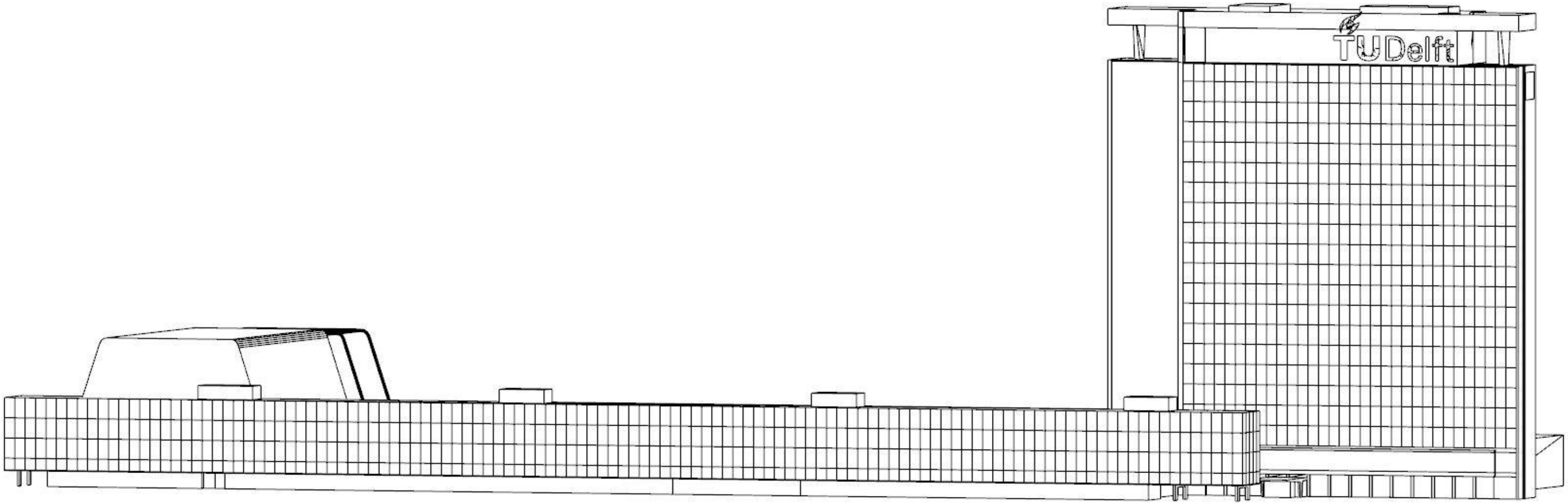
Cochleas

A tool to convert an existing layout into a residential one



Konstantina Chouliara 4744292

1st mentor: Dr.ir. P.Nourian | 2nd mentor: Dr.ir. W.van der Spoel | Delegate examiner: Dr. J. Hoekstra

01
Research formulation

02
Tool development

03
Layout development

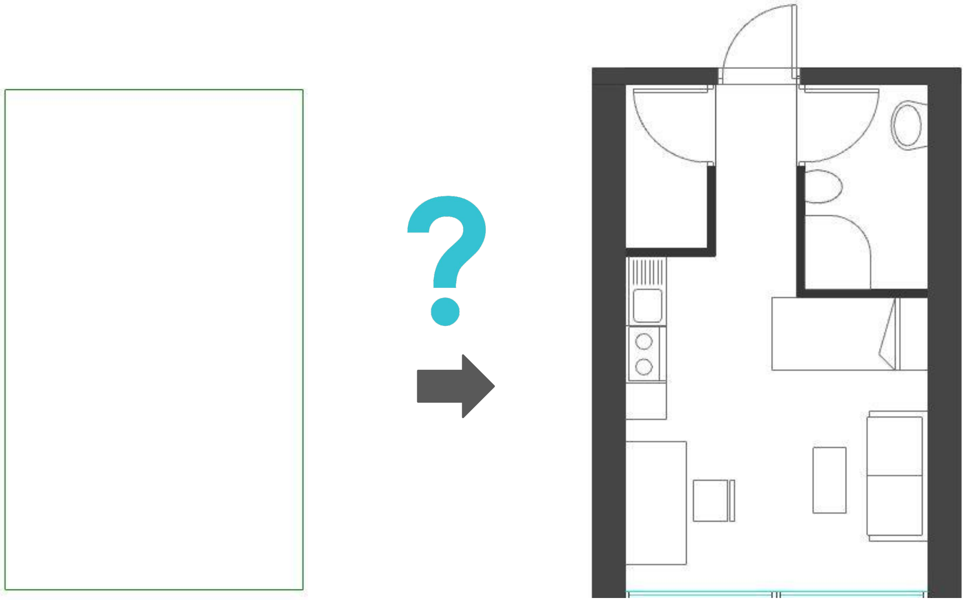
04
Evaluation

05
Conclusions

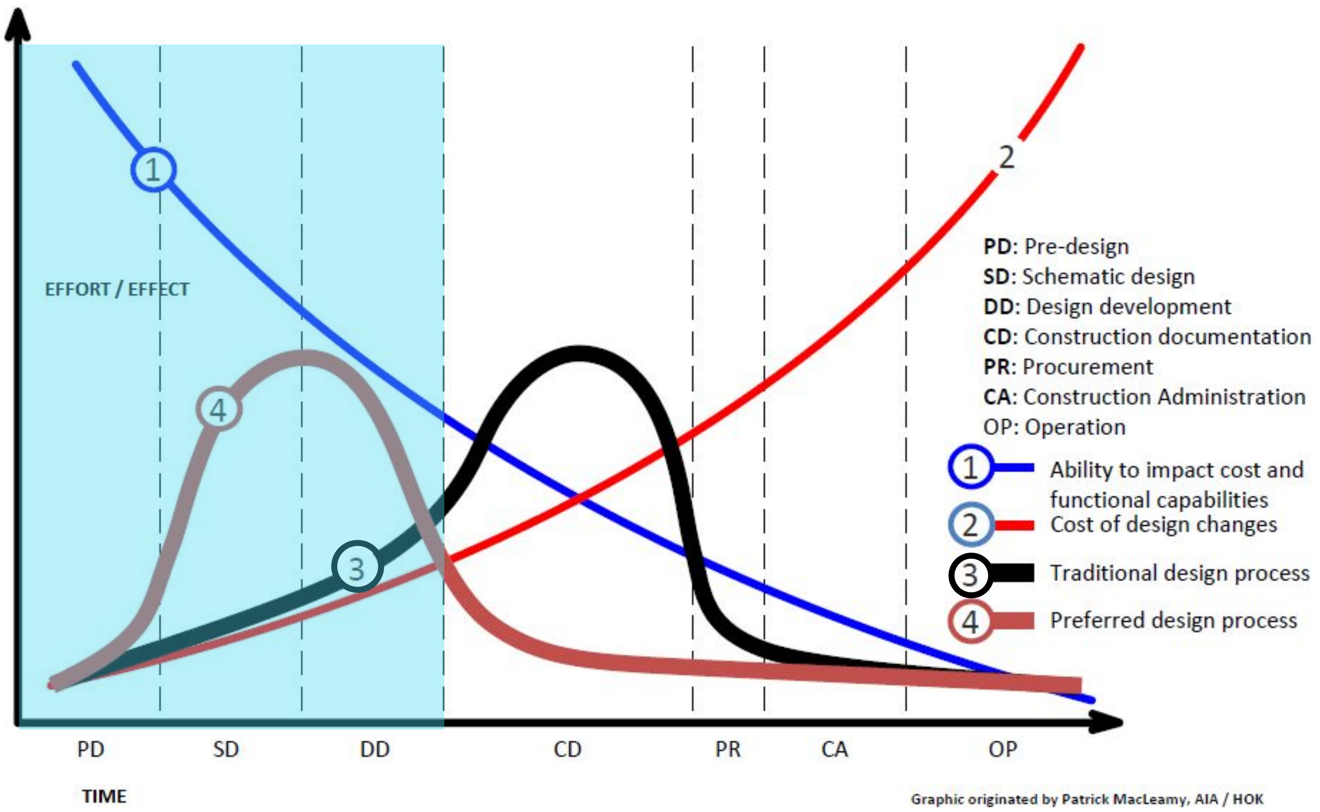
Motivation

Existing building

Residential layout

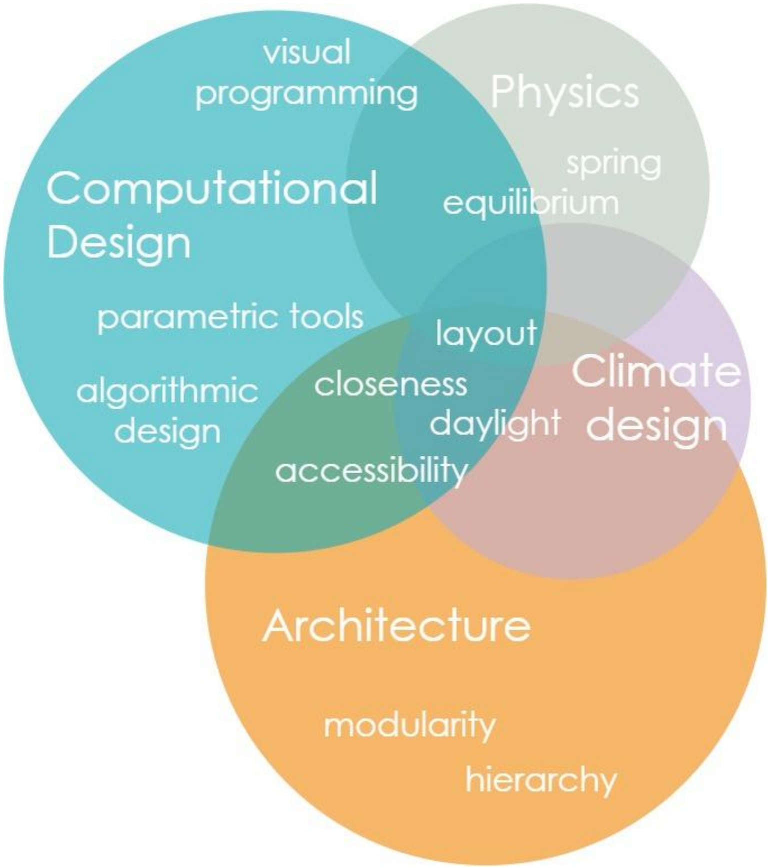


Design process



Source: http://division4triclinium.blogspot.com/2013/06/of-macleamy-curve-efficient-design-and.html

Scope


visual
programming
Physics
spring
equilibrium
Computational
Design
parametric tools
layout
algorithmic
design
closeness
daylight
Climate
design
accessibility
Architecture
modularity
hierarchy

Research question

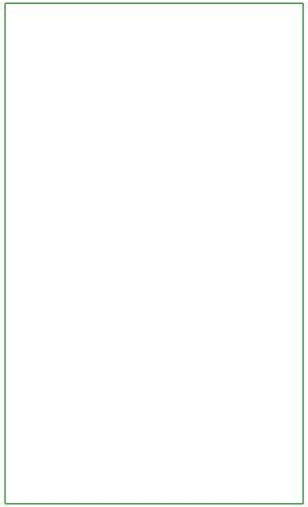
> "To what extent is it possible to convert an **existing layout** into a **residential** one regarding **proximity** relationships and **illuminance** requirements using **computational tools** during primary design stages?"

Subquestions

- How to find optimal configuration regarding **proximity**?
- How to find optimal configuration regarding **illuminance**?
- How to **combine** proximity and illuminance preferences?
- Are existing **plugins** for Grasshopper useful?

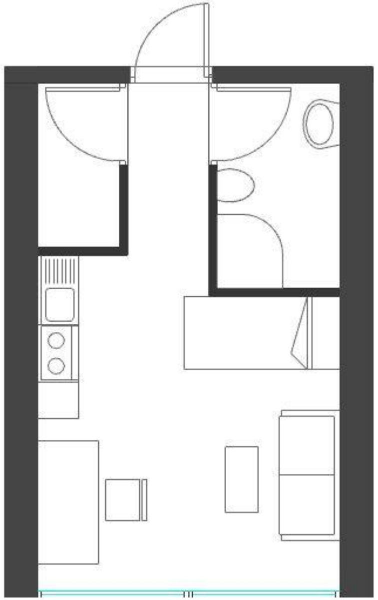
Problem statement

Existing building



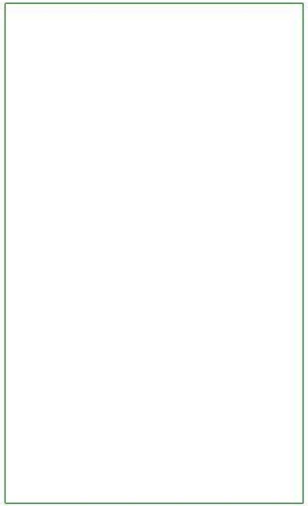
?

