ECE 6133
Physical Design Automation
Spring 2019

ILP Based Floorplanning

Soundarya Bhagi
Vydeeswaran Kannan

Project Overview

- Integer Linear Programming Based Floorplanning
- Problem Formulation
- Algorithm and Implementation
- Results
- Extension
- Conclusion

Integer Linear Programming Based Floorplanning

- Integer Linear Programming algorithm for floorplanning has been implemented to handle soft and hard modules under area minimization constraint.
- ILP algorithm has been implemented using Python language.
- lp_solve, a free mixed integer linear programming solver by SourceForge.net, has been used to solve the linear constraints.
- Python's GUI Tkinter is used to demonstrate the final floorplan.

Problem Formulation

- **Objective:**

To find the optimal dimensions of flexible blocks and rotation of fixed blocks and their locations on a chip such that the total area is minimized and none of the blocks are overlapped.

- **Input:**

The dimensions of a set of fixed blocks and the area and aspect ratio of the flexible blocks are provided by the user as a *.ilp* file.

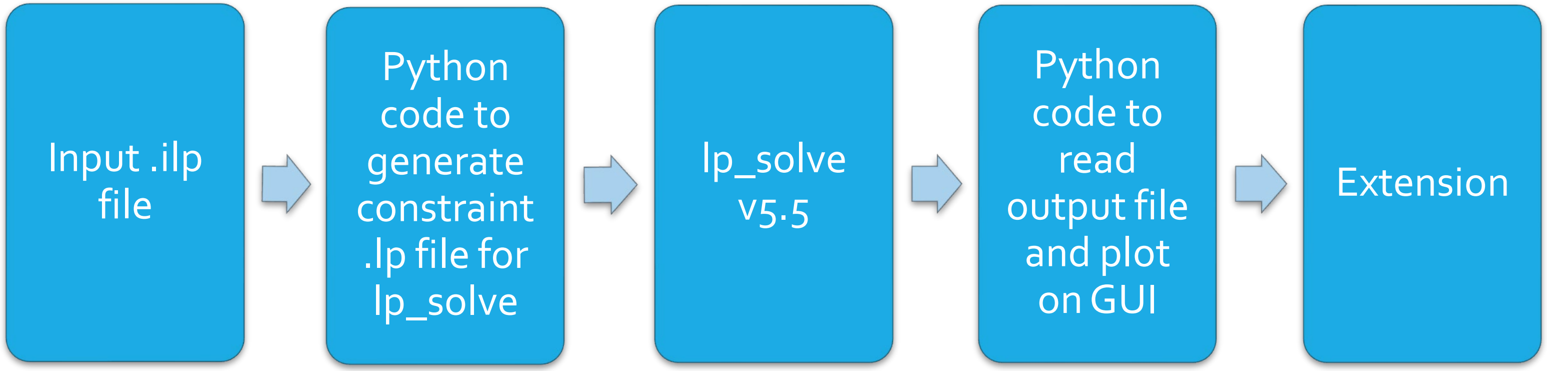
- **To Do:**

Parse the input file and generate the input constraints which are fed to a linear programming solver. The generated output file with values satisfying the input constraints are used to generate the final floorplan layout.

- **Output:**

Generate the final floorplan layout on a GUI.

Algorithm and Implementation



Input .ilp file

Python code to generate constraint .lp file for lp_solve

lp_solve v5.5

Python code to read output file and plot on GUI

Extension

Results

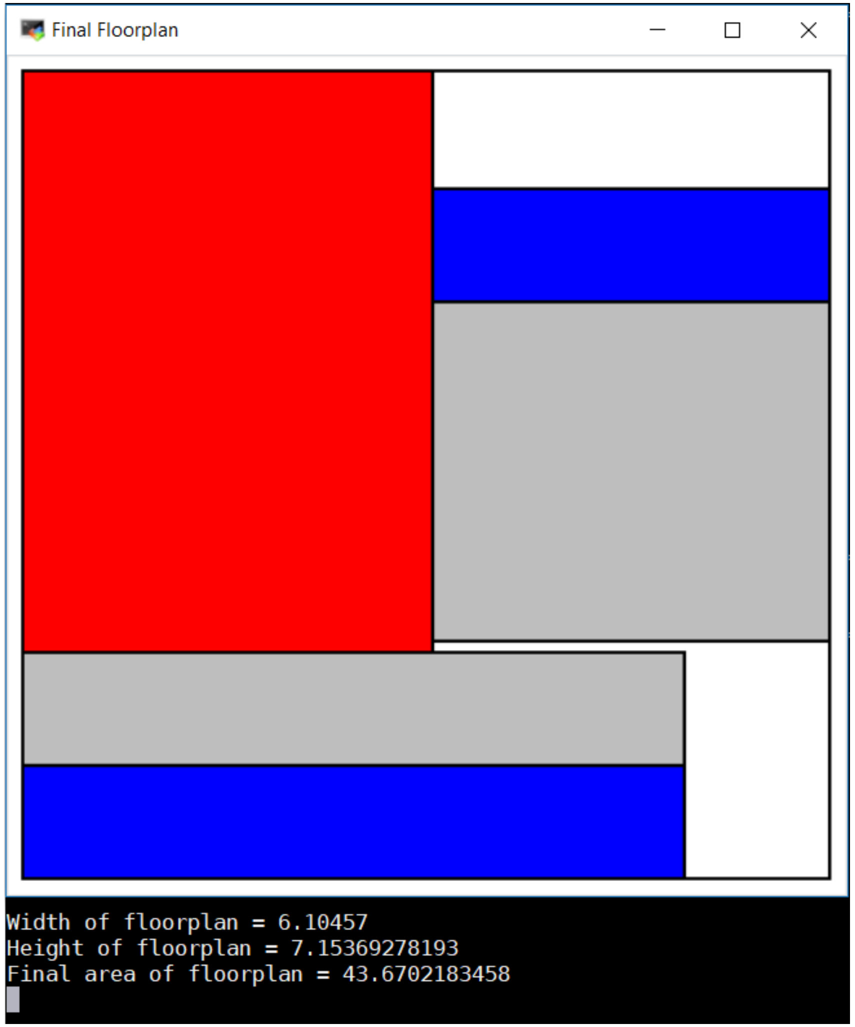
- We ran our code for 5_block.ilp, 10_block.ilp, 30_block.ilp, 50_block.ilp and 100_block.ilp.
- The output GUI is shown in the following slides and the extension where whitespace and overlap is removed, is shown beside it.
- Run time is the time we allowed the lp_solve to run. The higher the run time, better the result. Thus, the noted run times and final floorplan are not the optimal values since we can get better values if we let the lp_solve run for longer time.
- We do not report the percentage whitespace for the above noted reason.
- Hard non-rotated modules are represented by grey color.
- Hard rotated modules are represented by blue color.
- Soft modules are represented by red color.

Extension

- We extended our project by running ILP algorithm again on the floorplan but by setting all the modules to be rigid and fixing their relative positions. Thus only the location of each module is calculated by lp_solve.
- The area improvement is not much, but it would be significant if the initial floorplan solution is optimal.

