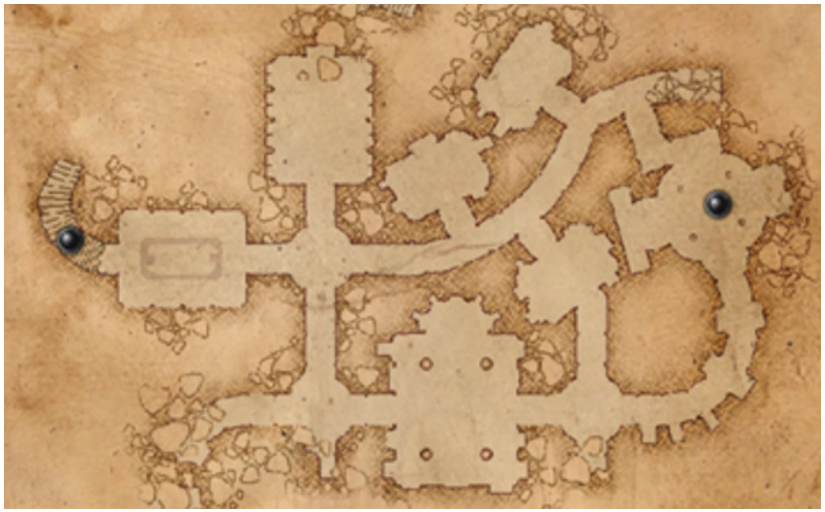
Eurographics 2014 – Strasbourg – France

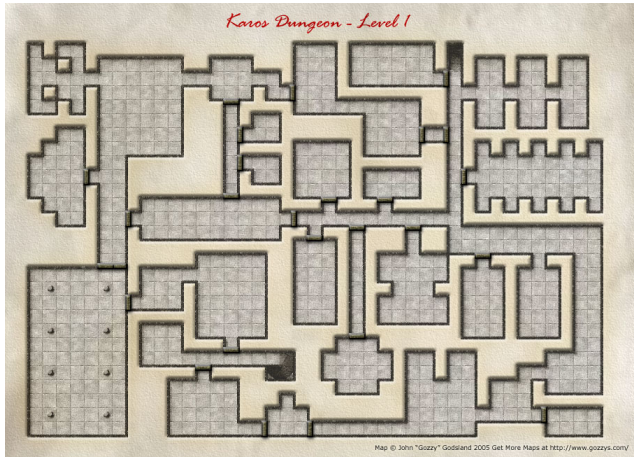
Game Level Layout from Design Specification

[Eurographics 2014]