Residential layout



Problem statement

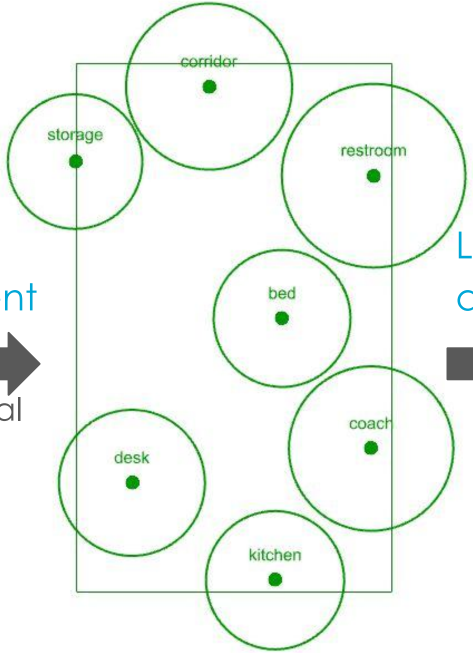
Existing building



Tool development

computational process

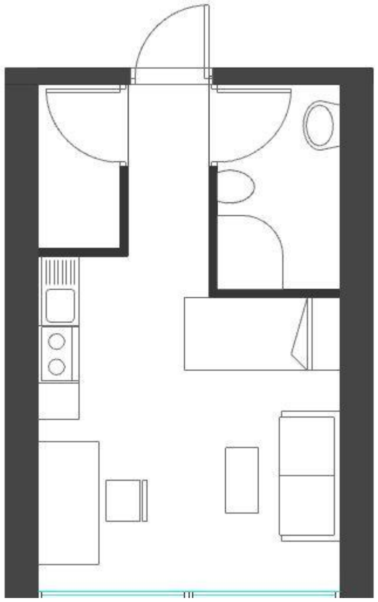
Bubble diagram



Layout development

manual process

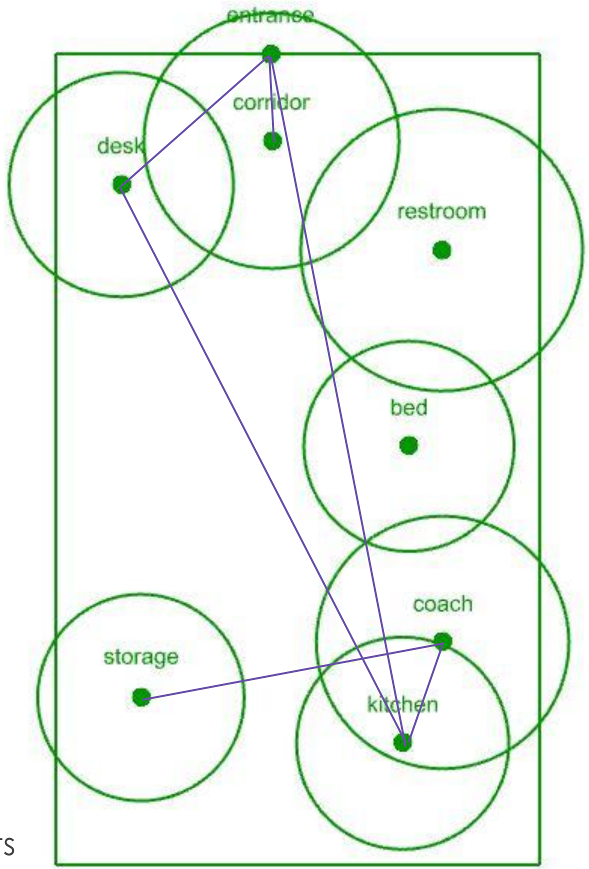
Residential layout



Requirement: Proximity

Adjacency matrix - Proximity

	entrance	kitchen	couch	bed	restroom	desk	corridor	storage
entrance		1	0	0	0	0	1	1
kitchen	1		1	0	0	0	0	1
couch	0	1		0	0	1	0	0
bed	0	0	0		0	0	0	0
restroom	0	0	0	0		0	0	0
desk	0	0	1	0	0		0	0
corridor	1	0	0	0	0	0		0
storage	1	1	0	0	0	0	0	



proximity requirements

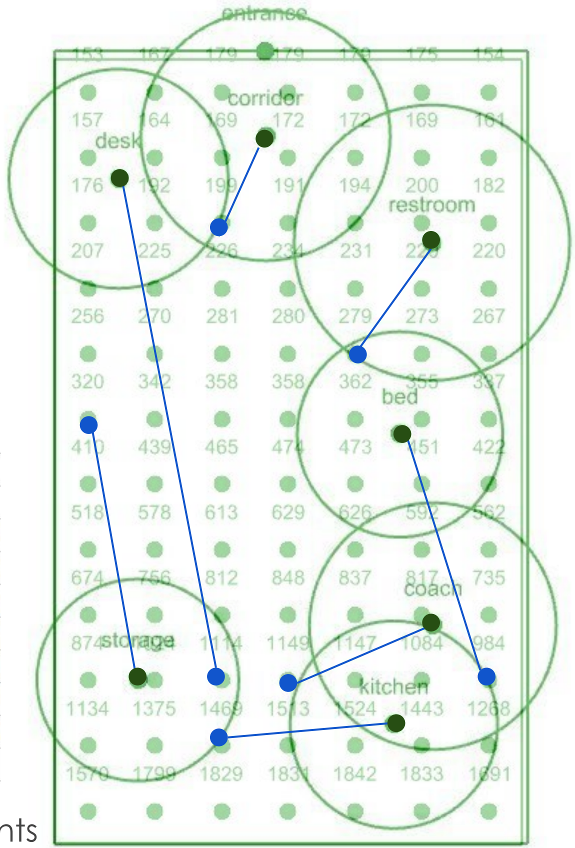
Requirement: Illuminance

Illuminance recommendation

Room	category	min lux	target lux
entrance	1	100	300
kitchen	2	300	500
couch	2	300	500
bed	2	300	500
restroom	1	100	300
desk	3	500	750
corridor	1	100	300
storage	1	100	300

Adjacency matrix - Illuminance

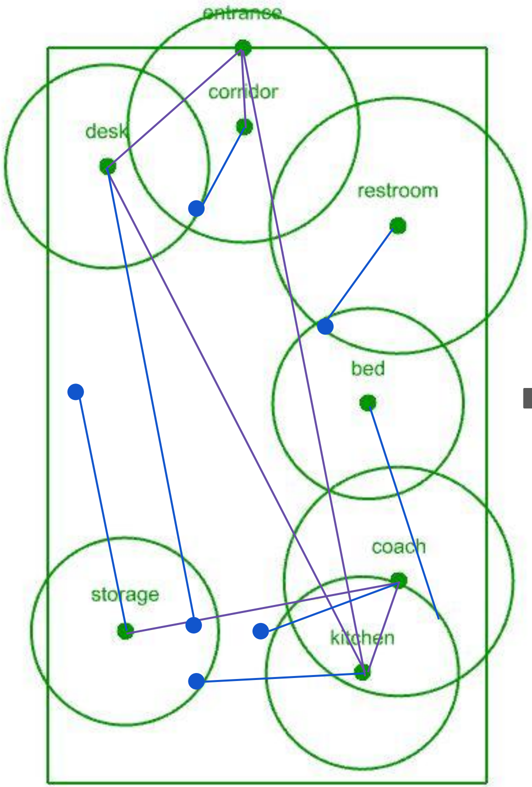
	position for entrance	position for kitchen	position for couch	position for bed	position for restroom	position for desk	position for corridor	position for storage
entrance	1	0	0	0	0	0	0	0
kitchen	0	1	0	0	0	0	0	0
couch	0	0	1	0	0	0	0	0
bed	0	0	0	1	0	0	0	0
restroom	0	0	0	0	1	0	0	0
desk	0	0	0	0	0	1	0	0
corridor	0	0	0	0	0	0	1	0
storage	0	0	0	0	0	0	0	1



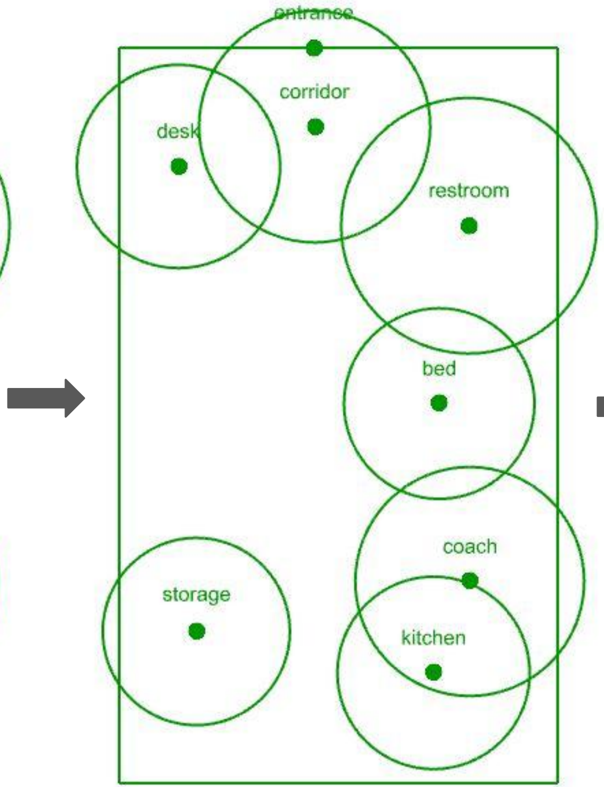
illuminance requirements

Objective

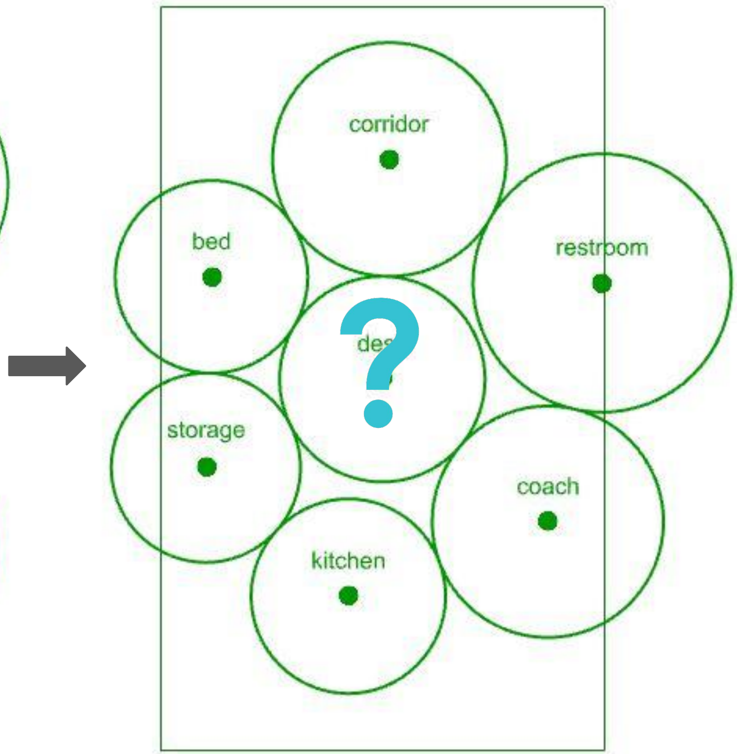
Requirements



Overlap avoidance

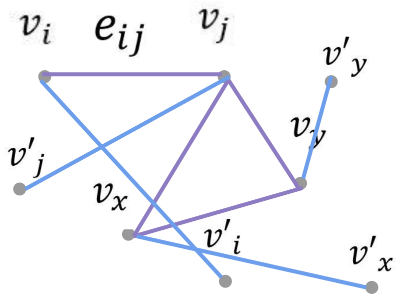


Optimal configuration



Spring system: inputs

1. Vertices
2. Edges



Proximity & illuminance requirements as adjacency matrix

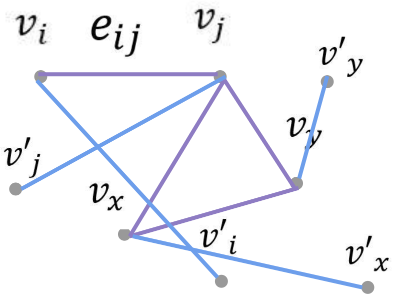
	v_i	v_j	v_x	v_y	v'_i	v'_j	v'_x	v'_y
v_i		1	0	0	1	0	0	0
v_j	1		1	1	0	1	0	0
v_x	0	1		1	0	0	1	0
v_y	0	1	1		0	0	0	1
v'_i	1	0	0	0		0	0	0
v'_j	0	1	0	0	0		0	0
v'_x	0	0	1	0	0	0		0
v'_y	0	0	0	1	0	0	0	

proximity

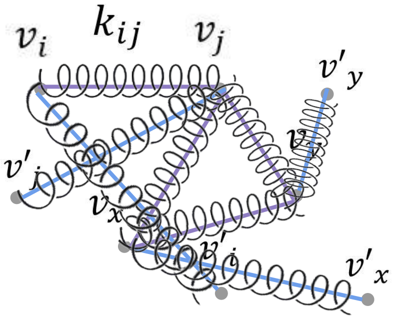
illuminance

Spring system: inputs

1. Vertices
2. Edges



3. Stiffness



Proximity & illuminance requirements as weight factors

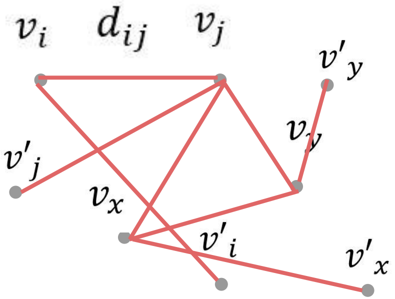
	v_i	v_j	v_x	v_y	v'_i	v'_j	v'_x	v'_y
v_i		0.9	0	0	0.9	0	0	0
v_j	0.9		0.6	0.6	0	0.9	0	0
v_x	0	0.6		0.3	0	0	0.9	0
v_y	0	0.6	0.3		0	0	0	0.9
v'_i	0.9	0	0	0		0	0	0
v'_j	0	0.9	0	0	0		0	0
v'_x	0	0	0.9	0	0	0		0
v'_y	0	0	0	0.9	0	0	0	

