Assignment 2B - Simulated Annealing

In this assignment, you will run simulations, comparing Hill Climbing and Simulated Annealing.

TODO

Enter your information below.

Name: ...
CWID: ...

In [1]:
```
import numpy as np

import pandas as pd

from matplotlib import pylab
import matplotlib.pyplot as plt
pylab.rcParams['figure.figsize'] = (10.0, 8.0)

from sa_utils import Node
from sa_utils import hill_climbing, simulated_annealing
```

In [2]:
```
from tsp_utils import City, TSPNode, read_cities, subsample_cities, create_initial_
from tsp_utils import plot_cities, plot_path, compare_sols
```

Read Data

In [3]:
```
all_cities = read_cities('berlin52.tsp')
TSPNode._cities = all_cities
```

Simulations

You will run simulations for multiple subsets and multiple initializations, like you have done in Assignment 2a. One major difference here is that, in addition to hill climbing (HC), you will also run simulations using simulated annealing (SA).

Important: You need to pick the temperature schedule and the respective parameters to achieve the best results you can get. The temperature schedule and its parameters depend on the domain, as well as the range of the optimization value (in this case, the -1*cost of the travel plan).

The best parameters might also depend on the size of the problem. For example, the best parameters might be different for 10 cities versus 20 cities.

In [4]:
```
# Temperature Schedules for SA

# Here, you are given one temperature schedule.
# You are free to add other schedules, or experiment with only this.

## Exponential multiplicative schedule

def exp_schedule(T_0, l):
    """Returns a function whose input is iter."""
    return lambda iter: T_0 * np.exp(-l * iter)
```

Example call to simulated annealing:

`_, sa_sol_n = simulated_annealing(initial_n, temp_schedule=exp_schedule(T_0=20, l=0.2), max_iter=100)`

Notes:

1. `simulated_annealing` returns the final state and the best state. Given large enough temperature and enough iterations, these should be the same states, but they might not be. We will work with the best state, which is saved in `sa_sol_n` in this call.

2. These parameter settings are just examples and not the best settings :)

Simulations 1 - Subsample of 10

1. Create multiple subsamples of cities, of size 10 each.
2. Create multiple initializations for each.
3. Run HC for each.
4. Run SA for each. Try different temperature schedules and parameters. Do NOT share everthing you tried; show us only the setting that achieved the best performance (on average).
5. Present the initial, the HC, and SA results as a table. Your table should include a final row that includes the average values for Initial, HC, and SA.

Use

- subsample_size = 10
- subsample_seeds of `[0, 1, 2, 3, 4]`
- initial_seeds = `[11, 12, 13, 14, int(last_three_digits_of_your_CWID)]`

In [5]:
```
# TODO - Write code and run the simulation

CWID='A12345678'
```

```python
subsample_size = 10
subsample_seeds = range(0, 5)
initial_seeds = [11, 12, 13, 14, 15, int(CWID[6:])]

#Write code
```

