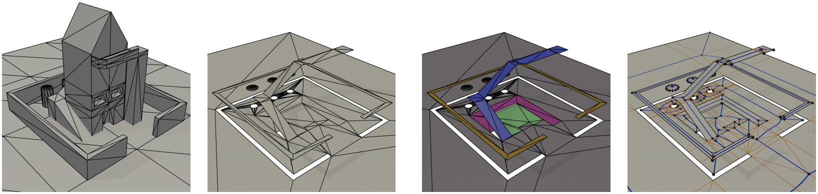
Separating a Walkable Environment into Layers

Arne Hillebrand Marjan van den Akker Roland Geraerts
Han Hoogeveen

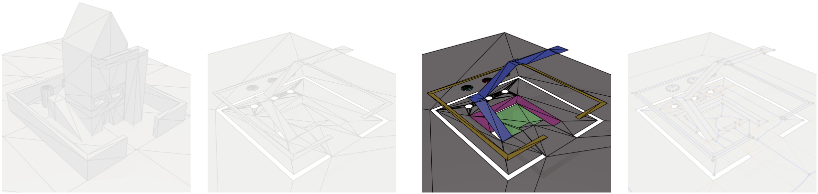
Department of Information and Computing Sciences
Utrecht University, the Netherlands

What?



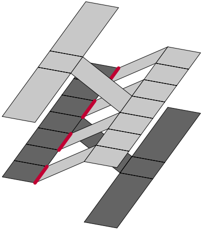
1. Obtain a 3D-model of a building;
2. Filter and repair to obtain the **walkable environment**;
3. Obtain a **multi-layered environment**;
4. Do something useful (e.g. generate a **navigation mesh**).

What?



1. Obtain a 3D-model of a building;
2. Filter and repair to obtain the **walkable environment**;
3. Obtain a **multi-layered environment**;
4. Do something useful (e.g. generate a **navigation mesh**).

Multi-layered environment



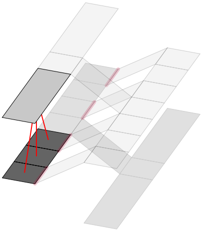
Definition (Multi-layered environment [1])

A multi-layered environment (MLE) is a set $\mathbf{L} = \{L_1, \ldots, L_l\}$ of layers and a set $\mathbf{C}$ of connections, such that:

- No two polygons P and Q in a layer L_i **overlap**;
- Iff polygons P and Q are **connected** and in different layers, the shared edge between P and Q is a **connection** in $\mathbf{C}$.
- Every polygon P is assigned to exactly one **layer**.

[1] van Toll, Cook, and Geraerts, "Navigation meshes for realistic multi-layered environments"

Multi-layered environment



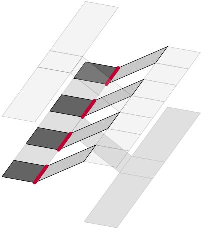
Definition (Multi-layered environment [1])

A multi-layered environment (MLE) is a set $\mathbf{L} = \{L_1, ..., L_l\}$ of layers and a set $\mathbf{C}$ of connections, such that:

- No two polygons P and Q in a layer L_i **overlap**;
- Iff polygons P and Q are **connected** and in different layers, the shared edge between P and Q is a **connection** in $\mathbf{C}$.
- Every polygon P is assigned to exactly one **layer**.

[1] van Toll, Cook, and Geraerts, "Navigation meshes for realistic multi-layered environments"

Multi-layered environment



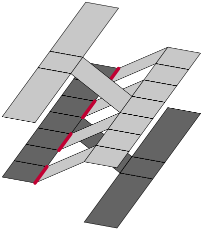
Definition (Multi-layered environment [1])

A multi-layered environment (MLE) is a set $\mathbf{L} = \{L_1, ..., L_l\}$ of layers and a set $\mathbf{C}$ of connections, such that:

- No two polygons P and Q in a layer L_i **overlap**;
- Iff polygons P and Q are **connected** and in different layers, the shared edge between P and Q is a **connection** in $\mathbf{C}$.
- Every polygon P is assigned to exactly one **layer**.

[1] van Toll, Cook, and Geraerts, "Navigation meshes for realistic multi-layered environments"

Multi-layered environment



Definition (Multi-layered environment [1])

A multi-layered environment (MLE) is a set $\mathbf{L} = \{L_1, \ldots, L_l\}$ of layers and a set $\mathbf{C}$ of connections, such that:

- No two polygons P and Q in a layer L_i **overlap**;
- Iff polygons P and Q are **connected** and in different layers, the shared edge between P and Q is a **connection** in $\mathbf{C}$.
- Every polygon P is assigned to exactly one **layer**.

