Fiber Optic Network Topology Optimization

Under the Guidance of: Prof. Abhijit Gogulapati

Course: AE 775 *Optimization of Engineering Design*

Section A: Verification and Benchmarking

Verification and benchmarking each algorithm(number them)

Approach 1: Comparing against a standard library implementation

1. Benchmarking Algorithm 1 Details:
   Language: Julia
   Library: JuMP
   Solver: GLPK Solver (GNU Linear-Programming Kit)
   Method: Interior point method.

2. Algorithm 2: (for testcase2, testcase3, testcase4)
   Language: Python
   Library: Ortools
   Solver: Pywraplp.Solver()

Approach 2: Taking a problem whose optima is known (for testcase1)

Reference link: http://www.math.wsu.edu/faculty/genz/364/lessons/l603.pdf (Page no. 6)

Standard or Known Test Cases:

Test Case 1:

Maximise $5x_1 + 6x_2$, with integer $x_1, x_2 \geq 0$

$$\text{subject to} \quad 10x_1 + 3x_2 \leq 52$$
$$2x_1 + 3x_2 \leq 18$$

[x1,x2] = [3,4]
opt_value = 39

Test Case 2:

$$\begin{aligned} \textit{Maximize} \quad & 8x_1 + 11x_2 + 6x_3 + 4x_4 \\ \textit{subject to} \quad & 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14 \\ & x_j \in \{0, 1\} \quad j = 1, \ldots, 4 \end{aligned}$$

[x1,x2,x3,x4] = [0,1,1,1]
opt_value = 21

Test Case 3:

```
Maximize
 obj: 4 X1 + 7 X2 + 6 X3 + 5 X4 + 4 X5
Subject To
 c1: 5 X1 + 8 X2 + 3 X3 + 2 X4 + 7 X5 <= 112
 c2: X1 + 8 X2 + 6 X3 + 5 X4 + 4 X5 <= 109
```

[x1,x2,x3,x4,x5] = [14,0,0,19,0]
opt_value = 151

Test Case 4:

$$
\begin{array}{rl}
\text{Minimize} & x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11} \\
\text{subject to} & x_1 + x_2 + x_3 + x_4 \geq 1 \\
& x_1 + x_2 + x_3 + x_5 \geq 1 \\
& x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \geq 1 \\
& x_1 + x_3 + x_4 + x_6 + x_7 \geq 1 \\
& x_2 + x_3 + x_5 + x_6 + x_8 + x_9 \geq 1 \\
& x_3 + x_4 + x_5 + x_6 + x_7 + x_8 \geq 1 \\
& x_4 + x_6 + x_7 + x_8 \geq 1 \\
& x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} \geq 1 \\
& x_5 + x_8 + x_9 + x_{10} + x_{11} \geq 1 \\
& x_8 + x_9 + x_{10} + x_{11} \geq 1 \\
& x_9 + x_{10} + x_{11} \geq 1 \\
& x_j \in \{0, 1\} \quad j = 1, \ldots, 11
\end{array}
$$

```
Number of variables = 11
Number of constraints = 11
Solution:
Objective value = 3.0
x1 = -0.0
x2 = -0.0
x3 = 0.0
x4 = 1.0
x5 = 1.0
x6 = 0.0
x7 = -0.0
x8 = -0.0
x9 = 1.0
x10 = -0.0
x11 = 0.0
```

Branch and Bound Algorithm:

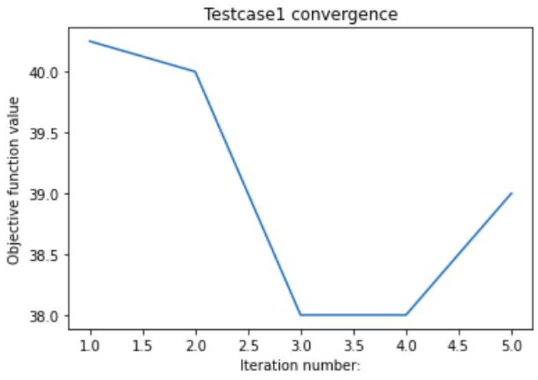
1. Values for accuracy, (actual results vs obtained results)

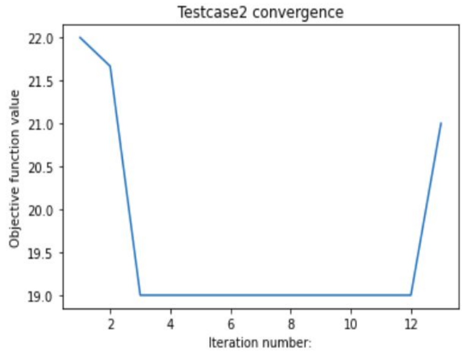
Test case	Optima point	Actual Solution	Obtained Optima point	Obtained solution	Accuracy
1	[3,4]	39	[3,4]	38.99999939340455	99.9999%
2	[0,1,1,1]	21	[0,1,1,1]	21.0	100%
3	[14,0,0,19,0] [15,0,1,17,0]	151	[15,0,1,17,0]	150.9999999507497	99.9999%
4	[1,0,0,0,0,0,0,1,0,0,1] [0,1,0,1,0,0,0,0,0,1,0]	3	[1,0,0,0,0,0,0,1,0,0,1]	3.000000000082103	99.9999%

