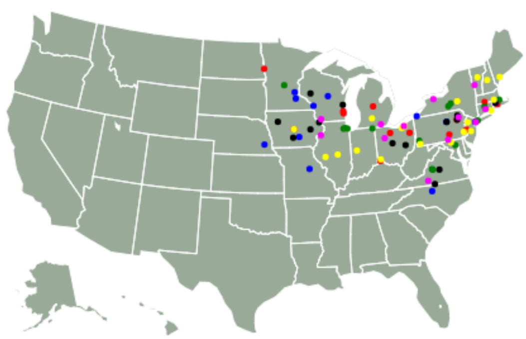
[12]:
```
usa_map( { 'black' : [ school_latlng( school( id ) ) for id in all_ids() ] },␣
↪'All Schools' )
```



[13]:
```
def partition_map ( schools_partition, title = 'Partition of All Schools' ):
```

```
    colors = [ 'red', 'blue', 'green', 'black', 'yellow', 'magenta', 'cyan',␣
↪'white', 'gray', 'orange' ]
    def points_in_part ( part_index ):
        return [ school_latlng( s )
                  for s in schools_in_part_in_partition( part_index,␣
↪schools_partition ) ]
    return usa_map( {
        colors[i] : points_in_part( i ) for i in range( max( schools_partition␣
↪) )
    }, title )
partition_map( random_partition(), 'Plotting a random parition as an example' )
```

Plotting a random parition as an example



1.5 Components of the Objective Function

First, we will want to experiment with the range of the various components of the objective function, to see how we should rescale them to match each other.

[14]:
```
def range_experiment ( func, num_tries=100 ):
    data = [ func( random_partition() ) for i in range( num_tries ) ]
    return min( data ), max( data )
```

1.5.1 Component 1: Variance of size of parts in the partition

[15]:
```
def part_size_variance ( partition ):
    return statistics.variance( [
        size_of_part_in_partition( part, partition )
```

```
        for part in range( num_parts_in_partition )
    ] )
# print( range_experiment( part_size_variance ) ) # gives a max in the 50s
# plus we want size variance to be bad, so we need a -1 multiplier, so:
def obj_fn_component_1 ( partition ):
    return part_size_variance( partition ) * -1.0 / 50
```

1.5.2 Component 2: Total distance between schools in each part of the partition

[16]:
```
def total_distance_in_one_part ( part_index, partition ):
    indices = indices_for_part_in_partition( part_index, partition )
    return sum( [ distance_lookup( i, j )
                  for i in indices for j in indices if i < j ] )
def total_distance_of_all_parts ( partition ):
    return sum( [ total_distance_in_one_part( index, partition )
                  for index in range( num_parts_in_partition ) ] )
# print( range_experiment( total_distance_of_all_parts ) ) # gives a max in to␣
↪50000s
# plus we want travel distance to be bad, so we need a -1 multiplier, so:
def obj_fn_component_2 ( partition ):
    return total_distance_of_all_parts( partition ) * -1.0 / 50000
```

1.5.3 Component 3: Variance of mean powers of each part in partition

[17]:
```
def mean_power_of_part ( part_index, partition ):
    powers = [ s[SCH_POW2] for s in schools_in_part_in_partition( part_index,␣
↪partition ) ]
    if len( powers ) > 0:
        return statistics.mean( powers )
    else:
        return 0
def part_power_variance ( partition ):
    return statistics.variance( [
        mean_power_of_part( part, partition )
        for part in range( num_parts_in_partition )
    ] )
# print( range_experiment( part_power_variance ) ) # gives a max around 5
# plus we want power variance to be bad, so we need a -1 multiplier, so:
def obj_fn_component_3 ( partition ):
    return part_power_variance( partition ) * -1.0 / 5
```

1.5.4 Objective function: sum of 3 components

[18]:
```
def objective_function ( partition ):
    return obj_fn_component_1( partition ) \
        + obj_fn_component_2( partition ) \
        + obj_fn_component_3( partition )
```

1.6 Solving the problem with Genetic Algorithms

[19]:
```
num_generations = 10000
def progress_bar ( name="Progress", size=num_generations ):
    bar = tqdm( range( size ), desc=name )
    def step ( *args ):
        bar.update( 1 )
        bar.display()
    return step
best, fitness_curve = optimize_partition(
    objective_function = objective_function,
    initial_pool = [ random_partition() for i in range( num_parts_in_partition␣
 ↪) ],
    size_of_partition = num_parts_in_partition,
    prob_mutate = 0.1,
    num_generations = num_generations,
    progress_callback = progress_bar()
)
```

```
HBox(children=(FloatProgress(value=0.0, description='Progress', max=10000.0, style=ProgressSty

After 10000 generations: max score =  -1.4149   100% done, 11:11/11:11 (00:00)
```