proximity

illuminance

Spring system: calculations

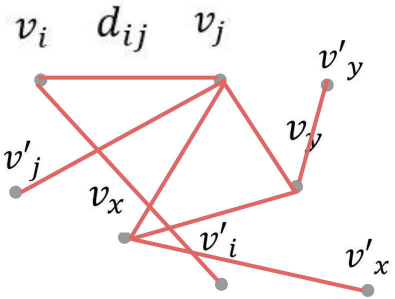
4. Distances



$$d_{ij} = v_i - v_j$$

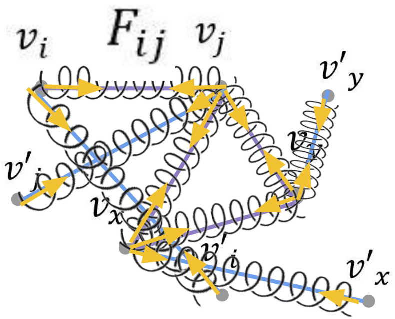
Spring system: calculations

4. Distances



$$d_{ij} = v_i - v_j$$

5. Forces



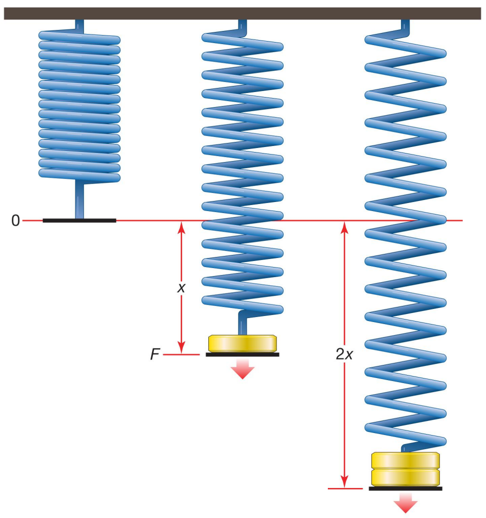
Hooke's law

$$F = -k\,x$$

F: force

k: stiffness

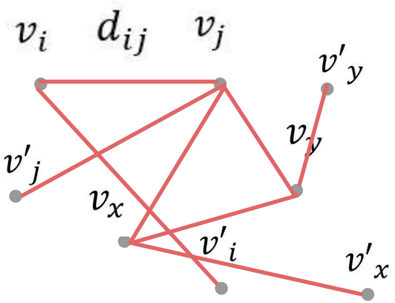
x: elongation



Source: https://www.britannica.com/science/Hookes-law

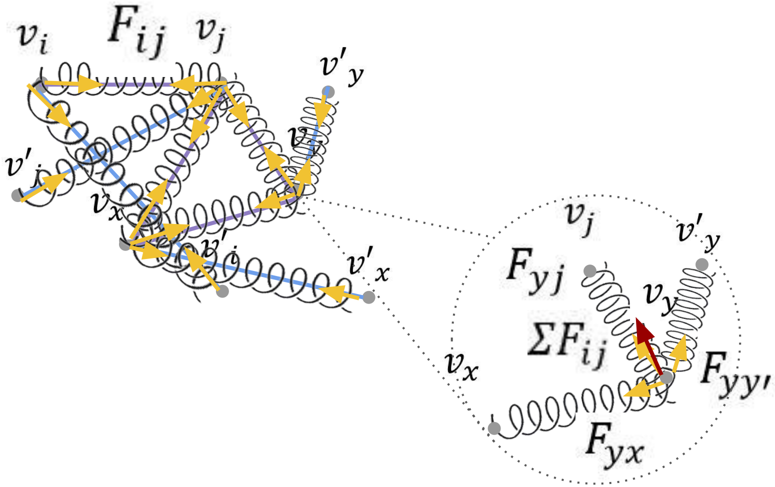
Spring system: calculations

4. Distances



$$d_{ij} = v_i - v_j$$

5. Forces



6. Oscillation

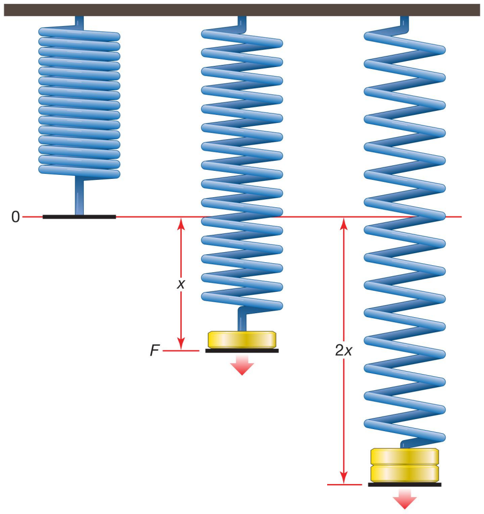
Hooke's law

$$F = -k\,x$$

F: force

k: stiffness

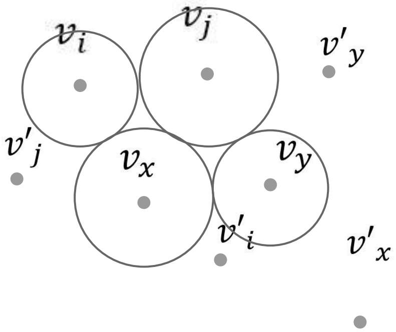
x: elongation



Source: https://www.britannica.com/science/Hookes-law

Spring system: objective

7. Equilibrium



Objective

$$min\ U_{(x)}$$

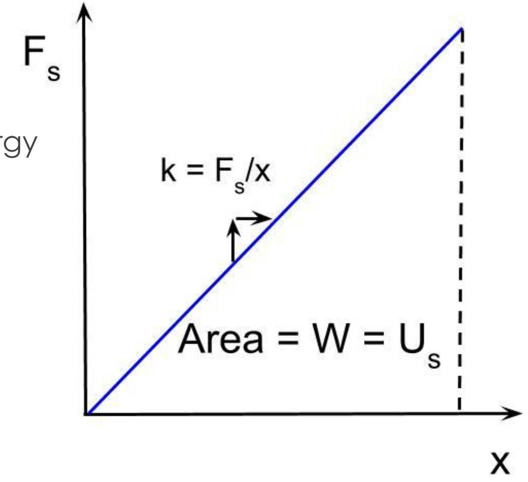
Elastic potential energy

$$U = \frac{1}{2} k\, x^2$$

U: elastic potential energy

x: elongation

k: stiffness



Source: https://www.khanacademy.org/science/ap-physics-1/ap-work-and-energy/spring-potential-energy-and-hookes-law-ap/a/spring-force-and-energy-ap1

Gradient descent: theory

Function $U(x)$ to be minimized

1. Start with some random values for x (eg $x=1$)
2. Change x to reduce $U(x)$:

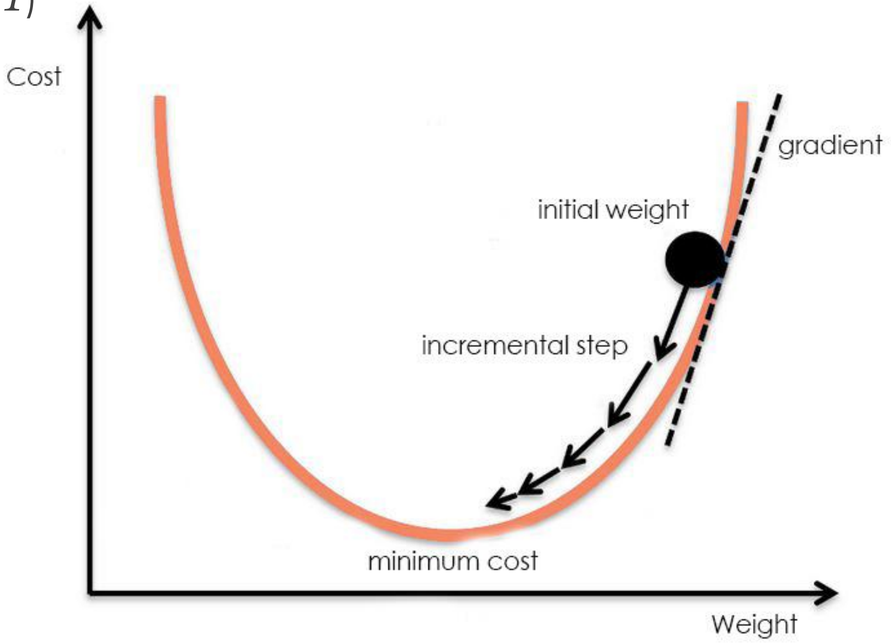
$$x^{t+1} = x^t - \eta * U'(x)$$

η: learning rate

$U'(x)$: partial derivative

3. Repeat until convergence:

x^{t+1} similar to x^t



Source: https://blog.clairvoyantsoft.com/the-ascent-of-gradient-descent-23356390836f

Gradient descent: application

Function $U(d_{ij})$ to be minimized

$$U_{(d_{ij})} = \frac{1}{2} k_{ij}\, d_{ij}^{\;2}$$

$$U'_{(d_{ij})} = k_{ij} * d_{ij}$$

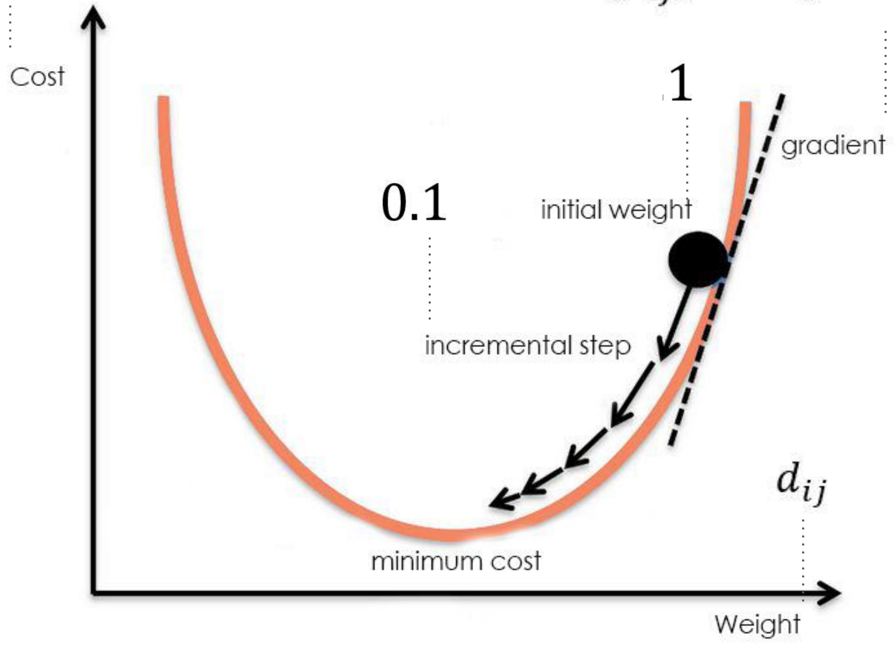
1. Start with some random values for d_{ij} (eg $d_{ij}=1$)
2. Change d_{ij} to reduce $U(d_{ij})$: → Partial derivative

   $d^{t+1}_{ij} = d^{t}_{ij} - 0.1 * U'_{ij}$

   learning rate

3. Repeat until convergence:

   d^{t+1}_{ij} similar to d^{t}_{ij}



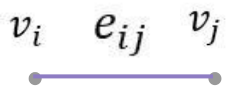
Source: https://blog.clairvoyantsoft.com/the-ascent-of-gradient-descent-23356390836f

Force directed graph drawing

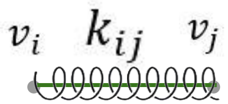
Input:

1.vertex v_i v_j

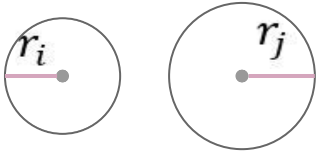
2. edge e_{ij}



3. stiffness k_{ij}

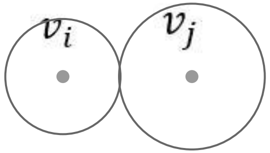


4. radius r_i r_j



Output:

1.kissing-disc drawing



Source: P. Nourian and S. Azadi, "Dynamic Relaxation & Force-Directed Graph Drawing Configraphics: Graph Theoretical Methods for Design and Analysis of Spatial Configurations" DOI: 10.13140/RG.2.2.10147.09761

Force directed graph drawing

Step 1: Compute **distance**: $d_{ij} = v_i - v_j$

Step 2: Compute **attractive force**: $F_{ij} = k_{ij} * d_{ij}$

Step 3: Check **condition**: $if \quad \frac{d_{ij}}{r_i + r_j} - 1 < 0.0001$

Step 4: Compute **repulsive force**:

if true: $R_{ij} = 0$

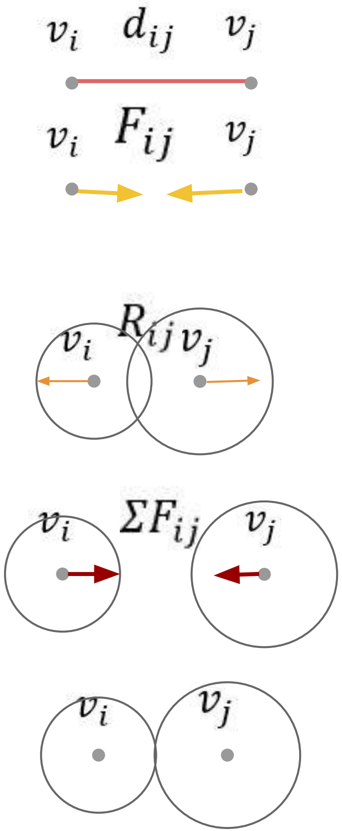
If false: $R_{ij} = -F_{ij}$

Step 5: Compute **resultant force**: $\Sigma F_{ij} = F_{ij} + R_{ij}$

Step 6: Update **position**: $v^{t+1}{}_i = v^t{}_i + 0.1 * \Sigma F_{ij}$

Step 7: Compute **count**: $count = count + 1$

Until: $condition == false \ or \ count > maximum$



Source: P. Nourian and S. Azadi, "Dynamic Relaxation & Force-Directed Graph Drawing Configraphics: Graph Theoretical Methods for Design and Analysis of Spatial Configurations" DOI: 10.13140/RG.2.2.10147.09761

Tools

- 2D drawing → Rhinoceros
- Visual programming → Grasshopper
- Spring simulation → Kangaroo
- Illuminance analysis → Honeybee
- Weather analysis → Ladybug
- Programming → GhPython

01
Research formulation

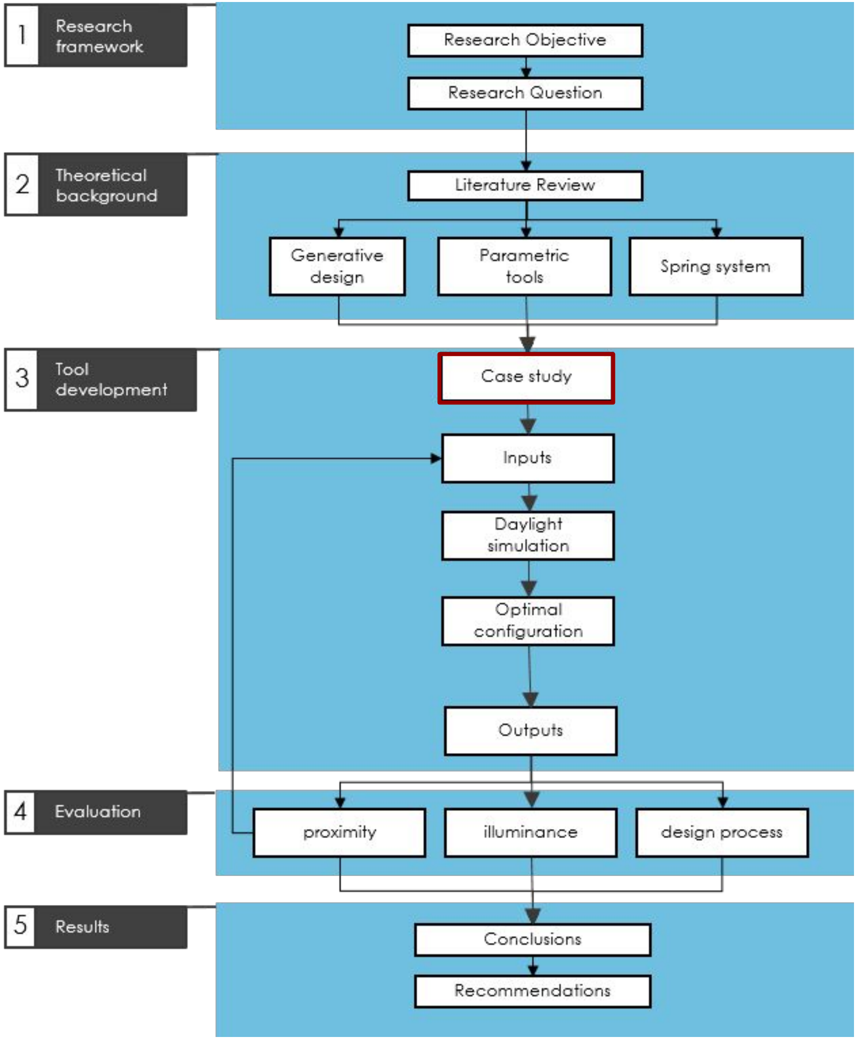
02
Tool development

03
Layout development

04
Evaluation

05
Conclusions

Research methodology



Case study: EEMC

EEMC: Faculty of Electrical Engineering, Mathematics & Computer science

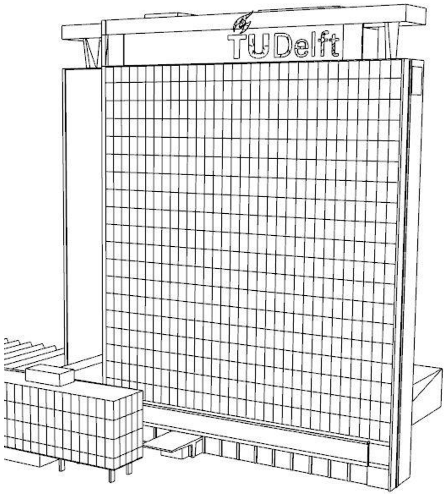
- Landmark of TU Delft
- Facade replacement
- Student housing