Chongyang Ma
Sylvain Lefebvre
Nicholas Vining
Alla Sheffer

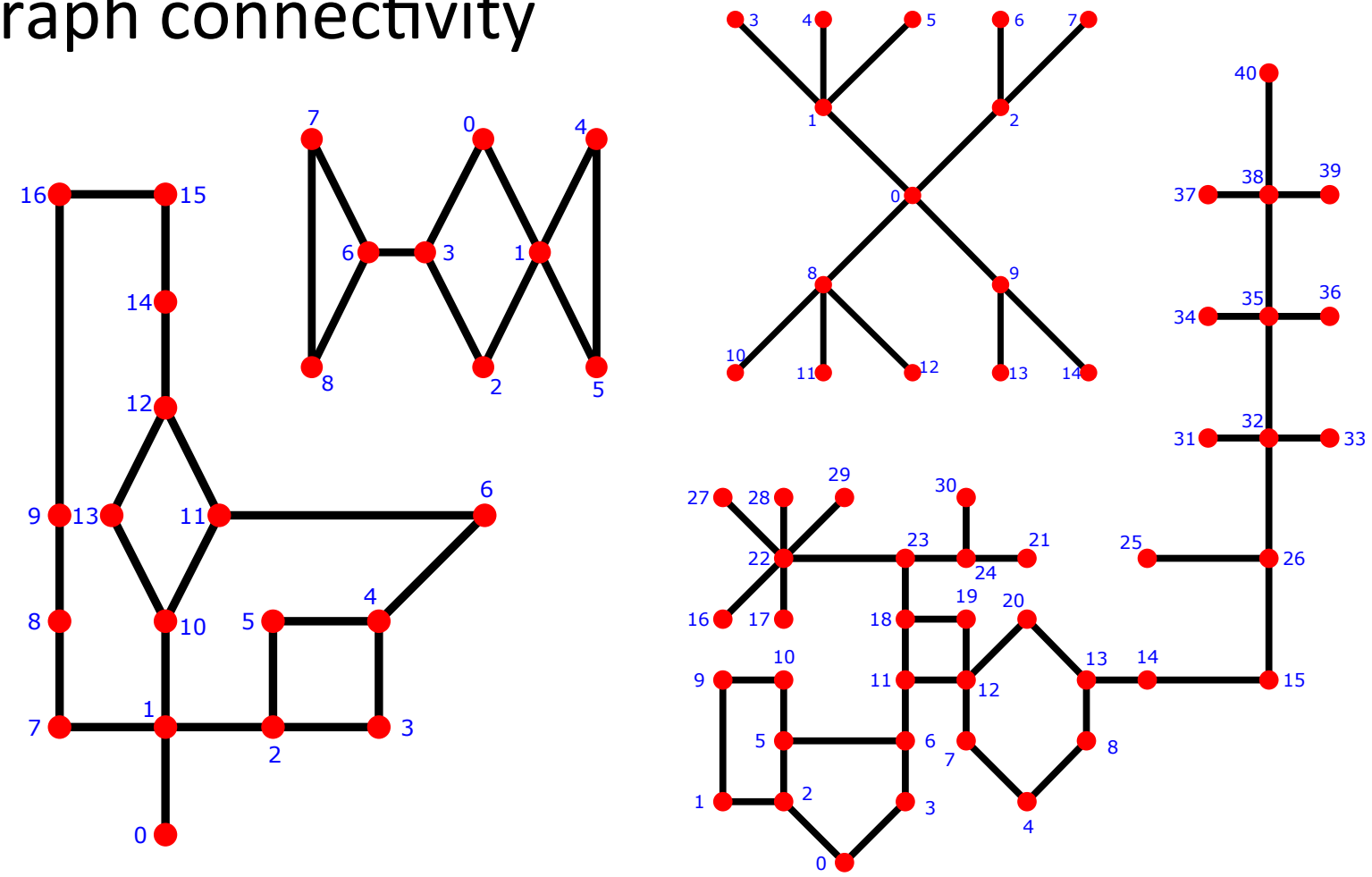
Game level layouts





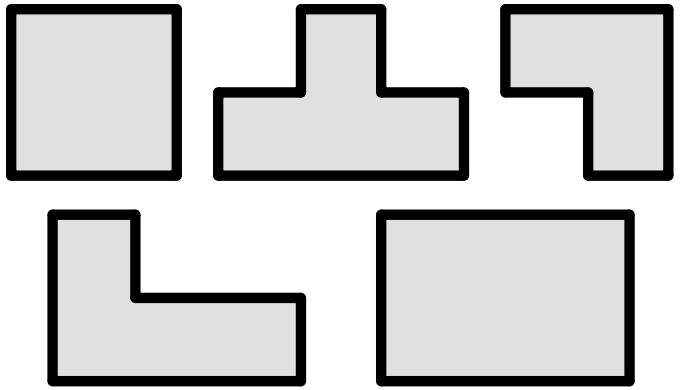
Design specification

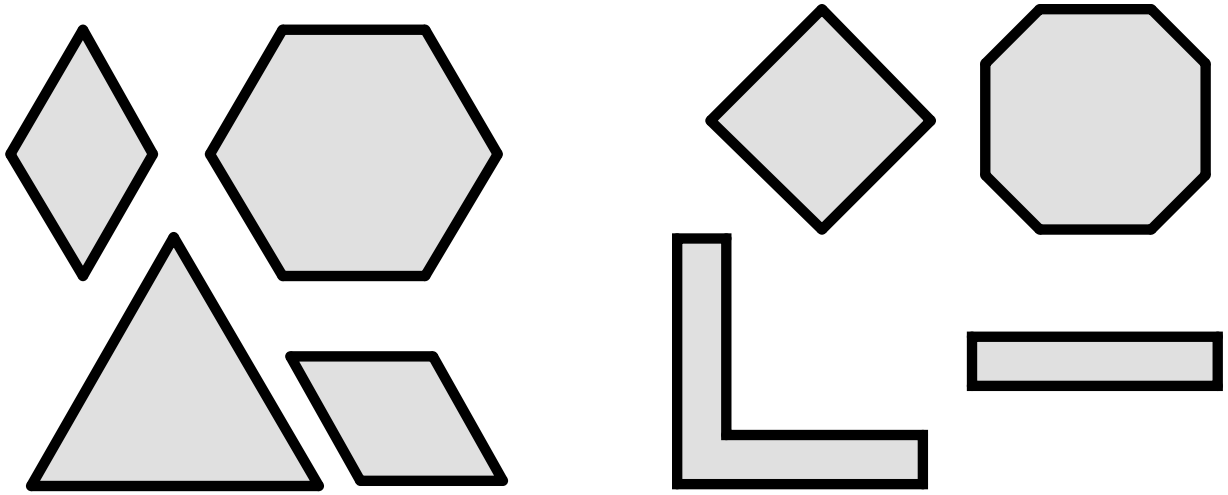
- Graph connectivity



Design specification

- Graph connectivity
- Building blocks



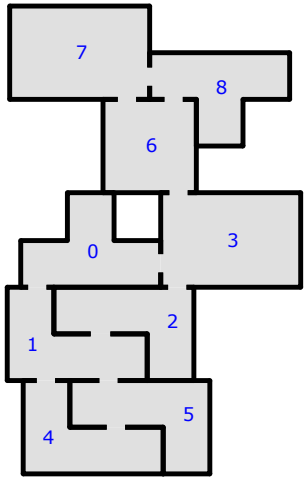


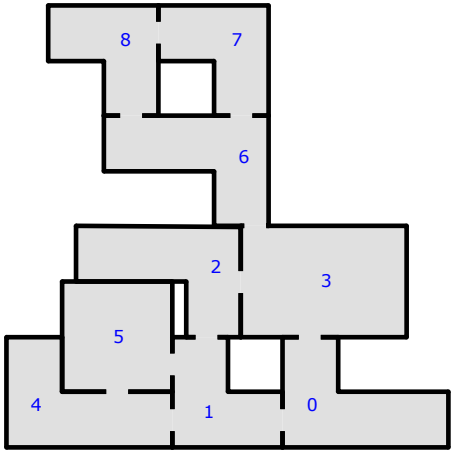
Design specification

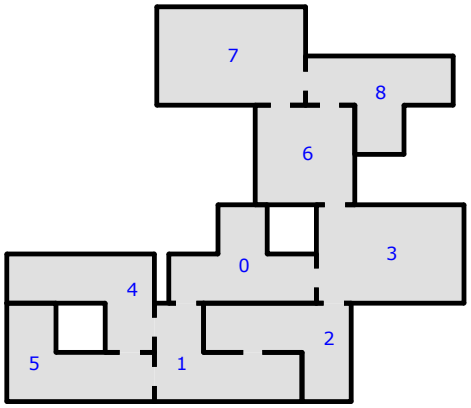
- Graph connectivity
- Building blocks
- Additional constraints
  - Intersection-free
  - Pairwise contacts
  - Boundary obstacles

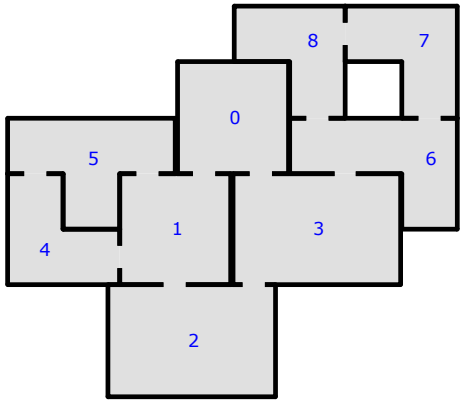
Design specification

- Graph connectivity
- Building blocks
- Additional constraints
- Diverse outputs

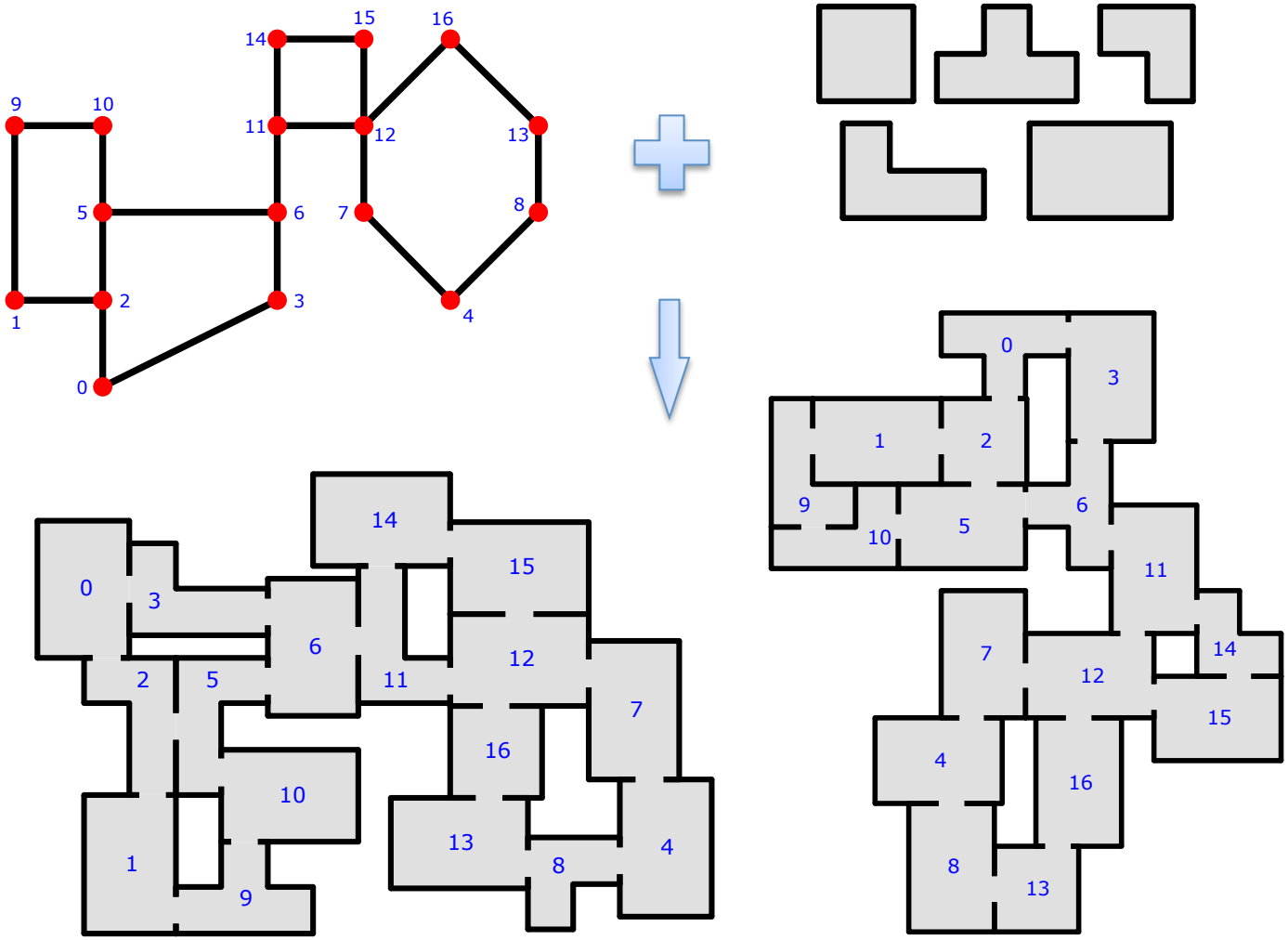






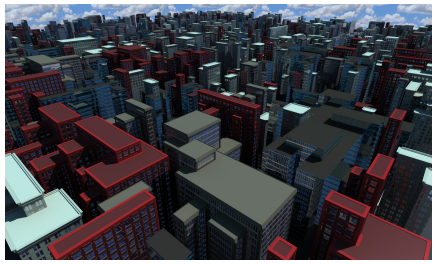


Game level layout synthesis



Procedural geometry modeling

- 3D architectural shapes
  - [WWSR03, MWH*06, MM08, LCOZ*11]



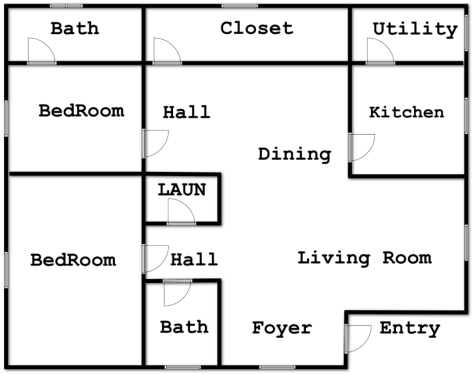
- 2D building layouts
  - [PM01, CEW*08, VKW*12, YYW*12]

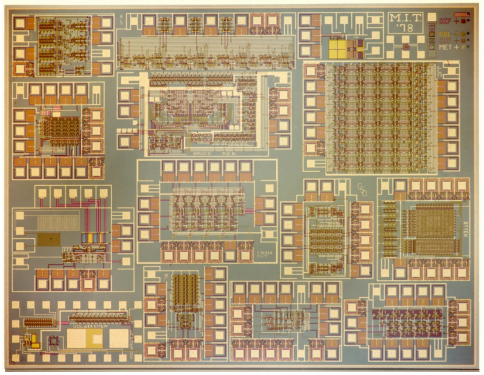
✔ Plausibility and aesthetics

✖ Do NOT control contacts or adjacencies

Procedural geometry modeling

- Floor plans
  - [MS74, Sha87, Lig00, MSK10, LYAM13, BYMW13]
- VLSI circuits
  - [She98]





✔ Controlled envelope

✖ Axis-aligned elements only

Technical challenges

- High-dimensional
  - Large number of blocks
- Mixed continuous-discrete search space
  - Continuous: block positions
  - Discrete: node-to-block associations

Naïve stochastic optimization

✖ Low convergence rates

Algorithm

- Configuration space
- Incremental layout
  - Graph decomposition
  - Backtracking
- Chain layout
  - Iterative optimization