[20]:
```
print_partition( best )
```

```
Region 1, 11 schools, mean power 2.36036 (stdev 3.62114):
---------------------------------------------------------
    Centroid:  41.834 lat, -92.081 lon
    Mean distance to centroid:  141.262 miles
        Buena Vista                   610 W 4th St, Storm Lake, IA     169.0
miles
        Central College               812 University St, Pella , IA     52.4
miles
        Coe                           1220 First Avenue NE, Cedar Ra    22.7
miles
        Loras                         450 Alta Vista St,, Dubuque, I    85.3
miles
        Luther                        700 College Dr, Decorah, IA      102.6
```

```
miles
        MacMurray College              447 E College Ave, Jacksonvill   174.7
miles
        Nebraska Wesleyan University   5000 St Paul Ave, Lincoln, NE    246.6
miles
        North Central (IL)             30 North Brainard Street, Nape   202.0
miles
        St. Olaf                       1520 St Olaf Ave, Northfield,    189.3
miles
         Westminster (add 2017)        501 Westminster Ave, Fulton, M   206.3
miles
        Wisconsin-Platteville          1 University Plaza, Plattville   103.1
miles

Region 2, 11 schools, mean power 2.47845 (stdev 6.81742):
---------------------------------------------------------
    Centroid:  41.031 lat, -77.598 lon
    Mean distance to centroid:  217.364 miles
        Hunter                         695 Park Ave, New York, NY       190.6
miles
        Keystone College               1 College Rd, La Plume, PA       102.2
miles
        Rochester Institute of Technol Lomb Memorial Dr, Rochester, N   158.5
miles
        Scranton                       800 Linden St, Scranton, PA      103.7
miles
        Southern Virginia University   1 University Hill Dr, Buena Vi   246.6
miles
        SUNY-Oneonta                   08 Ravine Pkwy, Oneonta, NY      158.5
miles
        SUNY-Oswego                    7060 New York 104, Oswego, NY    158.5
miles
        The College of New Jersey      2000 Pennington Rd, Ewing Town   164.3
miles
        Wartburg                       100 Wartburg Blvd, Waverly, IA   773.0
miles
        Washington and Lee             204 W Washington St,, Lexingto   245.1
miles
        Wilkes                         84 W South St, Wilkes-Barre, P    90.1
miles

Region 3, 12 schools, mean power 2.78942 (stdev 3.08961):
---------------------------------------------------------
    Centroid:  43.969 lat, -90.151 lon
    Mean distance to centroid:  113.287 miles
        Augsburg                       2211 Riverside Ave, Minneapoli   168.3
miles
        Concordia (WI)                 12800 N Lake Shore Dr, Mequon,   119.8
```