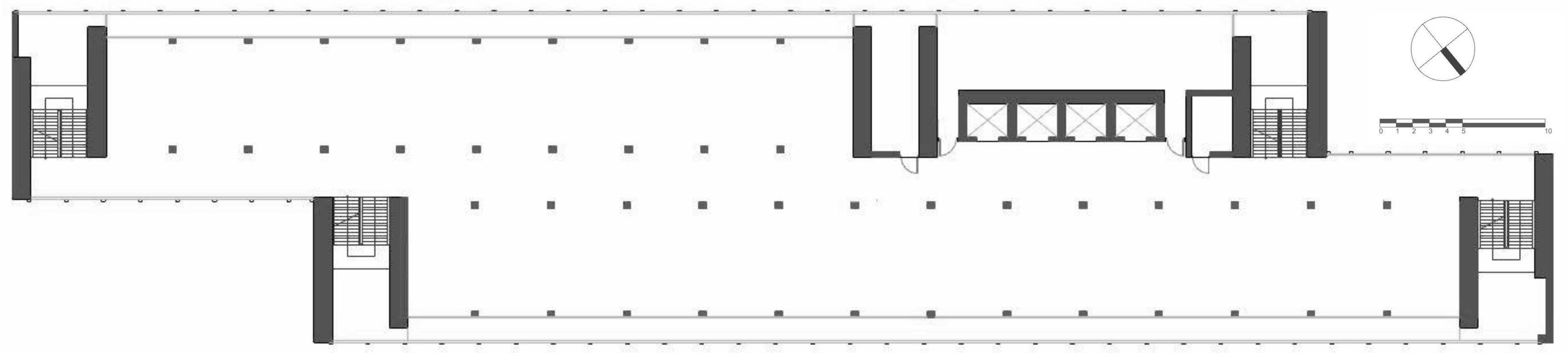
Source: Behoud het markante EWI-gebouw, voor Delft, Nederland en de rest van de wereld - Petities.nl

Design assignment

Architectural competition:
Convert tower of EEMC into student housing

500 students
4000m^2 common facilities



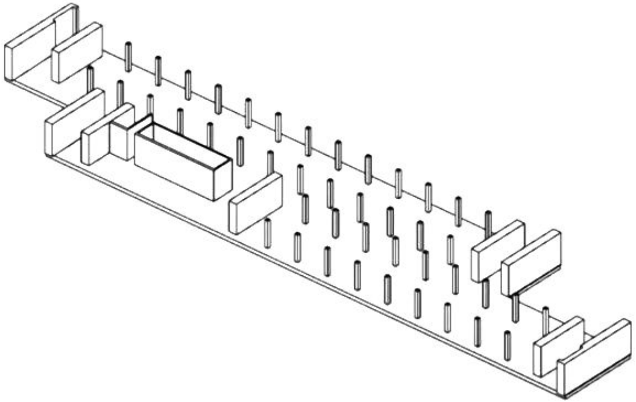


Source: Case study EWI: low rise facade details - AR1A075 - TU Delft - StuDocu

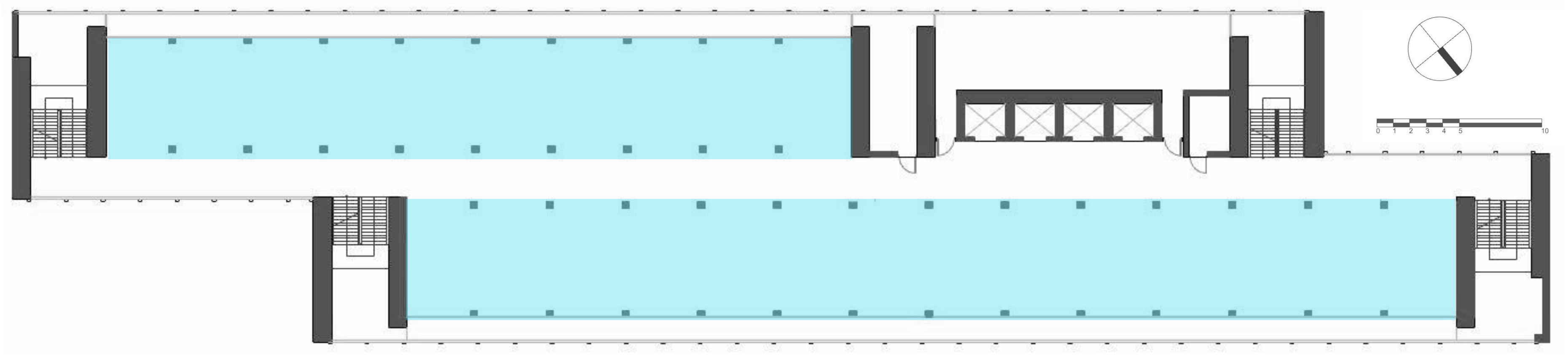
Design assignment

Architectural competition:
Convert tower of EEMC into student housing

500 students
$4000m^2$ facilities
one floor area: 1035 m^2
area of interest: 746 m^2



Source: Delft Seminars of Building Technology - group 10-
User controlled EWI



Source: Case study EWI: low rise facade details - AR1A075 - TU Delft - StuDocu

Program of requirements

Unit A: Studio

room	area (m2)
kitchen	2
coach	3
bed	2
restroom	3.75
desk	2.25
storage	2
corridor	3
tot. area	23

Unit B: Shared apartment

room	area (m2)
kitchen	2
living room	5
bedroom 1	7.5
bedroom 2	7.5
shower	2.25
restroom	2.25
dining room	3
storage	2.25
hall	3
tot. area	49

Unit C: Common facilities

room	area (m2)
kitchen	8.75
living room	100
leisure room	48.5
study room	171
workout area	90
WC	8.25
dining room	12.5
laundry	14
corridor	73
tot. area	858

Unit D: Common facilities (ground floor)

room	area (m2)
reception	50
living room	125
hall	45
lecture room	190
restaurant	150
WC	40
café	95
storage	60
cinema	245
entrance	5
offices	180
tot. area	2960

Vertical distribution

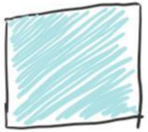
Floortype A

Unit A

Studio ($23m^2$)
24 studios/floor

Floortype B

Unit B



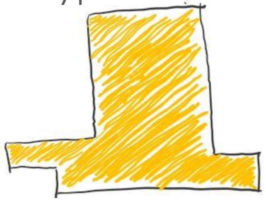
Shared apartment ($49m^2$)
12 apartments/floor

Floortype C

Unit C

Common facilities ($858m^2$)

Floortype D



Unit D

Common facilities ($2960m^2$)

floor	use
23	common
22	shared
21	studio
20	studio
19	studio
18	shared
17	shared
16	studio
15	studio
14	studio
13	studio
12	shared
11	shared
10	studio
9	studio
8	studio
7	studio
6	shared
5	shared
4	studio
3	studio
2	studio
1	common
0	common

Tool development

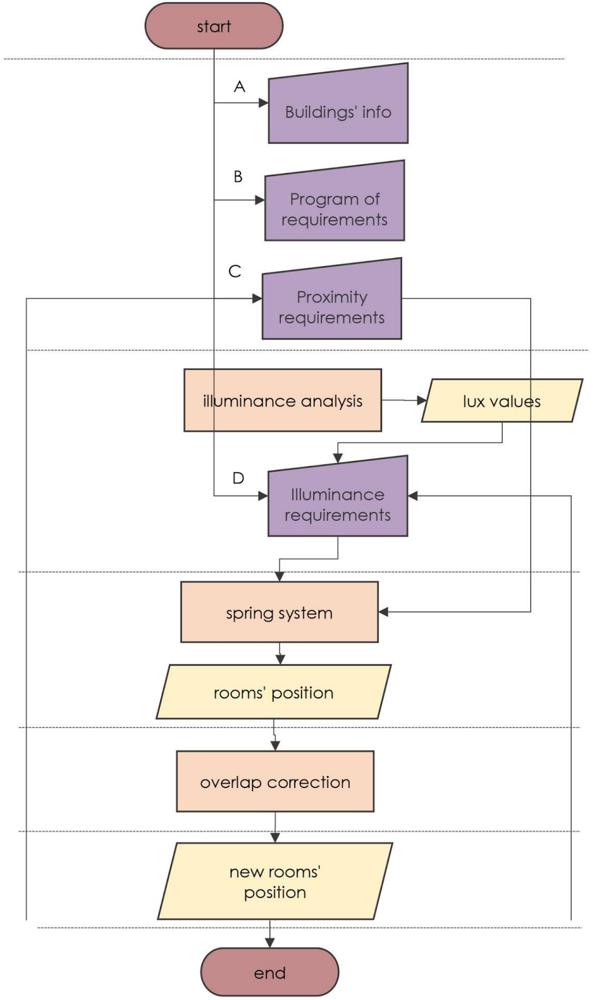
1. Inputs

2. Daylight simulation

3. Optimal configuration

4. Overlap

5. Outputs



1. Inputs: building's info & PoR

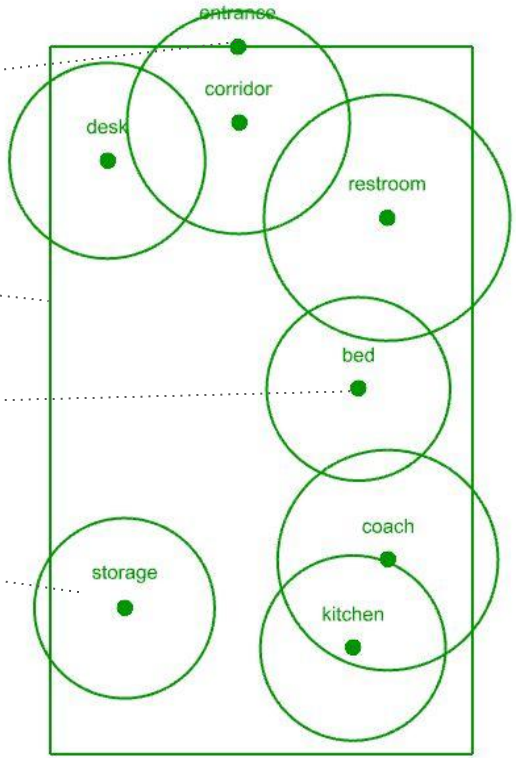
Buildings info

- entrance's position
- boundary

Program of requirements

- rooms' centroids
- rooms' areas

Unit A: Studio

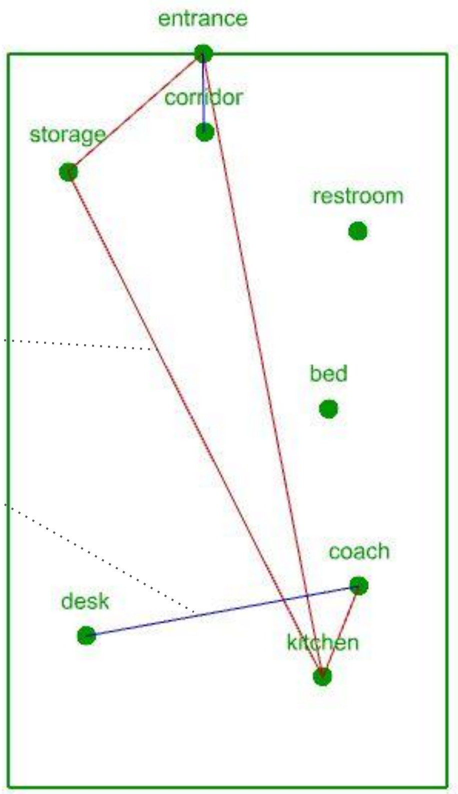


1. Inputs: proximity requirements

Hierarchy of connections

Category	Prox. factor	Connection
0	0.0	No
1	0.3	Weak
2	0.6	Medium
3	0.9	Strong

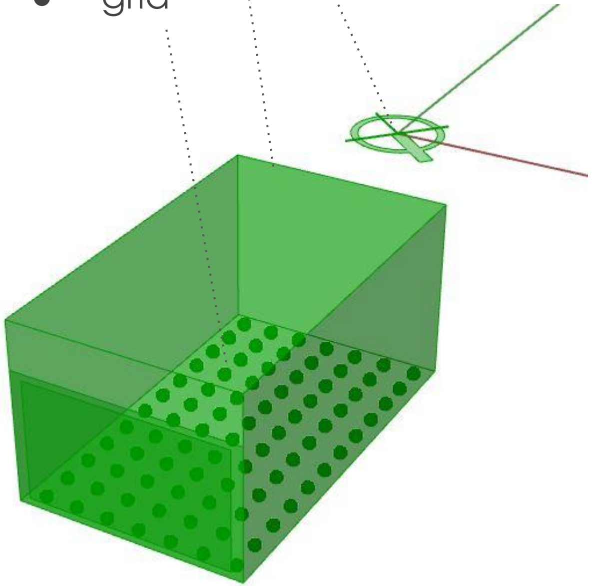
- strong connections
- medium connections