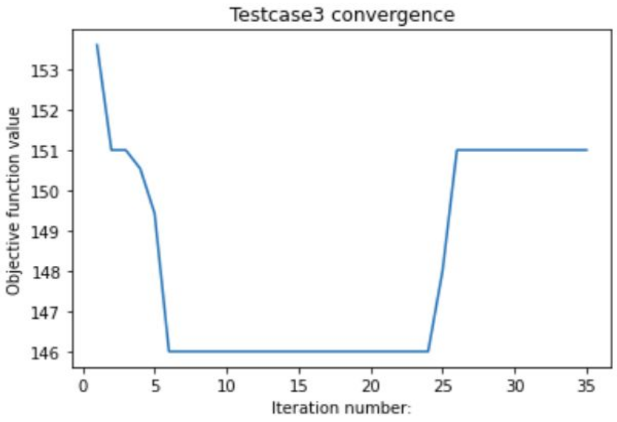
Branch and Bound Algorithm + linprog : Computational expense and Execution Time

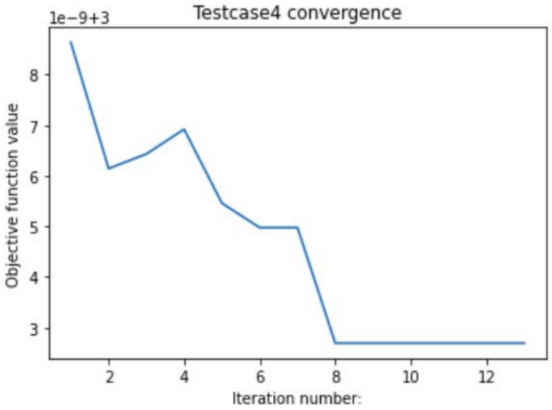
Test case	Number of design variables	Number of constraints	Problem type	Execution time
1	2	4	IP	.000268 s
2	4	9	Binary	.051096 s
3	5	7	IP	.21007 s
4.	11	33	Binary	.06226 s

Branch and Bound









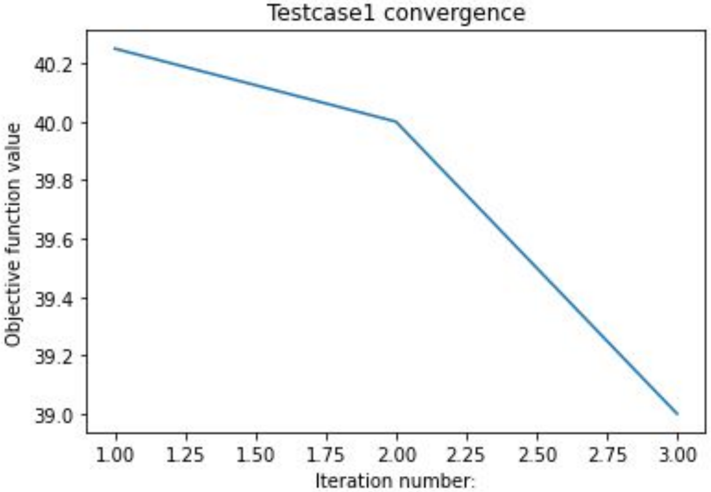
Cutting plane Algorithm:

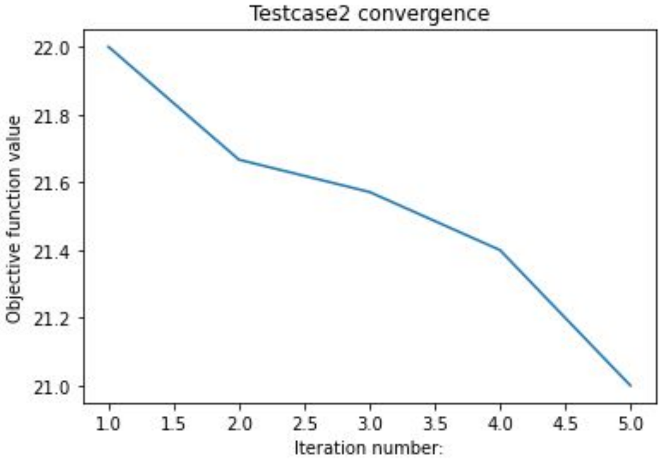
1. Values for accuracy, (actual results vs obtained results)

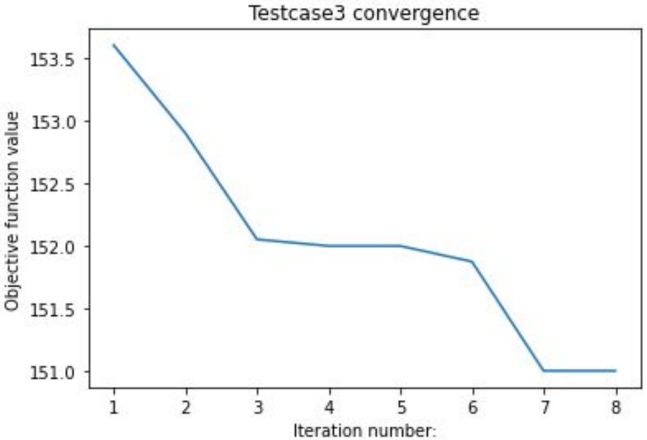
Test Case	Actual Optima point	Actual Solution	Obtained Optima point	Obtained Solution	Accuracy
1	[3,4]	39	[3,4]	39.0	100%
2	[0,1,1,1]	21	[0,1,1,1]	21.0	100%
3	[14,0,0,19,0] [15,0,1,17,0]	151	[15,0,1,17,0]	151.0000000000006	99.9999%
4	[1,0,0,0,0,0,0,1,0,0,1] [0,1,0,1,0,0,0,0,0,1,0]	3	[0,1,0,1,0,0,0,0,0,1,0]	3.0	100%