Algorithm

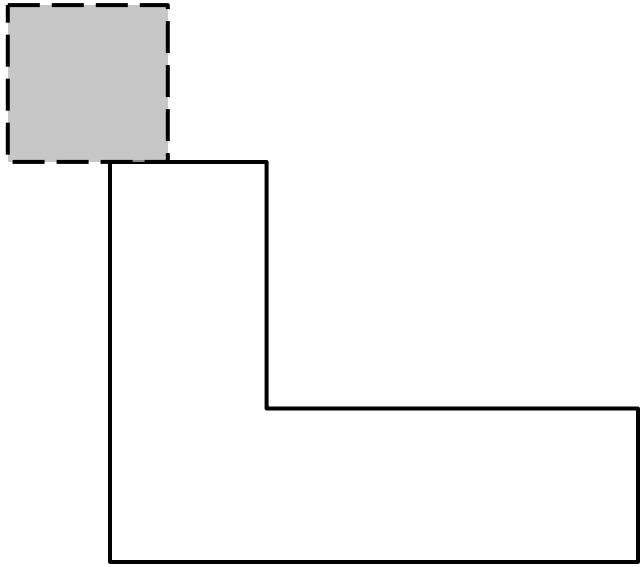
- Configuration space

- Incremental layout
  - Graph decomposition
  - Backtracking

- Chain layout
  - Iterative optimization

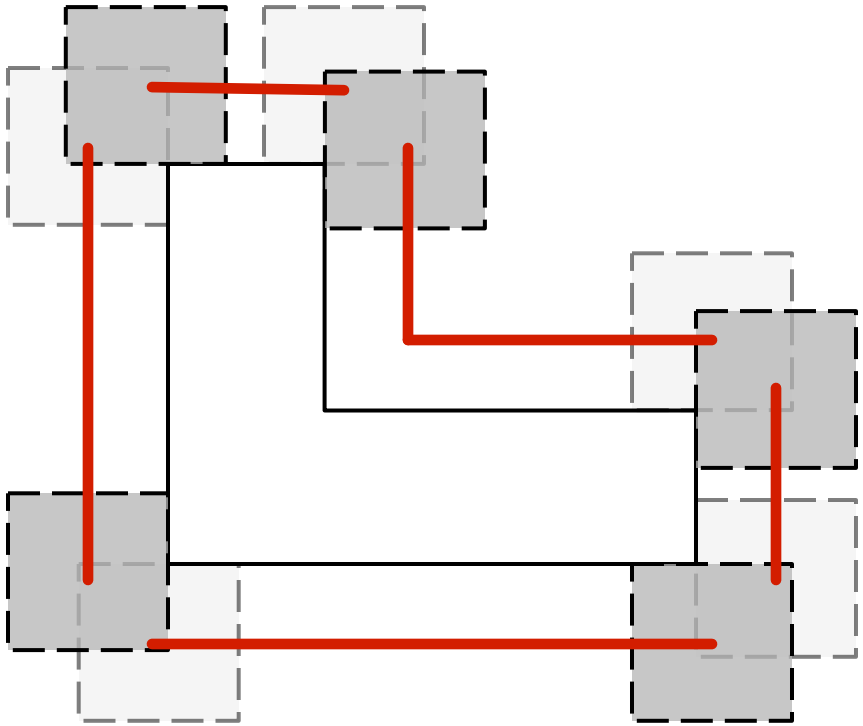
Configuration space

- For a pair of building blocks
- Enough contact area but no intersection



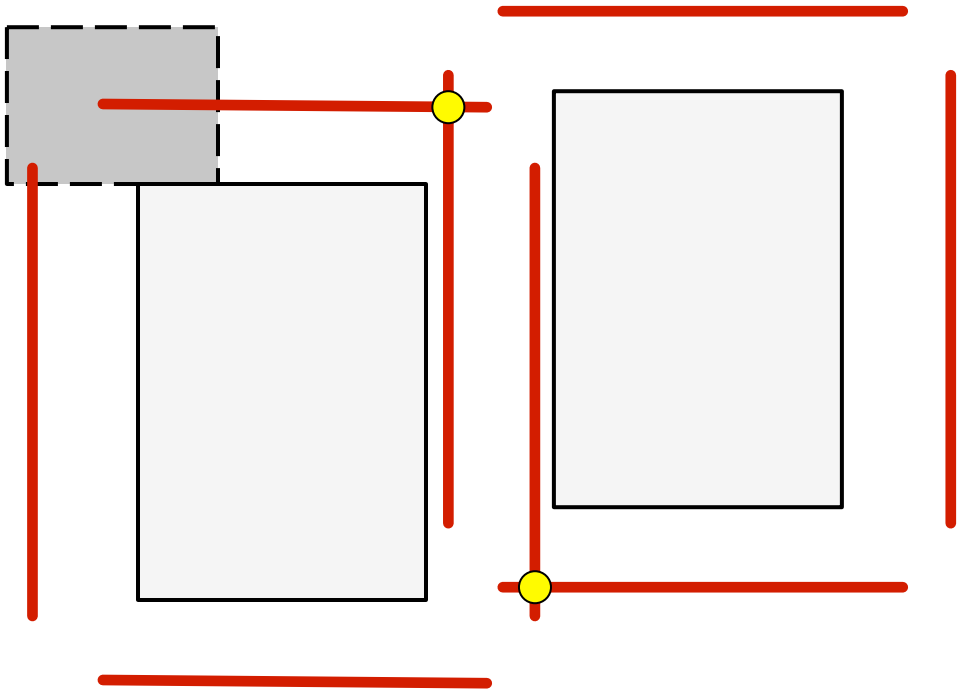
Configuration space

- For a pair of building blocks
- Enough contact area but no intersection



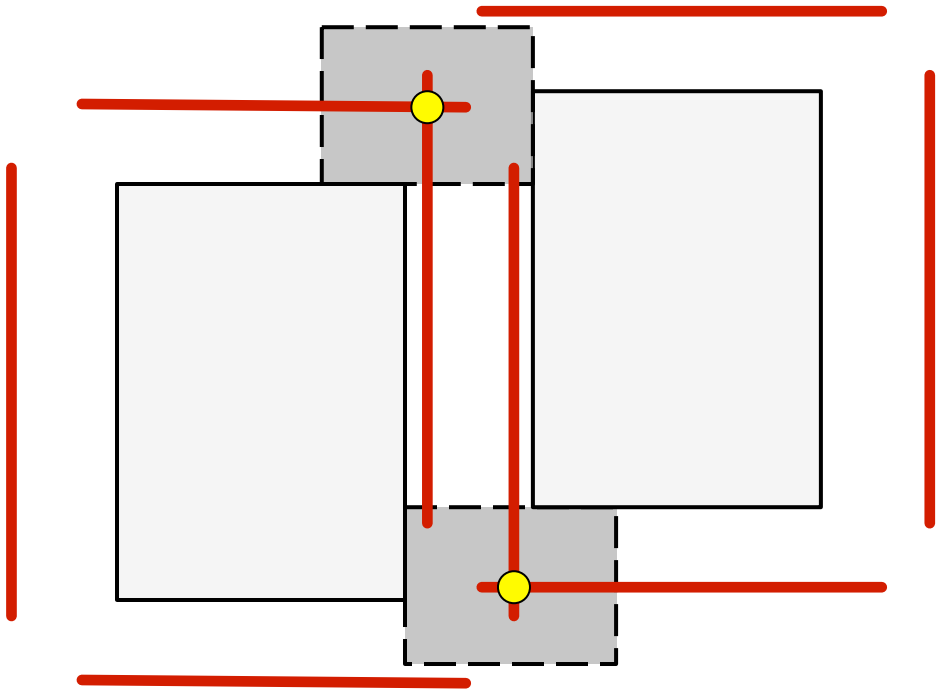
Configuration space

- Intersection of multiple config. spaces



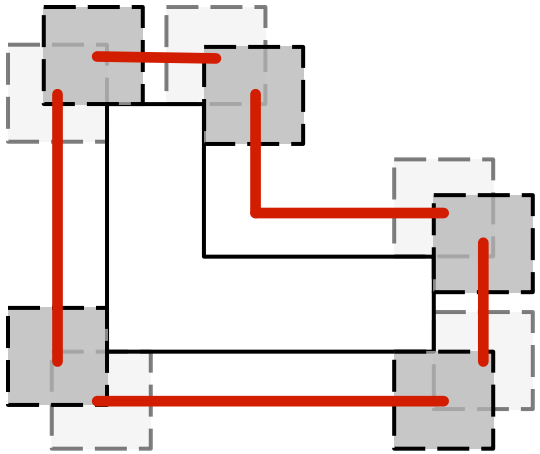
Configuration space

- Intersection of multiple config. spaces

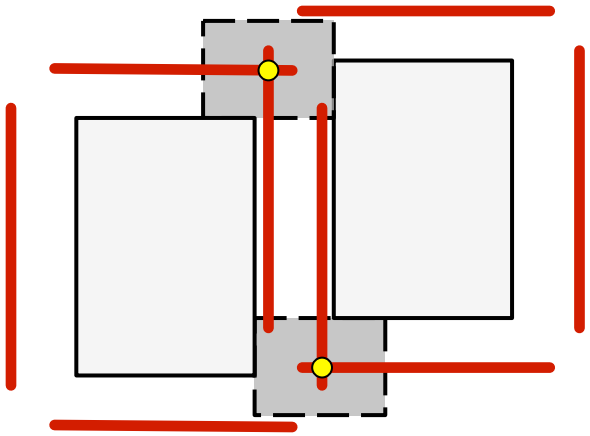


Algorithm

- Configuration space



- Incremental layout
  - Graph decomposition
  - Backtracking



- Chain layout
  - Iterative optimization

Algorithm

- Configuration space

- Incremental layout
  - Graph decomposition
  - Backtracking

- Chain layout
  - Iterative optimization

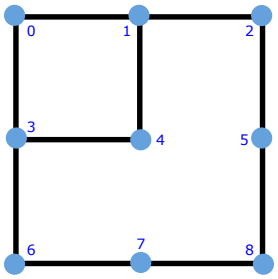
Incremental layout

Input: Planar graph $\mathcal{G}$, building blocks $\mathcal{B}$, layout stack $\mathcal{S}$

```
 1: procedure IncrementalLayout(G, B, S)
 2:     Push empty layout into S
 3:     repeat
 4:         s ← S.pop()
 5:         Get the next chain c to add to s
 6:         AddChain(c, s)   //extend the layout to contain c
 7:         if extended partial layouts were generated then
 8:             Push new partial layouts into S
 9:         end if
10:     until target # of full layouts is generated or S is empty
11: end procedure
```

