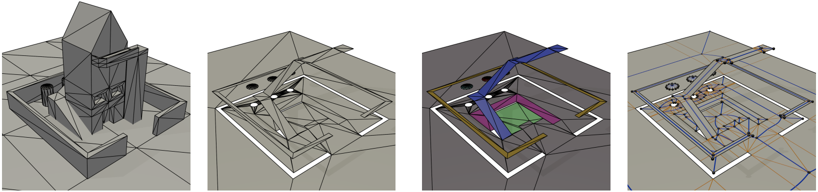
Performing multicut on walkable environments

Arne Hillebrand Marjan van den Akker Roland Geraerts
Han Hoogeveen

Department of Information and Computing Sciences
Utrecht University, the Netherlands

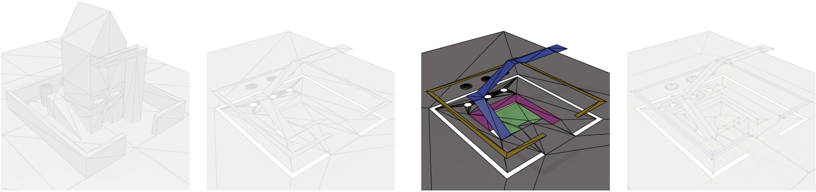
10th Annual International Conference on Combinatorial Optimization and Applications, December 16, 2016

What?



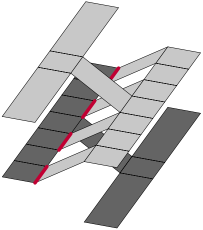
1. Obtain a 3D-model of a building;
2. Filter and repair to obtain the **walkable environment**;
3. Obtain a **multi-layered environment**;
4. Do something useful (e.g. generate a **navigation mesh**).

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



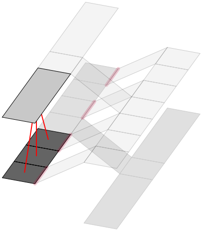
Definition (Multi-layered environment [1])

A multi-layered environment (MLE) is a set $\mathbf{L} = \{L_1, ..., L_l\}$ 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 IV, and Geraerts, "Navigation meshes for realistic multi-layered environments"

Multi-layered environment



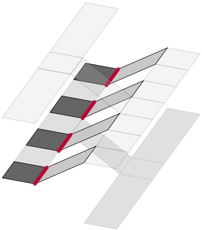
Definition (Multi-layered environment [1])

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 IV, and Geraerts, "Navigation meshes for realistic multi-layered environments"

Multi-layered environment



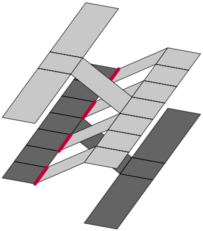
Definition (Multi-layered environment [1])

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 IV, and Geraerts, "Navigation meshes for realistic multi-layered environments"

Multi-layered environment



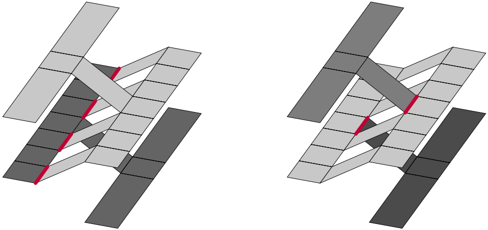
Definition (Multi-layered environment [1])

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 IV, and Geraerts, "Navigation meshes for realistic multi-layered environments"

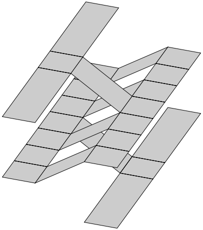
Minimally connected multi-layered environment



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

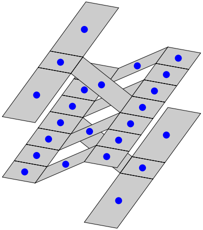


Definition (Walkable environment graph)

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;
- An **overlap** annotation between every distinct pair of overlapping polygons.

Walkable environment graph

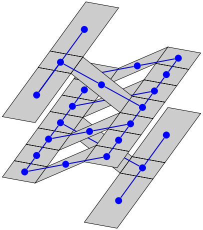


Definition (Walkable environment graph)

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;
- An **overlap** annotation between every distinct pair of overlapping polygons.

Walkable environment graph

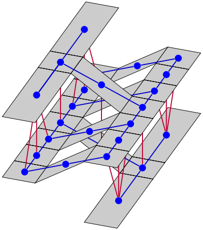


Definition (Walkable environment graph)

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;
- An **overlap** annotation between every distinct pair of overlapping polygons.

Walkable environment graph



Definition (Walkable environment graph)

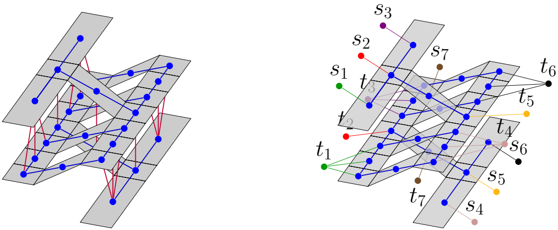
A walkable environment graph (WEG) is a graph representing a walkable environment with:

- A **vertex** for every polygon;
- An **edge** between every distinct pair of connected polygons;
- An **overlap** annotation between every distinct pair of overlapping polygons.

How hard can it be? (1/2)

Finding a MICLE can be formulated as a **MULTICUT** problem [1].

1. Create source s_x and sink t_x for a vertex v with overlaps;
2. Connect s_x to v;
3. Connect vertices overlapped by v to t_x and remove overlaps;
4. Repeat while there are still overlaps.



