JAGIELLONIAN UNIVERSITY
IN KRAKOW

Evolutionary methods in automatic floor layout generation

Barbara Strug

Jagiellonian University, Institute of Applied Computer Science

AIRA Seminar, 4.04.2024

JAGIELLONIAN UNIVERSITY IN KRAKOW

Outline

1. Problem formulation, motivation and background
2. Evolutionary computations
3. Design representation
4. Evolutionary operators
5. Fitness evaluation
6. Examples and results
7. Open problems - future work

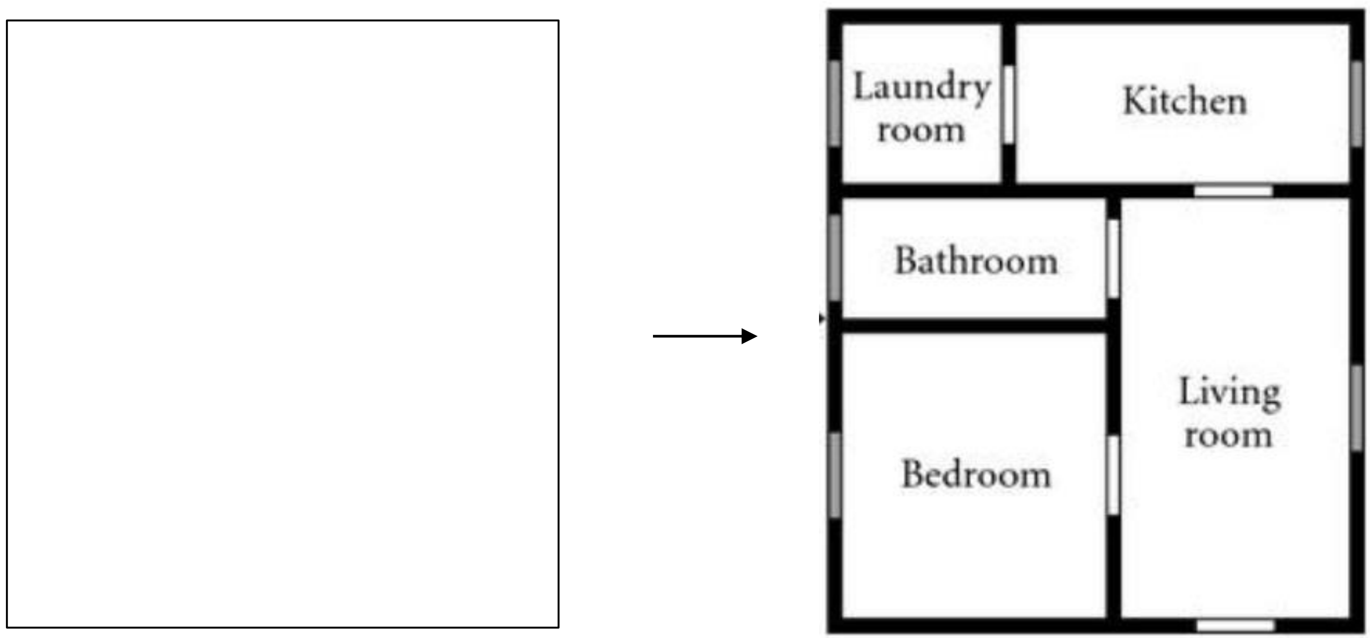
Based mainly on:

Barbara Strug, Ewa Grabska, Grażyna Ślusarczyk: Supporting the design process with hypergraph genetic operators. Adv. Eng. Informatics 28(1): 11-27 (2014)

Grzesiak-Kopeć, Katarzyna, Barbara Strug, and Grażyna Ślusarczyk. 2021. "Evolutionary Methods in House Floor Plan Design" Applied Sciences 11, no. 17: 8229. https://doi.org/10.3390/app11178229

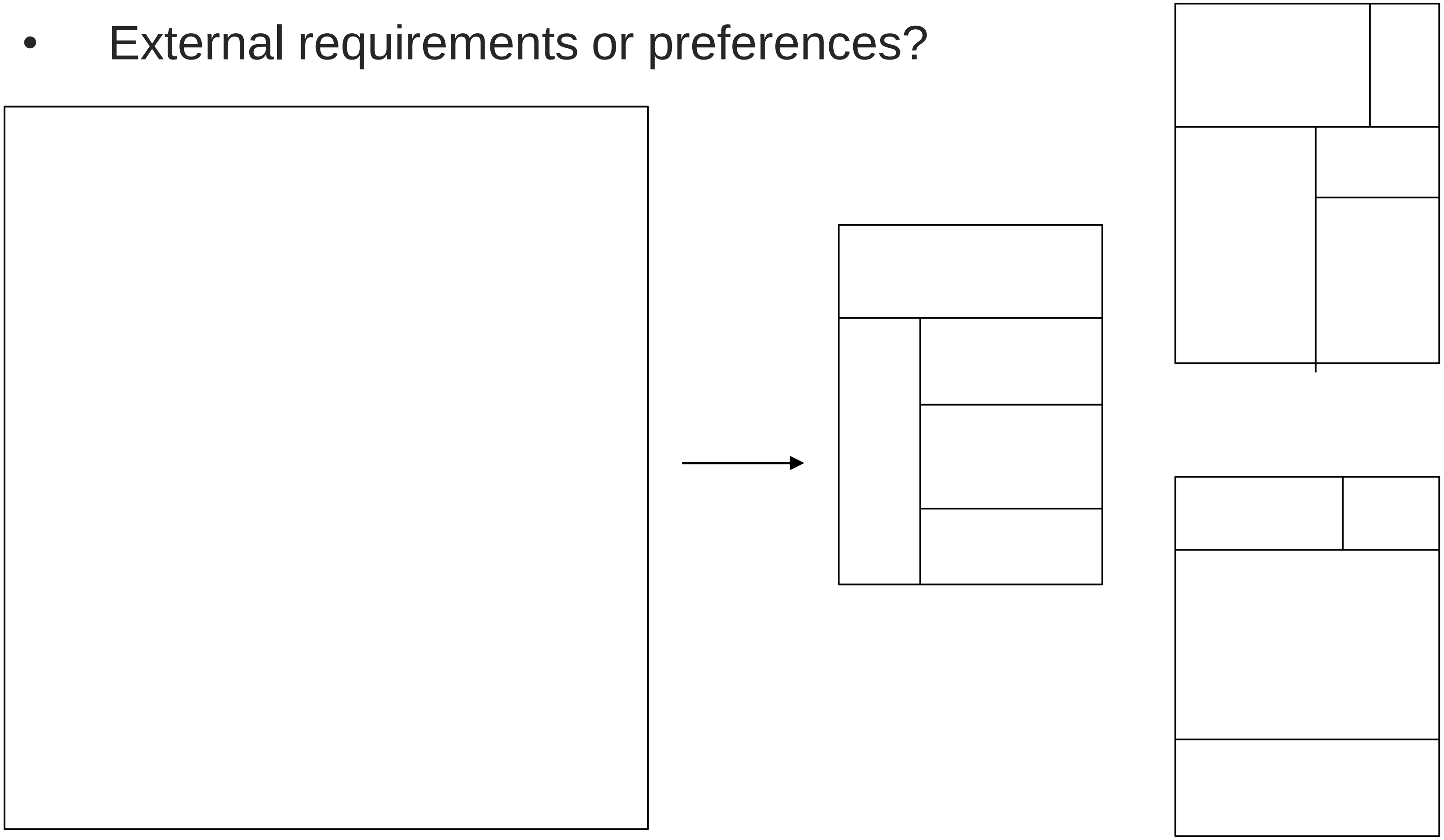
Poblem formulation (I)

- Geometric area as a base for the floor layout
- External knowledge – constraints, requirements
- Preferences



Poblem formulation (II)

- Same geometric area as a base -> different layouts
- External requirements or preferences?