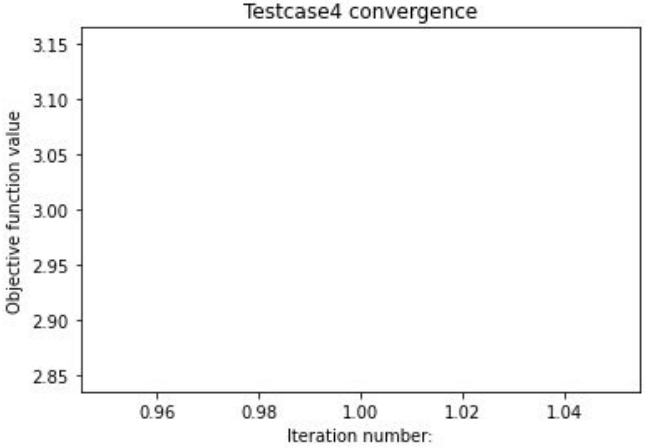
Cutting Plane Algorithm: Computational expense and Execution Time

Test case	Number of design variables	Number of constraints	Problem type	Execution time
1	2	4	IP	4.56 ms
2	4	9	Binary	6.43 ms
3	5	7	IP	10.53 ms
4.	11	33	Binary	4.02 ms


Testcase1 convergence
Objective function value
40.2
40.0
39.8
39.6
39.4
39.2
39.0
1.00
1.25
1.50
1.75
2.00
2.25
2.50
2.75
3.00
Iteration number:


Testcase2 convergence
Objective function value
22.0
21.8
21.6
21.4
21.2
21.0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Iteration number:


Testcase3 convergence
Objective function value
153.5
153.0
152.5
152.0
151.5
151.0
1
2
3
4
5
6
7
8
Iteration number:


Testcase4 convergence
Objective function value
3.15
3.10
3.05
3.00
2.95
2.90
2.85
0.96
0.98
1.00
1.02
1.04
Iteration number:

1. Accuracy:

   The accuracy for cutting plane algorithm is very high(almost 100%), and the slight differences correspond the floating point precision rules.

2. Convergence Rate:

   As the number of iterations for each test case in cutting plane algorithm is quite less, it is hard to come to a conclusion, however, as we can see in the plots below, as the objective value converges to optimum integer objective value, there is a steep decrease towards the optima.

Genetic algorithm: Verifying the accuracy of the algorithm

1. Values obtained for test cases

Test case	Optima point	Actual Optima	Obtained Optima point	**Obtained optima**	**Accuracy**
1	[3,4]	39	[3,4]	39	100%
2	[0,1,1,1]	21	[0,1,1,1]	21,19	99%
3	[14,0,0,19,0] [15,0,1,17,0]	151	[14,0,0,19,0] [15,0,1,17,0]	151,147, 148	96.29%
4.	[1,0,0,0,0,0,0,1,0,0,1] [0,1,0,1,0,0,0,0,0,1,0]	3	[1,0,0,0,0,0,0,1,0,0,1]	3	100%

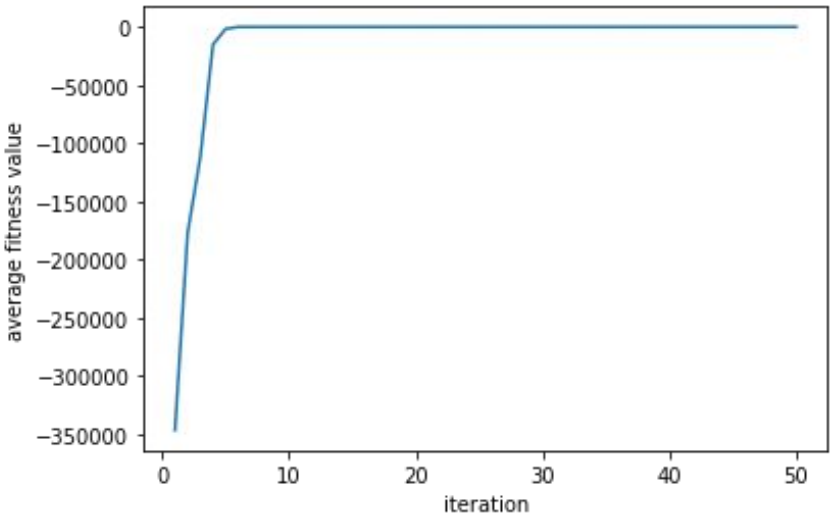
Accuracy calculated as ((average obtained value on 10 iterations)/optimal_value)*100%