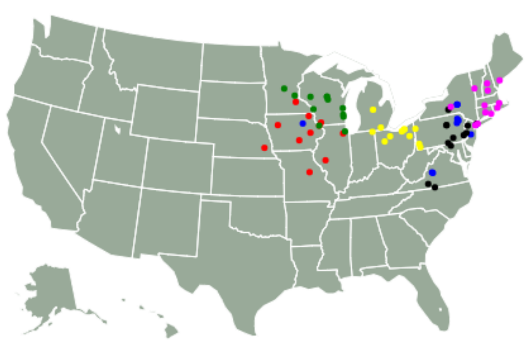
```
miles
        Dubuque                        2000 University Ave, Dubuque,    105.2
miles
        Elmhurst                       190 S Prospect Ave, Elmhurst,    181.5
miles
        Lakeland                       W3718 South Dr, Plymouth, WI     110.0
miles
        Milwaukee School of Engineerin 1025 N Broadway, Milwaukee, WI   126.0
miles
        St. Johns (MN)                 2850 Abbey Plaza, Collegeville   237.1
miles
        Wisconsin-Eau Claire           105 Garfield Ave, Eau Claire,     88.7
miles
        Wisconsin-La Crosse            1725 State St, La Crosse , WI     54.5
miles
        Wisconsin-Oshkosh              800 Algoma Blvd, Oshkosh, WI      59.9
miles
        Wisconsin-Stevens Point        100 Main St, Stevens Point, WI    48.5
miles
        Wisconsin-Whitewater           800 W Main St, Whitewater, WI     59.9
miles

Region 4, 12 schools, mean power 2.12350 (stdev 2.02969):
---------------------------------------------------------
    Centroid:  40.015 lat, -76.674 lon
    Mean distance to centroid:  116.517 miles
        Averett University (add 2017)  420 W Main St, Danville, VA      279.7
miles
        Centenary (NJ)                 400 Jefferson St, Hackettstown   113.2
miles
        College of Mount Saint Vincent 6301 Riverdale Ave, Bronx , NY   159.1
miles
        Delaware Valley                700 E Butler Ave, Doylestown,     84.2
miles
        Elizabethtown                  1 Alpha Dr, Elizabethtown, PA     10.3
miles
        Ferrum College                 215 Ferrum Mountain Rd, Ferrum   279.4
miles
        Gettysburg                     300 N Washington St, Gettysbur    32.3
miles
        Ithaca                         953 Danby Rd, Ithaca, NY         166.5
miles
        Lycoming                       700 College Pl, Williamsport,     87.1
miles
        McDaniel                       2 College Hill, Westminster, M    34.9
miles
        Pennsylvania College (add 2017 1 College Ave, Williamsport, P    86.2
miles
```

```
        Ursinus                       01 E Main St, Collegeville, PA    65.3
miles

Region 5, 12 schools, mean power 2.04617 (stdev 1.89098):
---------------------------------------------------------
    Centroid:  41.297 lat, -82.304 lon
    Mean distance to centroid:   92.856 miles
        Adrian College                10 S Madison St, Adrian, MI       99.1
miles
        Alma College                  614 W Superior St, Alma, MI      187.4
miles
        Baldwin Wallace               275 Eastland Rd, Berea, OH        23.4
miles
        Case Western Reserve          10900 Euclid Ave, Cleveland, O    38.8
miles
        Heidelberg                    310 E Market St, Tiffin, OH       47.2
miles
        John Carroll                  1 John Carroll Boulevard, Univ    42.2
miles
        Mount Union                   1972 Clark Ave, Alliance, OH      67.2
miles
        Ohio Northern                 525 S Main St, Ada, OH            87.3
miles
        Thiel                         College Ave, Greenville, PA       99.9
miles
        Trine University              1 University Ave, Angola, IN     141.8
miles
        Washington and Jefferson      60 S Lincoln St, Washington, P   132.7
miles
        Waynesburg                    51 W College S, Waynesburg, PA   147.3
miles

Region 6, 12 schools, mean power 1.87958 (stdev 2.50535):
---------------------------------------------------------
    Centroid:  42.255 lat, -72.538 lon
    Mean distance to centroid:   96.471 miles
        Bridgewater State University  131 Summer Street, Bridgewater    82.3
miles
        Castleton University          62 Alumni Dr,, Castleton, VT      98.9
miles
        Coast Guard                   31 Mohegan Ave Pkwy, New Londo    67.7
miles
        Merchant Marine               300 Steamboat Rd, Kings Point,   117.2
miles
        New England College           98 Bridge St,, Henniker, NH       73.7
miles
        Plymouth State                17 High St, Plymouth, NH         111.2
miles
```

```
        Roger Williams                 1 Old Ferry Road, Bristol, RI     76.3
miles
        Southern Maine                 96 Falmouth St, Portland, ME     153.6
miles
        Springfield                    263 Alden Street, Springfield,    10.6
miles
        Stevens Institute Of Technolog 1 Castle Point Terrace, Hoboke   129.7
miles
        SUNY-Cortland                  2 Graham Ave, Cortland, NY       187.2
miles
        Wesleyan (CT)                  45 Wyllys Ave, Middletown, CT     49.3
miles
```