JAGIELLONIAN UNIVERSITY IN KRAKOW

Optimization problem?

YES

Constraint based optimization problem

Case-based design

Possible well defined numerical boundaries (area, price, use of materials)

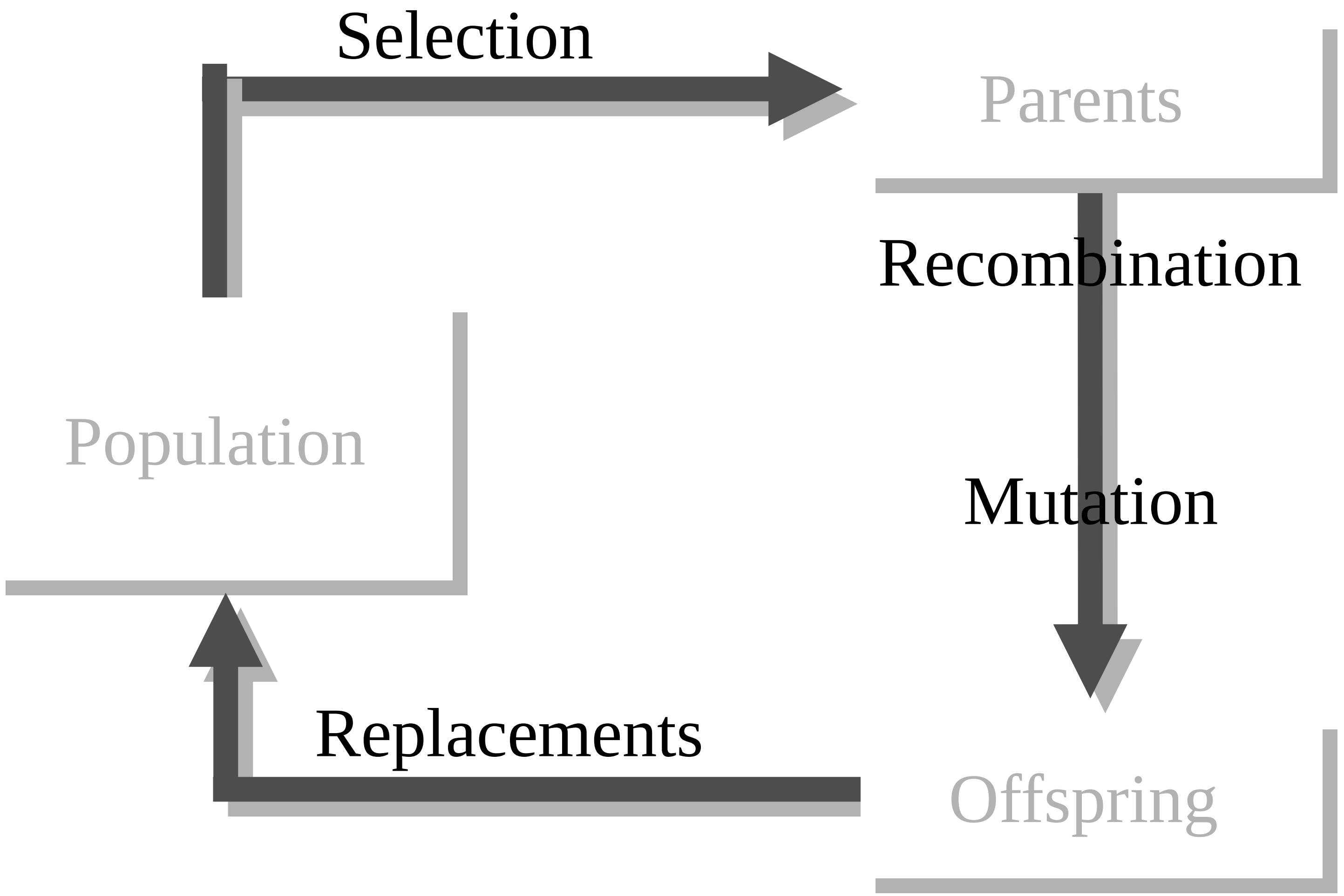
NO

Soft requirements

Personal preferences

Evolutionary computations (I)

Population algorithms

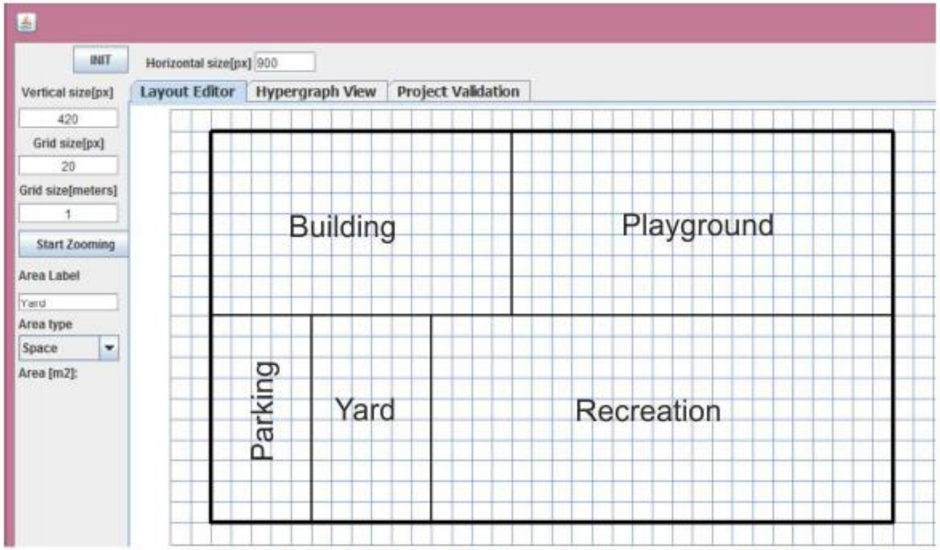


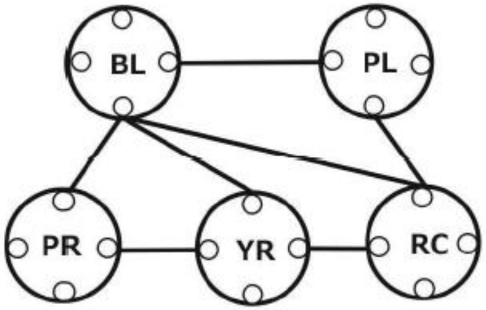
Population algorithms

- Initialization
- Representation
- Recombination and/or mutation
- Fitness evaluation
- Selection

Graph representation (CP)