[1] van Toll, Cook, and Geraerts, "Navigation meshes for realistic multi-layered environments"

Related work

Creators

- "Render" a multi-layered environment top down [1]
- Modelling tool where layer information is encoded [2]
- Create navigation mesh with a multi-layered environment as intermediate result [3]

Users

- Use a multi-layered graph of a building to clear buildings [4]
- Use a multi-layered environment to create navigation mesh [5]

---

[1] Deusdado, Fernandes, and Belo, "Path planning for complex 3D multilevel environments"

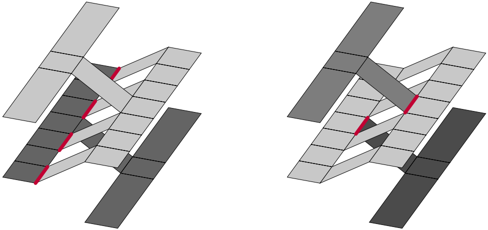
[2] Jiang et al., "A semantic environment model for crowd simulation in multilayered complex environment"

[3] Oliva and Pelechano, "NEOGEN: Near optimal generator of navigation meshes for 3D multi-layered environments"

[4] Rodriguez and Amato, "Roadmap-based level clearing of buildings"

[5] van Toll, Cook, and Geraerts, "Navigation meshes for realistic multi-layered environments"

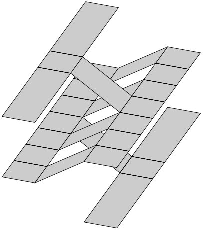
Minimally connected multi-layered environment



Definition (Minimally connected multi-layered environment)

A minimally connected multi-layered environment (MICLE) is a multi-layered environment where the number of connections is minimal.

Walkable environment graph

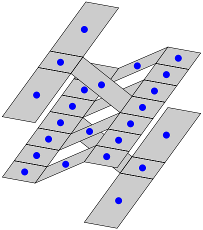


Definition (Walkable environment graph)

A walkable environment graph (WEG) is a graph representing a walkable environment with:

- A **vertex** for every polygon;
- An **edge** between every distinct pair of connected polygons;
  Associated with each edge e is also an integer **weight** $w(e)$.
- An **overlap** annotation between every distinct pair of overlapping polygons.

Walkable environment graph

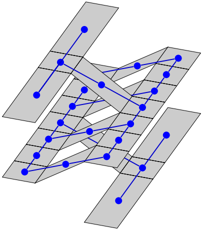


Definition (Walkable environment graph)

A walkable environment graph (WEG) is a graph representing a walkable environment with:

- A **vertex** for every polygon;
- An **edge** between every distinct pair of connected polygons; Associated with each edge e is also an integer **weight** $w(e)$.
- An **overlap** annotation between every distinct pair of overlapping polygons.

Walkable environment graph

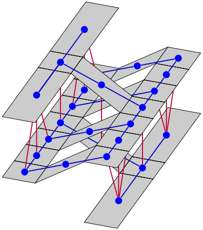


Definition (Walkable environment graph)

A walkable environment graph (WEG) is a graph representing a walkable environment with:

- A **vertex** for every polygon;
- An **edge** between every distinct pair of connected polygons;
  Associated with each edge e is also an integer **weight** $w(e)$.
- An **overlap** annotation between every distinct pair of overlapping polygons.

Walkable environment graph



Definition (Walkable environment graph)

A walkable environment graph (WEG) is a graph representing a walkable environment with:

- A **vertex** for every polygon;
- An **edge** between every distinct pair of connected polygons; Associated with each edge e is also an integer **weight** $w(e)$.
- An **overlap** annotation between every distinct pair of overlapping polygons.

Finding "good" MLEs

Minimize the cumulative weight of connections.

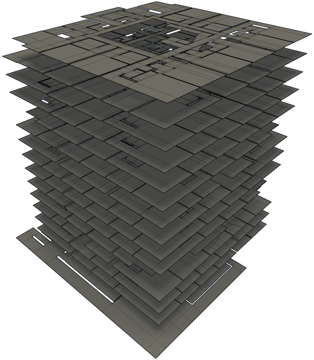
Developed three algorithms:

- ILP;
- Local search;
- Height heuristic.

Height heuristic (1/3)

Observation:

- Polygons traditionally clustered at height levels.