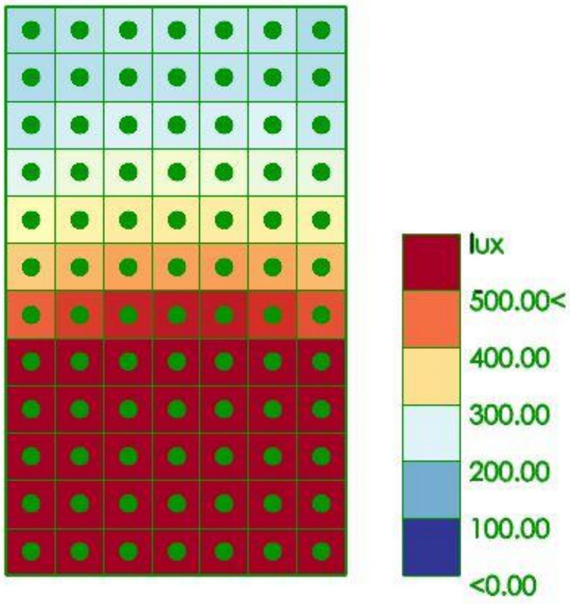
2. Daylight simulation

Inputs

- orientation
- geometry
- grid

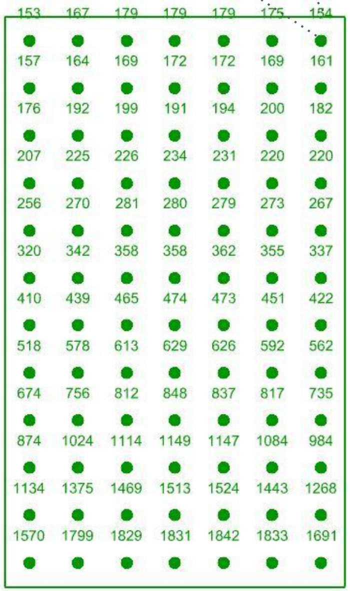


illuminance analysis

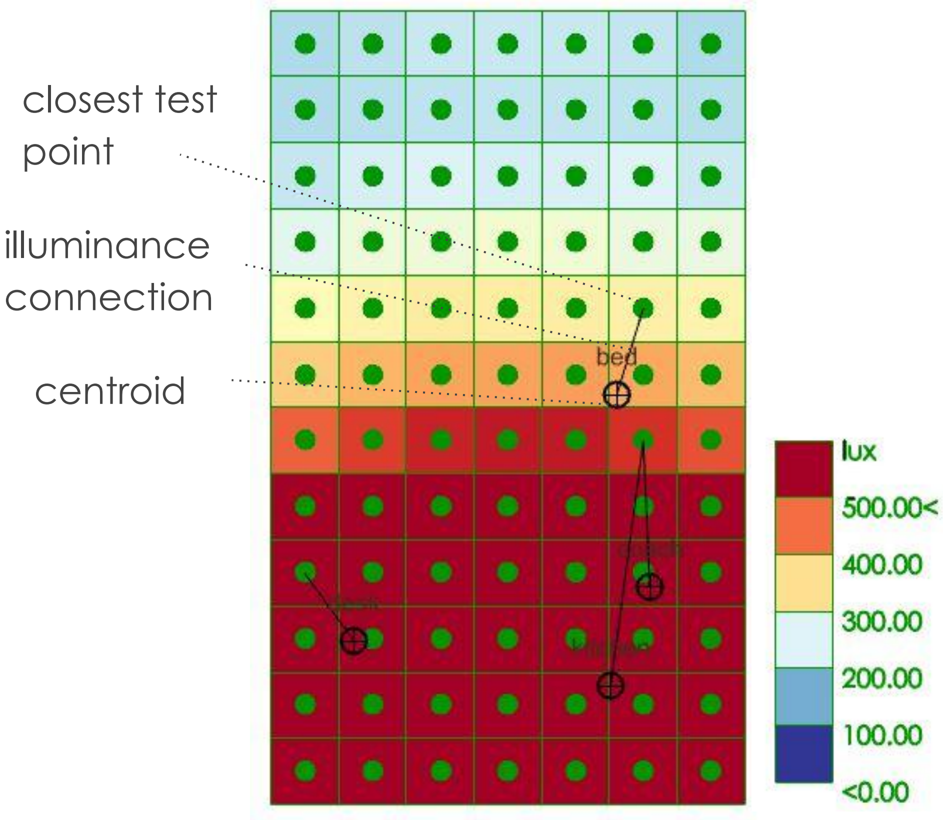


Outputs

- test points
- lux values



1. Inputs: illuminance requirements



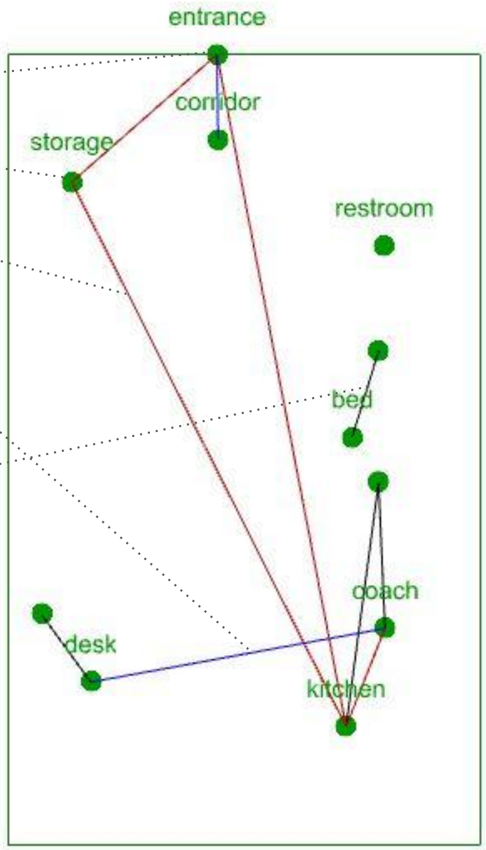
Illuminance recommendations

Category	min lux	max lux
1	100	300
2	300	500
3	500	750

3. Optimal configuration

Inputs

- Fixed points
- Points
- Strong proximity connections
- Medium proximity connections
- Illuminance connections



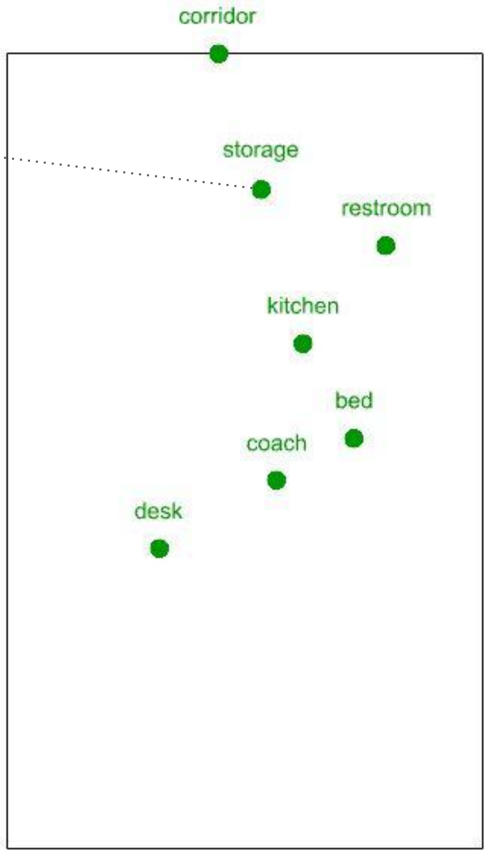
stiffness k_{ij}

type	stiffness
Strong connections	0.9
Medium connections	0.6
Illuminance connections	0.9

3. Optimal configuration

Output

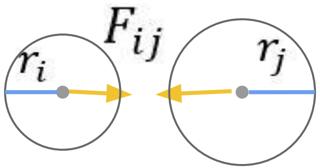
- New points' position



stiffness k_{ij}

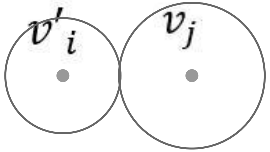
attractive force:

$$F_{ij} = k_{ij} * d_{ij}$$



move vertex:

$$v^{t+1}{}_i = v^t{}_i + 0.1 * \Sigma F_{ij}$$

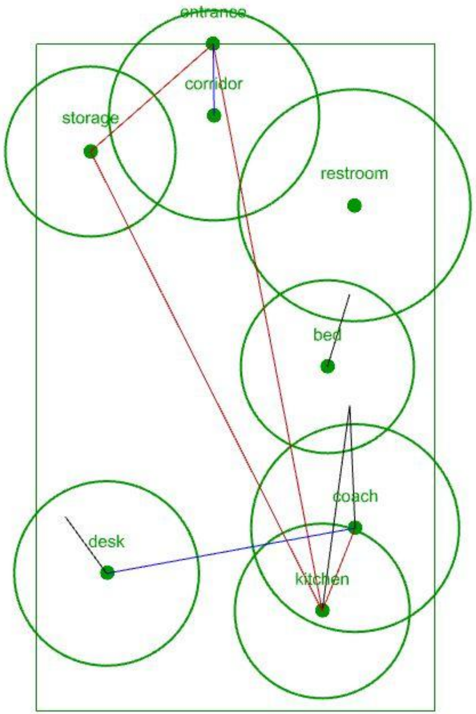


Converged when:

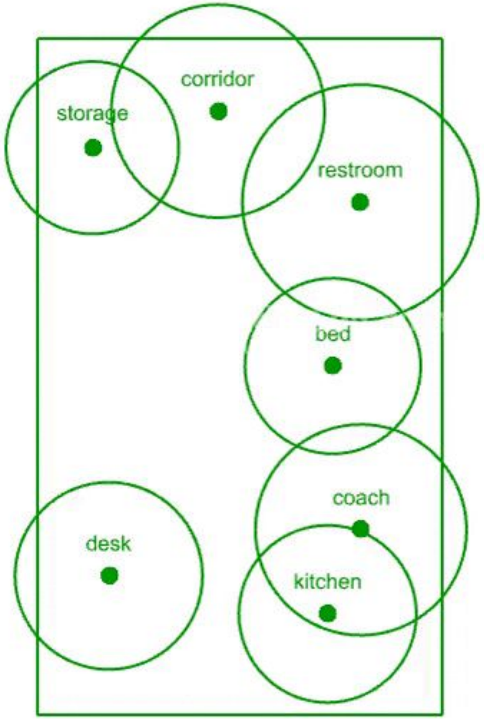
$$\left| \frac{d_{ij}}{r_i + r_j} \right| - 1 < 0.0001$$