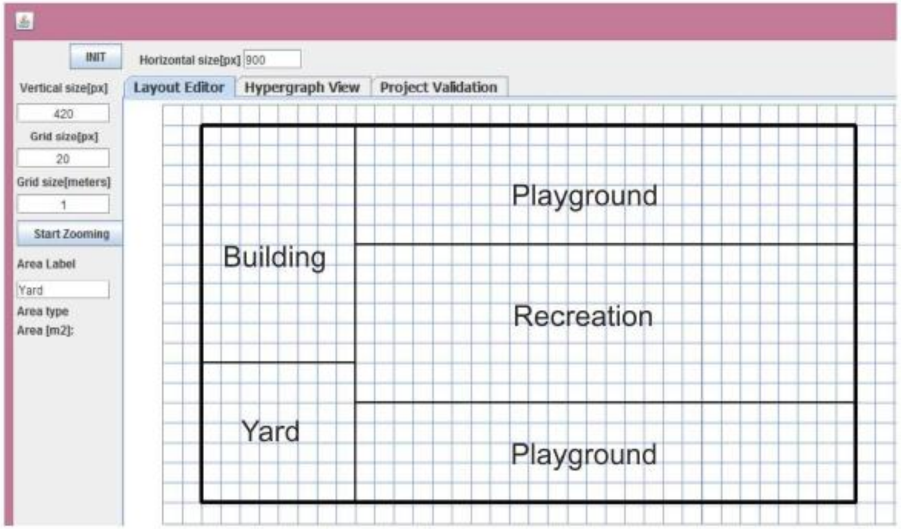
Structural relations between components – graphs
CP-Graph - nodes, edges, bonds
Nodes - components
Edges - relations between components
Bonds - potential connections („placeholders”)
Attributes – semantic information

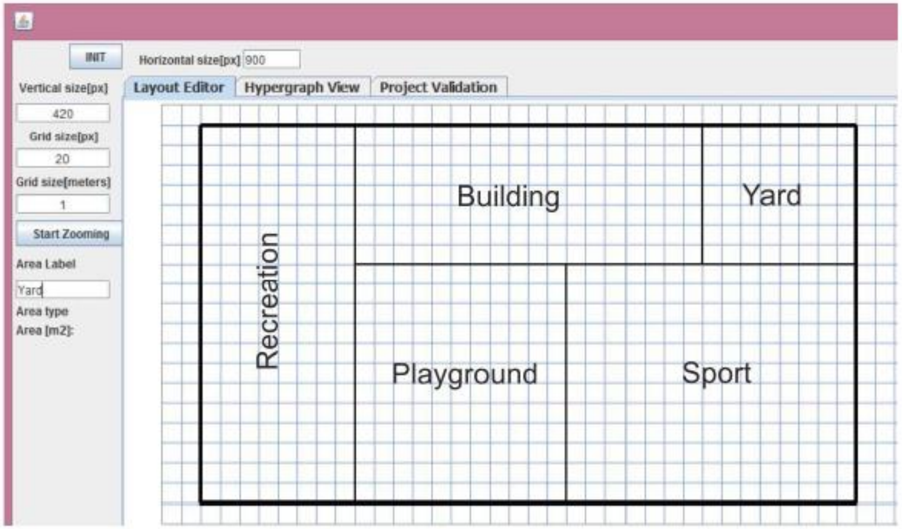


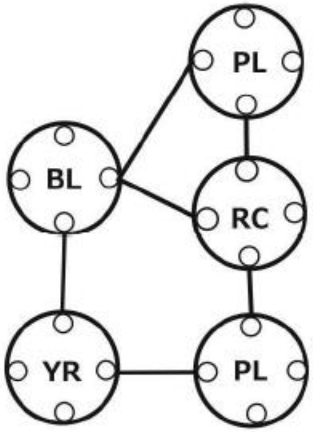


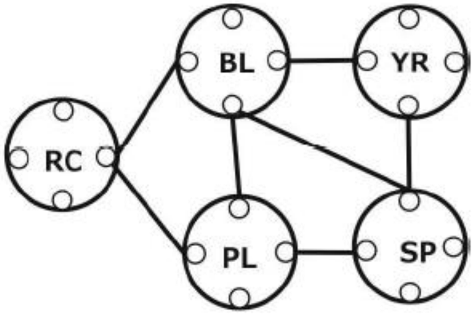
Graph – representa a potential solution of a design task

Graph representation (CP – II)





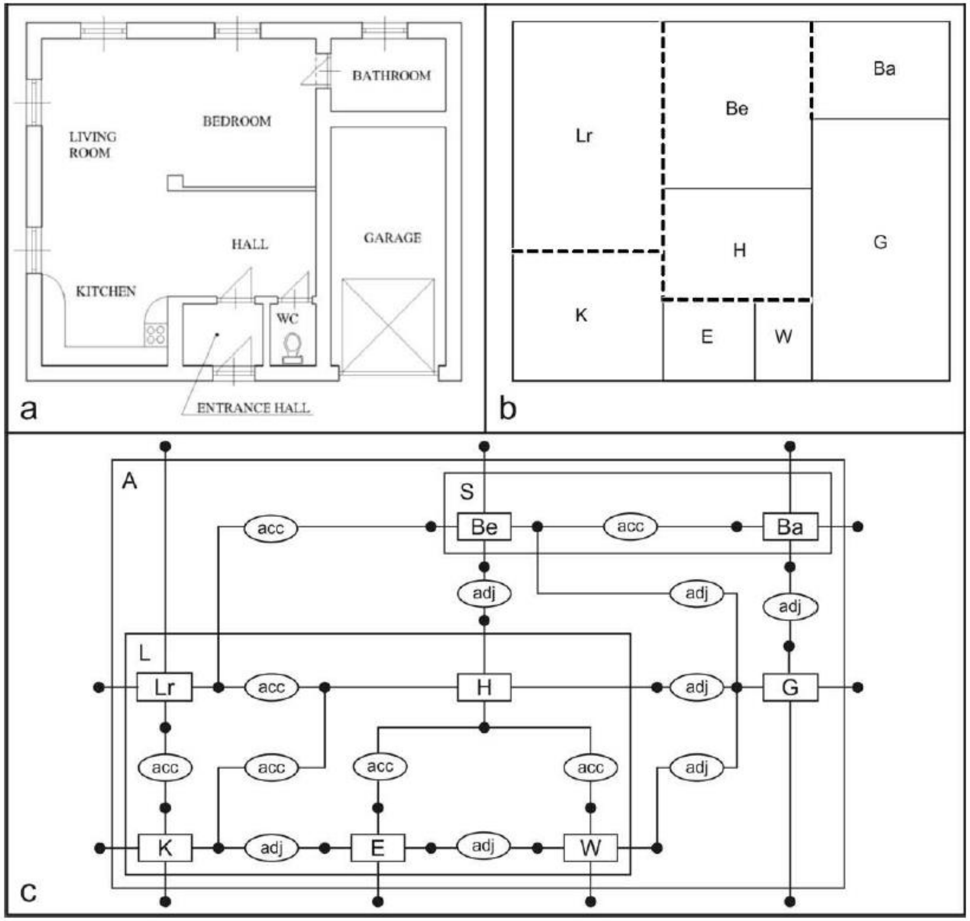




Graph representation (Hypergraph)

Relations between components – hypergraphs (hierarchical or not)