Incremental layout

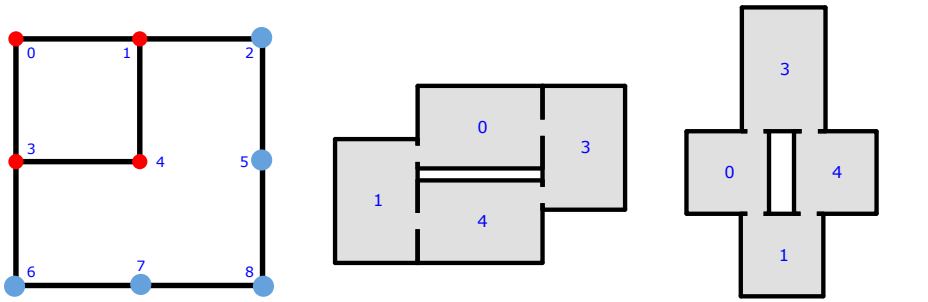
- Graph decomposition



(a) Input graph

Incremental layout

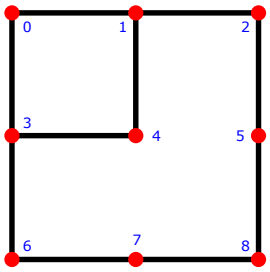
- Graph decomposition



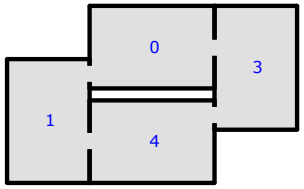
(a) Input graph (b) Partial solution 1 (c) Partial solution 2