Height heuristic (1/3)

Observation:

- Polygons traditionally clustered at height levels.

`CLUSTER`

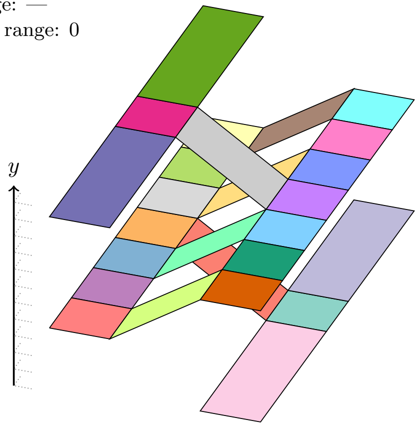
1. Create components
2. Scan bottom to top
3. Merge components in range
4. Increase range and repeat

`LOCALMIN`

1. Keep active list
2. Merge or redistribute
3. Repeat while active $\neq \emptyset$

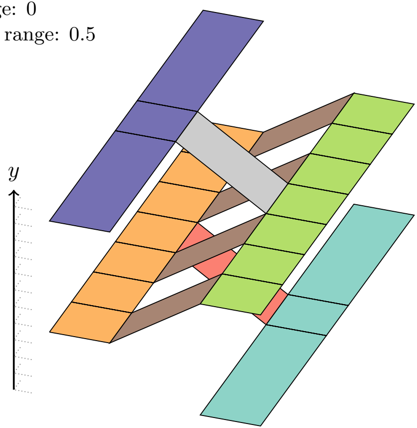
Height heuristic (2/3): `CLUSTER`

range: —
new range: 0



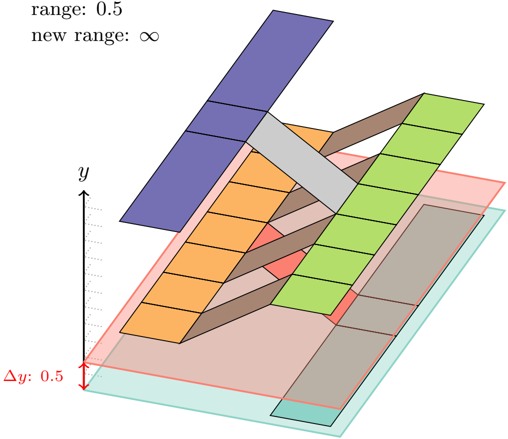
Height heuristic (2/3): CLUSTER

range: 0
new range: 0.5

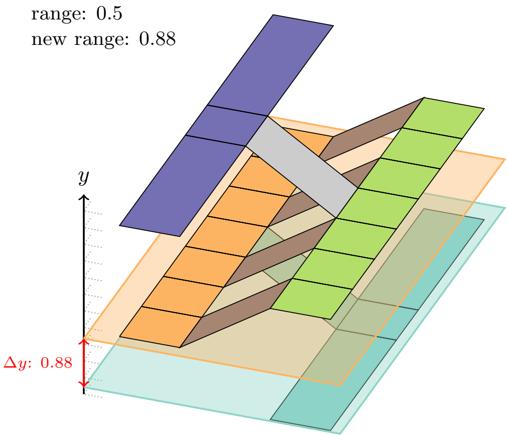


Height heuristic (2/3): CLUSTER