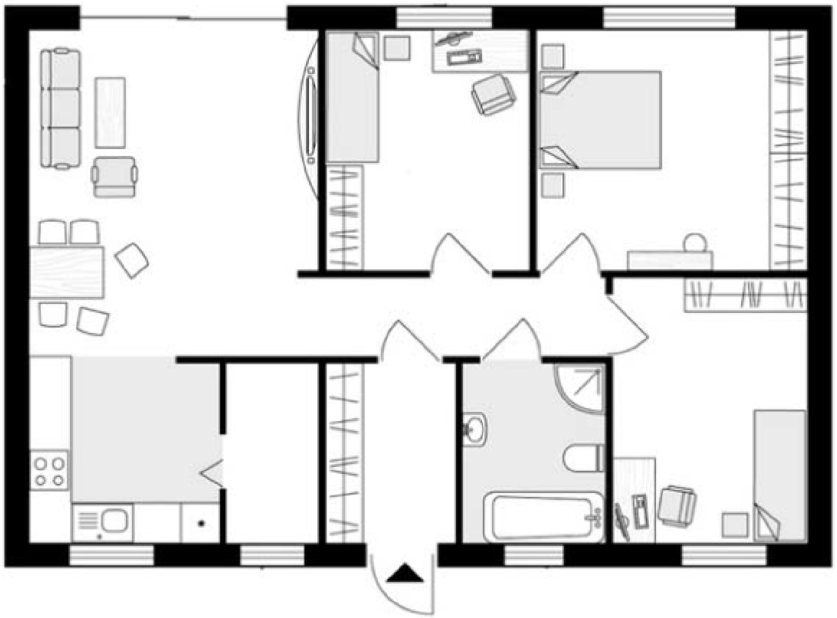
- hypergraph - nodes, hyperedges,
- nodes - walls
- hyperedges - components
  and relations between
  components
- attributes – semantic
  information

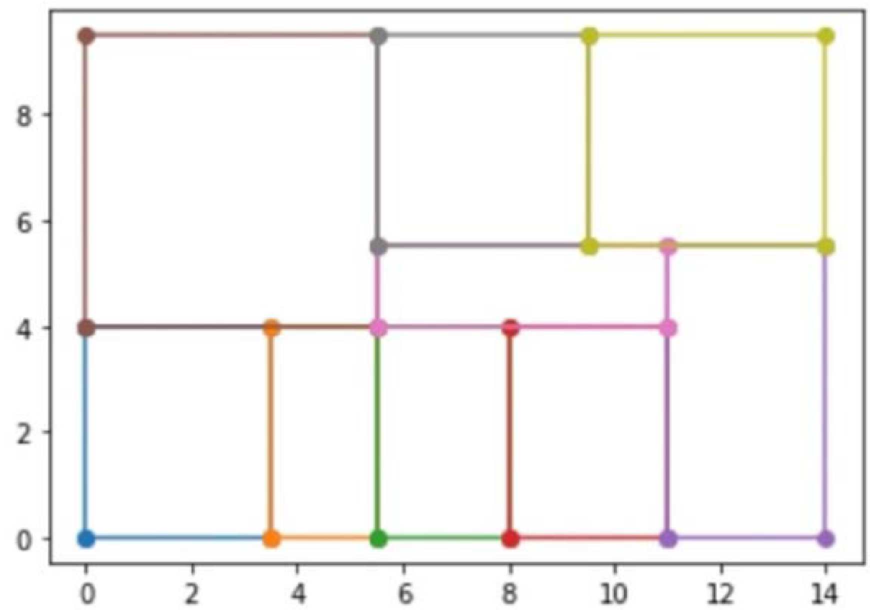


Hypergraph – representa
a potential solution of a
design task

Vector representation

No structural information

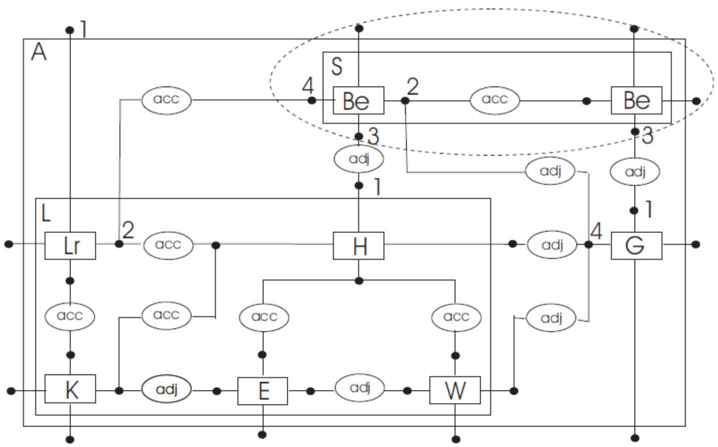


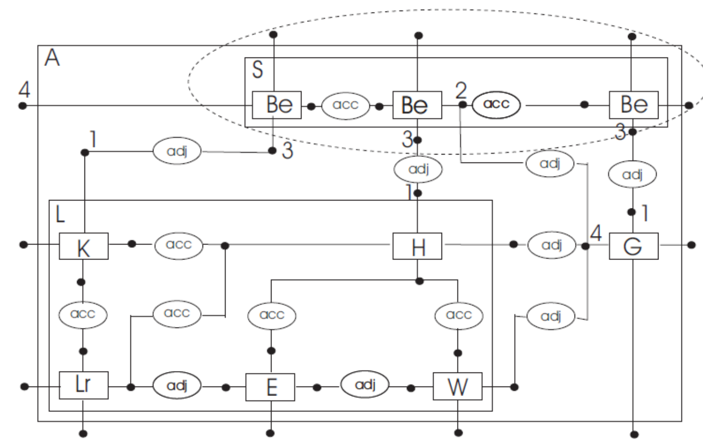


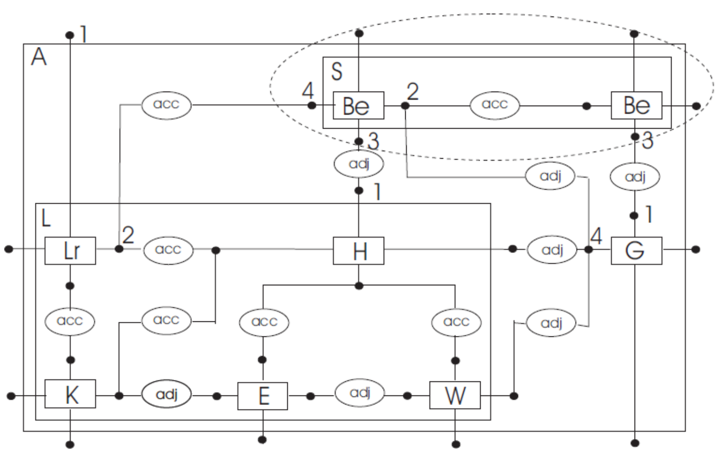

```
points = [(0.0,0.0),(3.5,0.0),(5.5,0.0),(8.0,0.0),(11.0,0.0),
          (14.0,0.0),(11.0,4.0),(8.0,4.0),(5.5,4.0),(3.5,4.0),
          (0.0,4.0),(5.5,5.5),(9.5,5.5),(11.0,5.5),(14.0,5.5),
          (14.0,9.5),(9.5,9.5),(5.5,9.5),(0.0,9.5)]

individual = {'R1':[0,1,9,10],'R2':[1,2,8,9],'R3':[2,3,7,8],
              'R4':[3,4,6,7], 'R5':[4,5,14,13], 'R6':[10,8,17,18],
              'R7':[8,6,13,11], 'R8':[11,12,16,17], 'R9':[12,14,15,16]}
```