Incremental layout

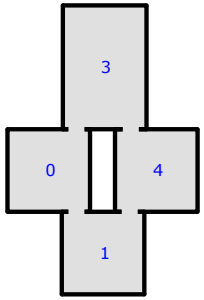
- Graph decomposition



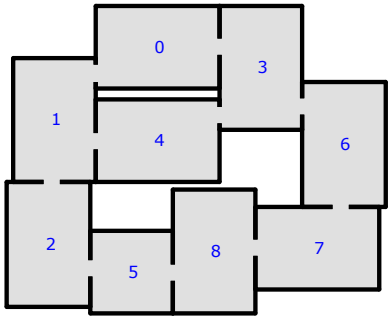
(a) Input graph



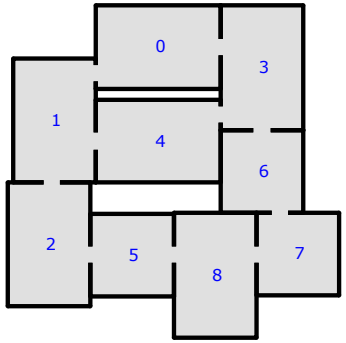
(b) Partial solution 1



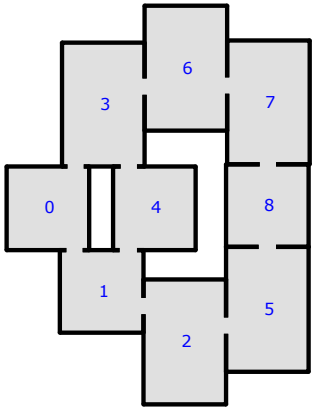
(c) Partial solution 2



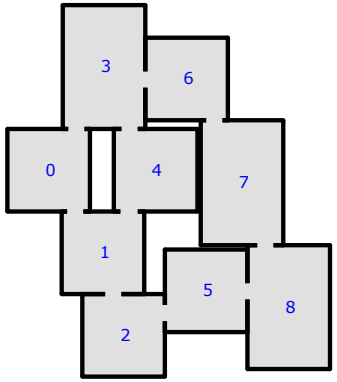
(d) Full solution 1



(e) Full solution 2



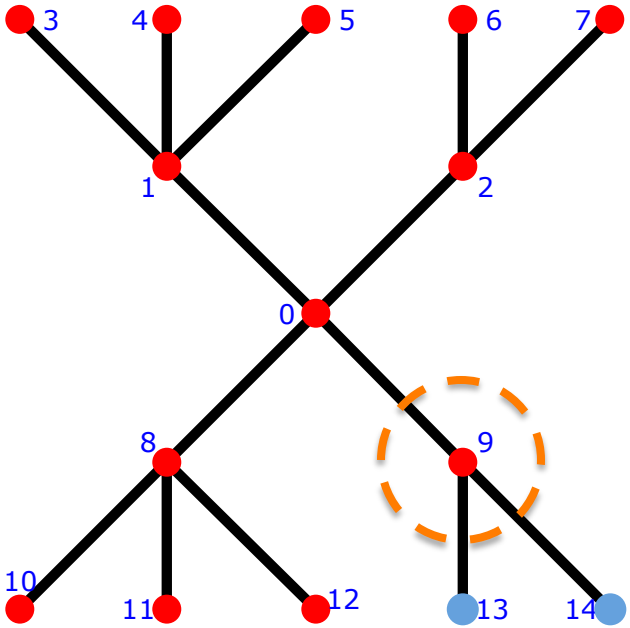
(f) Full solution 3



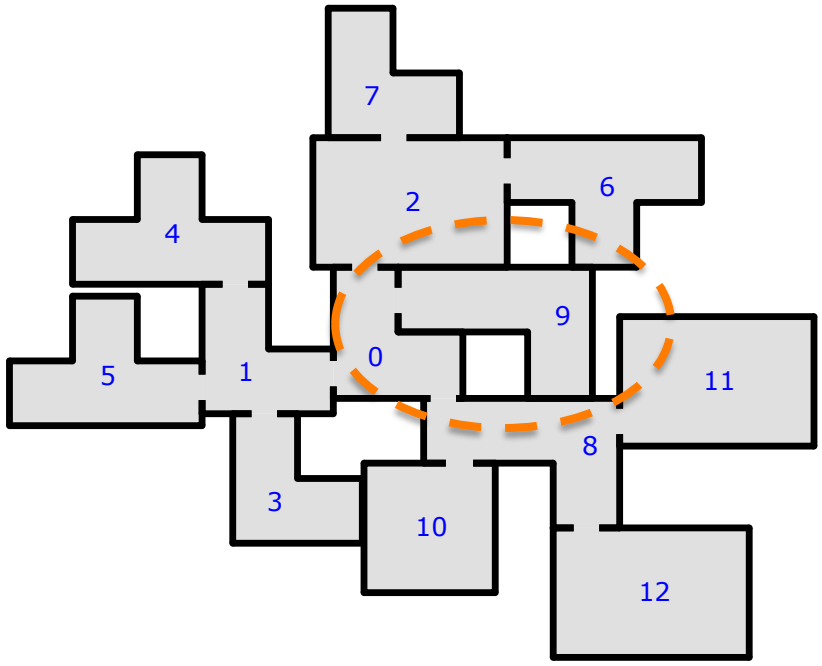
(g) Full solution 4

Incremental layout

- Backtracking



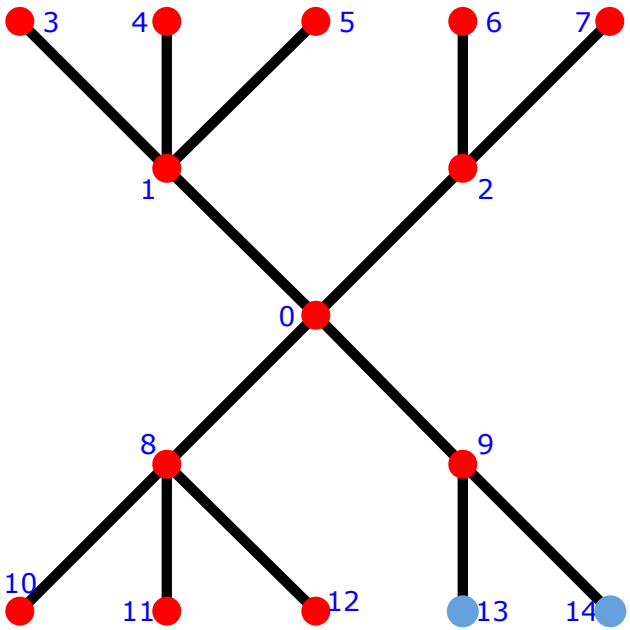
input graph



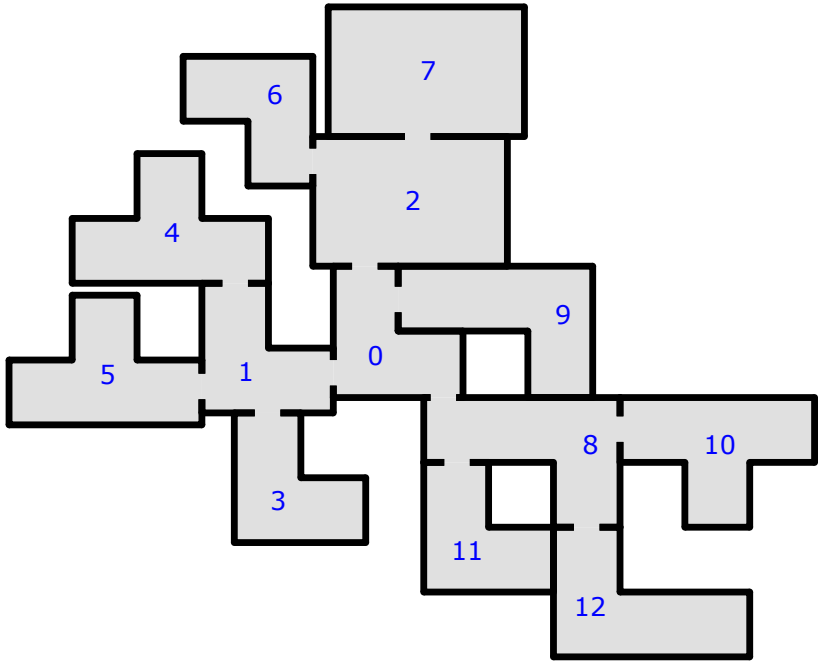
bad partial layout

Incremental layout

- Backtracking



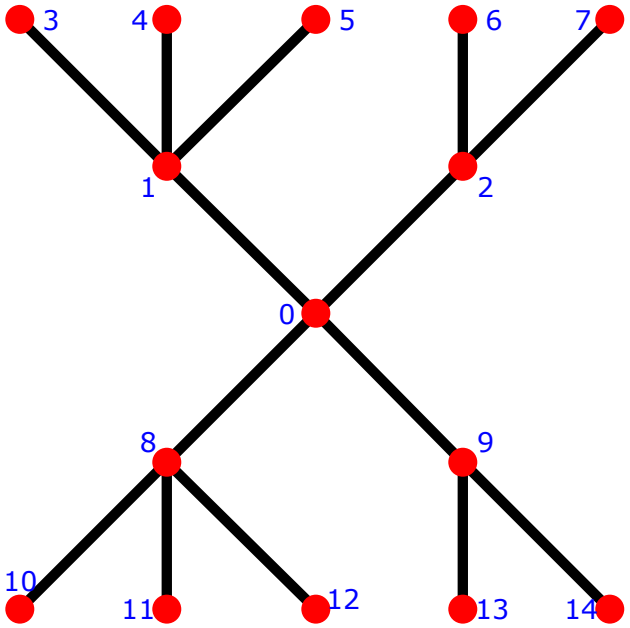
input graph



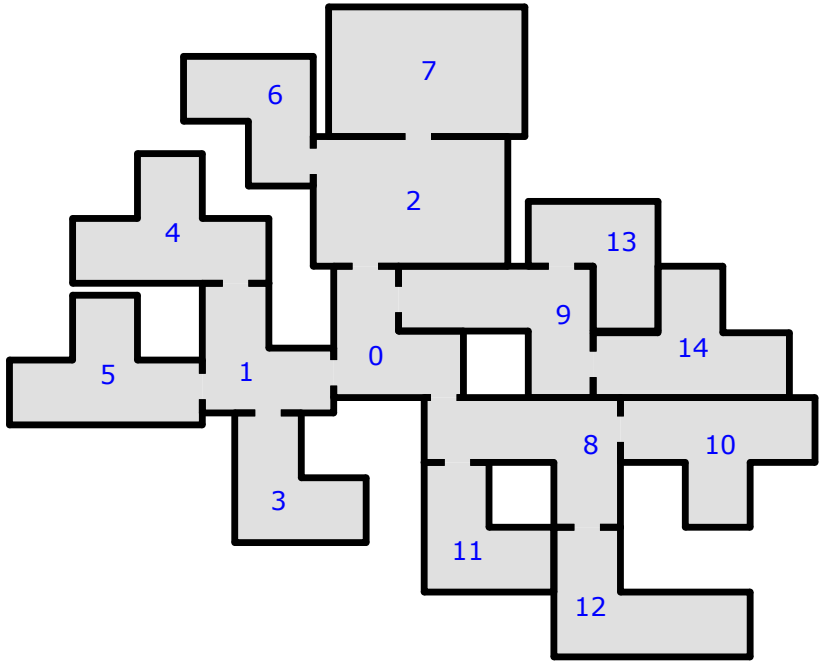
good partial layout
after backtracking

Incremental layout

- Backtracking



input graph



final solution

Algorithm

- Configuration space

- Incremental layout
  - Graph decomposition
  - Backtracking

- Chain layout
  - Iterative optimization

Chain layout

- Energy formulation
  - A : total area of intersection
  - D : sum of squared distances of pairs should be but not in contact

$$E = e^{A/\sigma} \cdot e^{D/\sigma} - 1$$

- Iterative optimization
  - Simulated annealing

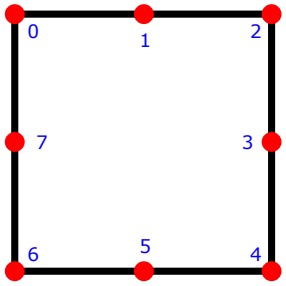
Chain layout

Pseudocode 2 Extend partial layout s adding the chain $\mathbf{c}$

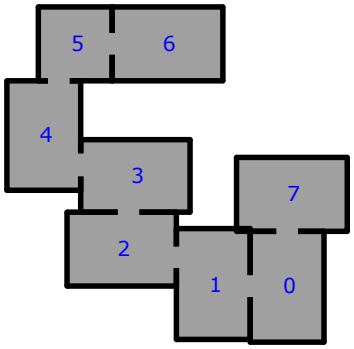
```
1:  procedure AddChain(G, B, S, c, s)
2:      t ← t_0                                          // Initial temperature
3:      for i ← 1, n do                                  // n: # of cycles in total
4:          for j ← 1, m do                              // m: # of trials per cycle
5:              s' ← Locally perturb s ∪ c
6:              if s' is valid then
7:                  if s ∪ c is full layout then output it
8:                  else if s' passes variability test
9:                      Push s' into S
10:                     Return if enough extended layouts computed
11:                 end if
12:             end if
13:             if ΔE < 0 then                           // ΔE = E(s') − E(s)
14:                 s ← s'
15:             else if rand() < e^{−ΔE/(k∗t)} then
16:                 s ← s'
17:             else
18:                 Discard s'
19:             end if
20:         end for
21:         t ← t × ratio                                // Cool down temperature
22:     end for
23: end procedure
```