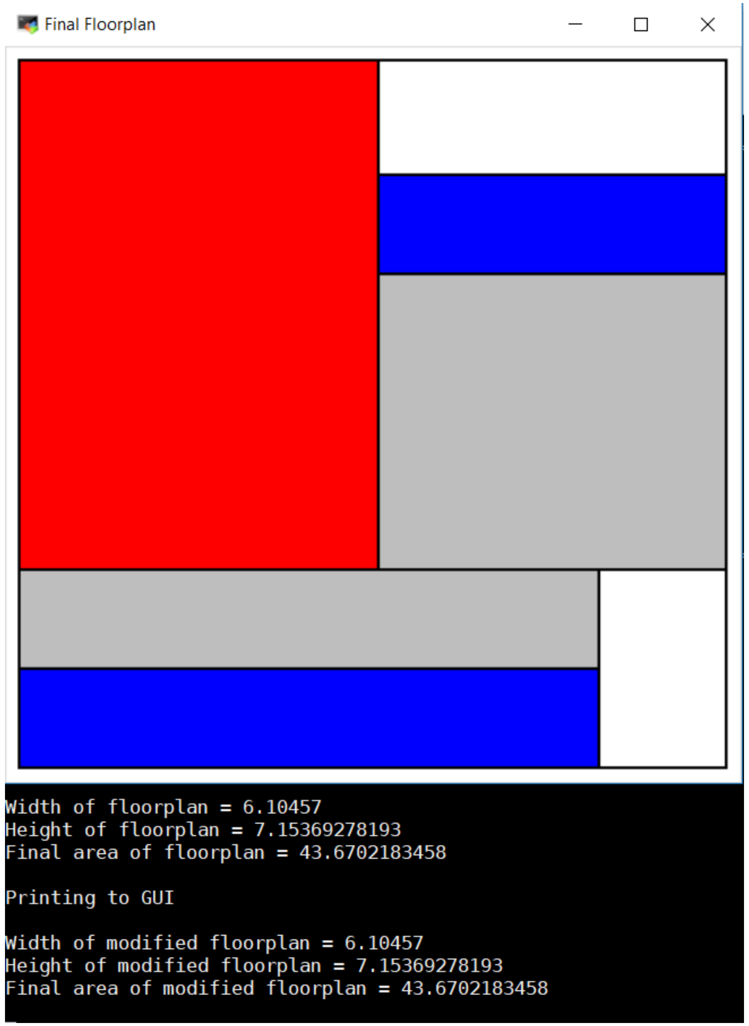
5 Blocks – Underestimation with runtime 0.86s

With overlap



```
Width of floorplan = 6.10457
Height of floorplan = 7.15369278193
Final area of floorplan = 43.6702183458
```

After Extension



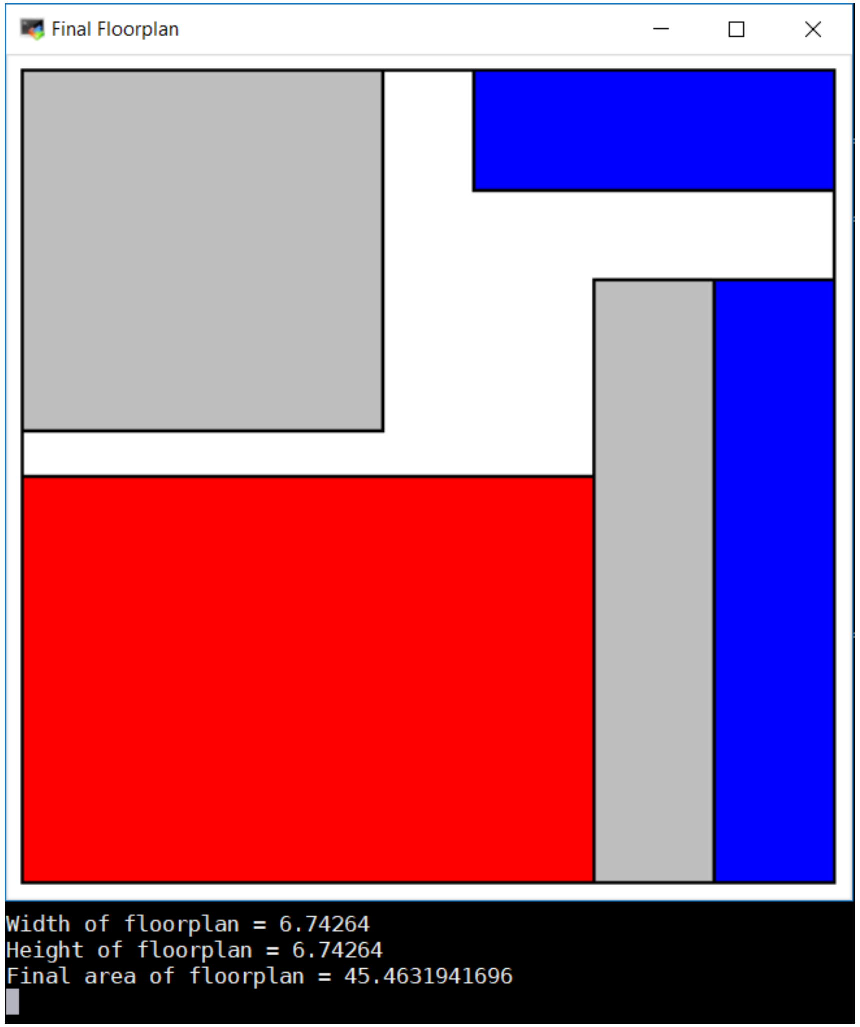
```
Width of floorplan = 6.10457
Height of floorplan = 7.15369278193
Final area of floorplan = 43.6702183458

Printing to GUI

Width of modified floorplan = 6.10457
Height of modified floorplan = 7.15369278193
Final area of modified floorplan = 43.6702183458
```

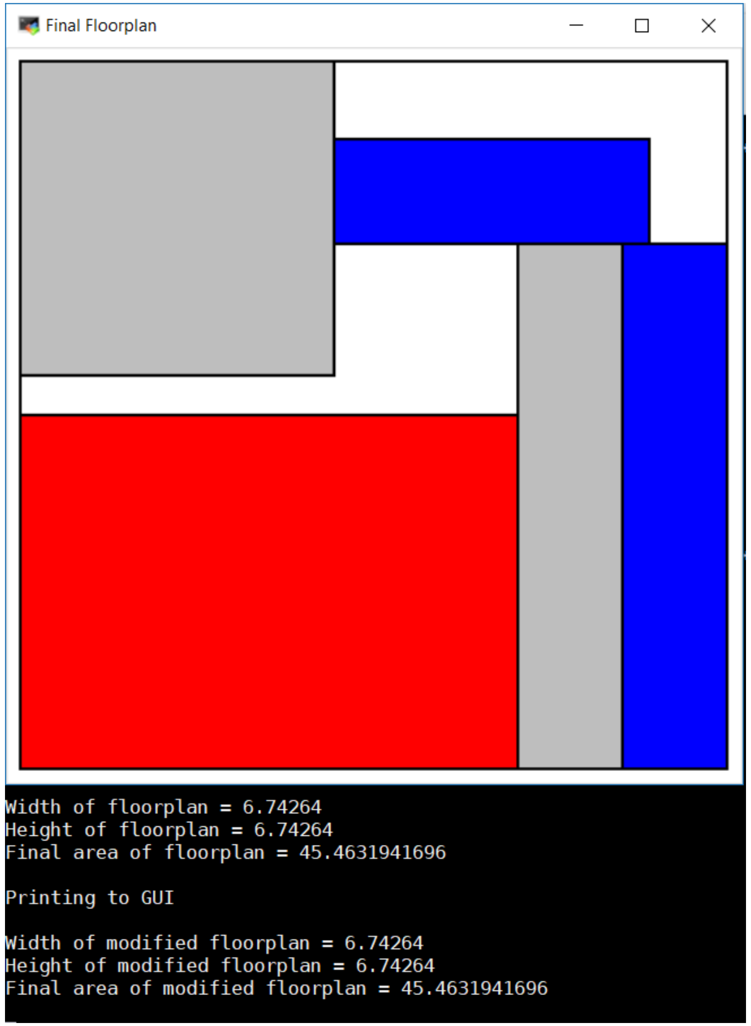
5 Blocks – Overestimation with runtime 0.86s

With whitespace



```
Width of floorplan = 6.74264
Height of floorplan = 6.74264
Final area of floorplan = 45.4631941696
```