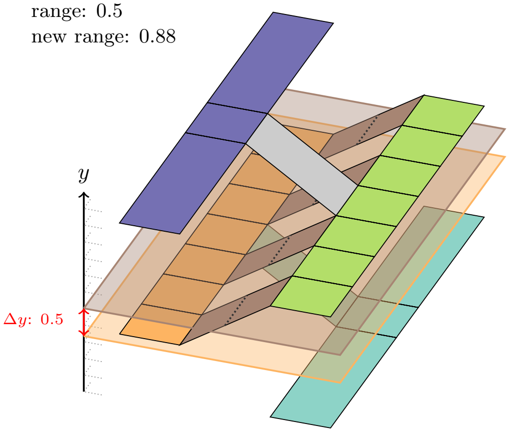
range: 0.5
new range: ∞



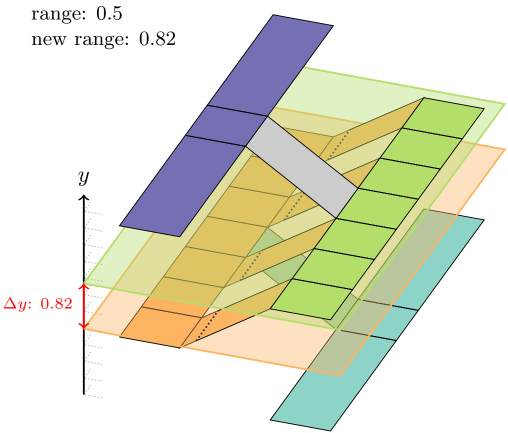
Height heuristic (2/3): CLUSTER



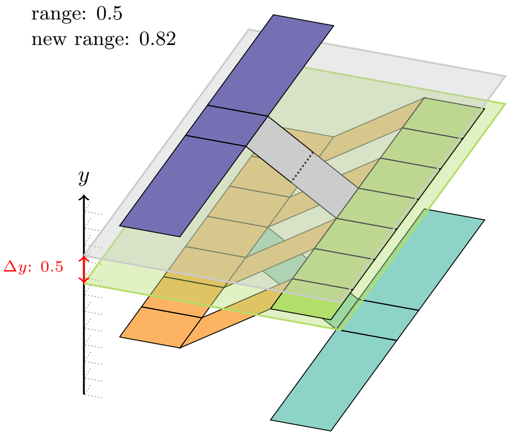
Height heuristic (2/3): `CLUSTER`



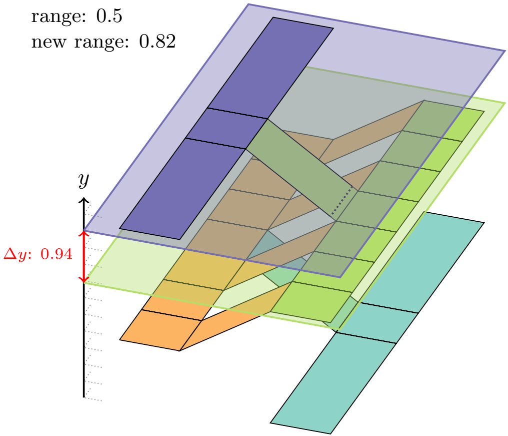
Height heuristic (2/3): CLUSTER



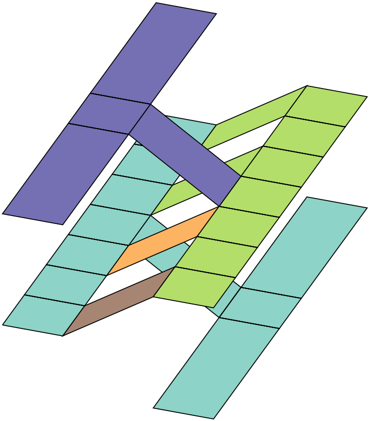
Height heuristic (2/3): CLUSTER



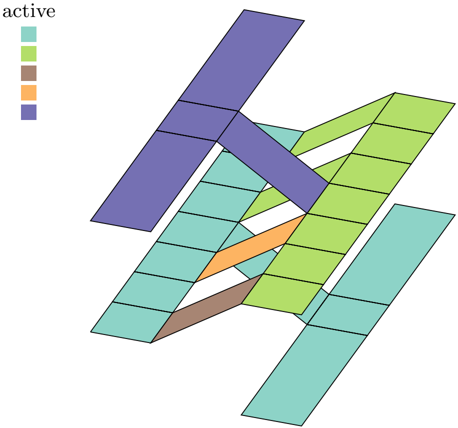
Height heuristic (2/3): CLUSTER



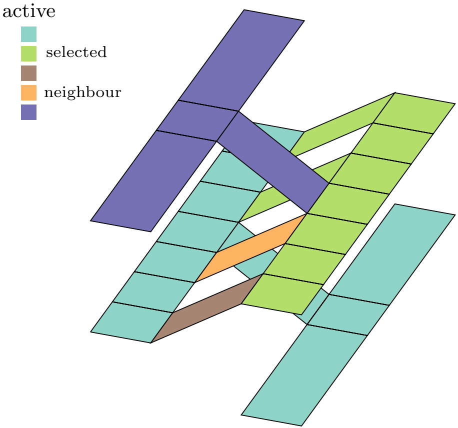
Height heuristic (3/3): `LOCALMIN`



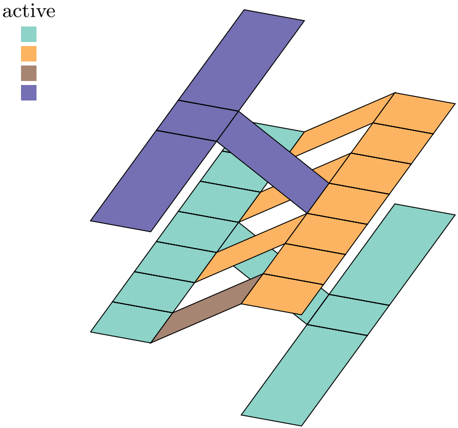
Height heuristic (3/3): LOCALMIN



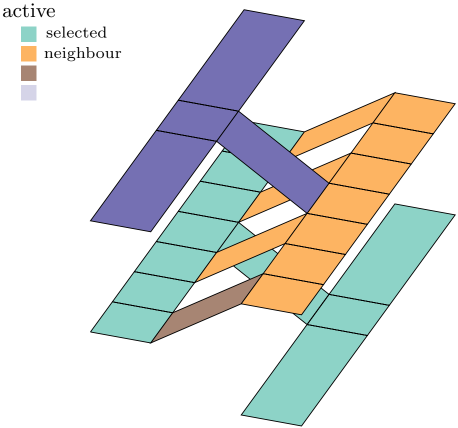
