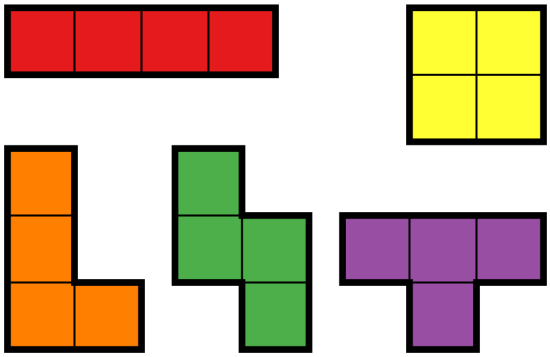
INF219 – COWI Research



Jakob Sverre Alexandersen
Thone Anlaug Stordal Støa
Hanna Søndenaa Rasmussen
Håvard Kattem Lunde
Hermann Holstad Walaunet

April 2025

GitHub Repository

Contents

1 Summary

Click me: for the full repo

This project considers techniques for placing light fixtures in any given room, minimizing the amount of lights used whilst still being compliant with regulatory and aesthetic requirements.

There are many different approaches to such a challenge. For instance, COWI presented a genetic algorithm/reinforcement learning ML-based solution. Their apprach entailed maximizing coverage area while maintaining a pattern by penalizing variations in x/y coordinates.

We chose to take a different approach, using Linear Programming instead. The main reason behind this choice is the guaranteed optimality involved in such a solution. Using a solver, we are able to find the single most mathematically optimal solution that satisfies all constraints, given that a solution exists.

In addition to the optimality, we are also presented with several other advantages using the LP approach. The deterministic nature of our solution ensures that given the same inputs and constraints, we always get identical, reproducible results — eliminating the randomness inherent in genetic algorithms. This consistency is particularly valuable in professional settings where repeatability is essential.

Our LP formulation also offers greater transparency and explainability, making it clear exactly what we're optimizing for and what constraints we're satisfying. For typical room dimensions, the LP solver finds optimal solutions relatively quickly, without requiring the multiple generations of evolution that genetic algorithms need. This efficiency translates to faster design iterations and project delivery. The precise constraint handling in our approach allows for explicit modeling of critical requirements like minimum light levels and fixture spacing. Rather than approximating these through penalty functions as in the genetic algorithm approach, we incorporate them directly into the mathematical model. Finally, our solution offers straightforward adaptability to changing requirements. The LP model parameters can be easily adjusted to meet different regulatory standards or design preferences without needing to retrain an entire ML model. While the ML-based approach might offer advantages in terms of continuous learning from historical data, our LP solution provides immediate, verifiable results with mathematical guarantees—a critical factor when dealing with safety-related regulatory compliance in building design. This becomes a bigger deal if we choose to also incorporate positioning of a sprinkler system.

2 Motivation and Introduction (Thone)

3 Mathematics (jakob se over)

In this section, we will discuss the mathematical model behind the program.

3.1 Sets and Indices

$$G = (x, y) \mid (x, y) \in \mathbb{R} \tag{1}$$

$$A = (x, y) \mid (x, y) \in \mathbb{R} \tag{2}$$

3.2 Decision Variables

$$z_{x,y} \in \{0, 1\} \text{ for } (x, y) \in G \tag{3}$$

$$c_{x,y} \geq 0 \text{ for } (x, y) \in G \tag{4}$$

3.3 Parameters

$$\alpha_{(x_g,y_g),(x_a,y_a)} \in \{0, 1\} \tag{5}$$

$$L_{\min} \geq 0 \tag{6}$$

$$S_{\min} \geq 0 \tag{7}$$

$$A_g \tag{8}$$

3.4 Objective Function

$$\min \sum_{(x,y)\in G} z_{x,y} \tag{9}$$

3.5 Constraints

$$c_{x_a,y_a} = \sum_{(x,y)\in G} \alpha_{(x_g,y_g),x_a,y_a} \cdot z_{x_g,y_g} \quad \forall (x_a, y_a) \in A \tag{10}$$

$$\sum_{(x_a,y_a)\in A_g} c_{x_a,y_a} \geq L_{\min} \cdot |A_g| \quad \forall g \in G \text{ where } |A_g| > 0 \tag{11}$$

$$\max\left(\frac{|x_1 - x_2|}{\text{grid size}}, \frac{|y_1 - y_2|}{\text{grid size}}\right) \leq S_{\min} \tag{12}$$

$$z_{x_1,y_1} + z_{x_2,y_2} \leq 1 \tag{13}$$

3.6 Equations explainations

1. List of grid points where we may place light fixtures
2. List of smaller grid points used to calculate the coverage
3. Binary variable indicating whether a light is placed at point (x, y) – 1 if yes, 0 otherwise
4. Continuous variable representing the light level at area cell (x, y)
5. Coverage value representing the amount of light that a fixture at grid point (x_g, y_g) contributes to area cell (x_a, y_a)
6. Minimum average light level required in each grid cell
7. Minimum spacing required between fixtures (measured in grid units)
8. Set of area cells contained within the grid cell $g \in G$
9. **Minimize the total amount of light fixtures**
10. For each area cell $(x_a, y_a) \in A$, the light level is determined by the sum of contributions from all placed fixtures
11. For each grid cell $(x_g, y_g) \in G$, the average light level across all contained area cells must meet or exceed the minimum requirement
12. For any two grid points $(x_1, y_1), (x_2, y_2) \in G$ that are closer than the minimum spacing,
13. We add the constraint that at most one of them can have a fixture
14. TODO: add neigh constraint

4 Execution

4.1 Method (jakob)

4.2 What we tried: (hanna)

5 Experiments (TBD)

5.1 Light distribution plot

5.2 Small room

5.3 Large room

5.4 Runtime Discussion

6 Discussion (Håvard)

6.1 Potential Improvements

6.1.1 Caching

7 Conclusion (Hermann)