After Extension



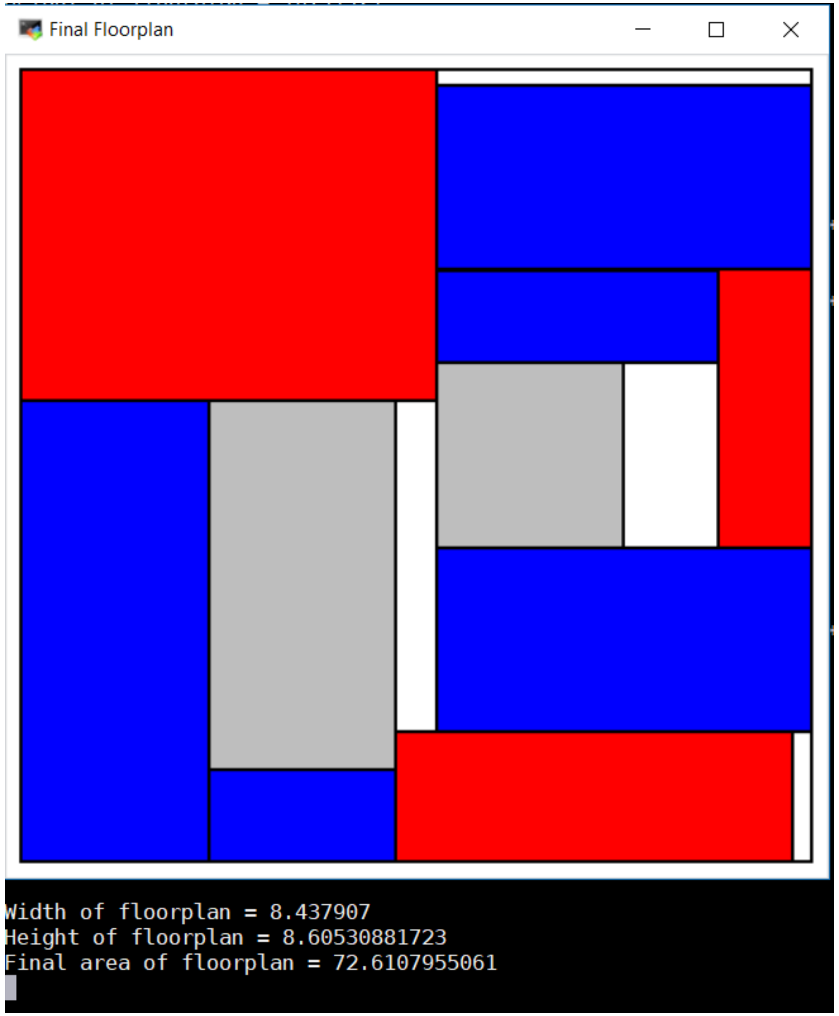
```
Width of floorplan = 6.74264
Height of floorplan = 6.74264
Final area of floorplan = 45.4631941696

Printing to GUI

Width of modified floorplan = 6.74264
Height of modified floorplan = 6.74264
Final area of modified floorplan = 45.4631941696
```

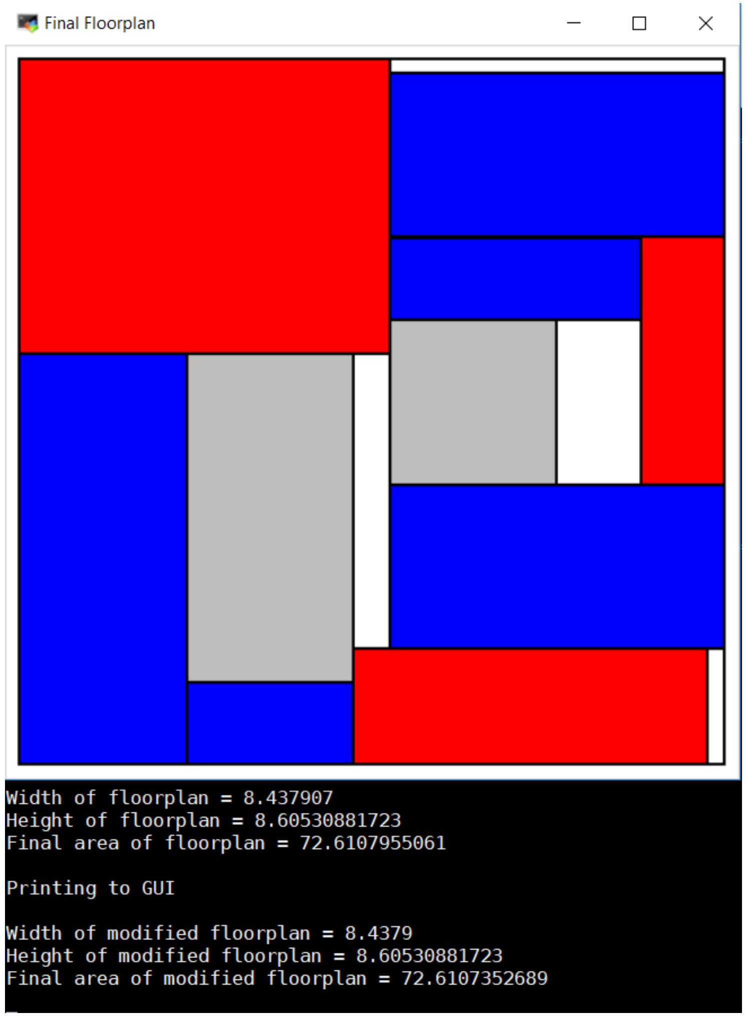
10 Blocks – Underestimation with runtime 60s

With overlap



```
Width of floorplan = 8.437907
Height of floorplan = 8.60530881723
Final area of floorplan = 72.6107955061
```

After Extension



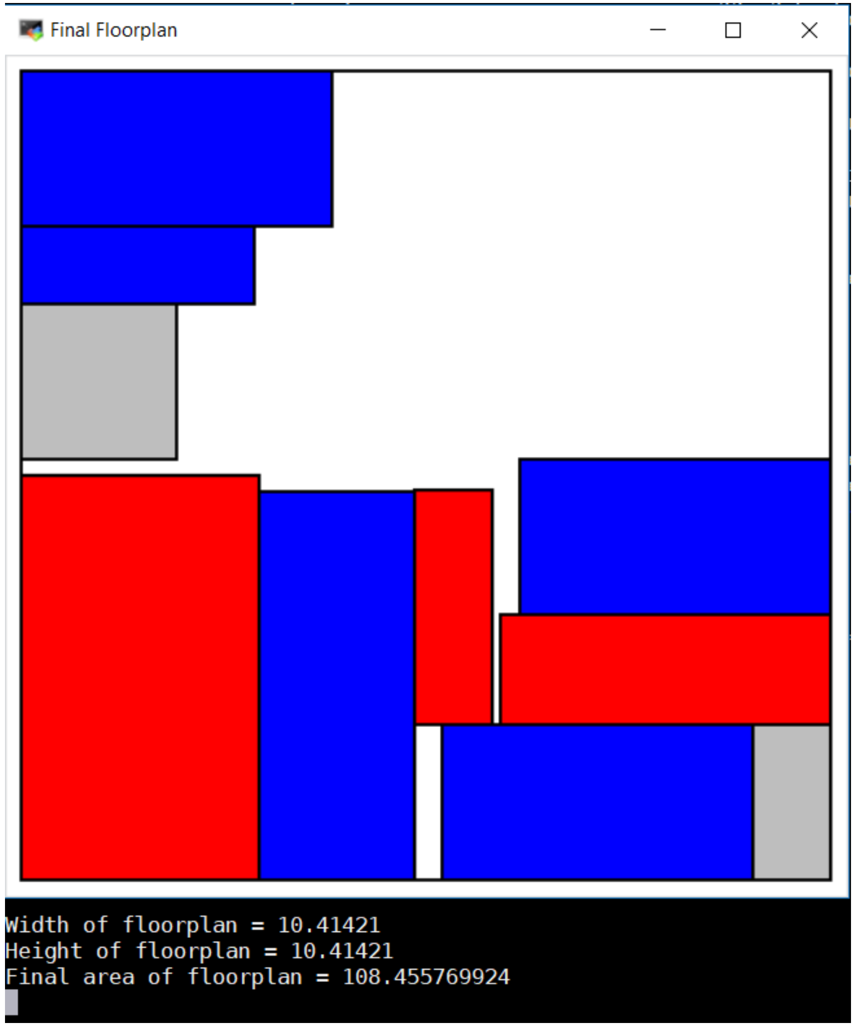
```
Width of floorplan = 8.437907
Height of floorplan = 8.60530881723
Final area of floorplan = 72.6107955061

Printing to GUI

Width of modified floorplan = 8.4379
Height of modified floorplan = 8.60530881723
Final area of modified floorplan = 72.6107352689
```

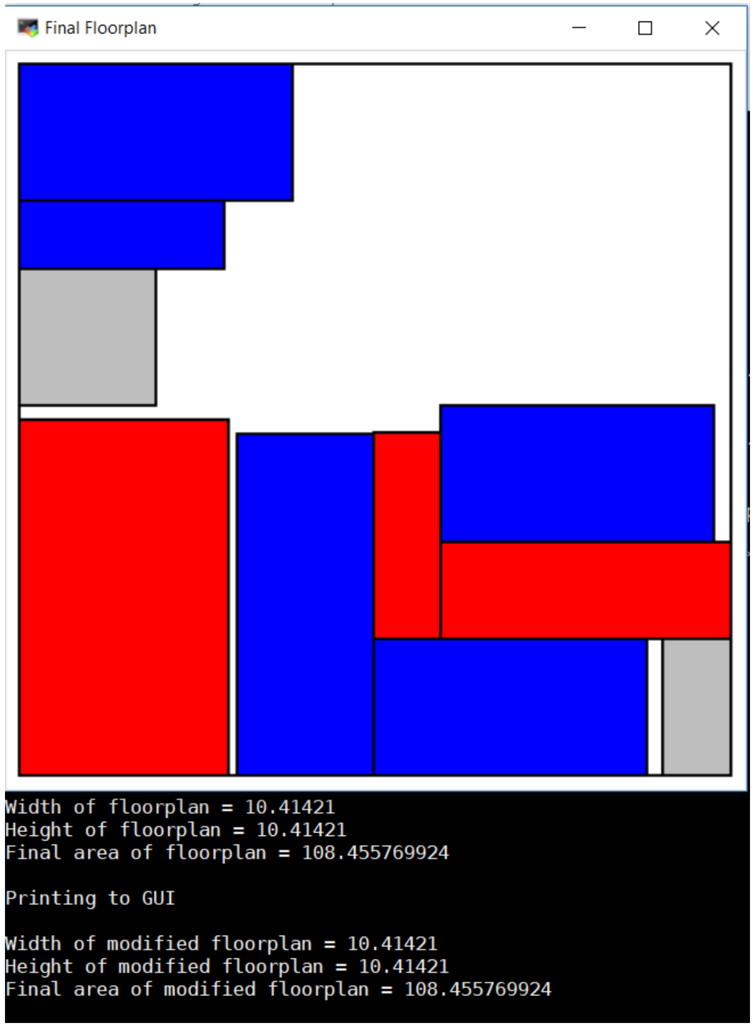
10 Blocks – Overestimation with runtime 60s

With whitespace



```
Width of floorplan = 10.41421
Height of floorplan = 10.41421
Final area of floorplan = 108.455769924
```

After Extension



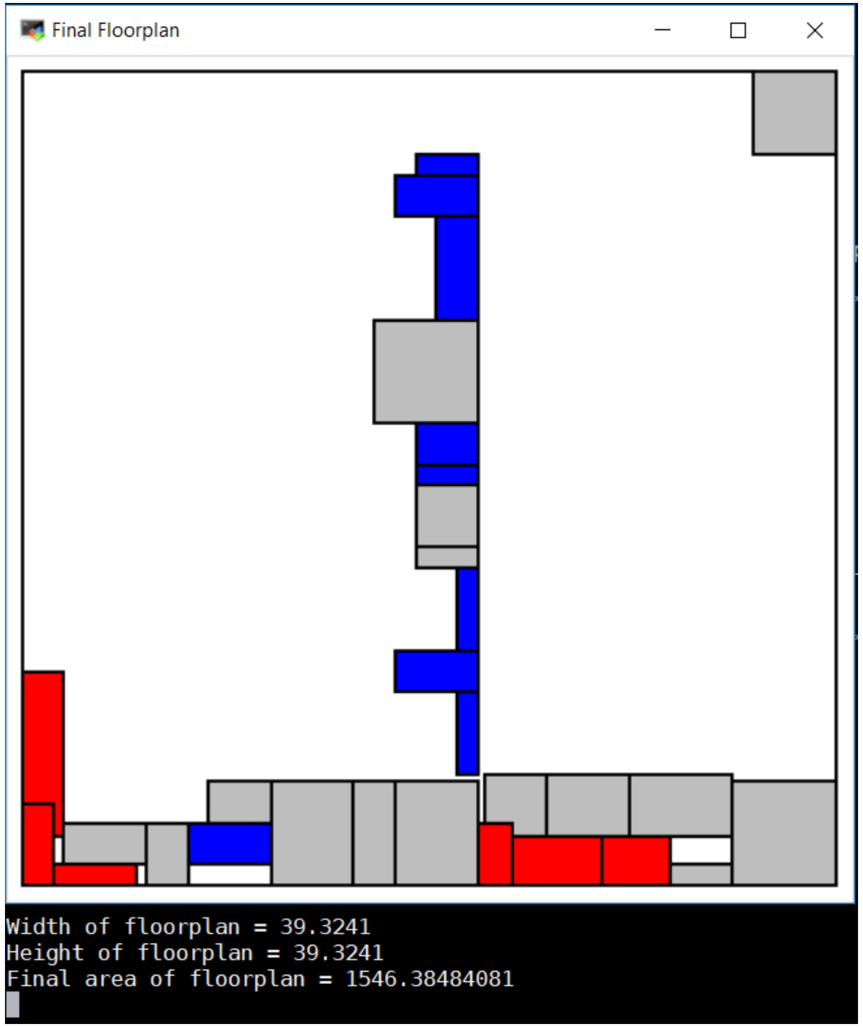
```
Width of floorplan = 10.41421
Height of floorplan = 10.41421
Final area of floorplan = 108.455769924

Printing to GUI

Width of modified floorplan = 10.41421
Height of modified floorplan = 10.41421
Final area of modified floorplan = 108.455769924
```

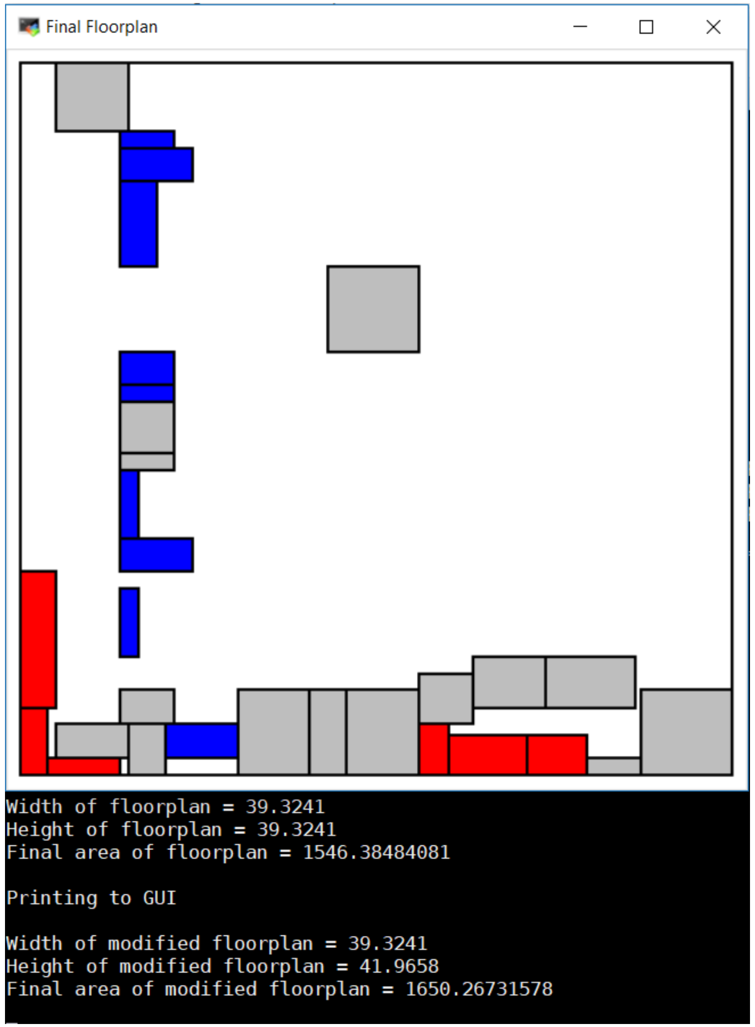
30 Blocks – Underestimation with runtime 600s

With overlap



```
Width of floorplan = 39.3241
Height of floorplan = 39.3241
Final area of floorplan = 1546.38484081
```

After Extension



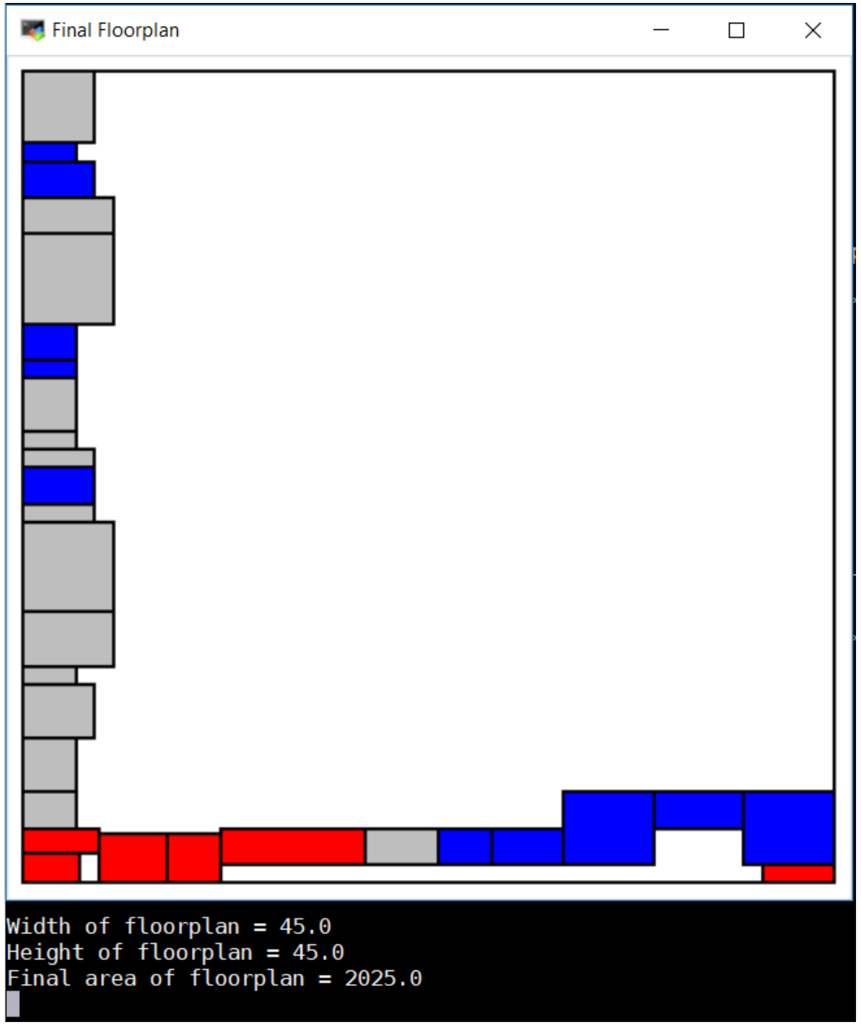
```
Width of floorplan = 39.3241
Height of floorplan = 39.3241
Final area of floorplan = 1546.38484081

Printing to GUI

Width of modified floorplan = 39.3241
Height of modified floorplan = 41.9658
Final area of modified floorplan = 1650.26731578
```

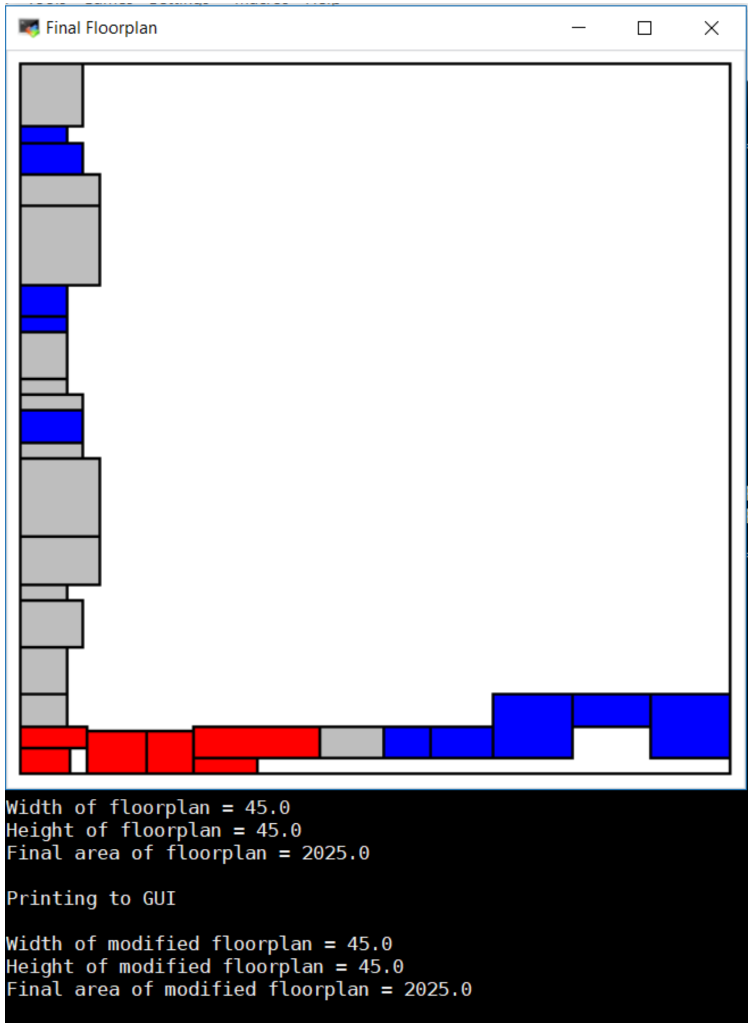
30 Blocks – Overestimation with runtime 600s

With whitespace



```
Width of floorplan = 45.0
Height of floorplan = 45.0
Final area of floorplan = 2025.0
```

After Extension



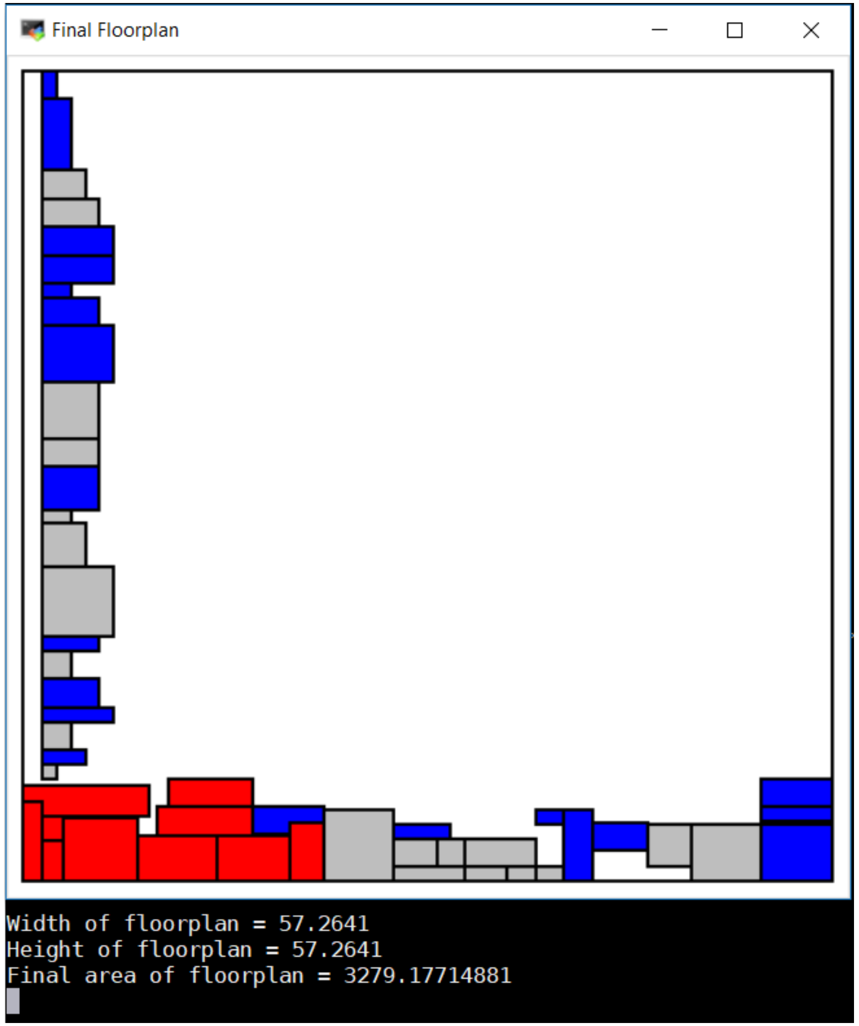
```
Width of floorplan = 45.0
Height of floorplan = 45.0
Final area of floorplan = 2025.0

Printing to GUI

Width of modified floorplan = 45.0
Height of modified floorplan = 45.0
Final area of modified floorplan = 2025.0
```

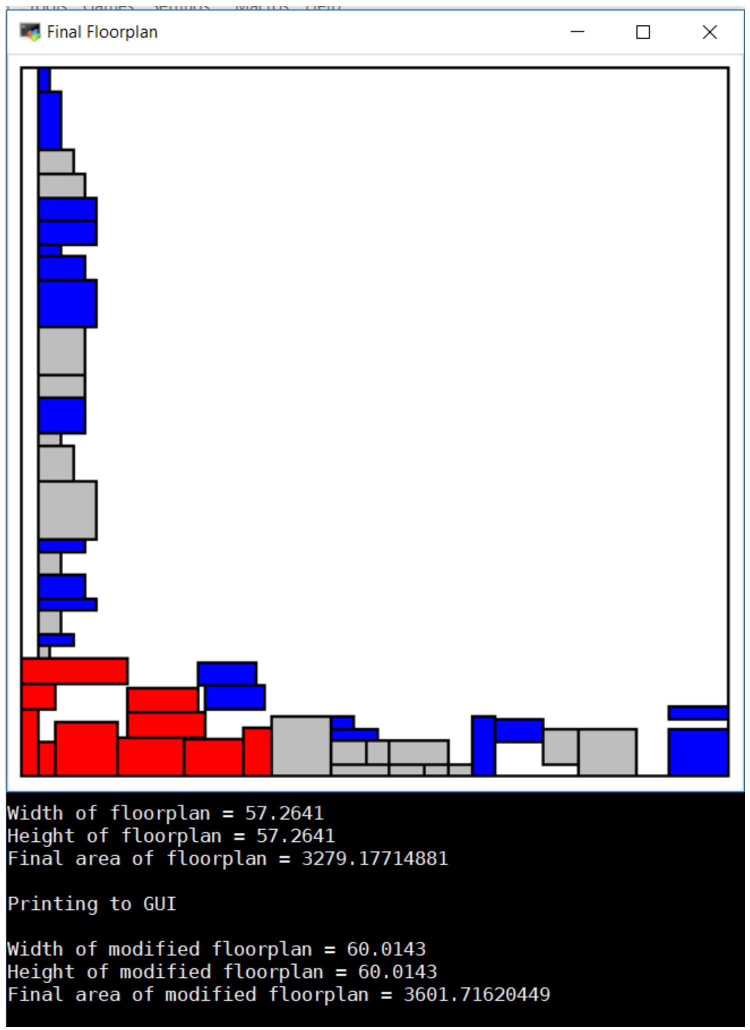
50 Blocks – Underestimation with runtime 1800s

With overlap



```
Width of floorplan = 57.2641
Height of floorplan = 57.2641
Final area of floorplan = 3279.17714881
```

After Extension



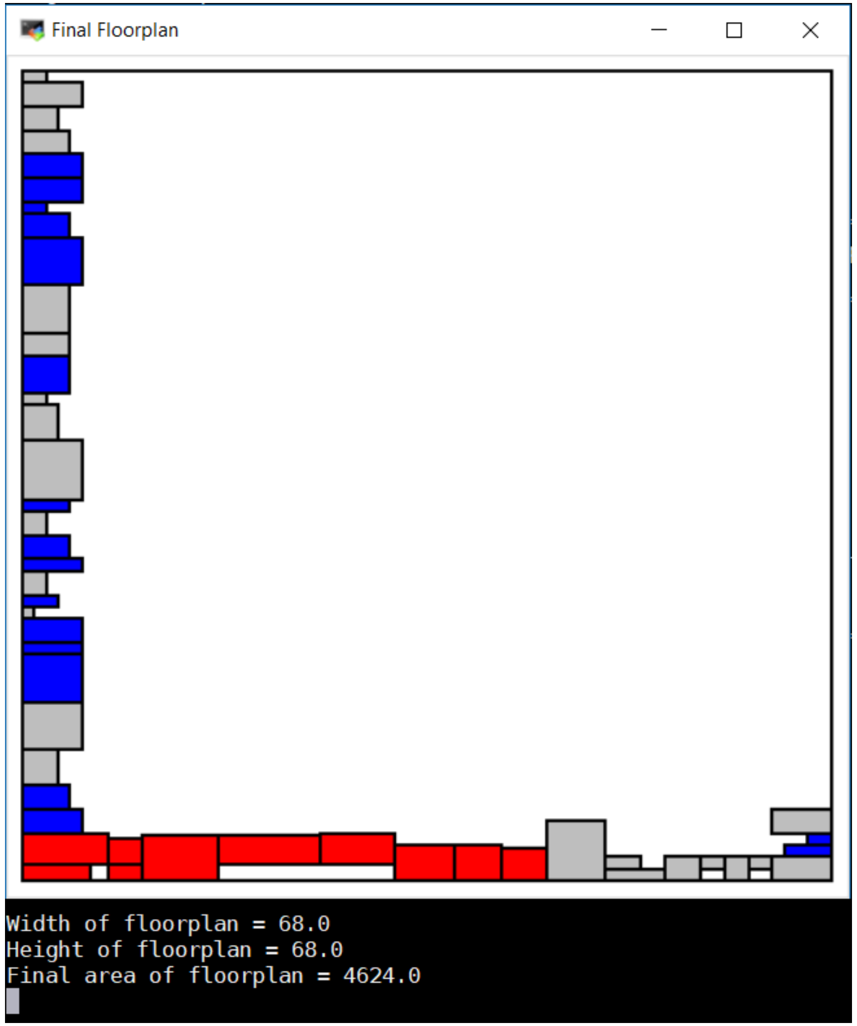
```
Width of floorplan = 57.2641
Height of floorplan = 57.2641
Final area of floorplan = 3279.17714881

Printing to GUI

Width of modified floorplan = 60.0143
Height of modified floorplan = 60.0143
Final area of modified floorplan = 3601.71620449
```

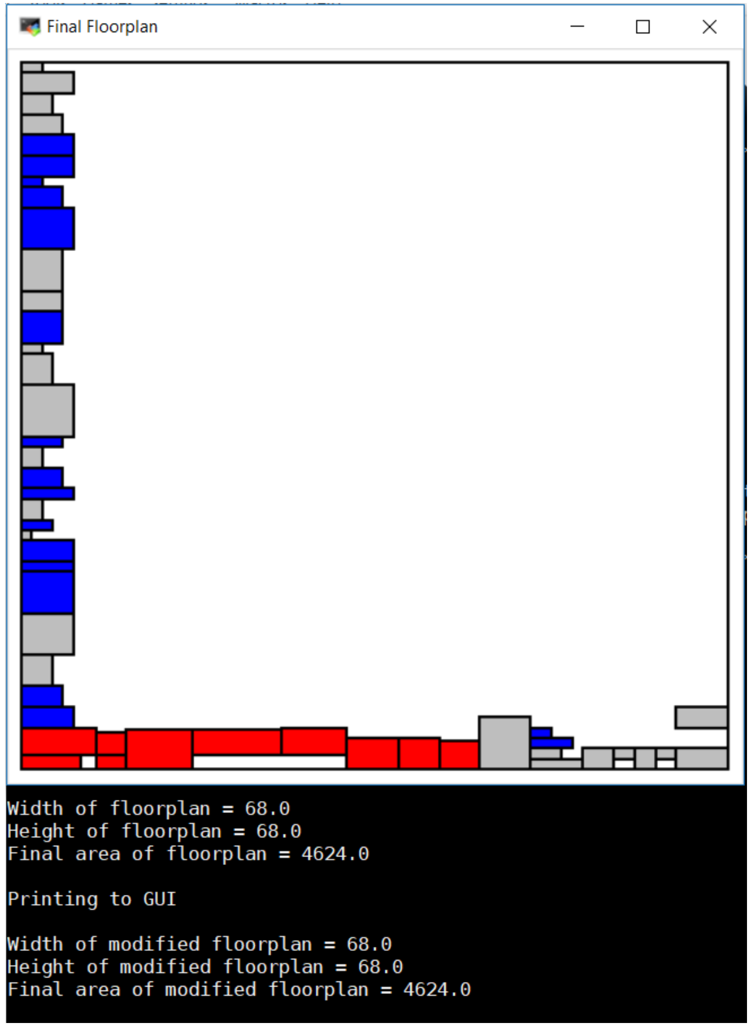
50 Blocks – Overestimation with runtime 1800s

With whitespace



```
Width of floorplan = 68.0
Height of floorplan = 68.0
Final area of floorplan = 4624.0
```

After Extension



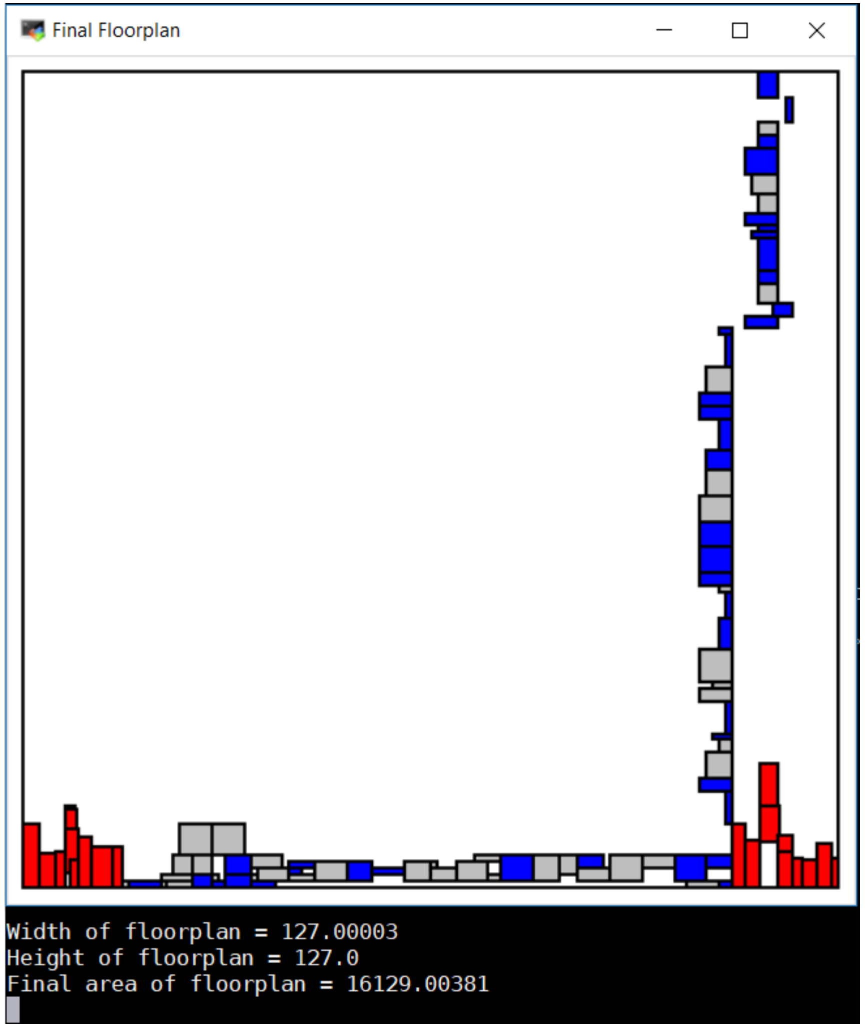
```
Width of floorplan = 68.0
Height of floorplan = 68.0
Final area of floorplan = 4624.0

Printing to GUI

Width of modified floorplan = 68.0
Height of modified floorplan = 68.0
Final area of modified floorplan = 4624.0
```

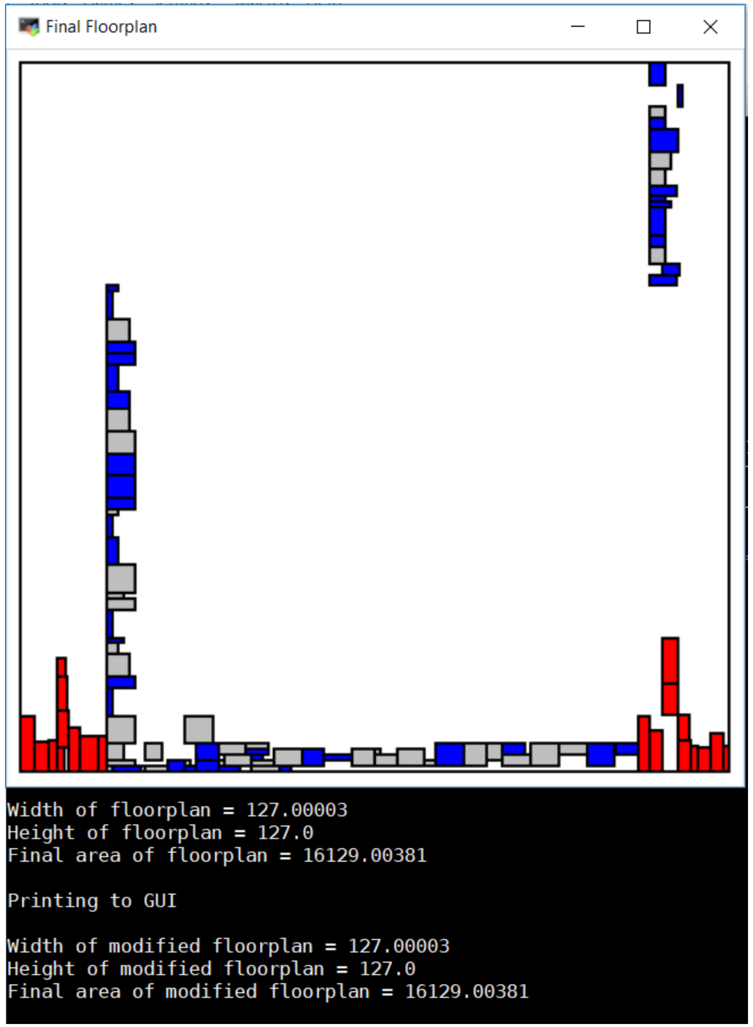
100 Blocks – Underestimation with runtime 3600s

With overlap



```
Width of floorplan = 127.00003
Height of floorplan = 127.0
Final area of floorplan = 16129.00381
```

After Extension



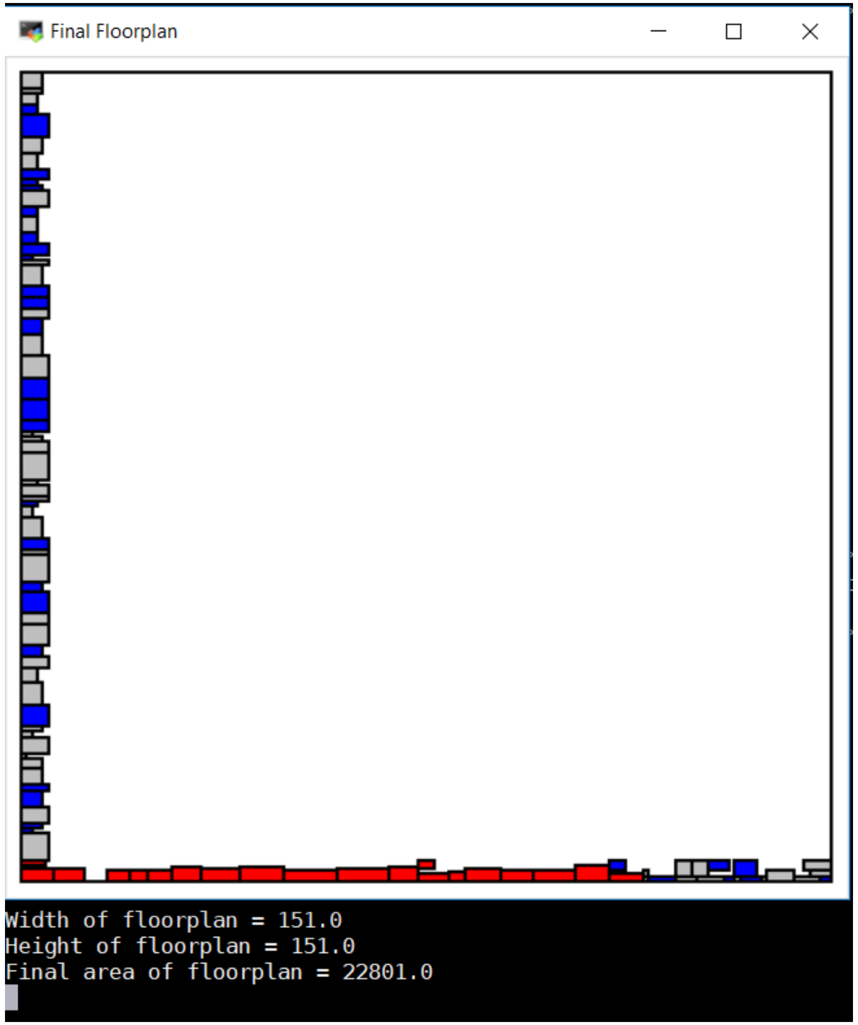
```
Width of floorplan = 127.00003
Height of floorplan = 127.0
Final area of floorplan = 16129.00381

Printing to GUI

Width of modified floorplan = 127.00003
Height of modified floorplan = 127.0
Final area of modified floorplan = 16129.00381
```

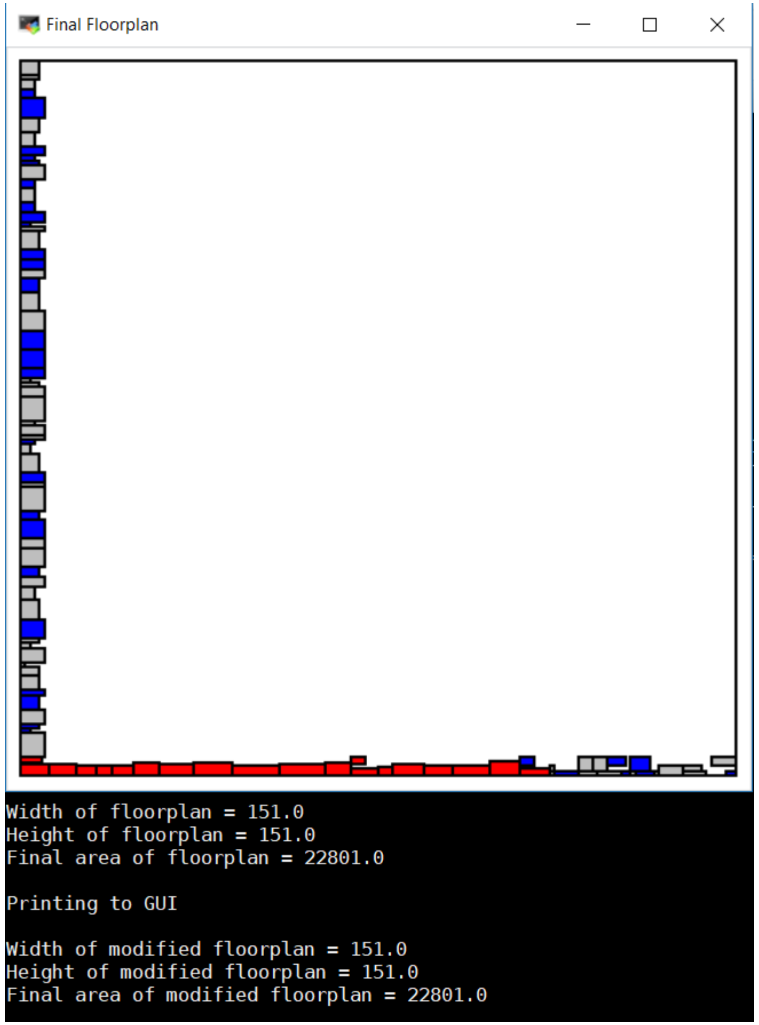
100 Blocks – Overestimation with runtime 3600s

With whitespace



```
Width of floorplan = 151.0
Height of floorplan = 151.0
Final area of floorplan = 22801.0
```

After Extension



```
Width of floorplan = 151.0
Height of floorplan = 151.0
Final area of floorplan = 22801.0

Printing to GUI

Width of modified floorplan = 151.0
Height of modified floorplan = 151.0
Final area of modified floorplan = 22801.0
```

Tabulation of Results

Number of Blocks	Hard Modules	Soft Modules	Area with Underestimation	Area with Overestimation	Runtime
5	3	2	43.6702	46.4631	0.86s
10	7	3	72.6107	108.4557	60s*
30	26	4	1546.3848	2025.0	600s*
50	40	10	3279.1771	4624.0	1800s*
100	80	20	16129.00381	22801.0	3600s*

*Indicates that these blocks were run with a timeout flag, and the program was forced to terminate after this amount of time

Percentage Whitespace

Percentage of Whitespace in the final floorplan		
Benchmarks	Under-estimation	Over-estimation
5_block	12.9841767698	16.4159036907
10_block	9.57929621032	37.3016299201
30_block	81.2465828462	85.6790123457
50_block	85.8196133085	89.9437716263
100_block	93.8371891302	95.6405420815

Run-time is the bottleneck, and with more run-time we could expect to see a reduction in overall whitespace in the floorplan.

Conclusion

The Mixed Integer Linear Programming approach is an analytical method to obtain an area efficient floorplan with runtime being the primary bottleneck. When the algorithm is run for a sufficient amount of time we get good solution quality in terms of area and whitespace, with percentage whitespace being as less as 10%.

Note:

We had also attempted implementing the algorithm in C++ but found no improvement in run-time over Python, so we chose to move ahead with Python due to the ease of integration with GUI.