GA: Computational Expense and Execution Time

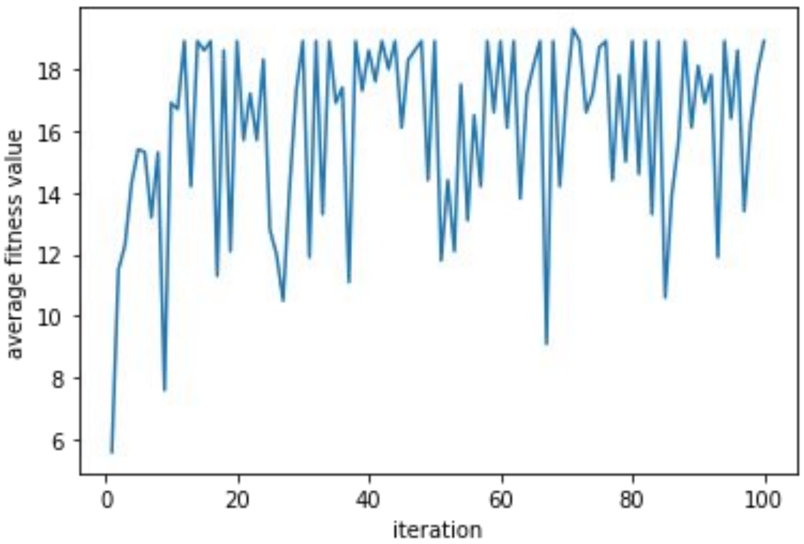
Test case	Number of design variables	Number of constraints	Problem type	Execution time
1	2	4	IP	0.110s
2	4	9	Binary	0.133s
3	5	7	IP	4.920s
4.	11	33	Binary	0.379s

Convergence plots: Average fitness vs iteration

Test case one



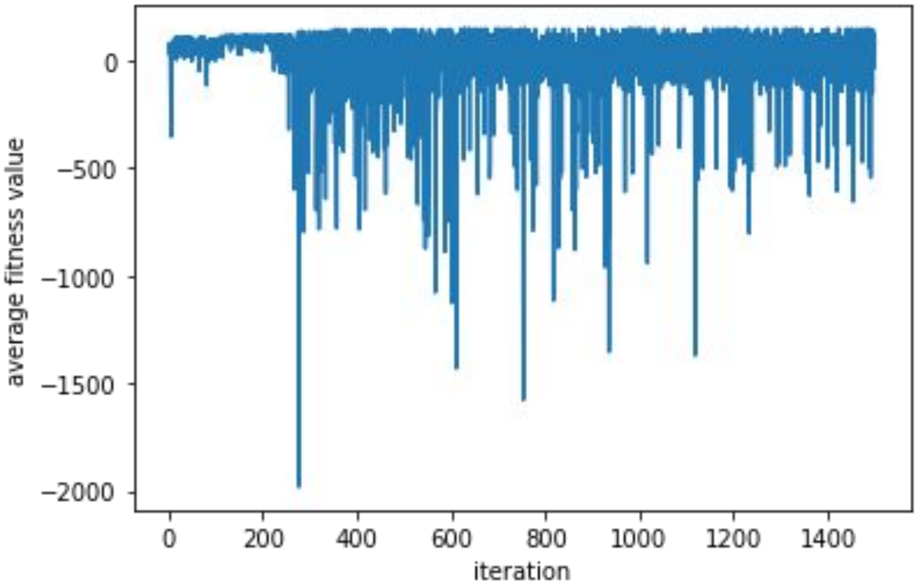
Test case two



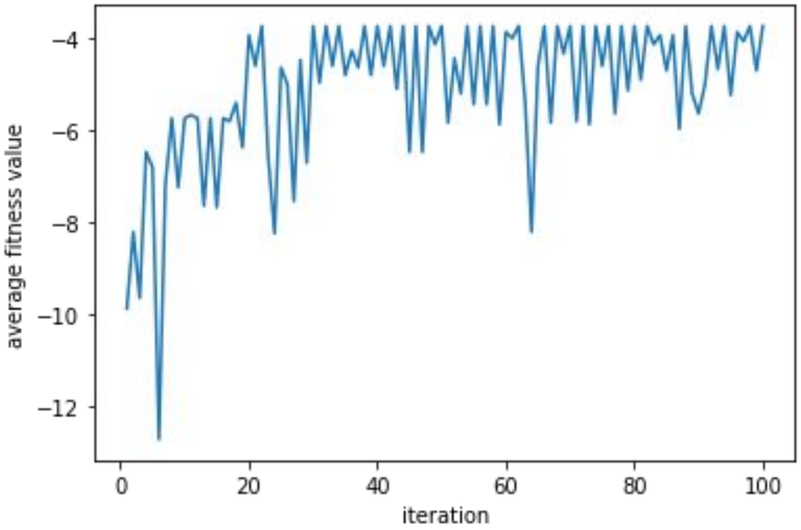
The values might fluctuate since average is sensitive to outliers, and there might be (a few) non-feasible solutions in a generation