3. Optimal configuration

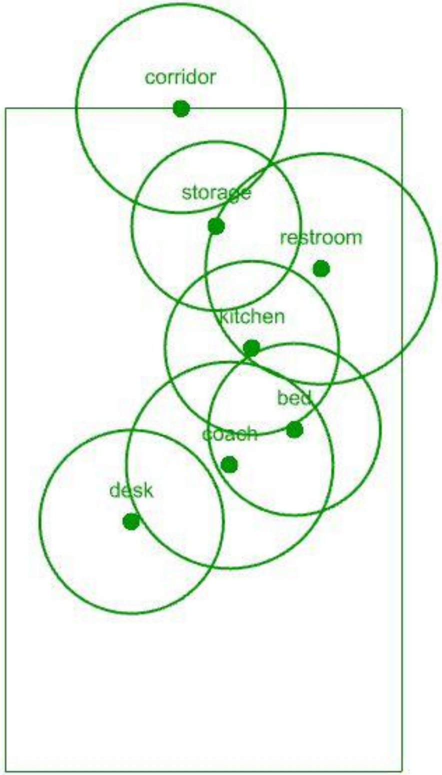
System's input



Spring system

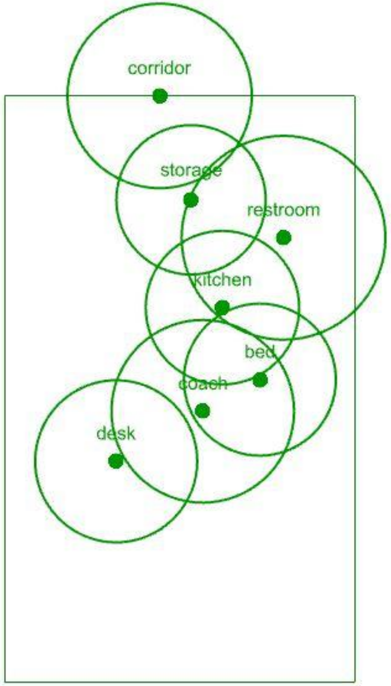


System's output

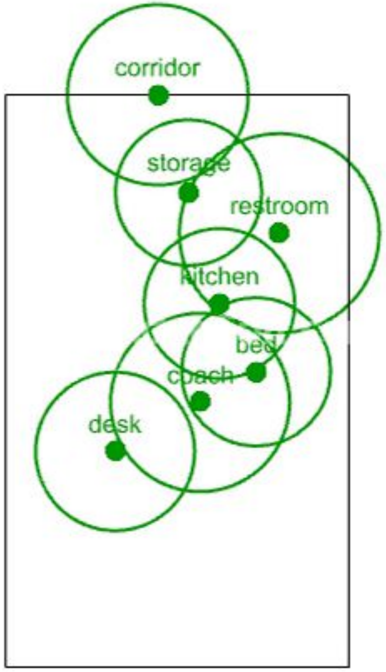


4. Overlap correction

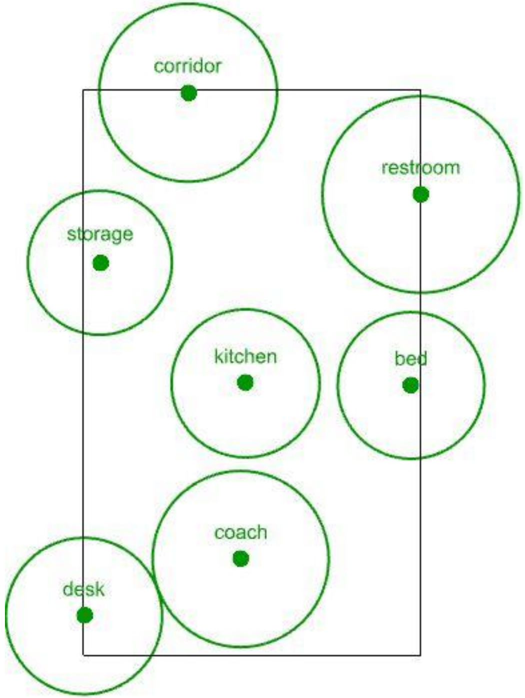
System's output



Move points

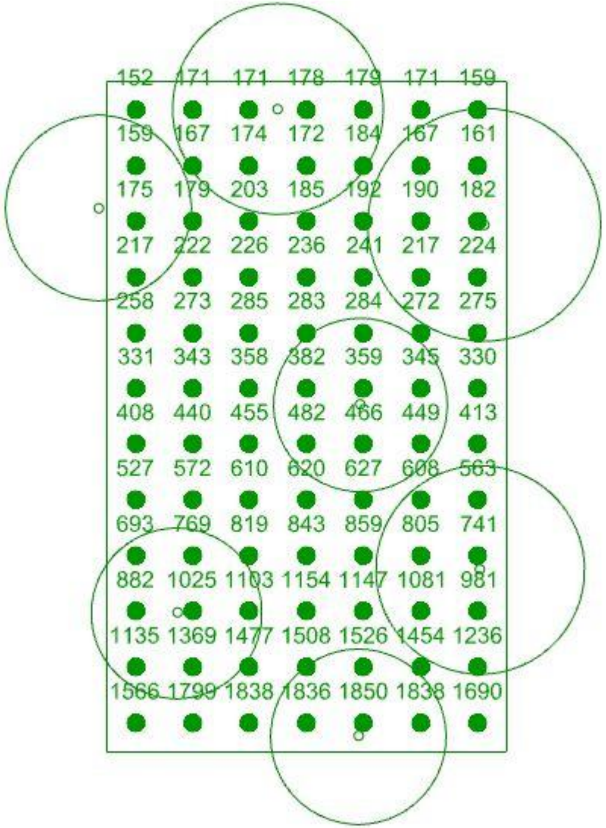


New positions

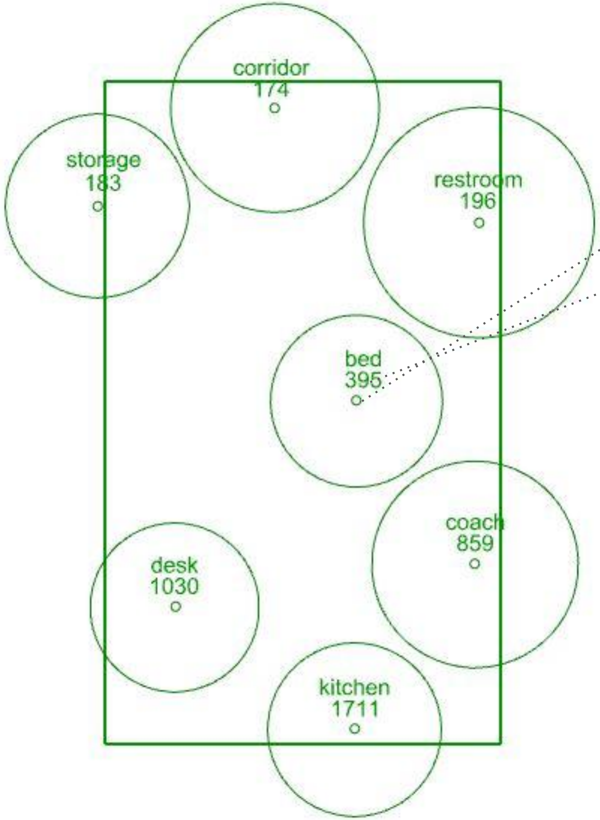


7. Outputs

Mean lux calculation



Bubble diagram



Output

- rooms' position
- mean lux values

01
Research formulation

02
Tool development

03
Layout development

04
Evaluation

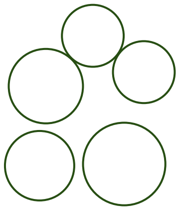
05
Conclusions

Layout development

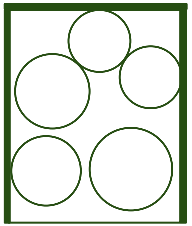
From bubble diagram to layout

1. Insert **bubble diagram**
2. Insert **existing walls**
3. Insert **desired walls**
4. Convert circles into **rectangles**
5. Move **closed rooms** towards inner corners
6. Move **rest rooms** beneath closed rooms
7. Insert **furniture** clusters
8. **Layout**

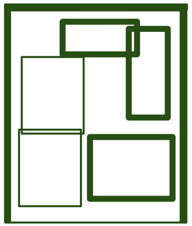
1



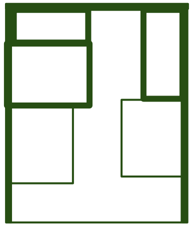
2



3,4

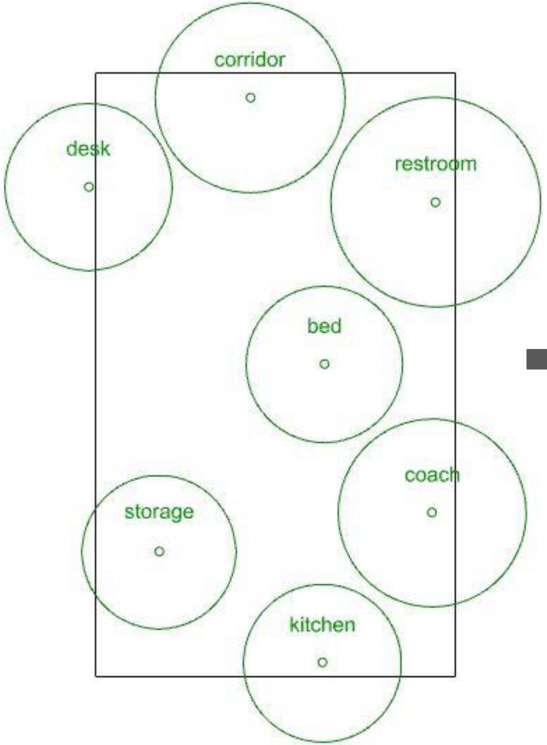


4,5

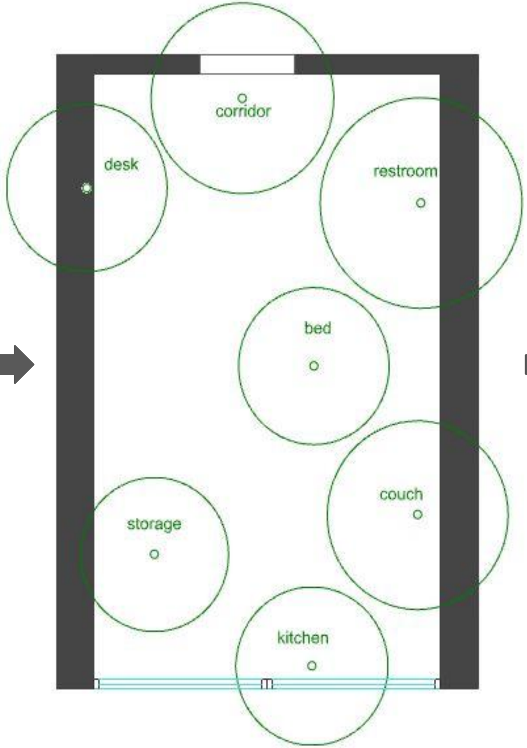


Design development: Unit A

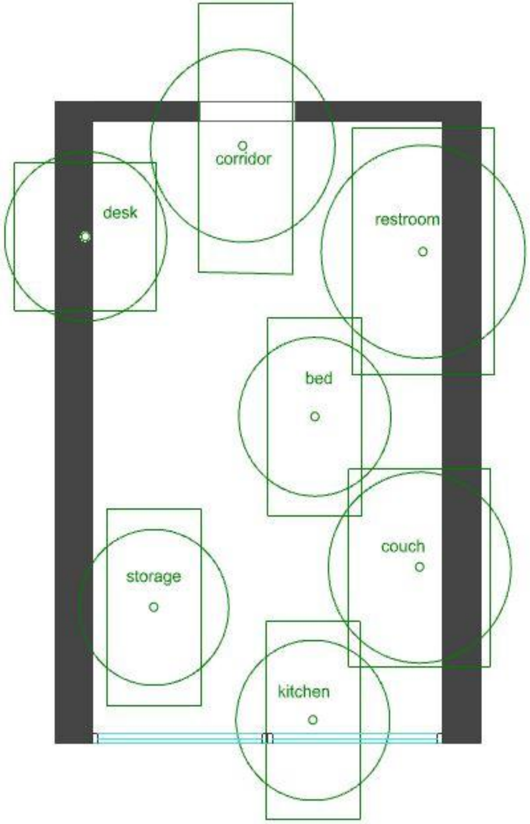
1.Bubble diagram



2. Exterior walls



3. Dimensions



Design development: Unit A

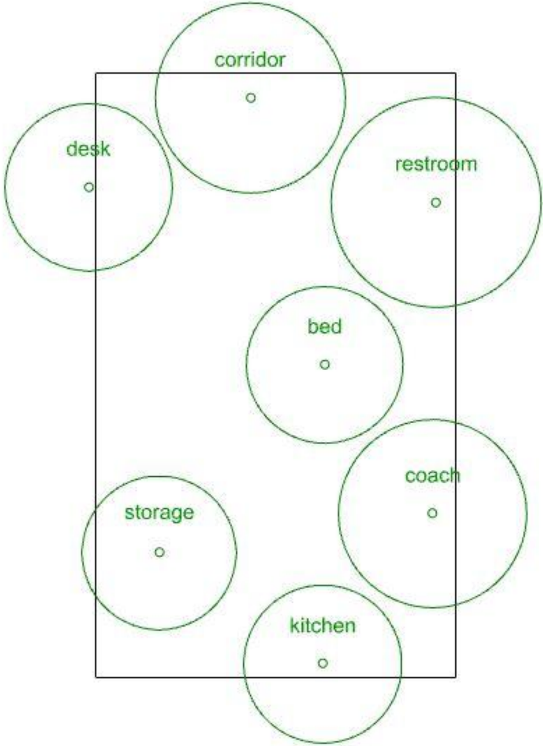
4. Interior walls

5. Alignment

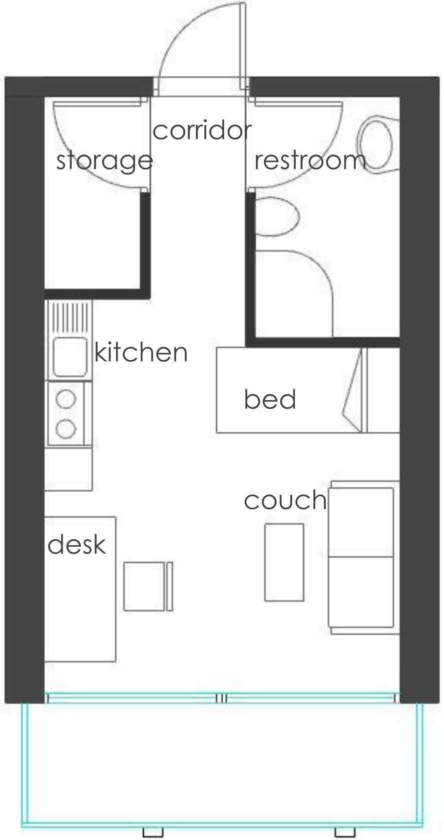
6. Furniture

Design development: Unit A

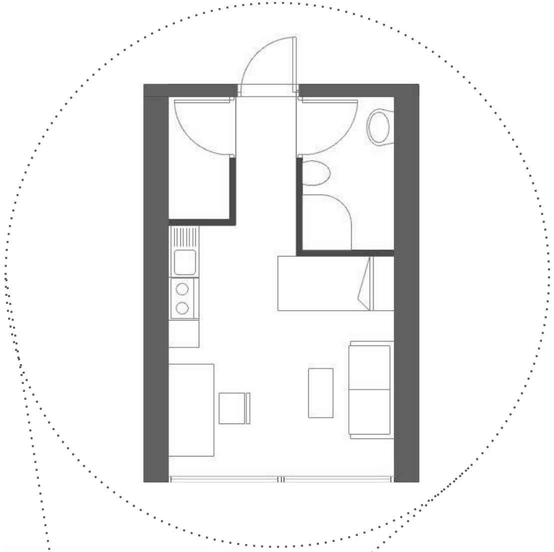
1.Bubble diagram



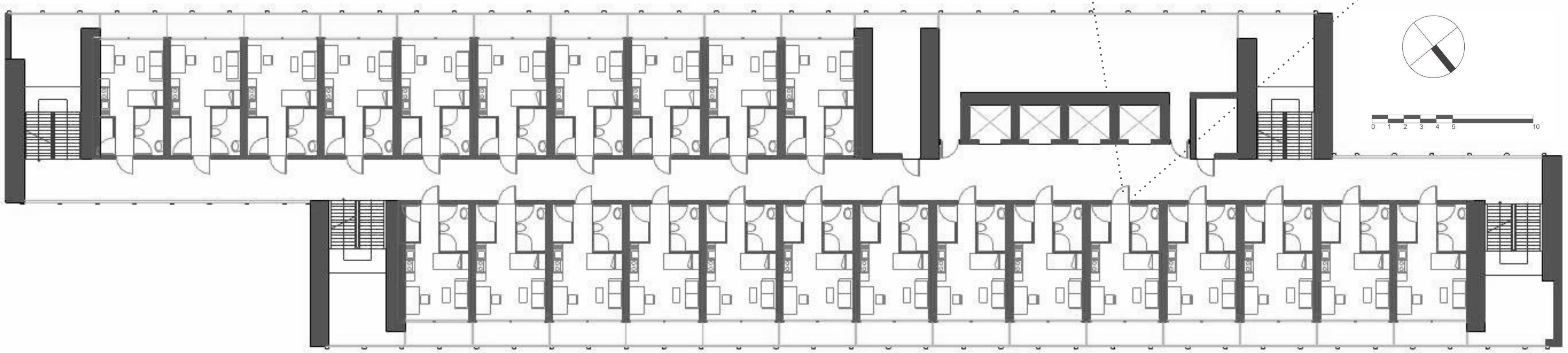
7. Residential layout



Design proposal: Floortype A

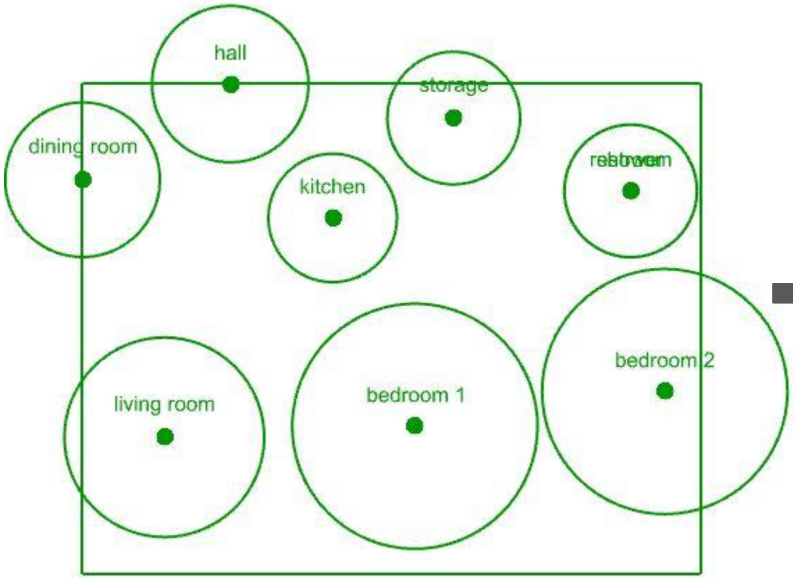


24 studios
23 m^2



Design development: Unit B

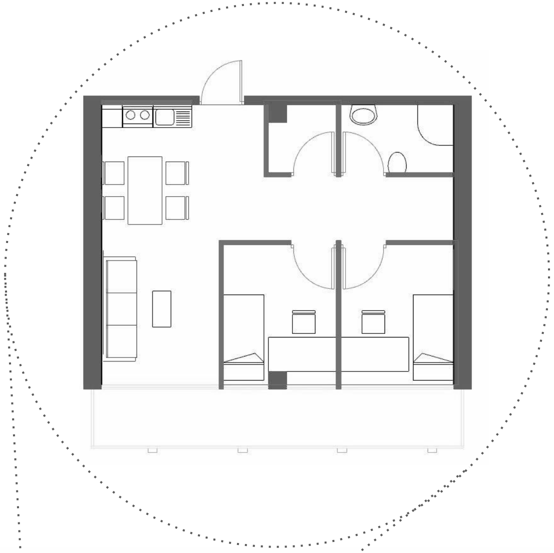
1.Bubble diagram



7. Residential layout

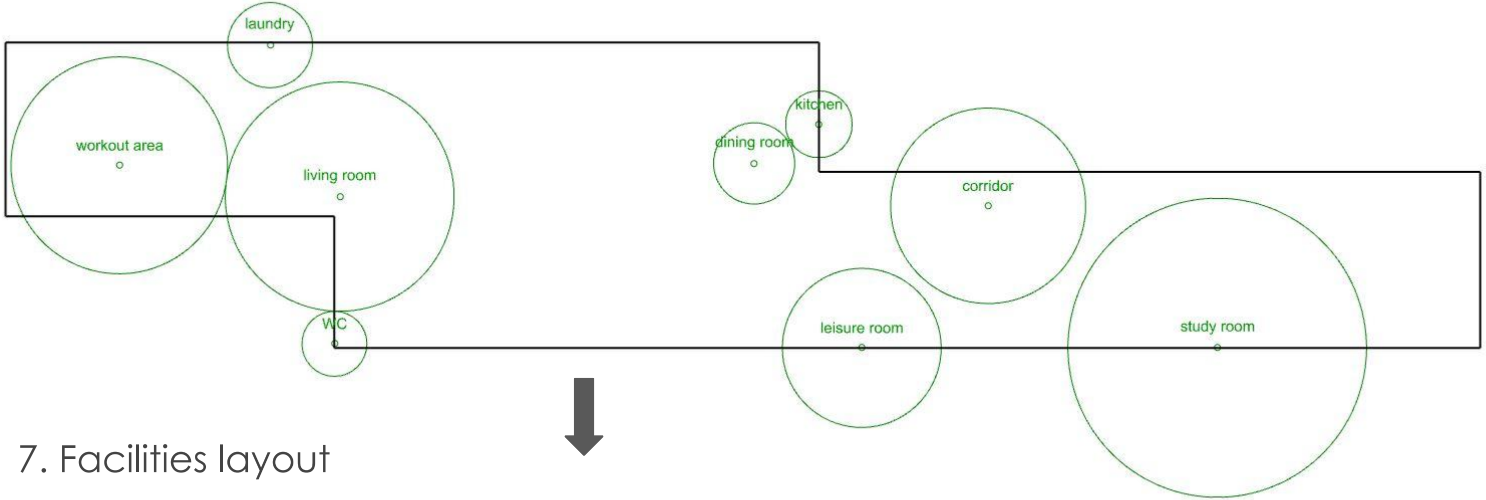
Design proposal: Floortype B

12 shared apartments
49 m^2

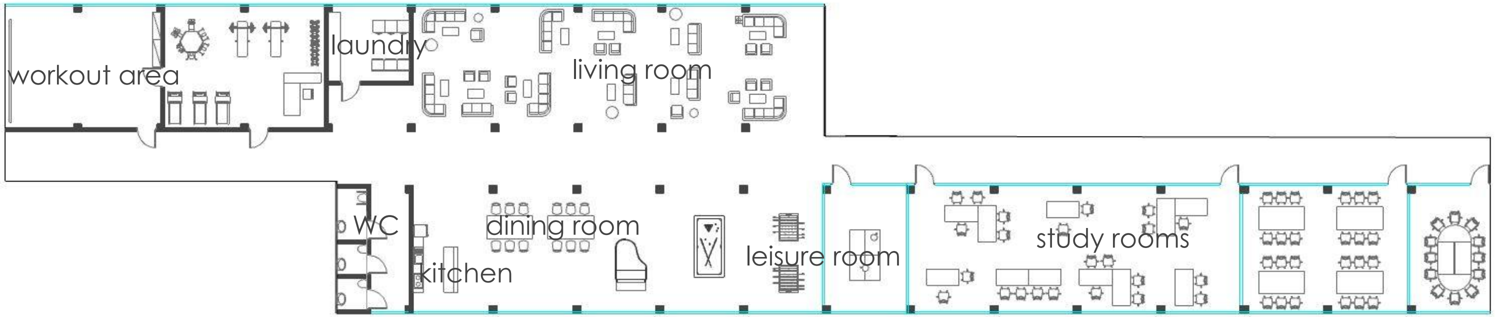


Design development: Unit C

1.Bubble diagram

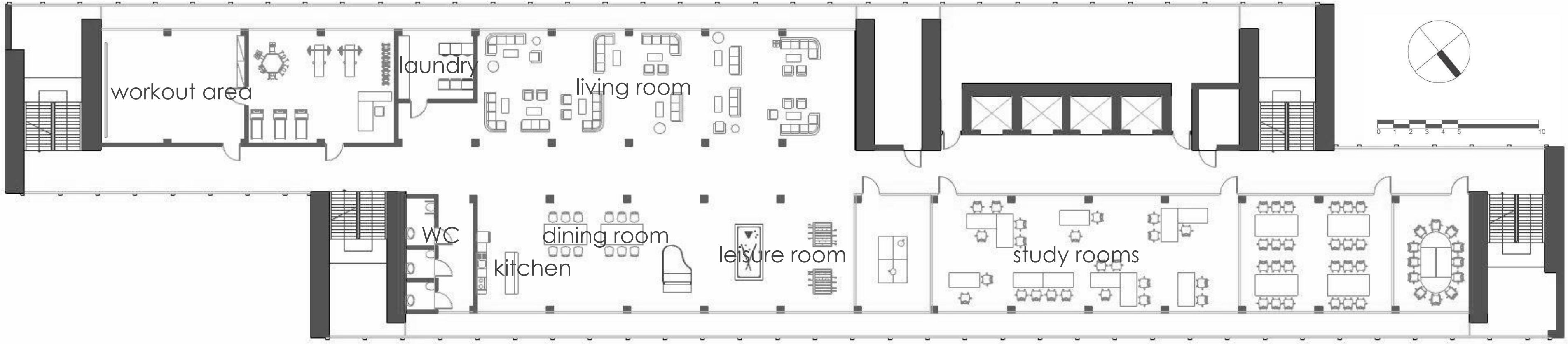


7. Facilities layout



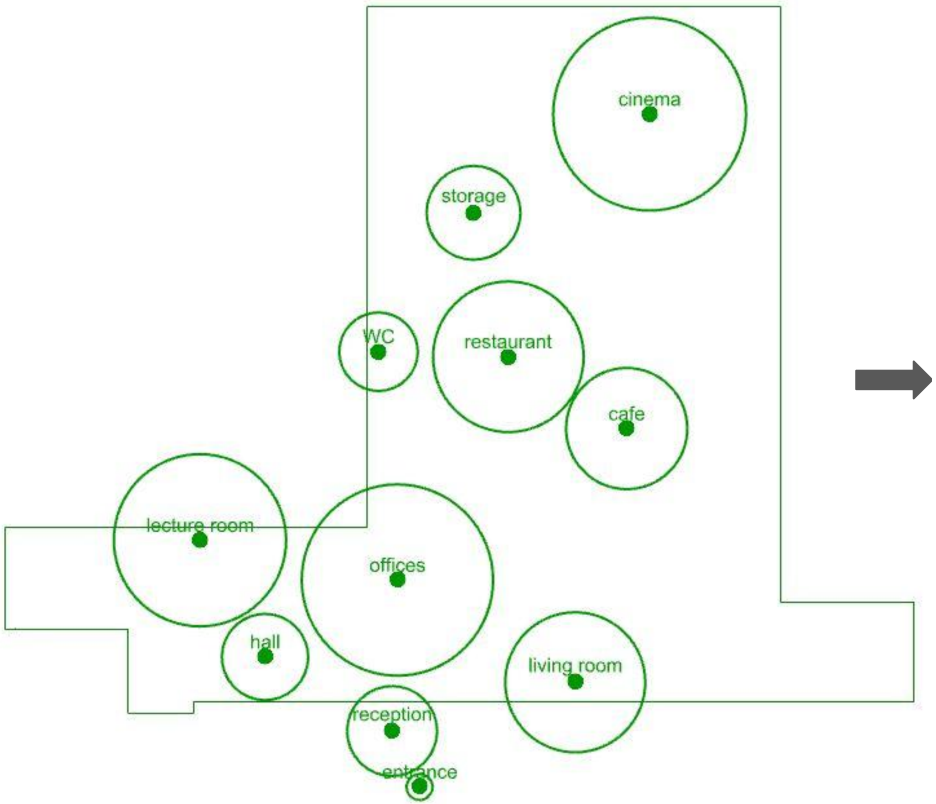
Design proposal: Floortype C

Common facilities
858 m^2

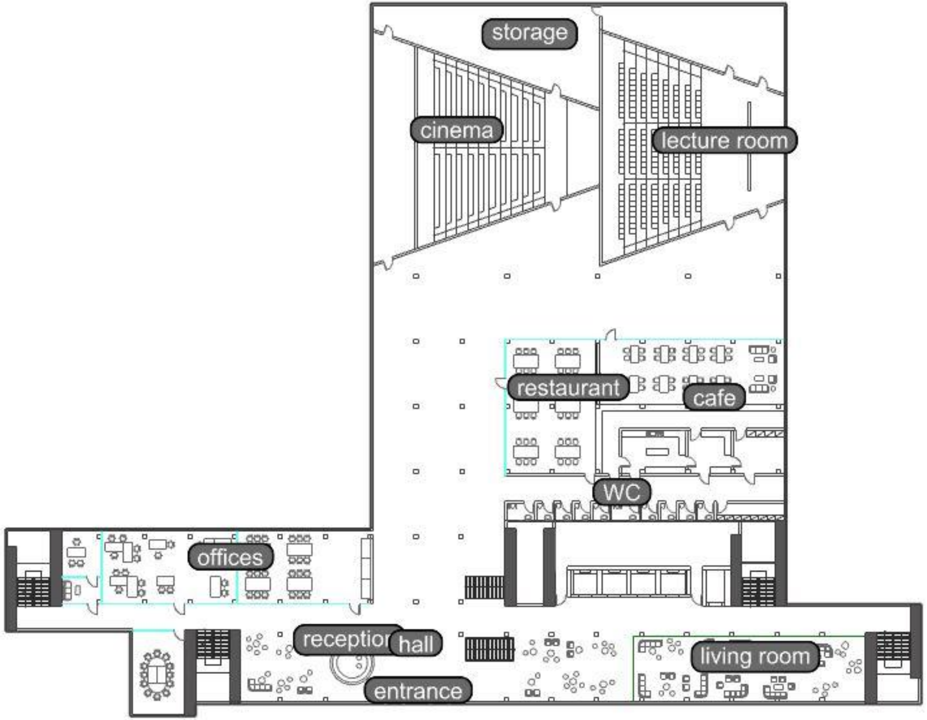


Design development: Unit D

1.Bubble diagram

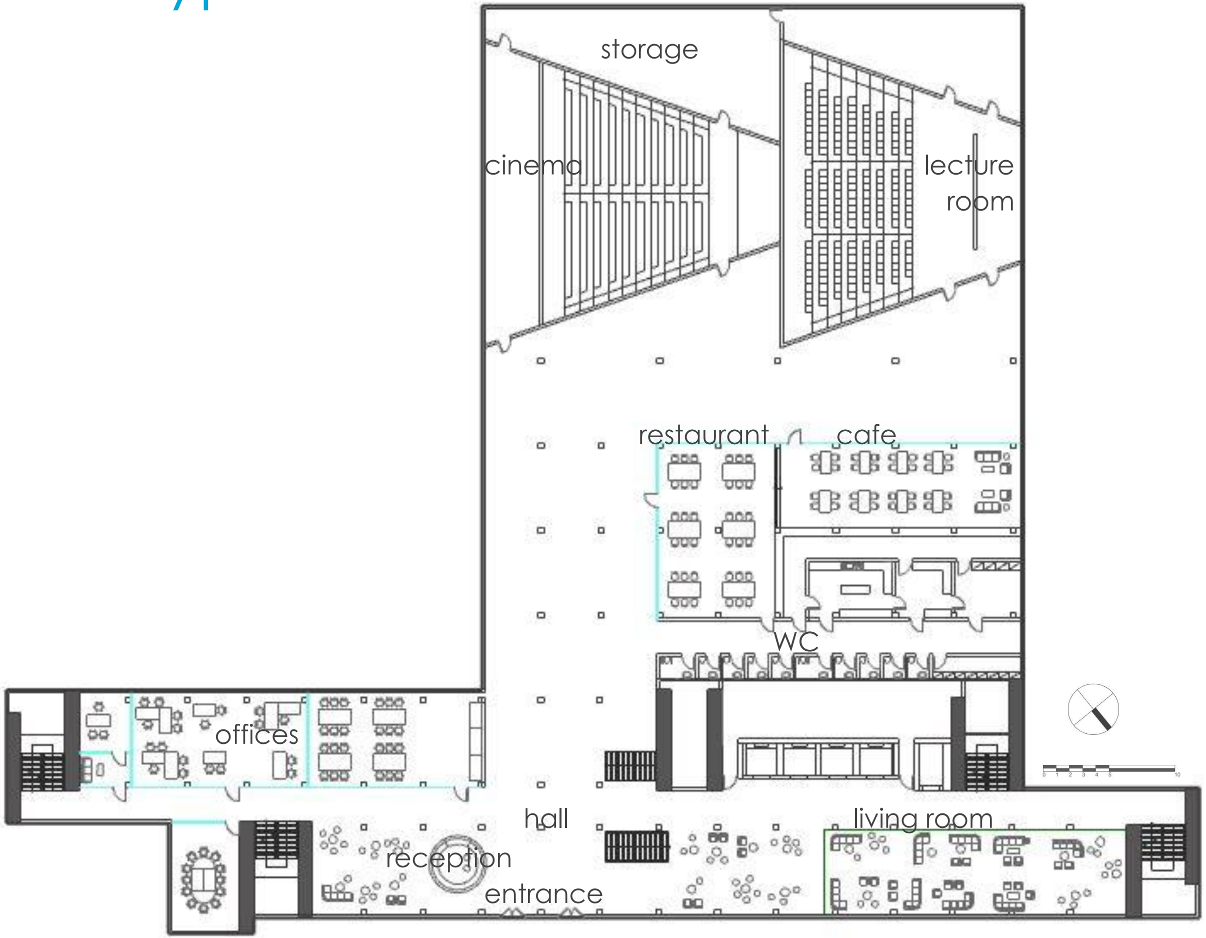


7. Facilities layout



Design proposal: Floortype D

Common facilities
2960 m^2



01
Research formulation

02
Tool development

03
Layout development

04
Evaluation

05
Conclusions

Evaluation of the tool: proximity

Condition:

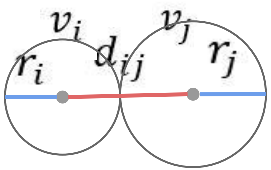
$$\frac{d_{ij}}{r_i + r_j} < t$$

$for\ t = 1.5$

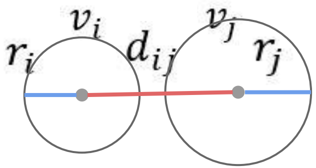
Hierarchy of connections

Category	Prox. factor	Connection	Tolerance
1	0.3	Weak	$4 * t$
2	0.6	Medium	$2 * t$
3	0.9	Strong	t

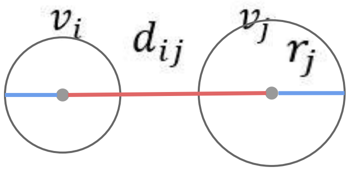
strong connection

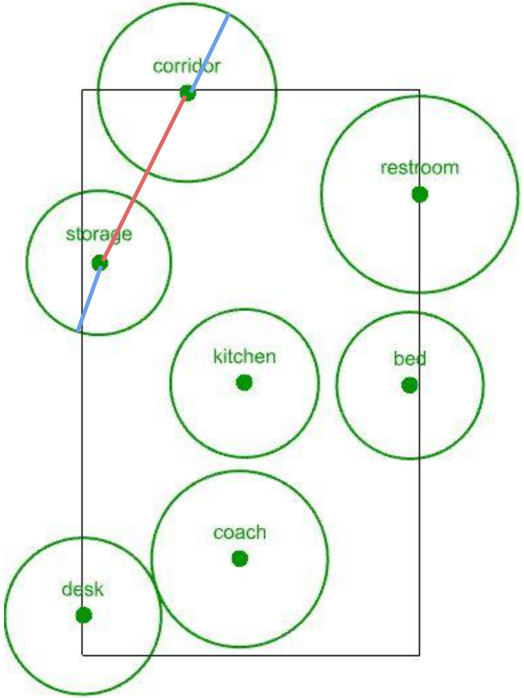


medium connection



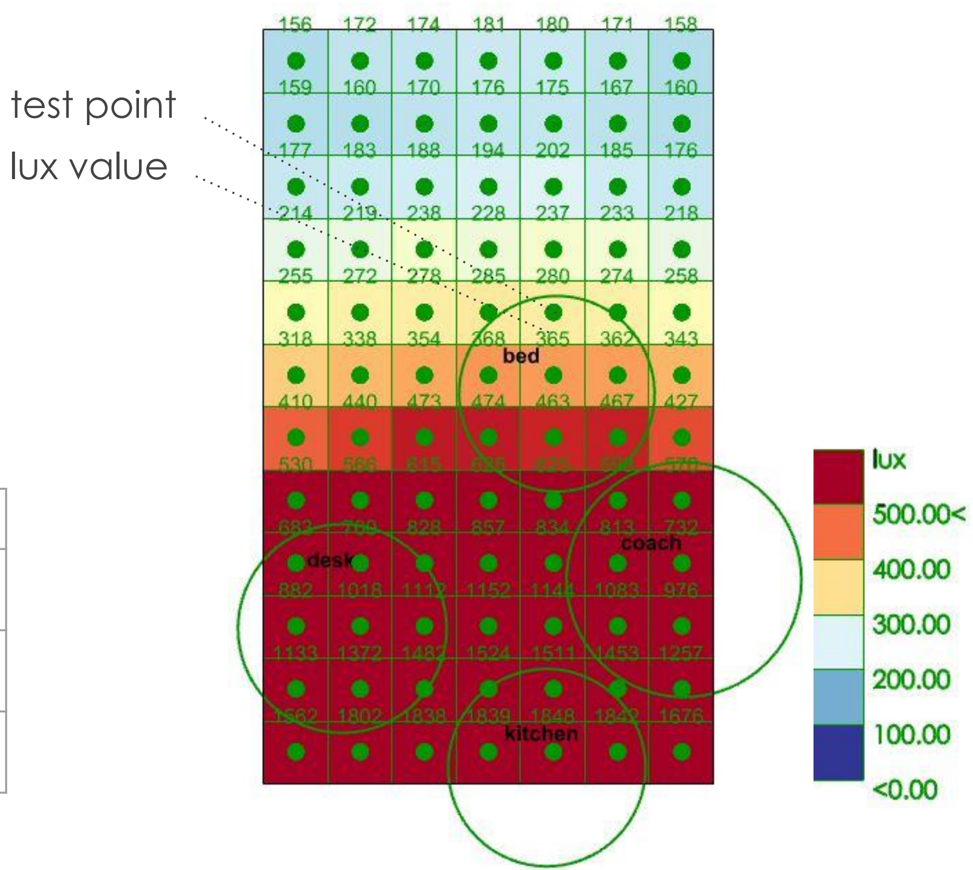
weak connection





Evaluation of the tool: illuminance

$$mean = \frac{\sum lux}{n}$$



Illuminance categorization

Category	Min lux	Max lux
1	100	300
2	300	500
3	500	max

Evaluation of the design process

Inputs

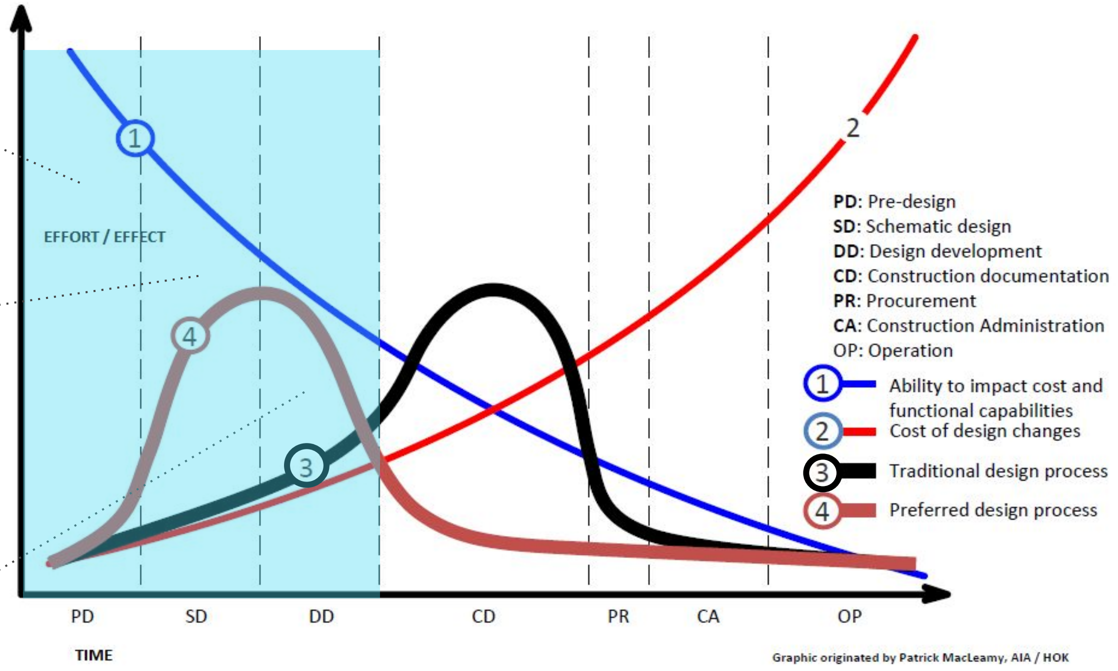
- building's info
- rooms
- areas

Bubble diagram (computational)

- proximity
- illuminance
- prioritization
- rooms' positions

Layout (manual)

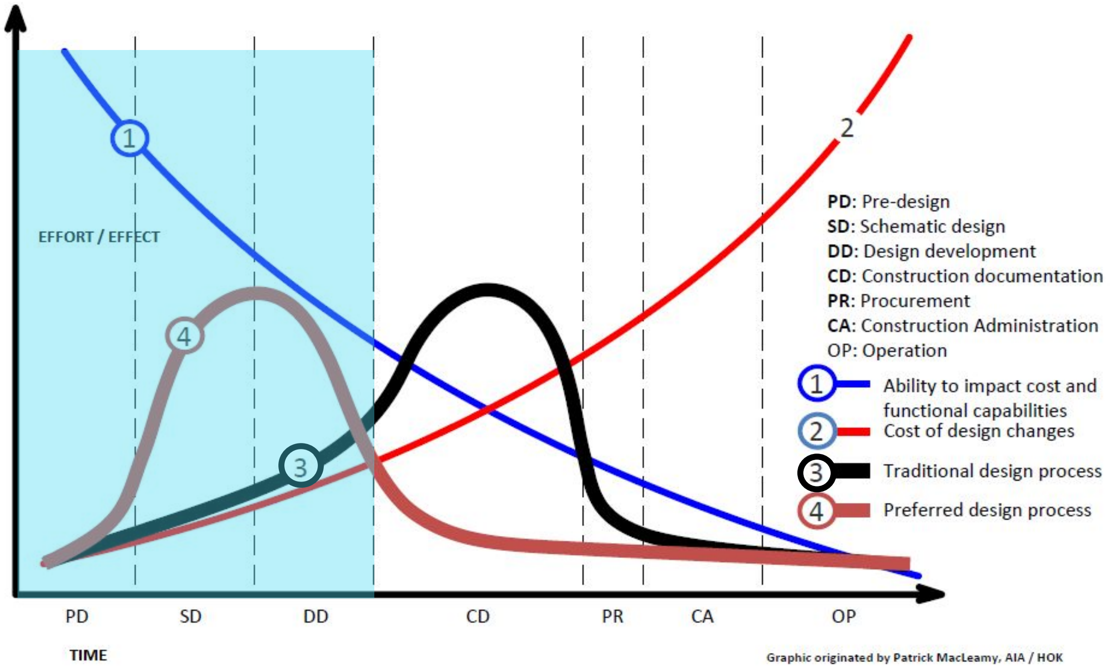
- walls
- dimensions
- furniture



Source: http://division4triclinium.blogspot.com/2013/06/of-macleamy-curve-efficient-design-and.html

Comparison with conventional process

criteria	conventional	proposed
time effort		3x less
decision making		prioritization
bubble diagram	slightly better	
layout	better	



Source: http://division4triclinium.blogspot.com/2013/06/of-macleamy-curve-efficient-design-and.html

Limitations

- The tool considers a limited amount of **design criteria.**
- If bubble diagram is not satisfactory the process has to be **repeated**.
- The tool ignores the existence of **obstacles.**
- The tool is designed for **2D drawings.**

01
Research formulation

02
Tool development

03
Layout development

04
Evaluation

05
Conclusions

Conclusions: Answer to subquestions

1 How to find optimal configuration regarding **proximity** requirements?

- **Spring system** approach proved to be **successful**.
- **Hierarchy** of requirements can be expressed by using different **stiffness** values.
- It is an **intuitive approach**.

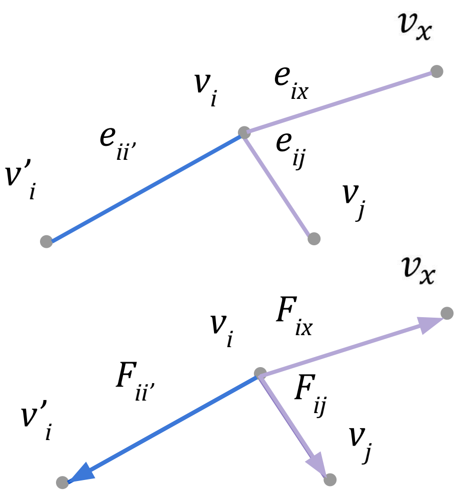
2 How to find optimal configuration regarding **illuminance** requirements?

- Illuminance requirements can be treated as **proximity requirements**.
- An **illuminance analysis** is needed.

Conclusions: Answer to subquestions

3 How to **combine** proximity and illuminance preferences?

- The **hierarchy** of proximity and illuminance connections is adjusted with stiffness values.
- In case illuminance has to have the **same importance** as proximity the sum of the stiffness values for proximity has to be equal to the stiffness value for illuminance.
  Example:



$$\Sigma F_i = F_{ij} + F_{ii'} + F_{ix}$$

$$\Sigma F_i = k_{ij} * d_{ii'} + k_{ii'} * d_{ii'} + + k_{ix} * d_{ix}$$

	v_i	v_x	v_y	v'_i
v_i		0.9	0.9	1.8
v_j	0.9		0	0
v_x	0.9	0		0
v'_i	1.8	0	0	

Conclusions: Answer to subquestions

4 Are existing **plugins for Grasshopper** useful?

- The computation **time** is short.
- The capabilities of existing tools are **limited**.
- It is recommended to use a **programming language**:
    - the user can have more **control**,
    - **repetitive** tasks can be dealt more effectively,
    - the tool can have better **compatibility**.

Conclusions: Answer to research question

"To what extent is it possible to convert an **existing layout** into a **residential** one regarding **proximity** relationships and **illuminance** requirements using **computational tools** during primary design stages?"

- To create a **schematic layout** with limited design criteria it is possible to use computational tools.
- The schematic layout is a very important step and it can be used as **guidelines** for a functional layout.
- To design a **functional layout** more design criteria have to be taken into consideration which can be difficult to systemize.

Further recommendations

- extra design criteria (noise, thermal)
- different applications (offices, shops)
- automate manual tasks (during design process)
- development considering z axis (two-storey apartment)
- layout from scratch

Thank you
for your attention