Height heuristic (3/3): `LOCALMIN`



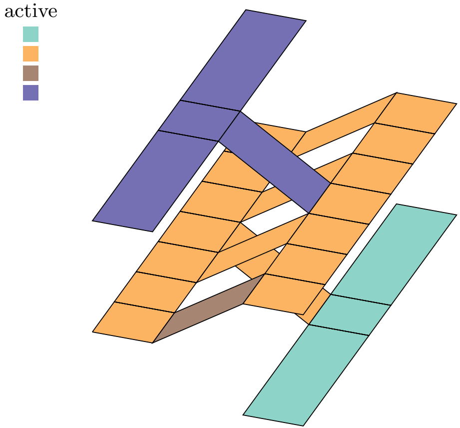
Height heuristic (3/3): `LOCALMIN`



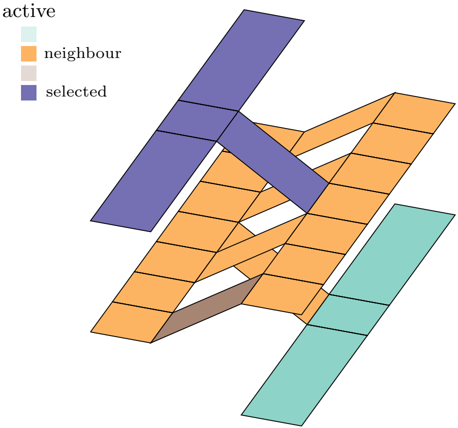
Height heuristic (3/3): LOCALMIN



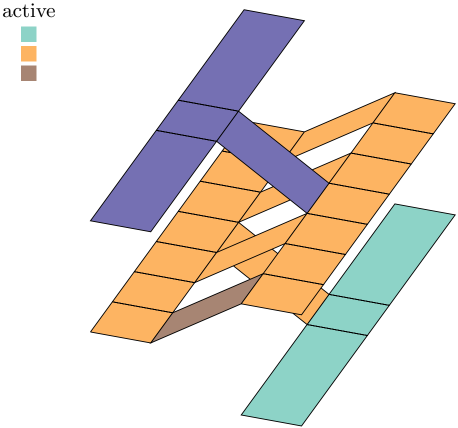
Height heuristic (3/3): `LOCALMIN`



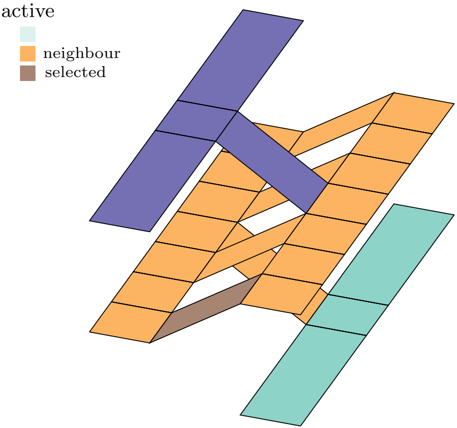
Height heuristic (3/3): LOCALMIN



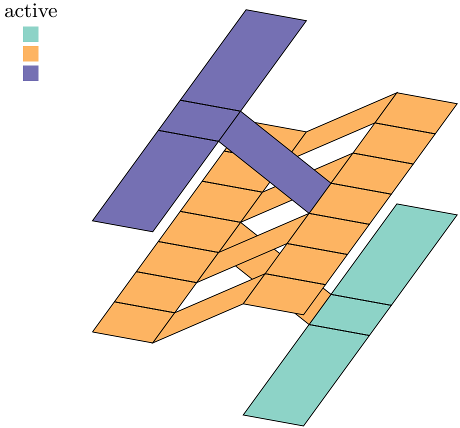
Height heuristic (3/3): LOCALMIN



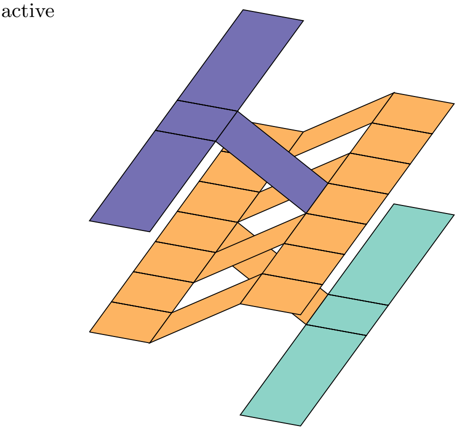
Height heuristic (3/3): `LOCALMIN`



Height heuristic (3/3): LOCALMIN



Height heuristic (3/3): LOCALMIN



What and how?