Convergence plots: Average fitness vs iteration

Test case 3



Test case 4

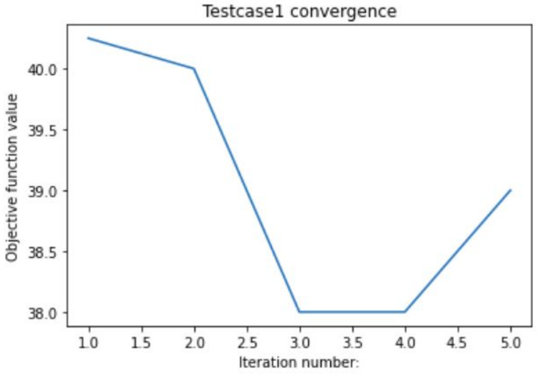


Section B: Comparative analysis

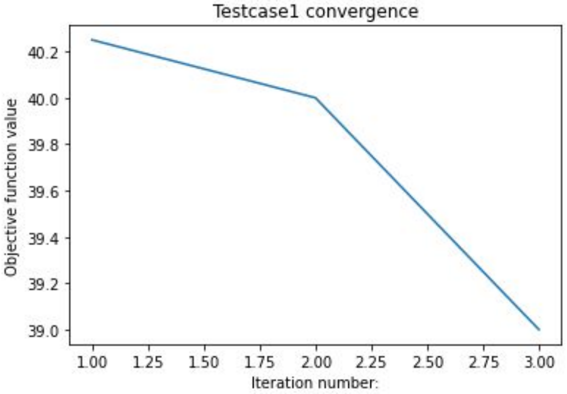
Comparative Accuracy analysis

Test Case 1	**Genetic Algorithm**	Cutting Plane	Branch and Bound
1	100%	100%	99.9999%
2	99%	100%	100%
3	96.29%	99.9999%	99.9999%
4	100%	100%	99.9999%