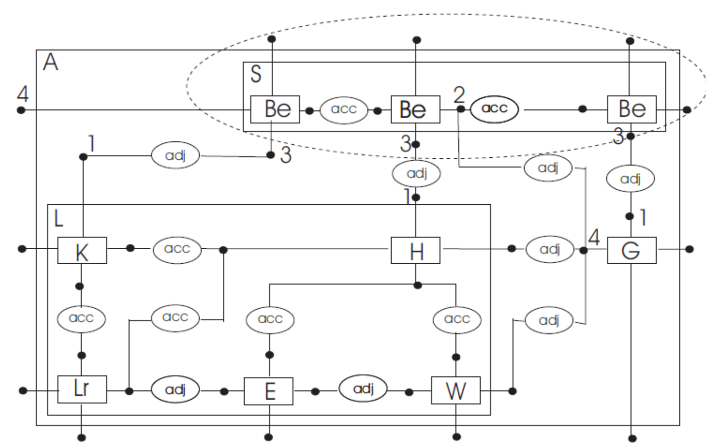
Operators (Hypergraph)

Crossover - The exchange of subgraphs between two different designs





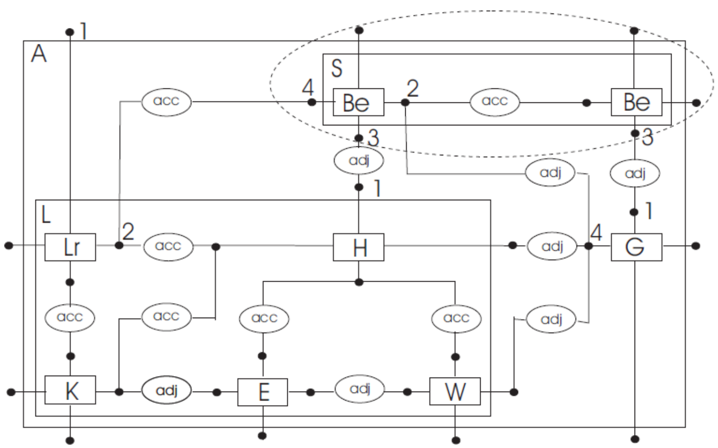


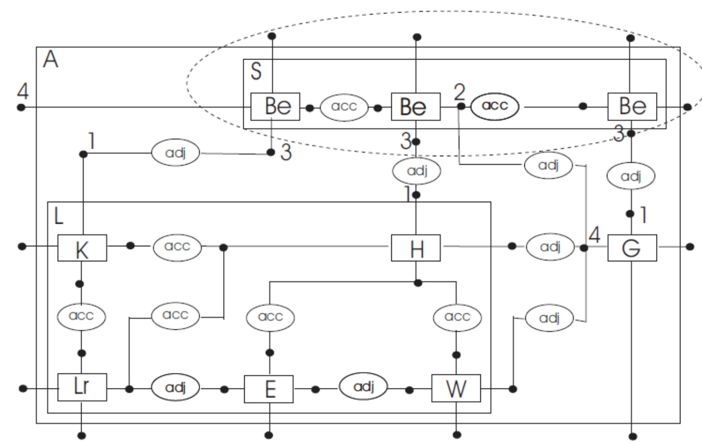


Operators (Hypergraph)

The exchange of subgraphs between two different designs

Limitations/problems
- embedding transformation (Ref*)
- computational complexity
- need for specialized algorithm(s)



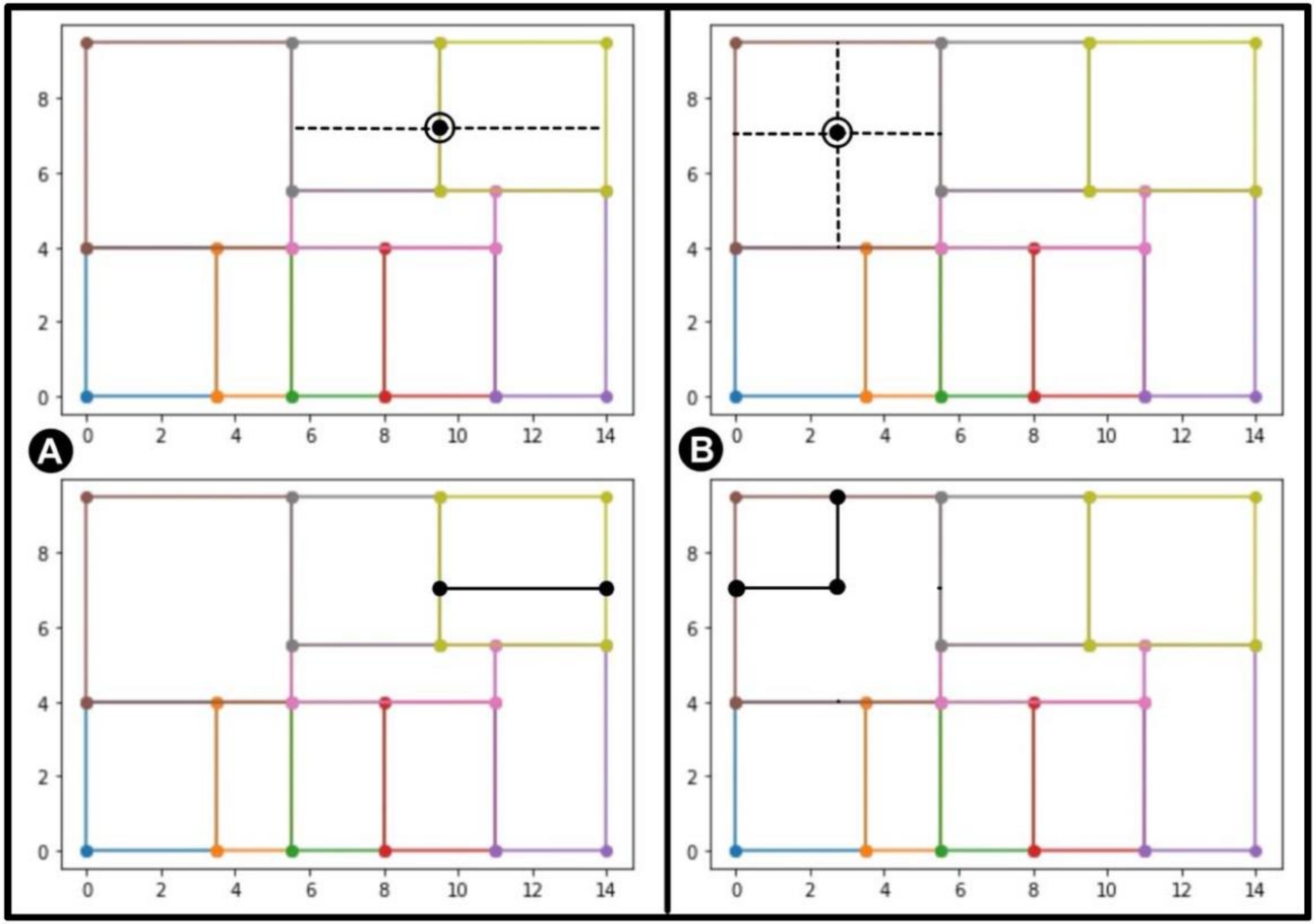


(*) *Grazyna Slusarczyk Barbara Strug , Anna Paszynska, Ewa Grabska, Wojciech Palacz ,Semantic-driven Graph Transformations in Floor Plan Design. Comput. Aided Des. 158: 103480 (2023)*