Chain layout

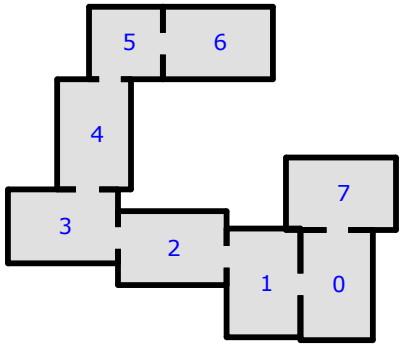
- Iterative optimization



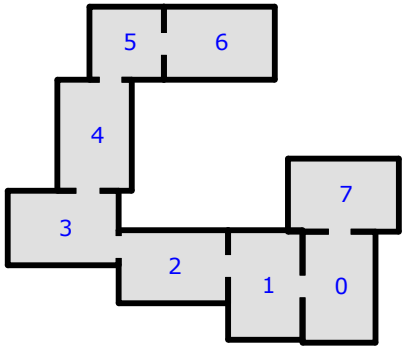
(a) Input graph



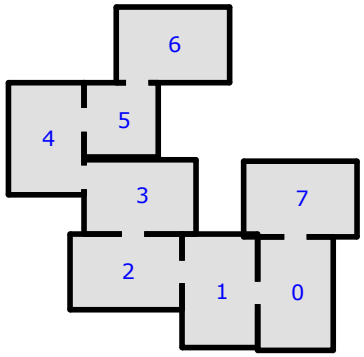
(b) Initialization



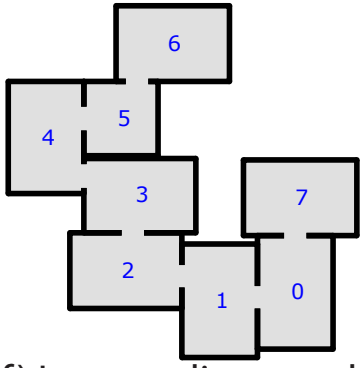
(c) Intermediate result 1



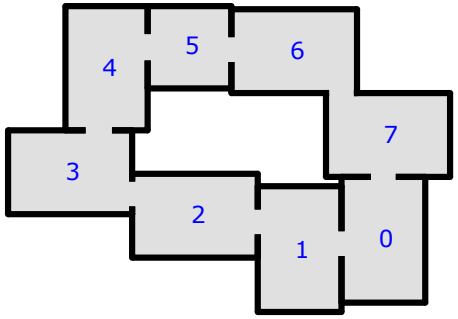
(d) Intermediate result 2



(e) Intermediate result 3

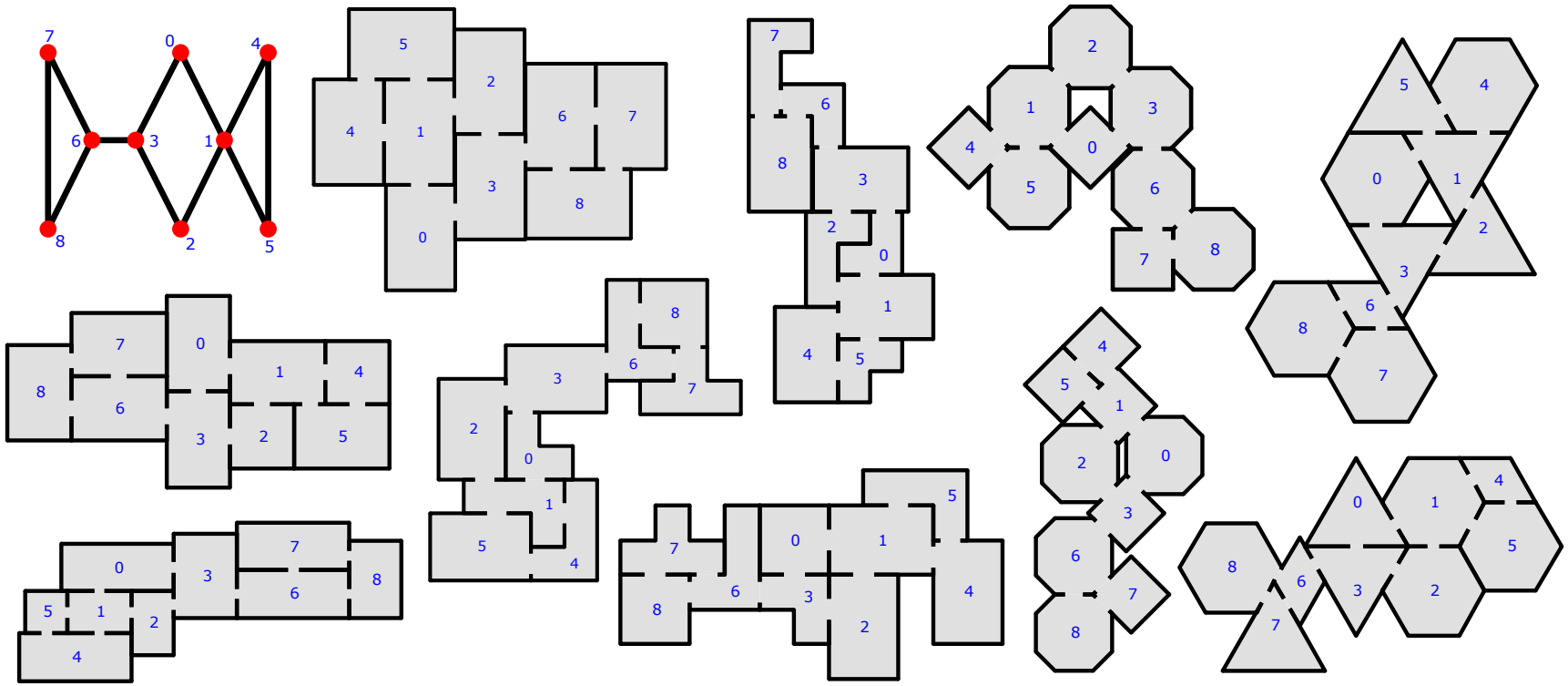


(f) Intermediate result 4

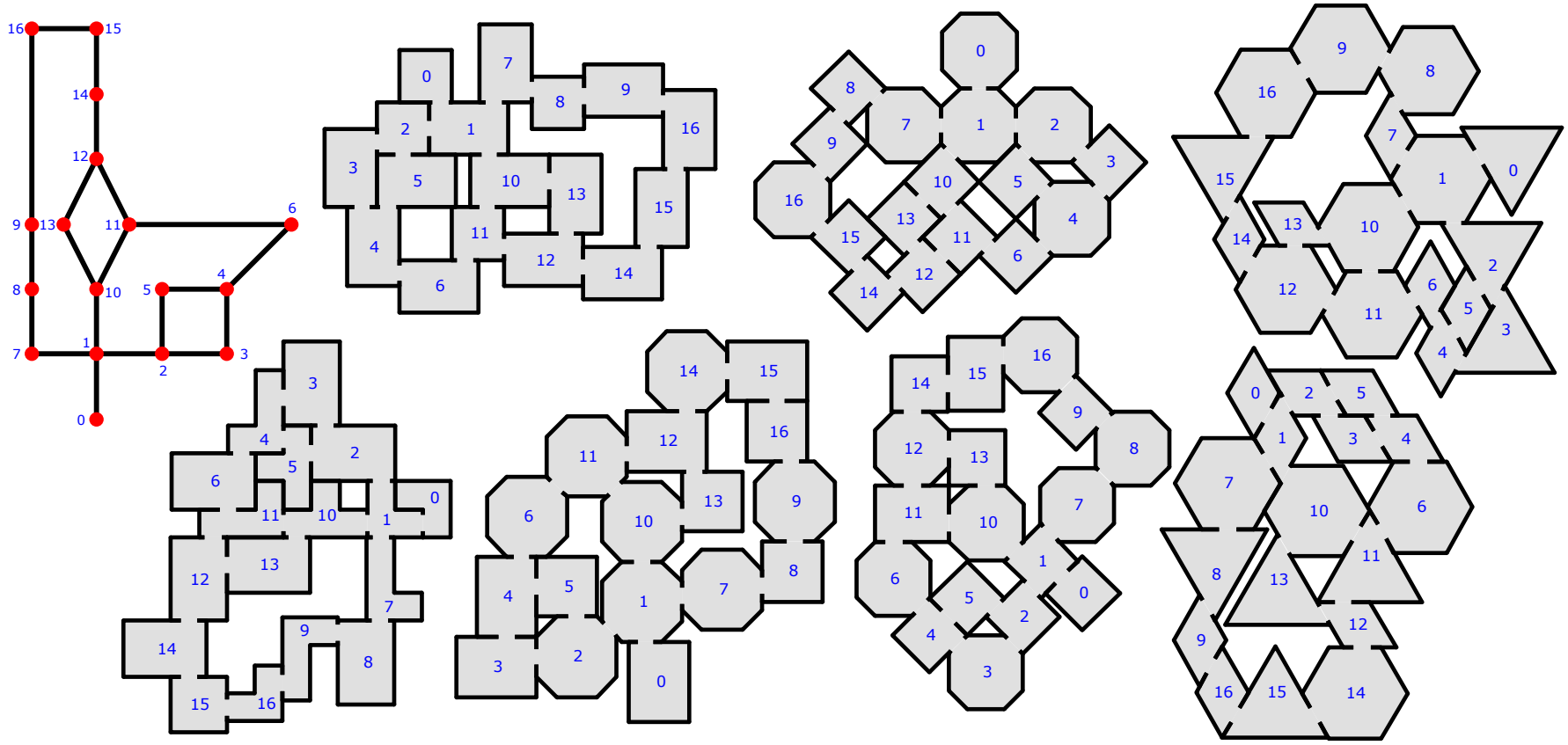


(g) Final output

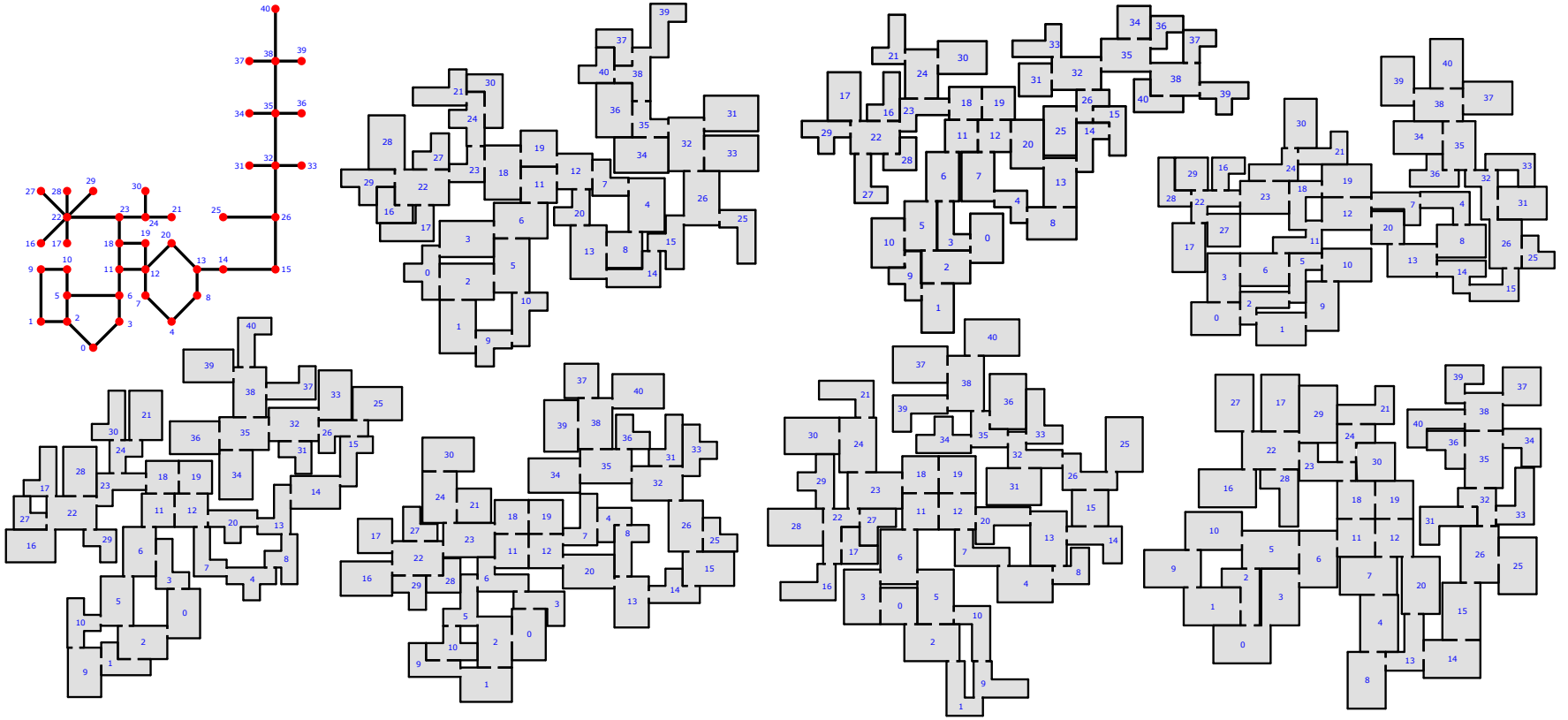
Results: different building blocks