Results

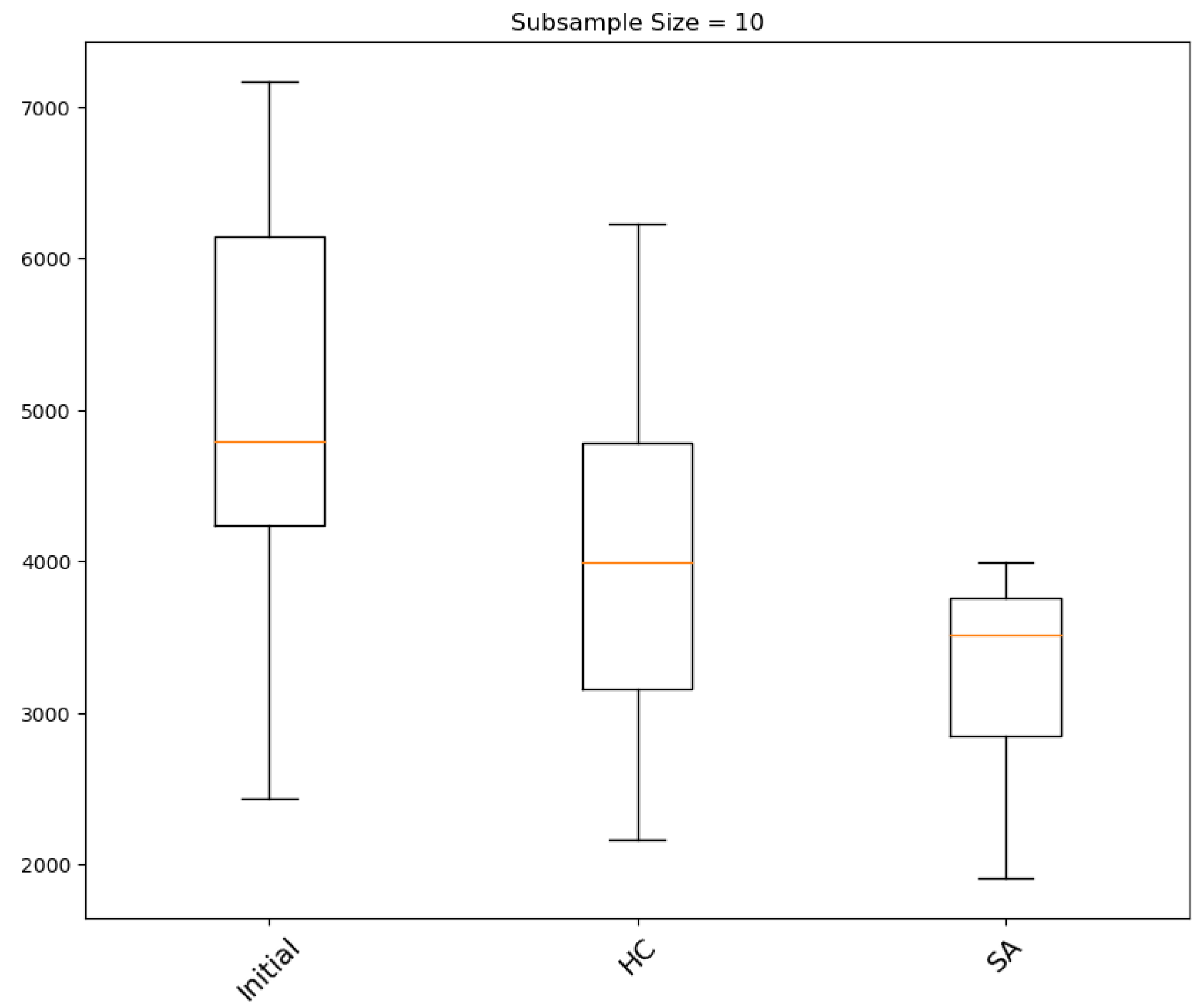
In [6]:
```python
# TODO: Present a table of the results.
# Note that the entries are costs (-1*value).
# Note that the last row are the mean values.
# Write code
```

Out[6]:

Trial	SubSeed	InitSeed	Initial	HC	SA
1	0	11	6910.9	4845.3	3726.5
2	0	12	4805.4	4578.8	3857.9
3	0	13	6281.9	6230.3	3825.8
4	0	14	6547.4	4943.6	3646.3
5	0	15	6567.3	3994.0	3845.8
6	0	678	6190.6	4844.2	3825.8
7	1	11	3897.9	2195.3	1910.7
8	1	12	3851.6	3140.5	1910.7
9	1	13	3343.3	2746.0	1910.7
10	1	14	2431.8	2160.0	1910.7
11	1	15	2541.9	2390.7	1910.7
12	1	678	4140.2	2790.4	1910.7
13	2	11	4544.4	4089.5	2850.5
14	2	12	4312.8	3211.7	2850.5
15	2	13	4733.3	3118.2	2850.5
16	2	14	3813.3	3334.9	2850.5
17	2	15	4394.0	2850.5	2850.5
18	2	678	4209.7	3197.8	2850.5
19	3	11	4800.0	3990.1	3517.6
20	3	12	5107.3	3764.1	3517.6
21	3	13	4792.2	3898.4	3517.6
22	3	14	4842.8	4342.7	3517.6
23	3	15	4777.7	4060.7	3517.6
24	3	678	5073.1	4475.0	3517.6
25	4	11	6451.0	4993.8	3763.0
26	4	12	7169.9	5886.3	3763.0
27	4	13	6146.6	4957.4	3716.9
28	4	14	4638.0	3997.5	3997.5
29	4	15	6147.9	6078.8	3796.9
30	4	678	5981.5	3960.5	3869.3
Mean	-	-	4981.5	3968.9	3176.9