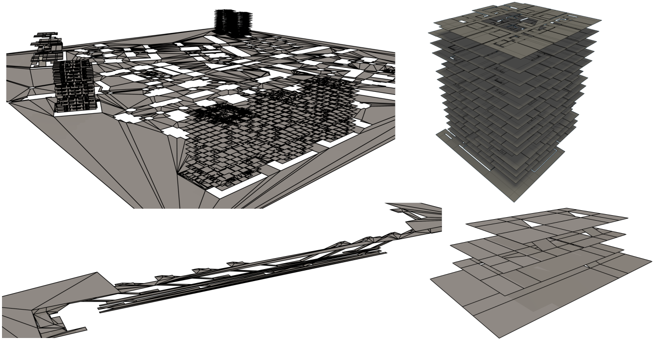
What did we want to determine?

- How do the algorithms scale;
- Quality of the solution (**number of connections**);
- Independence of geometric representation.

What and how?

How did we test this?

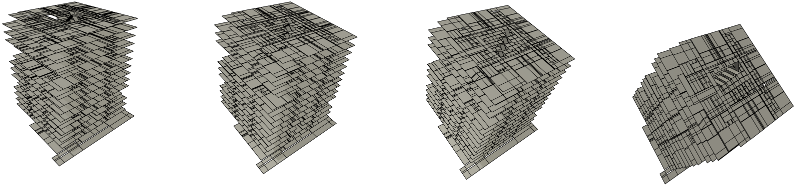
- Test different environments;
- Rotate environment;
- Compress the graph.



What and how?

How did we test this?

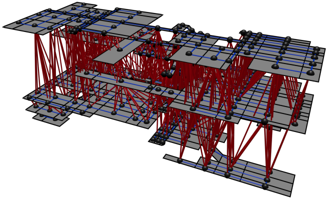
- Test different environments;
- Rotate environment;
- Compress the graph.

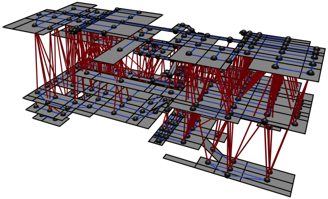


What and how?

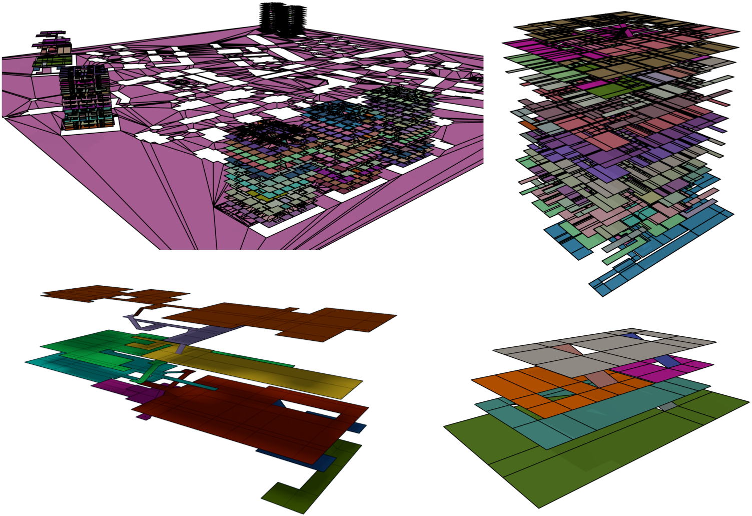
How did we test this?