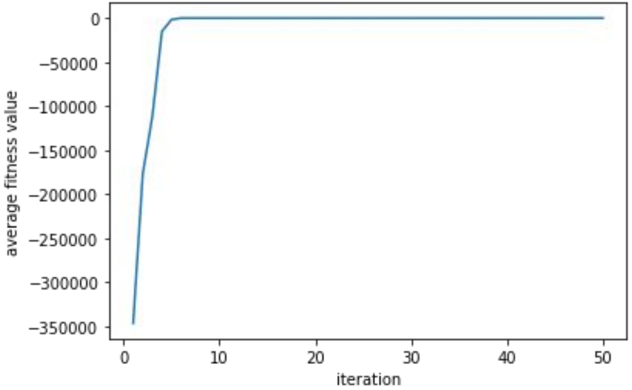
Comparing the convergence: Test case 1



Branch and Bound

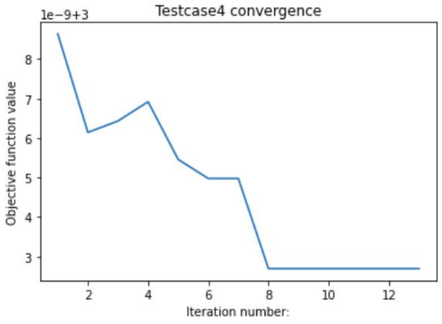


Cutting planes

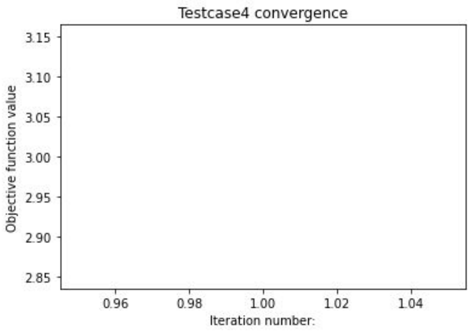


Genetic Algorithm

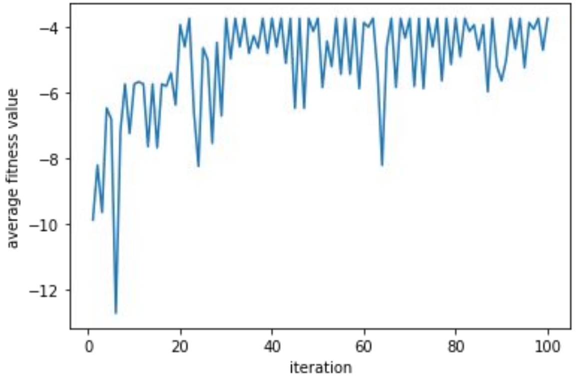
Comparing the convergence: Test case 4



Branch and Bound

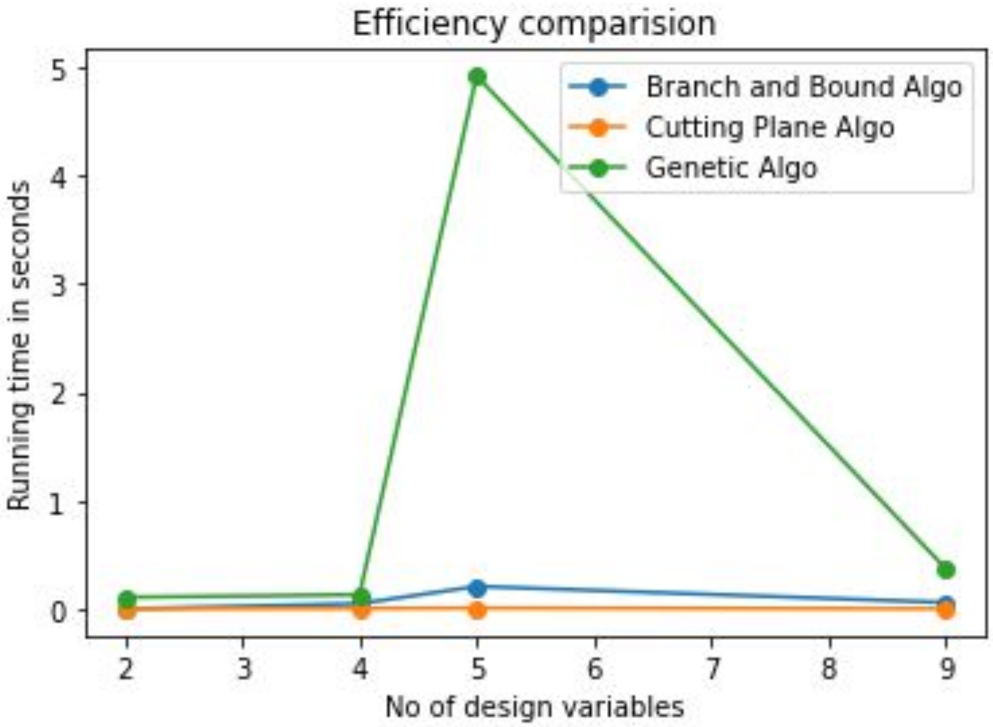


Cutting planes



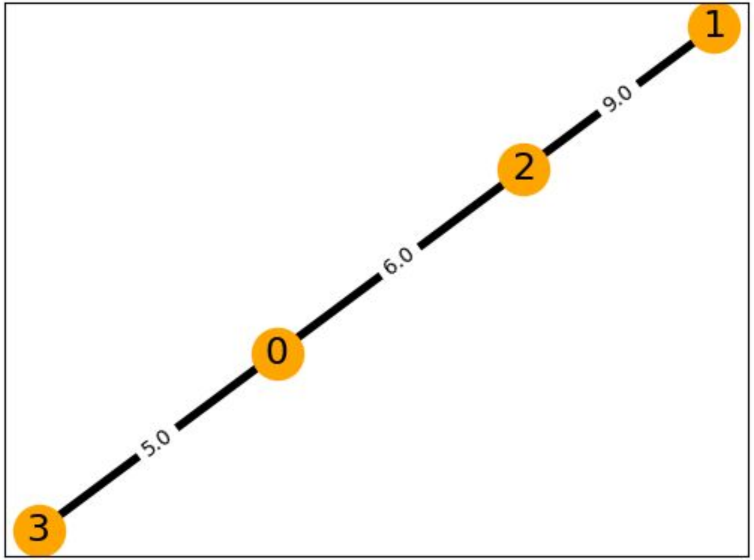
Genetic Algorithm

Plot of time taken by different methods for all test cases



Comparison Plots of Optimal value , computational time for Network Topology Problems

4 Cities Network Topology Problem:




```
weights = np.array([[ 0., 16.,  3.,  5.],
                    [16.,  0.,  9., 10.],
                    [ 3.,  9.,  0.,  3.],
                    [ 5., 10.,  3.,  0.]])
```

Objective function value: 20

Running time for:

Cutting Plane Algorithm:	2.672910690307617 ms
Branch and Bound Algorithm:	8.25047492980957 ms

Failure of G.A in the current implementation

Genetic Algorithm: Needs extremely large number of iterations.

1. The GA at its current form is not properly optimized for very large subspace such as the current problem.
2. Besides, for the current problem the space of feasible solutions are extremely small compared to the search space due to the way the problem is defined.