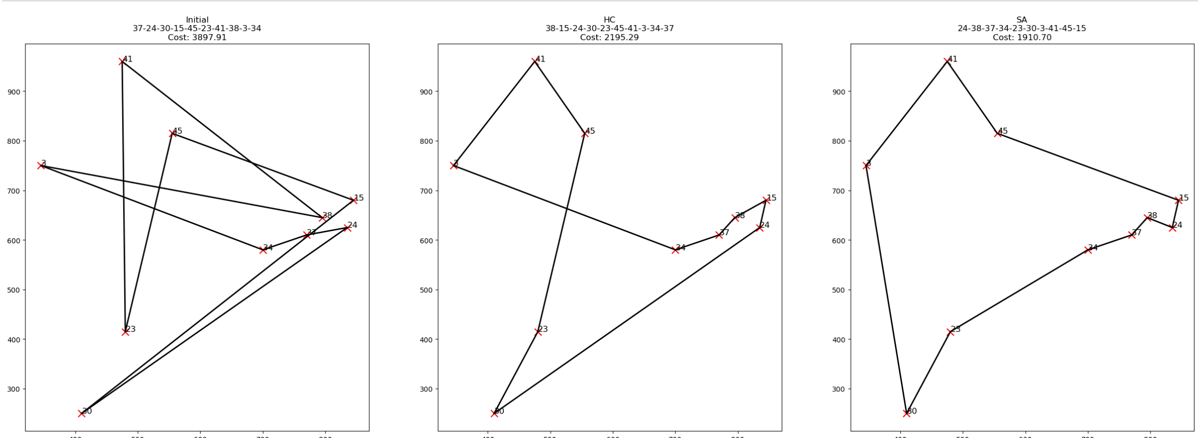
In [7]:

```
# TODO: Plot a boxplot of the cost values for Initial, HC, and SA.
# Write code
```



In [8]:
```
# TODO: Pick subsample seed and initial seed, and visualize Initialization, HC, and
# Use the compare_sols function.

compare_sols((("Initial", ...), ("HC", ...) ("SA", ...)), all_cities)
```



SA Parameter Settings for Subsample 10

The best parameter settings I found for subsample 10 are as follows.

TODO

EDIT THIS. For example `temp_schedule=exp_schedule(T_0=20, l=0.1), max_iter=100`

Simulations 20, 30, 40, 52

Repeat the above simulations, tables, and plots for

- subsample of 20
- subsample of 30
- subsample of 40
- Full data (no subsampling for full data, only random initialization).

You need to choose the best parameter settings and report them.

Simulations - Subsample 20

In [9]:
```
# Run simulations using different temperature settings and pick the best setting.
```

Results

Table, boxplot, and path visualizations.

In [ ]:
```

```

SA Parameter Settings for Subsample 20

The best parameter settings I found for subsample 20 are as follows.

TODO

EDIT THIS. For example `temp_schedule=exp_schedule(T_0=20, l=0.1), max_iter=100`

Simulations - Subsample 30

Simulations - Subsample 40

Simulations - Full Data

Note: for the full data, you do not perform subsampling. Run only the random initializations.

Report

Discuss your findings and your experience with these simulations. This part of the question is worth 5 points.

TODO

EDIT THIS.

In [ ]:
```

```