- Test different environments;
- Rotate environment;
- Compress the graph.





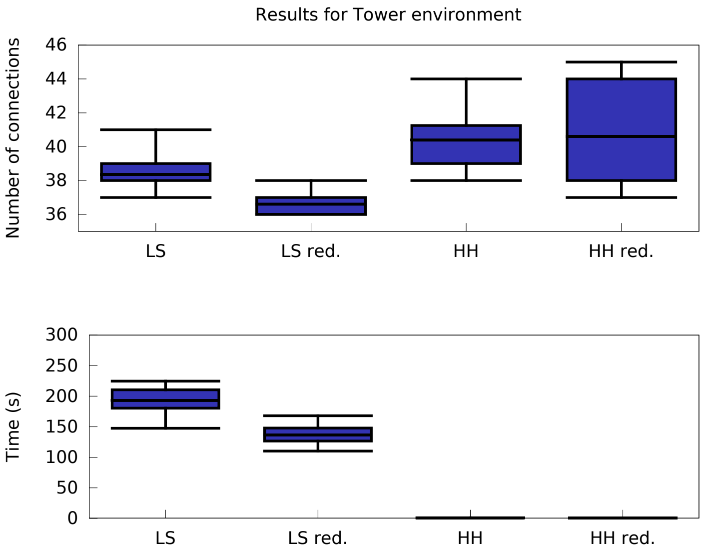
Results



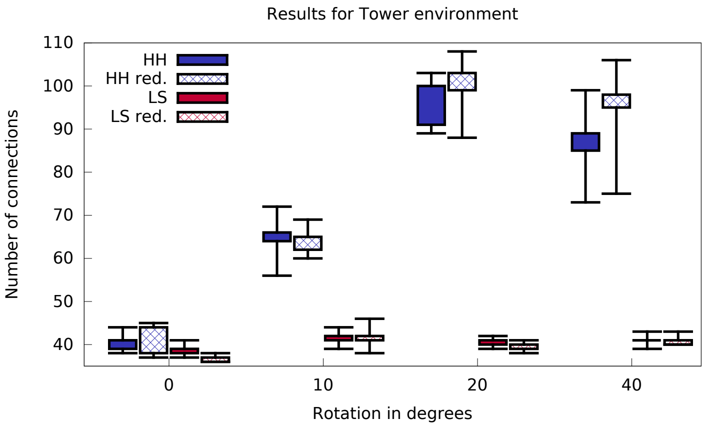
Results (2/2)

