# EE 456 | Traveling Salesperson Problem

## Aishwarye Omer | ado5146@psu.edu

In [2]:
```
import numpy as np
from random import random


#    calculating penalties   #


def cal_penalty(tour_matrix_1):


    # penalty for visiting multiple cities in a day #

    P_multiple_cities = 0
    combinations = 0
    summation = 0
    for i in range(10):
        for j in range(10):
            summation = summation + tour_matrix_1[j][i]
        combinations = (summation * (summation-1)) / 2
        summation = 0
        P_multiple_cities = P_multiple_cities + combinations
    P_multiple_cities = C/2 *  P_multiple_cities


    # penalty for multiple days in a city #

    P_multiple_day = 0
    for i in range(10):
        for j in range(10):
            summation = summation + tour_matrix_1[i][j]
        combinations = summation * (summation-1) / 2
        summation = 0
        P_multiple_day = P_multiple_day + combinations
    P_multiple_day = D/2 *  P_multiple_day


    # penalty for skipping a city ROW #

    P_skipping_city = 0
    for i in range(10):
        for j in range(10):
            P_skipping_city = P_skipping_city + tour_matrix_1[i][j]
    P_skipping_city = A/2 * (abs(10 - P_skipping_city))


    # penalty for skipping a day COLUMN #

    P_skipping_day = 0
    for i in range(10):
        for j in range(10):
            P_skipping_day = P_skipping_day + tour_matrix_1[i][j]
    P_skipping_day = B/2 * (abs(10 - P_skipping_day))

    # Calculate Distance #
```

```
    dist = 0
    for x in range(10):
        for y in range(10):
            for i in range(10):
                if x!= y:
                    if i == 0:
                        dist = dist + dist_data[x][y] * tour_matrix[x][i] * (tou
                    elif i == 9:
                        dist = dist + dist_data[x][y] * tour_matrix[x][i] * (tou
                    else:
                        dist = dist + dist_data[x][y] * tour_matrix[x][i] * (tou
                                                                             + t
    P_dist = E/2 * dist

    total_penalty = P_multiple_cities + P_multiple_day + P_skipping_city + P_ski

    return total_penalty
```

In [3]:

```
 ##########################
    ###    Flip Randomly   ###
    ##########################

def flip_rand(tour_opt_1, temp_row, temp_clmn):
    if tour_opt_1[temp_row][temp_clmn] == 0:
        tour_opt_1[temp_row][temp_clmn] = 1
    else:
        tour_opt_1[temp_row][temp_clmn] = 0

    ###################################
    ### Calculate change in penalty ###
    ###################################

def cal_change_penalty(a,b):
    delta_c = a-b
    return delta_c

    ##########################
    ###    Randomness      ###
    ##########################

def coin_flip(temp, del_c, tour_matrix_1, tour_opt_1, temp_row, temp_clmn):
    A = 1/(1+np.exp(-del_c/temp))
    rand_num = random()
    if(A > rand_num):
        tour_matrix_1 = tour_opt_1
    else:
        flip_rand(tour_opt_1, temp_row, temp_clmn)
    return tour_matrix_1
```

In [4]:

```
dist_data = np.asarray([(0, 104.7, 145.2, 242.9, 275.5, 56.3, 162.6, 185.6, 87.4
                        (104.7, 0, 238.1, 335.8, 377.0, 63.8, 264.1, 278.5, 94.9,
                        (145.2, 238.1, 0, 129.2, 200.7, 194.4, 55.7, 107.5, 153.0
                        (242.9, 335.8, 129.2, 0, 129.3, 292.9, 136.2, 94.1, 297.2
                        (275.5, 377.0, 200.7, 129.3, 0, 356.4, 153.3, 182.1, 314.
                        (56.3, 63.8, 194.4, 292.9, 356.4, 0, 205.3, 235.8, 53.8,
                        (162.6, 264.1, 55.7, 136.2, 153.3, 205.3, 0, 116.4, 161.9
                        (185.6, 278.5, 107.5, 94.1, 182.1, 235.8, 116.4, 0, 256.9
                        (87.4, 94.9, 153.0, 297.2, 314.3, 53.8, 161.9, 256.9, 0,
                        (97.9, 190.4, 41.0, 169.7, 212.9, 140.6, 60.5, 147.3, 112
```

```
    #   Initializing the tour matrix   #


tour_matrix = np.empty([10,10])
tour_opt = np.empty([10,10])
for i in range(10):
    for j in range(10):
        tour_matrix[i][j] = np.random.randint(2)

tour_opt = tour_matrix
print("Initial Tour matrix")
print(tour_matrix)
```

```
Initial Tour matrix
[[1. 1. 0. 0. 0. 1. 0. 1. 0. 1.]
 [0. 0. 1. 1. 1. 1. 1. 1. 1. 0.]
 [0. 1. 0. 0. 1. 1. 1. 1. 0. 1.]
 [0. 1. 1. 0. 0. 1. 1. 1. 0. 1.]
 [1. 0. 1. 1. 1. 1. 0. 1. 0. 0.]
 [0. 1. 1. 0. 0. 1. 1. 0. 0. 1.]
 [1. 0. 1. 1. 0. 1. 0. 1. 1. 1.]
 [0. 1. 1. 1. 1. 0. 0. 1. 1. 0.]
 [0. 1. 0. 0. 1. 0. 0. 0. 0. 0.]
 [0. 0. 1. 0. 1. 1. 1. 1. 1. 0.]]
```

In [5]:

```
#   Penalty for skipping a city   #

A = 100

#            skip a day           #

B = 100

#    multiple cities in a day    #

C = 200

#    multiple days in a city     #

D = 200

#           distance             #

E = 1

#     Starting Temperature       #

T = 10000000

#             TSP                #

for i in range(0,100):
    for j in range(0,100):
        total_penalty_prev = cal_penalty(tour_matrix)
        temp_row = np.random.randint(0,10)
        temp_clmn = np.random.randint(0,10)
        flip_rand(tour_opt,temp_row,temp_clmn)
        total_penalty_new = cal_penalty(tour_opt)
        delta_penalty = cal_change_penalty(total_penalty_prev, total_penalty_new
        if(delta_penalty > 0):
```

```
            tour_matrix = coin_flip(T, delta_penalty, tour_matrix, tour_opt, tem
        else:
            flip_rand(tour_opt,temp_row,temp_clmn)

    #     Decrease the temperaure    #

    T = T * 0.9
print("Optimized Tour -")
print("Penalty = ", total_penalty_new)
print(tour_matrix)
```

```
Optimized Tour -
Penalty =  762.3999999999999
[[0. 0. 0. 0. 0. 1. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 1. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 1. 1. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 1.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 1. 0. 0. 0.]]
```