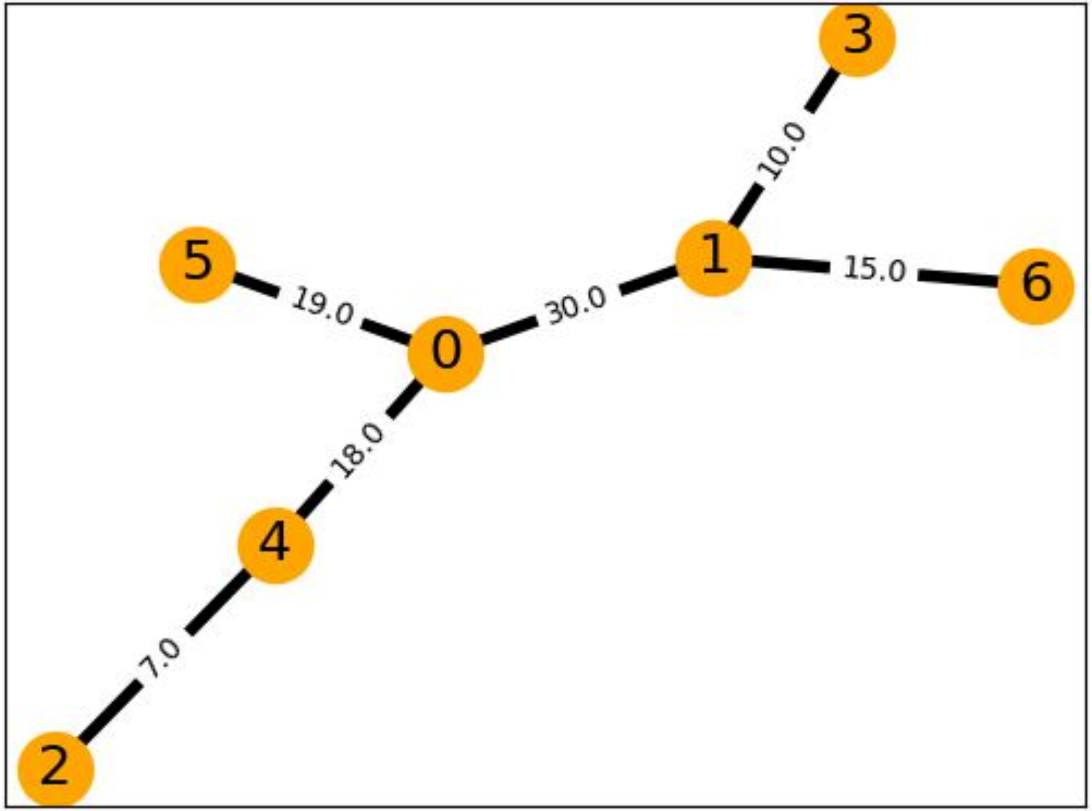
7 Cities Network Topology Problem:

```
weights = np.array([[ 0., 10., 34., 28.,  9., 19., 27.],
                    [10.,  0., 23., 10., 32., 20., 15.],
                    [34., 23.,  0., 22.,  7., 18.,  9.],
                    [28., 10., 22.,  0., 25., 31.,  8.],
                    [ 9., 32.,  7., 25.,  0., 10., 27.],
                    [19., 20., 18., 31., 10.,  0., 25.],
                    [27., 15.,  9.,  8., 27., 25.,  0.]])
```

Running time for:

Cutting Plane Algorithm: 13.128042221069336 ms

Branch and Bound Algorithm: 190.23886871337890 6 ms



Objective function value : 99

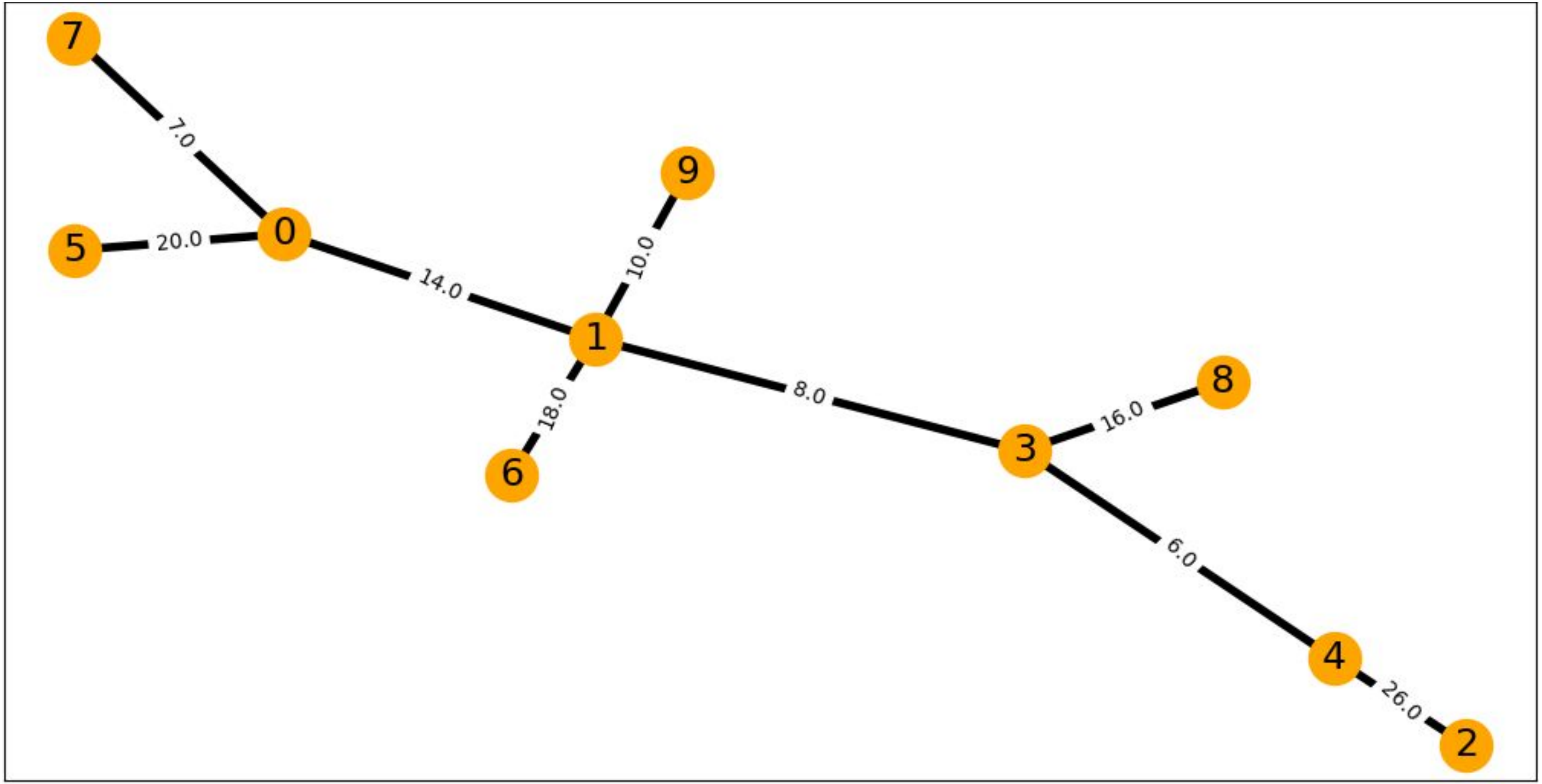
10 Cities Network Topology Problem:

```
weights = np.array([[ 0.  , 2., 34., 12., 48., 20., 40.  , 7., 46., 33.],
                    [ 2.,  0., 42.,  2., 41., 31., 18., 50., 26., 10.],
                    [34., 42.,  0., 48., 26., 18., 15., 44., 20., 32.],
                    [12.,  2. ,48.,  0.,  3., 32., 31., 44., 16., 28.],
                    [48., 41., 26.,  3.,  0., 28., 26., 23., 42., 33.],
                    [20., 31., 18., 32., 28.,  0., 22., 26., 45., 40.],
                    [40., 18., 15., 31., 26., 22.,  0., 27.,  1., 46.],
                    [ 7., 50., 44., 44., 23., 26., 27.,  0., 20., 23.],
                    [46., 26., 20., 16., 42., 45.,  1., 20.,  0., 16.],
                    [33., 10., 32., 28., 33., 40., 46., 23., 16.,  0.]])
```