	Environment	local search $\lvert C\rvert$	local search $\lvert C\rvert$ sd	local search t (ms)	local search t sd	HH $\lvert C\rvert$	HH $\lvert C\rvert$ sd	HH t (ms)	HH t sd	ILP $\lvert C\rvert$	ILP $\lvert C\rvert$ sd	ILP t (ms)	ILP t sd
Original WEG	As_oilrig	25.00	0.56	$5.27e^{4}$	$2.99e^{3}$	28.50	2.12	$9.27e^{1}$	$1.08e^{1}$	—	—	—	—
	Halo	4.00	0.00	$6.21e^{3}$	$3.47e^{2}$	4.35	0.67	$3.15e^{0}$	$3.66e^{-1}$	4.00	0.00	—	—
	Cliffsides	4.00	0.00	$1.99e^{4}$	$9.42e^{2}$	4.00	0.00	$3.19e^{1}$	$4.11e^{0}$	—	—	—	—
	Hexagon	29.20	0.70	$1.09e^{4}$	$6.51e^{2}$	31.00	1.45	$1.56e^{2}$	$8.96e^{0}$	57.50	8.02	—	—
	Library	8.25	0.44	$6.41e^{3}$	$1.71e^{2}$	9.00	0.00	$6.30e^{0}$	$5.71e^{-1}$	—	—	—	—
	Tower	38.35	1.14	$1.93e^{5}$	$2.06e^{4}$	40.40	1.60	$8.85e^{2}$	$8.65e^{1}$	65.00	0.00	—	—
	Station 1	15.20	0.41	$3.33e^{3}$	$1.95e^{2}$	17.25	0.55	$6.85e^{0}$	$3.66e^{-1}$	—	—	—	—
	Station 2	6.00	0.00	$1.62e^{3}$	$5.40e^{1}$	6.00	0.00	$1.00e^{0}$	$0.00e^{0}$	6.00	0.00	$4.56e^{6}$	$3.67e^{3}$
	Parking lot	8.00	0.00	$1.40e^{3}$	$4.61e^{1}$	8.00	0.00	$1.00e^{0}$	$0.00e^{0}$	8.00	0.00	$2.38e^{4}$	$4.56e^{2}$
	City	392.95	4.70	$6.88e^{5}$	$3.39e^{4}$	444.55	6.53	$3.05e^{4}$	$1.55e^{3}$	25773.25	359.29	—	—
	Tower 10	41.60	1.14	$1.53e^{5}$	$1.21e^{4}$	64.60	4.13	$8.77e^{2}$	$8.26e^{1}$	551.25	26.27	—	—
	Tower 20	40.30	0.73	$1.88e^{5}$	$1.17e^{4}$	95.00	4.87	$1.04e^{3}$	$1.86e^{2}$	870.00	48.87	—	—
	Tower 40	41.00	0.92	$1.28e^{5}$	$1.17e^{4}$	87.00	6.36	$7.06e^{2}$	$7.49e^{1}$	—	—	—	—
Reduced WEG	As_oilrig	26.05	0.89	$2.37e^{4}$	$1.69e^{3}$	27.95	0.89	$5.29e^{1}$	$2.31e^{0}$	—	—	—	—
	Halo	4.00	0.00	$2.65e^{3}$	$1.34e^{2}$	5.00	0.00	$2.00e^{0}$	$0.00e^{0}$	4.00	0.00	$4.39e^{6}$	$4.89e^{3}$
	Cliffsides	4.00	0.00	$1.01e^{3}$	$4.99e^{1}$	4.00	0.00	$6.05e^{0}$	$2.24e^{-1}$	—	—	—	—
	Hexagon	29.60	1.14	$2.07e^{3}$	$1.50e^{2}$	30.65	0.49	$3.69e^{1}$	$4.89e^{-1}$	21.33	9.29	—	—
	Library	8.10	0.31	$3.34e^{3}$	$1.46e^{2}$	9.00	0.00	$4.05e^{0}$	$2.24e^{-1}$	8.00	0.00	—	—
	Tower	36.60	0.68	$1.37e^{5}$	$1.53e^{4}$	40.60	3.02	$5.68e^{2}$	$4.19e^{1}$	845.25	85.80	—	—
	Station 1	16.25	0.72	$1.25e^{3}$	$8.81e^{1}$	19.00	0.00	$2.05e^{0}$	$2.24e^{-1}$	—	—	—	—
	Station 2	6.00	0.00	$5.91e^{2}$	$1.50e^{1}$	6.00	0.00	$0.00e^{0}$	$0.00e^{0}$	6.00	0.00	$9.00e^{2}$	$2.02e^{0}$
	Parking lot	8.00	0.00	$1.05e^{3}$	$2.88e^{1}$	8.65	0.49	$1.00e^{0}$	$0.00e^{0}$	8.00	0.00	$2.44e^{5}$	$2.14e^{3}$
	Tower 10	41.35	1.66	$1.03e^{5}$	$6.85e^{3}$	63.65	2.21	$4.90e^{2}$	$3.16e^{1}$	1232.75	47.80	—	—
	Tower 20	39.55	0.83	$1.03e^{5}$	$7.79e^{3}$	100.30	4.51	$5.94e^{2}$	$7.38e^{1}$	—	—	—	—
	Tower 40	40.90	0.97	$9.12e^{4}$	$6.21e^{3}$	95.55	7.05	$4.98e^{2}$	$3.96e^{1}$	—	—	—	—

Analysis (1/3): Tower



Analysis (2/3): Geometric independence



Analysis (3/3)

27 tables condensed in one normal table and two 3D tables.

Speed:

- HH is **faster** than LS for 12/13 environments;
- No statistical difference between HH red. and HH;
- LS is **slower** than LS red. for 12/13 environments.

Quality:

- HH finds **lower** $|\mathbf{C}|$ than HH red. in 5/13 cases;
- HH finds **higher** $|\mathbf{C}|$ than LS in 8/13 cases;
- LS and HH have **higher** $|\mathbf{C}|$ when environment is rotated.
  Although effect is less pronounced for LS

All above statements are statistically significant for $\alpha = 0.001$

Conclusion

We have:

- Identified a common (sub-)problem: finding an MLE;
- Implemented and tested three algorithms.
  Height heuristic, local search and ILP

In the future:

- Work on the first step in the pipeline (extracting a WE);
- Extend 2D algorithms to multi-layered environments.