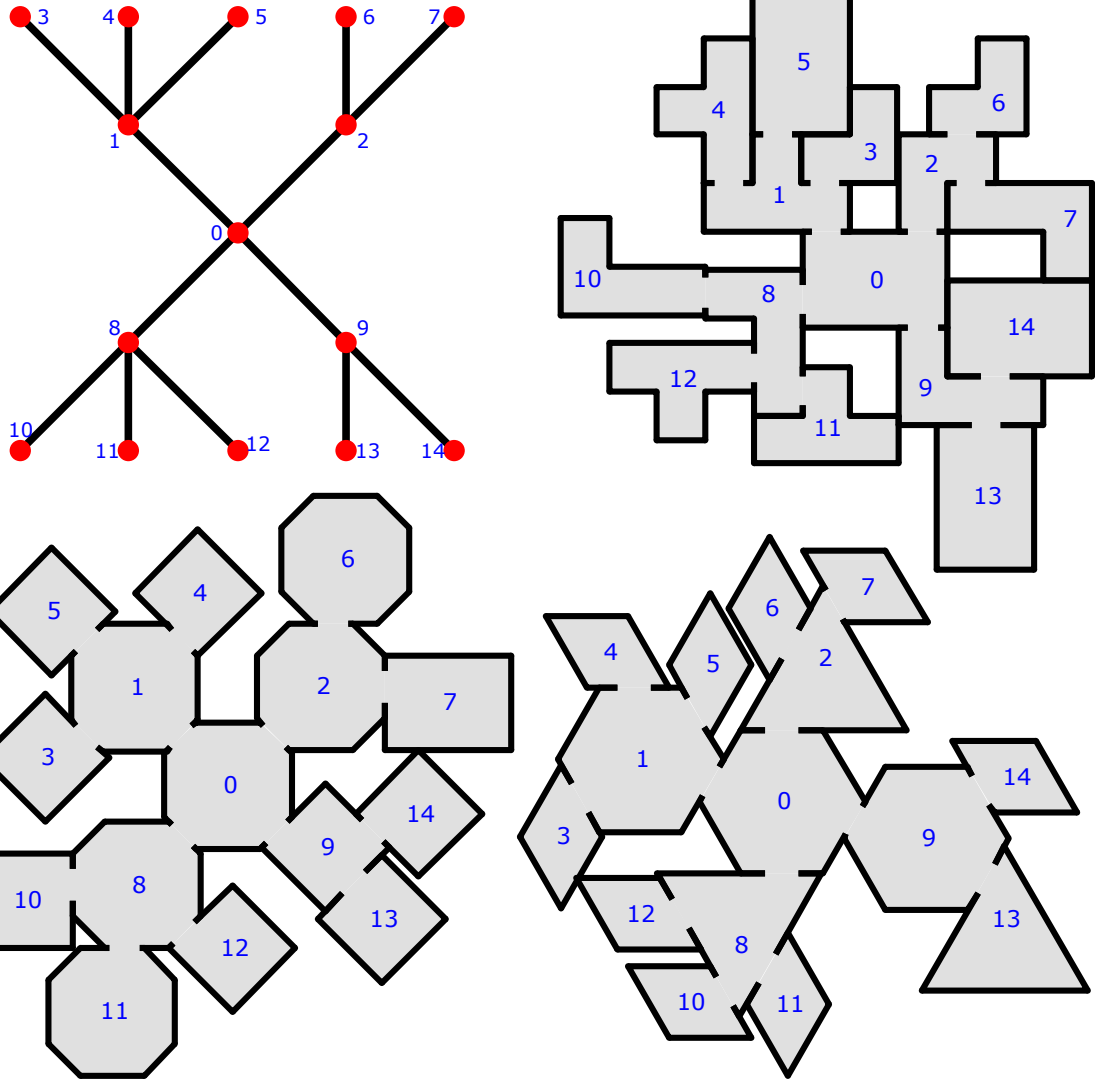
Results: different building blocks



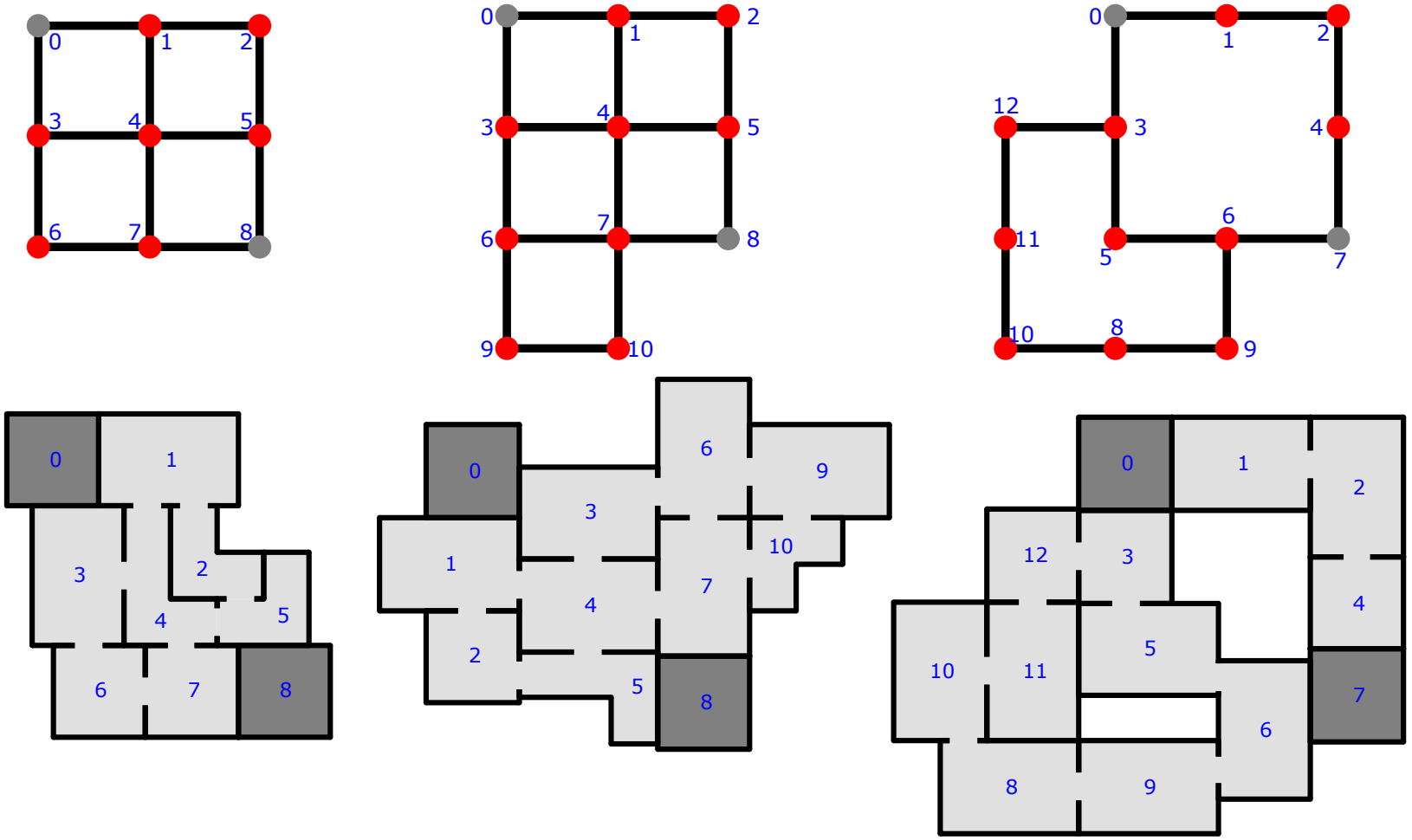
Results: large input graph



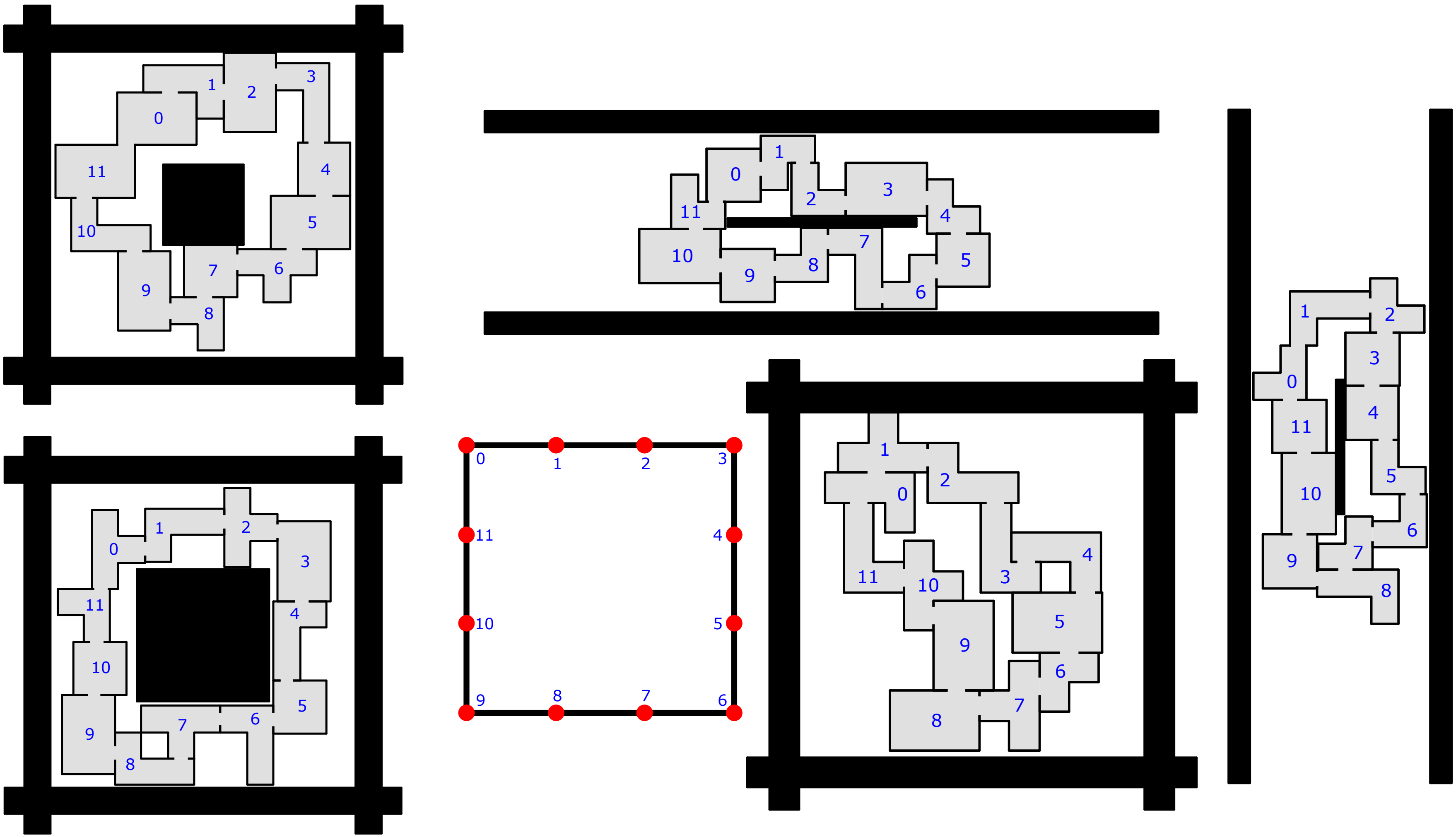
Results: high-valence tree



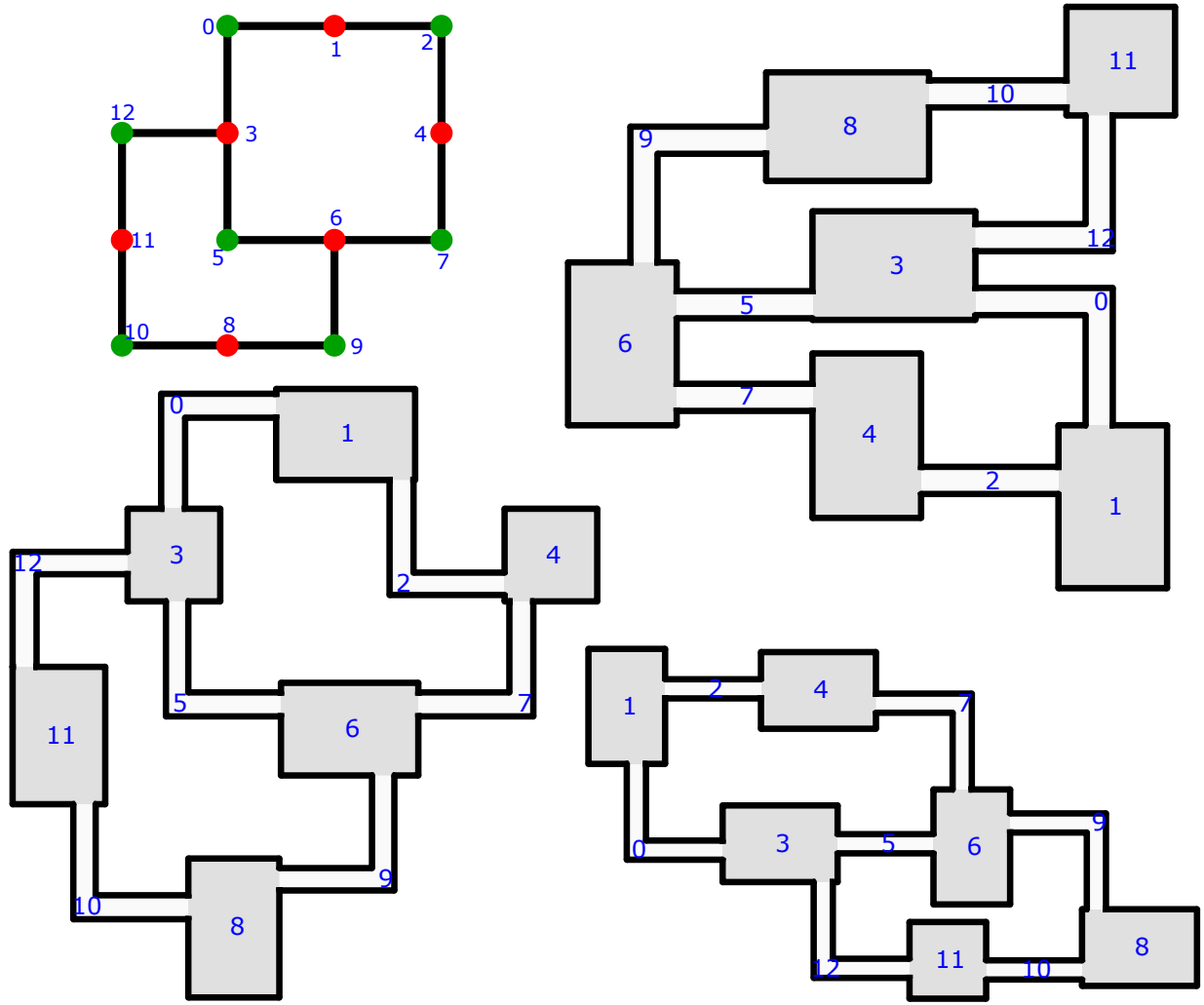
Results: multi-floor constraints



Results: boundary constraints



Results: restricted door positions



Statistics

	success rate first/10th	# of sol.	first solution	10th solution	
			time avg/med	time avg/med	iter. # avg/med
Fig 1	1/0.94	9.8	4.9/2.3	10.9/6.8	51k/33k
Fig 7, top	1/1	10	1.1/0.4	1.8/1.2	7k/4k
Fig 7, bot	0.94/0.84	9.3	23/18	48/40	229k/187k
Fig 8	0.98/0.98	10	80/55	94/73	385k/295k
Fig 9	1/1	10	1.7/0.3	2/0.6	22k/6k

Conclusion

- A novel level layout synthesis method for various design goals
- A *graph-decomposition* based layout strategy for complex connectivity
- A stochastic optimization algorithm based on *configuration space* for fast convergence

Future work

- Additional design goals
  - Production scenarios
- Speedup
  - More advanced stochastic search
- Increase output variability
  - Allow block deformation

Acknowledgements

- Anonymous reviewers

- NSERC, GRAND NCE and ERC ShapeForge (StG-2012- 307877).

Thank you!