Operators (Vector)

Mutation only

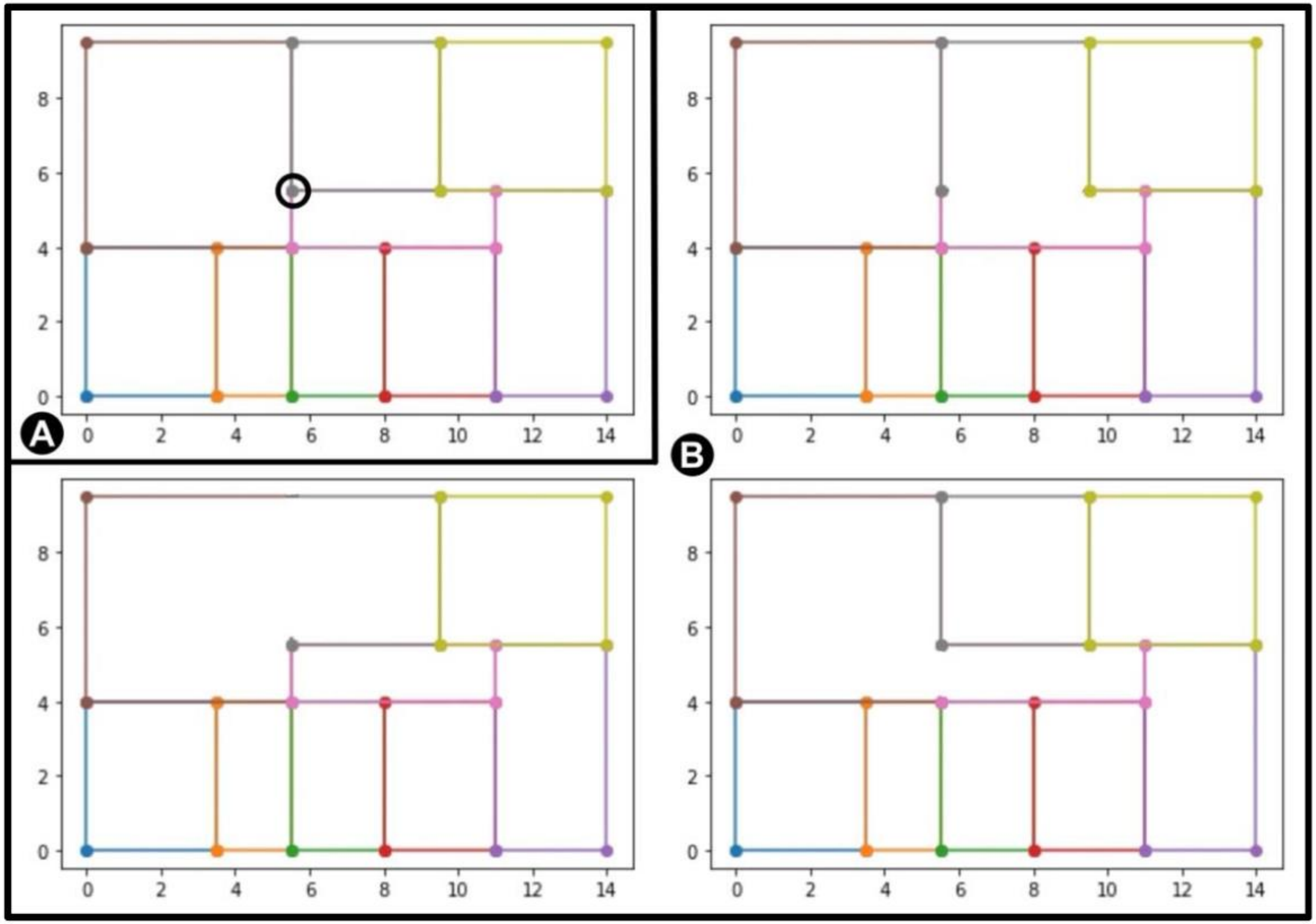
adding a point
deleting a point
moving a point



Operators (Vector)

Mutation only

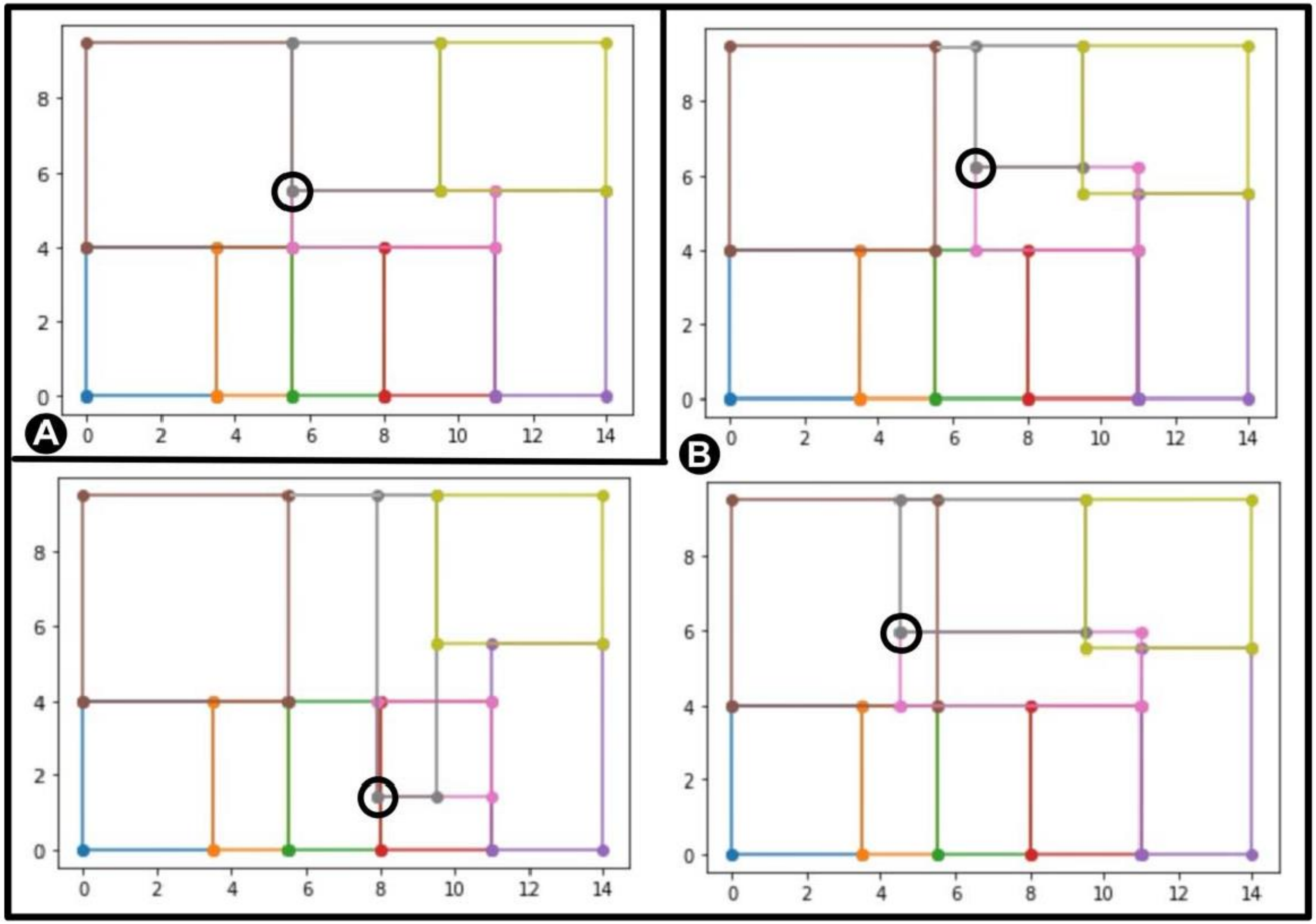
adding a point
deleting a point
moving a point