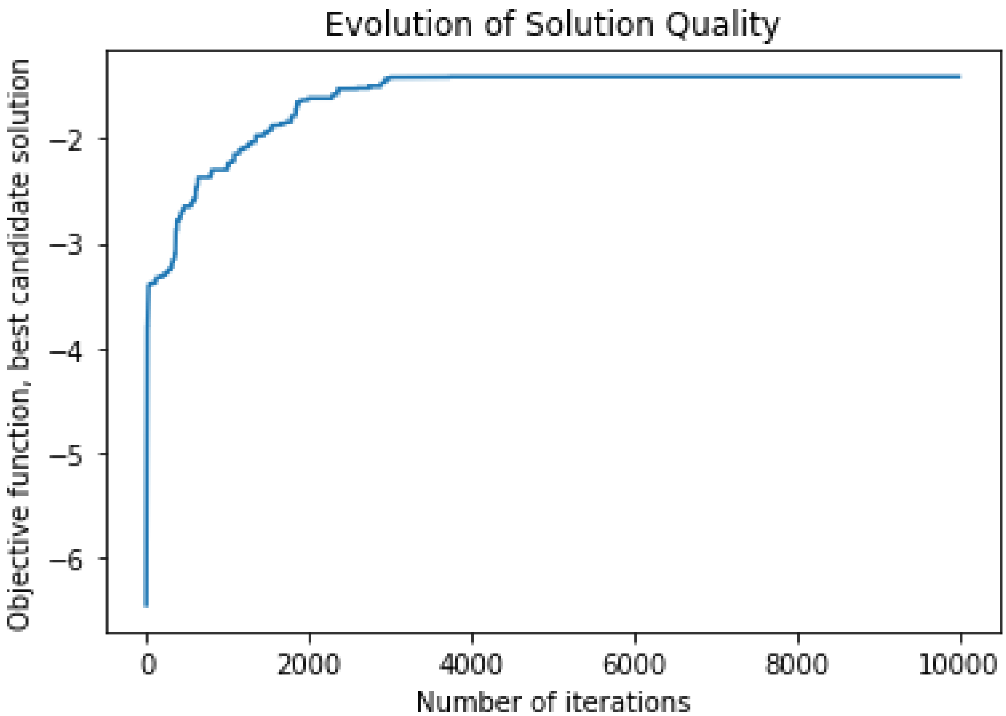
[21]:
```
partition_map( best )
```

Partition of All Schools



[22]:
```
plt.plot( range( len( fitness_curve ) ), fitness_curve )
plt.xlabel( 'Number of iterations' )
plt.ylabel( 'Objective function, best candidate solution' )
plt.title( 'Evolution of Solution Quality' )
```

[22]: Text(0.5, 1.0, 'Evolution of Solution Quality')



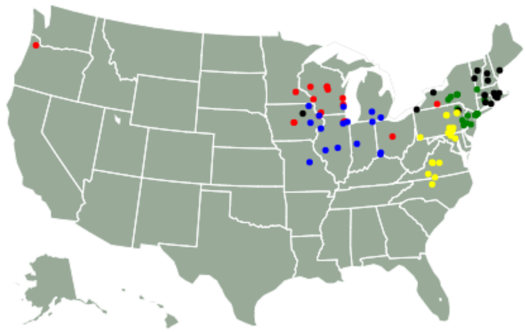
[23]:
```
objective_function( best )
```

[23]: -1.4148877257165877

1.7 Viewing Other Solutions

[24]:
```
partition = [ int( n ) - 1 if not math.isnan( n ) else 0 for n in df[SCH_ND1] ]
partition_map( partition )
```

Partition of All Schools

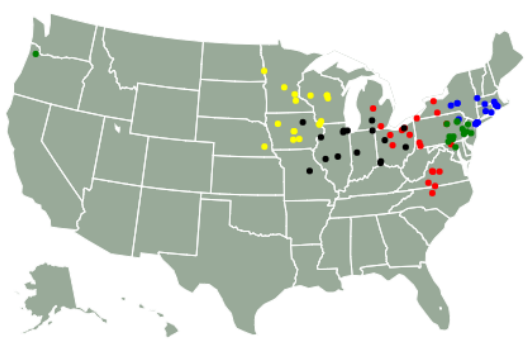


```
[25]: objective_function( partition )
```

[25]: -4.486573875023253

```
[26]: partition = [ int( n ) - 1 if not math.isnan( n ) else 0 for n in df[SCH_ND2] ]
      partition_map( partition )
```

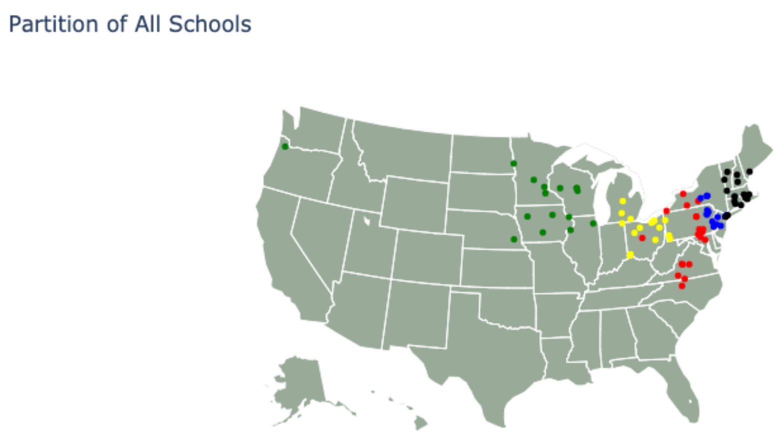
Partition of All Schools



```
[27]: objective_function( partition )
```

[27]: -4.940869315953698

[28]:
```
partition = [ int( n ) - 1 if not math.isnan( n ) else 0 for n in df[SCH_ND3] ]
partition_map( partition )
```



[29]:
```
objective_function( partition )
```

[29]: -3.3577779150908267

[ ]: