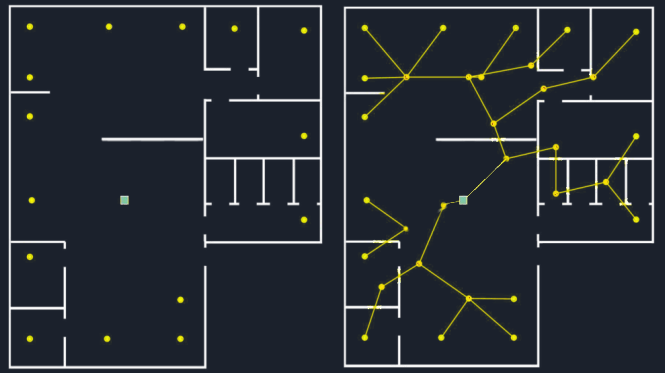
Router Placement Optimization



Given a building plan, a decision needs to be made on:

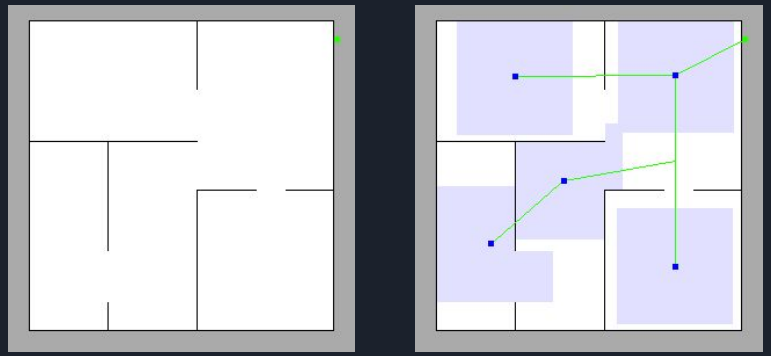
“ Where to put wireless routers and how to connect them to the fiber backbone to maximize coverage and minimize cost. ”

Eduardo Brito up201806271@fe.up.pt | Paulo Ribeiro up201806505@fe.up.pt | Rita Silva up201806527@fe.up.pt

Formulation as an Optimization Problem

- Solution Representation

Given a building blueprint, represented as a character matrix, the solution will consist of the same matrix of the building, with the routers and the backbone on their respective positions.



The cells used to link the backbone and the router will be highlighted, along with the cells inside the router range.

- Rigid Constraints

➔ To start, there is only one cell connected to the backbone, which can be of any type - target(.), void(-) or wall(#).

➔ A router can't be placed inside a wall and it has to be connected to the fiber backbone - a cable that delivers Internet to the router itself.

➔ Each router covers a square area of at most $(2 \times R + 1)^2$ cells around it, unless the signal is stopped by a wall cell:

- $|a - rx| \leq R$, and $|b - ry| \leq R$
- there is no wall cell inside the smallest enclosing rectangle of [a, b] and [rx, ry]. That is, there is no wall cells [w, v] where both $\min(a, rx) \leq w \leq \max(a, rx)$ and $\min(b, ry) \leq v \leq \max(b, ry)$.

➔ Placing a single router costs Pr and connecting a single cell to the backbone costs Pb. To ensure the budget constraint, the cost to place the routers and its connections cannot be higher than the budget:

$$Nb \times Pb + Nr \times Pr \leq B$$

Formulation as an Optimization Problem

- Neighborhood/Mutation

  One possible new-solution generator may be to start by randomly placing a number N of routers.

  The solution neighborhood is represented by the neighboring cells of each one of the routers' positions.

- Crossover Functions

  The crossover between two distinct solutions will consist on selecting and combining the routers with the highest coverage from each solution, without overlapping ranges.

- Evaluation Function

  Being c the total number of *target cells* covered, the value of a solution can be computed as follows:

  $$value = c + (B - (Nb \times Pb + Nr \times Pr))$$

  where:
  - ➔ B is the Budget
  - ➔ Nb is the number of Backbone cells used
  - ➔ Pb is the cost per Backbone cell
  - ➔ Nr is the number of Routers used
  - ➔ Pr is the cost per Router

  A solution found to have an higher value for a given building plan may be a solution that covers more *target cells*, while respecting the budget and having lower implementation costs.

Work already done and Tools in use

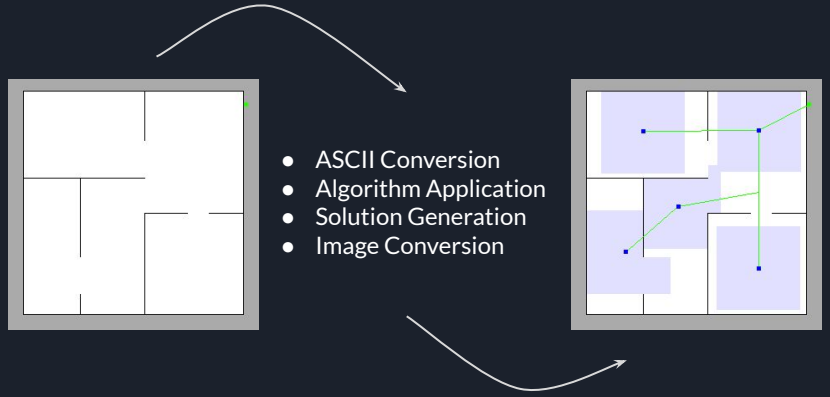
- Tools

➔ **Programming Language:** Python3

➔ **Development Environment:** JupyterLab & Visual Studio Code

➔ **File Structure:** The input file may be an image or an ASCII representation of the building, with only the void, target and wall cells. It can also contain the information about the budget and the costs. The solution file will consist on the same schema, but with the router(s) placed, along with the cells within the range and the backbone links. An image of the solution will be generated, with the representative colors.

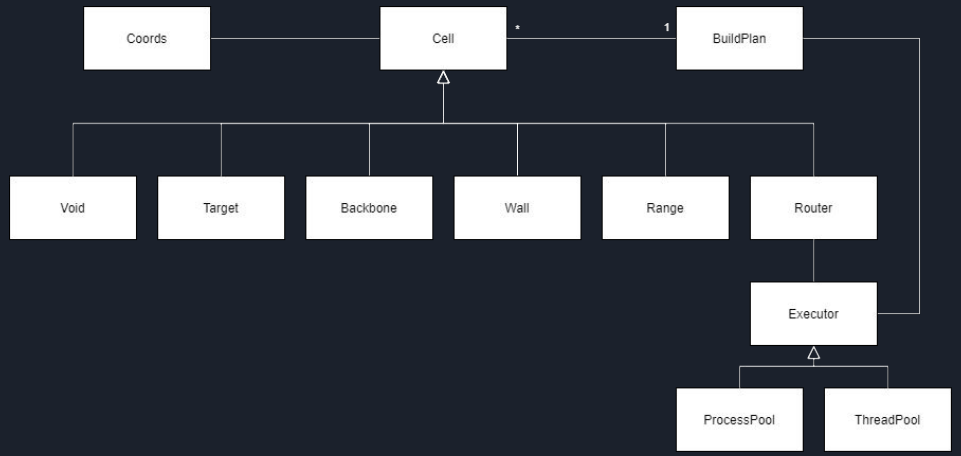
- Resolution Workflow



ASCII Symbol	Meaning	Color
-	void	grey
.	target cell	white
#	wall cell	black
R	router	blue
r	router range	light blue
b	backbone	green

Work already done and Additional Features

- Classes Diagram



- Additional Features

➔ **Algorithms:**

- Kruskal's algorithm - for generating a minimum spanning tree of backbone links between the routers and the initial backbone cell.
- Parallel flood fill algorithm - custom layered flood fill for generating the set of target cells in a router's range.

➔ **Parallel Computing**:

- ProcessPoolExecutor - for parallel generation of each router's set of target cells.
- ThreadPoolExecutor - for parallel calculation of the targeted cells of each layer in the router's range.

References and Related Work

- Mohammed A. Alanezi , Houssem R. E. H. Bouchekara, and Muhammad S. Javaid. (2020). Optimizing Router Placement of Indoor Wireless Sensor Networks in Smart Buildings for IoT Applications.
- Google #Hash Code 2017 Challenge "Router placement" - Team Gyrating Flibbittygibbitts.
- Introduction to Mutation, Crossover Operators, Survivor Selection - Genetic Algorithms, TutorialsPoint
- Google #Hash Code 2018 Challenge “City Plan - Optimization Problem for Public Projects Implementation”