Operators (Vector)

Mutation only

adding a point
deleting a point
moving a point



Fitness evaluation (Graph)

Graph based - low numer of produced solutions ;>

mainly human designer

graph pattern mining

Requires the process of expression (graph to design)

Fitness evaluation (Vector)

Point representation
based on the degree of fulfilment of requirements

$$F(I) = \begin{cases} -\propto, & \exists unfulfilled\ constraint \\ \sum_{i=1}^{n} w_i Req_i(I), & otherwise \end{cases}$$

Example (Vector)

Constraints

1 Six predefined rooms (3 Bedrooms, 1 bathroom, kitchen, living room)

2 No wall shorter than 0.8m

Requirements

1 There should be at least eight spaces and w1 = 0.8, Req1 from {0, 0.33, 0.5, 0.67,1}

2 The largest room should be bigger that 21 m2 and w2 = 0.7, Req2 from {0,1}.

3 There should exist a room larger than 7 m2 adjacent to the largest room and w3 = 0.6, Req3 from {0,1}.

4 The largest room should be oriented to the south and w4 = 0.5, Req4 from {0,1}.

5 There are not many spaces with areas less than 2 m2 and w5 = 0.5, Req5 from {0,0.2,0.4,0.6,0.8,1}

JAGIELLONIAN UNIVERSITY
IN KRAKOW

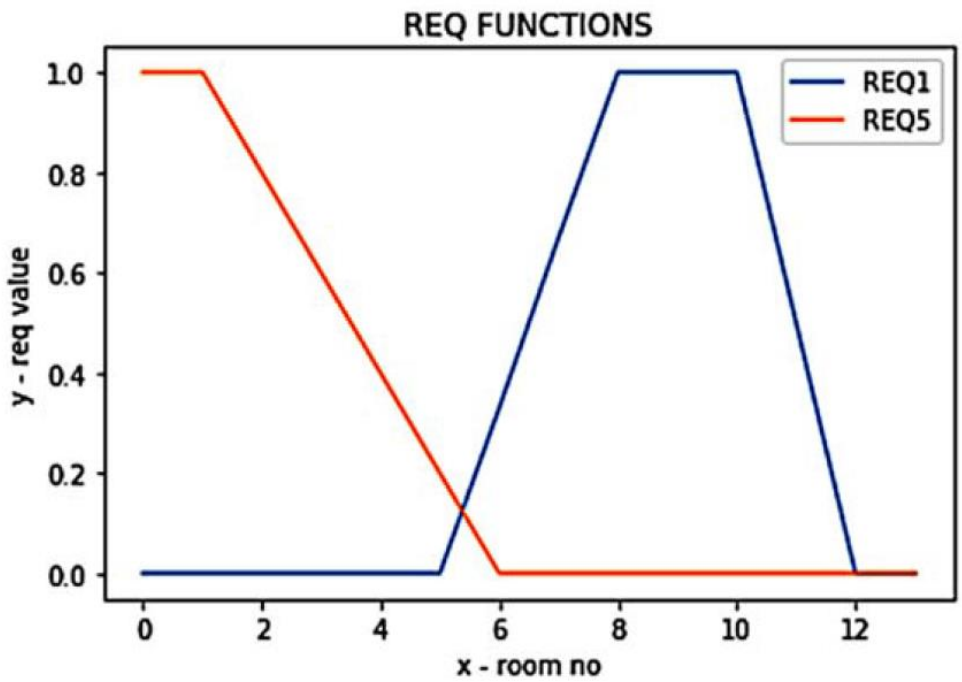
Example (Vector)

Constraints

1 Six predefined rooms (3 Bedrooms, 1 bathroom, kitchen, living room)
2 No wall shorter than 0.8m

Requirements

1 There should be at least eight spaces and w1 = 0.8, Req1 from {0, 0.33, 0.5, 0.67,1}
2 The largest room should be b {0,1}.
3 There should exist a room lar room and w3 = 0.6, Req3 f
4 The largest room should be c from {0,1}.
5 There are not many spaces w Req5 from {0,0.2,0.4,0.6,0.

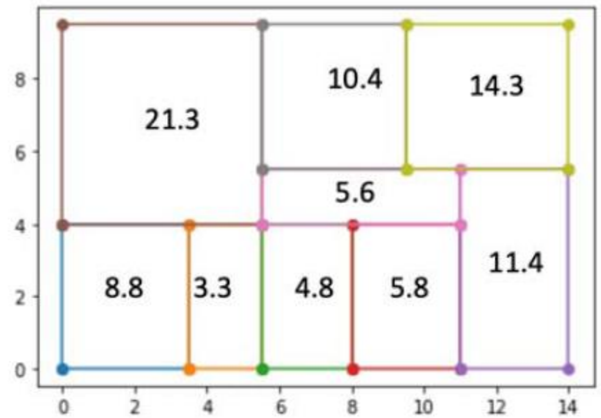


JAGIELLONIAN UNIVERSITY IN KRAKOW

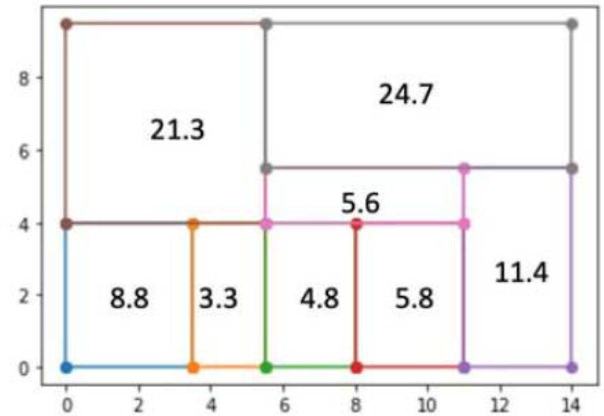
Example (Vector)

Constraints
1 Six predefined rooms (3 Bedrooms, 1 bathroom, kitchen, living room)
2 No wall shorter than 0.8m