Running time for:

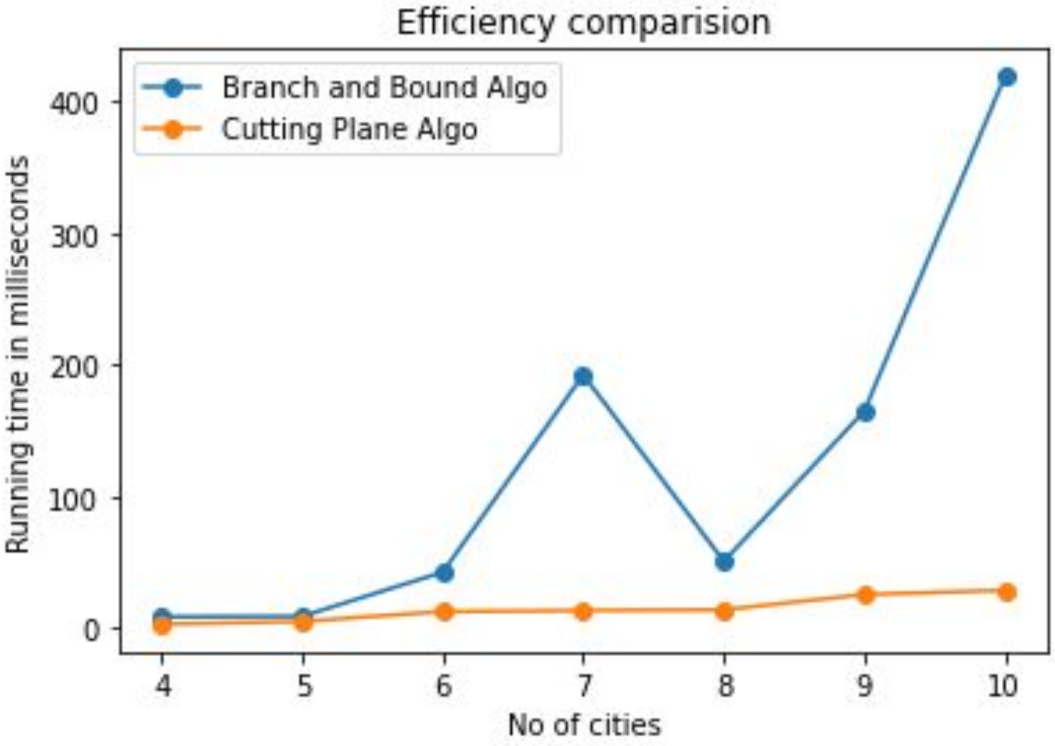
Cutting Plane Algorithm: 28.40900421142578 ms

Branch and Bound Algorithm: 219.3077087402344 ms



Objective function value : 125.0

Plot of time taken by different methods for num_cities = 4,5,6,7,8,9,10



Section C: Learning and recommendations

Observations while running various algorithms:

1. For all of the test cases run the GA since it is a probabilistic algorithm in certain runs (out of total number of runs) converges to a different local optima rather than going to global optima.

2. For Cutting planes and Branch and bound for the given test cases they are brute search algorithms and often do endup searching the entire search space for optima thus for the smaller problems in the current cases they always give the same optima.

3. There is a possibility of having more than one optimal integer solutions to an optimization problem, and different algorithms end up at different solutions based on their functionality. However, the objective function values are same for all optimal solutions.

Main takeaways from the Project:

- Algorithms may seem easy to implement based on pseudo codes but it is difficult to get them work completely well and efficiently.

- Non-gradient based methods are computationally less expensive per iteration but end up searching large space and often converge slowly to the global optimum