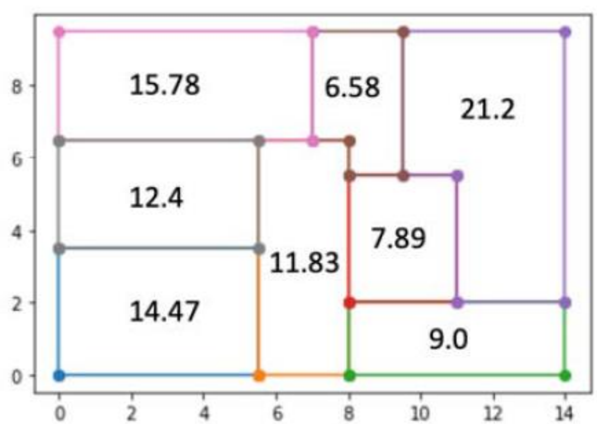
Requirements
1 There should be at least eigh
t spaces and w1 = 0.8,
Req1 from {0, 0.33, 0.5, 0.67,1}
2 The largest room should be bigger
that 21 m2 and w2 = 0.7,
Req2 from {0,1}.
3 There should exist a room larger
than 7 m2 adjacent to the largest
room and w3 = 0.6, Req3 from {0,1}.
4 The largest room should be
oriented to the south and w4 = 0.5,
Req4 from {0,1}.
5 There are not many spaces with
areas less than 2 m2 and w5 = 0.5,
Req5 from {0,0.2,0.4,0.6,0.8,1}



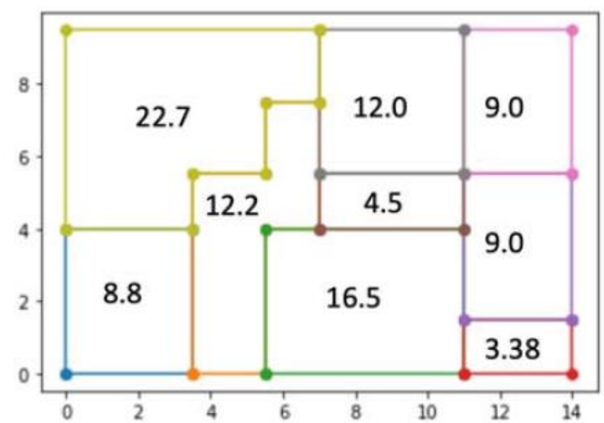
$F(I_1)$=1x0.8+1x0.7+1x0.6+1x0.5+1x0.5=3.1 **MAX**



$F(I_2)$=1x0.8+1x0.7+1x0.6+1x0.5+1x0.5=3.1 **MAX**



F (I_3)=1x0.8+1x0.7+1x0.6+1x0.5+1x0.5=3.1 **MAX**

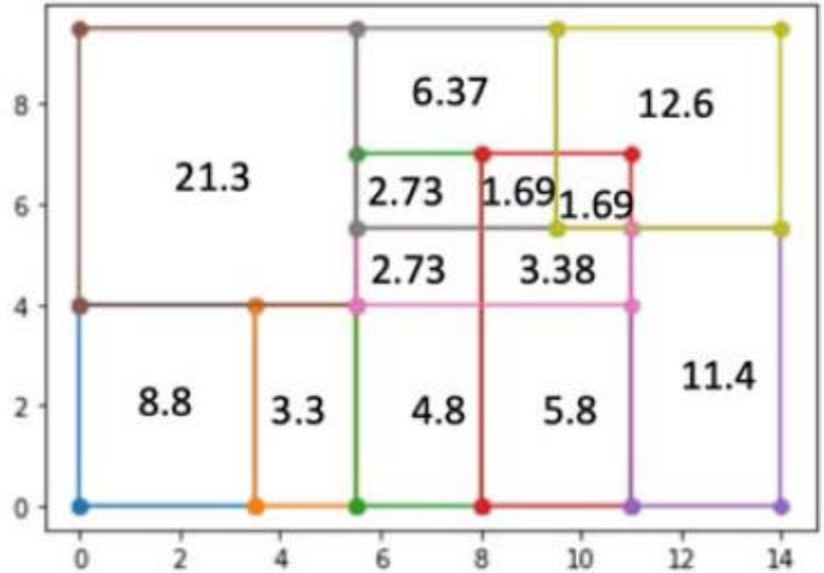


F (I_4)=1x0.8+1x0.7+1x0.6+1x0.5+1x0.5=3.1 **MAX**

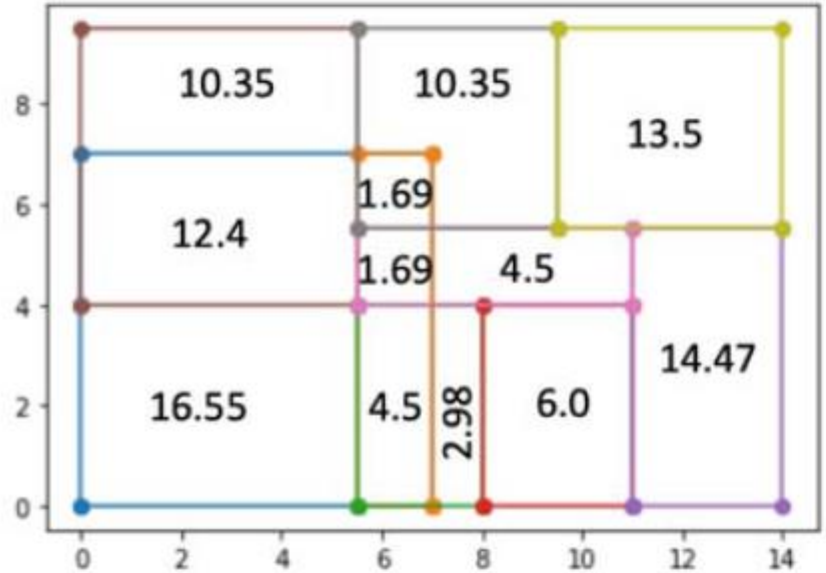
Example (Vector)

Constraints
1 Six predefined rooms (3 Bedrooms, 1 bathroom, kitchen, living room)
2 No wall shorter than 0.8m

Requirements
1 There should be at least eight spaces and w1 = 0.8, Req1 from {0, 0.33, 0.5, 0.67,1}
2 The largest room should be bigger that 21 m2 and w2 = 0.7, Req2 from {0,1}.
3 There should exist a room larger than 7 m2 adjacent to the largest room and w3 = 0.6, Req3 from {0,1}.
4 The largest room should be oriented to the south and w4 = 0.5, Req4 from {0,1}.
5 There are not many spaces with areas less than 2 m2 and w5 = 0.5, Req5 from {0,0.2,0.4,0.6,0.8,1}



$F(I_5)=0x0.8+1x0.7+1x0.6+1x0.5+0.8x0.5=2.2$



$F(I_6)=0x0.8+0x0.7+1x0.6+1x0.5+0.8x0.5=1.5$

JAGIELLONIAN UNIVERSITY IN KRAKOW

Conclusions

Graph based representation

+ better at prserving structural information
- complex operators
- smaller population

Vector based representation

+ faster computations
+ more flexible
- harder to add semantics

Other possibilities

Multi-storey buildings

Katarzyna Grzesiak-Kopeć,Barbara Strug , Grazyna Slusarczyk, Specification-Driven Evolution of Floor Plan Design. PPSN (2) 2022 368-381

Graph learning ?

JAGIELLONIAN UNIVERSITY
IN KRAKOW

Thank you for your attention