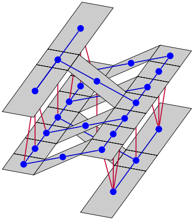
[1] Schrijver, *Combinatorial Optimization - Polyhedra And Efficiency*

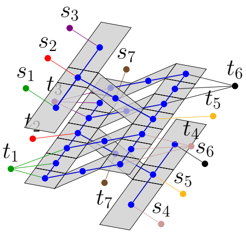
How hard can it be? (1/2)

Finding a MICLE can be formulated as a **MULTICUT** problem [1].

Unfortunately:

- MULTICUT is NP-Hard;
- MULTICUT is APX-Hard.

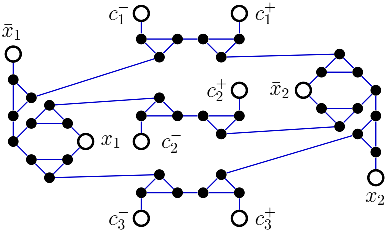




[1] Schrijver, *Combinatorial Optimization - Polyhedra And Efficiency*

How hard can it be? (2/2)

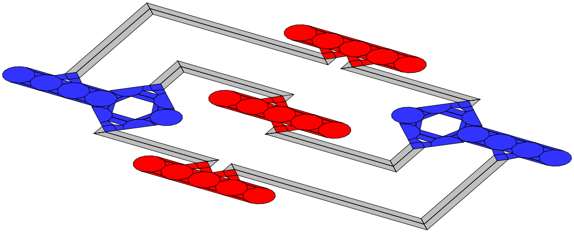
In this paper we have proven that finding a MICLE is NP-Hard. This proof is based on earlier work of Dahlhaus et al. [1]



[1] Dahlhaus et al., "The Complexity of Multiterminal Cuts"

How hard can it be? (2/2)

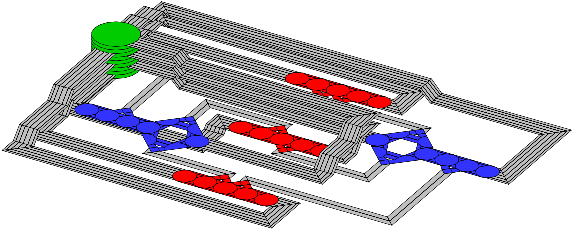
In this paper we have proven that finding a MICLE is NP-Hard.
This proof is based on earlier work of Dahlhaus et al. [1]



[1] Dahlhaus et al., "The Complexity of Multiterminal Cuts"

How hard can it be? (2/2)

In this paper we have proven that finding a MICLE is NP-Hard.
This proof is based on earlier work of Dahlhaus et al. [1]



[1] Dahlhaus et al., "The Complexity of Multiterminal Cuts"

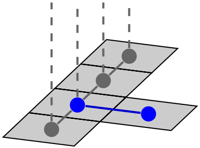
The general idea

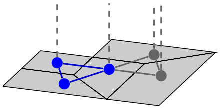
Observations:

- A WEG can contain a very large number of edges, vertices and overlaps.
- In some situations, vertices will always be part of the same layer.
- Some overlaps are redundant since their “constraint” is already enforced by other overlaps.

Try to efficiently **detect** these situations and remove them from a WEG. After each operation, keep track of possible new **candidate** vertices.

Vertex and edge reductions





1-CONTRACT

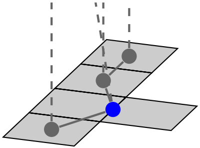
Detection: $O(1)$
Resolution: $O(p)$
Candidates: $O(1)$

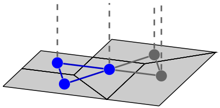
2-CONTRACT

Detection: $O(1)$
Resolution: $O(p)$
Candidates: $O(1)$

- p is the number of polygons a vertex represents.

Vertex and edge reductions





1-CONTRACT

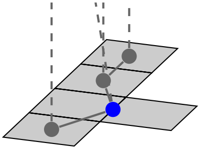
Detection: $O(1)$
Resolution: $O(p)$
Candidates: $O(1)$

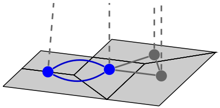
2-CONTRACT

Detection: $O(1)$
Resolution: $O(p)$
Candidates: $O(1)$

- p is the number of polygons a vertex represents.

Vertex and edge reductions





1-CONTRACT

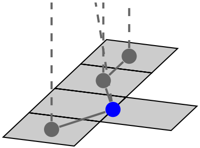
Detection: $O(1)$
Resolution: $O(p)$
Candidates: $O(1)$

2-CONTRACT

Detection: $O(1)$
Resolution: $O(p)$
Candidates: $O(1)$

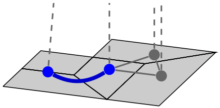
- p is the number of polygons a vertex represents.

Vertex and edge reductions



1-CONTRACT

Detection: $O(1)$
Resolution: $O(p)$
Candidates: $O(1)$



2-CONTRACT

Detection: $O(1)$
Resolution: $O(p)$
Candidates: $O(1)$

- p is the number of polygons a vertex represents.

Overlap removal

In general:

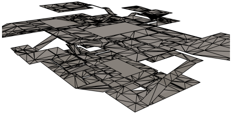
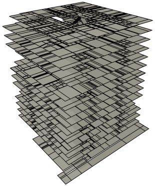
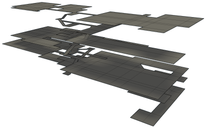
An overlap (v, w) can be removed if for every path connecting v and w there is a pair of overlapping vertices on that path.

! Very expensive to check

d-REMOVE

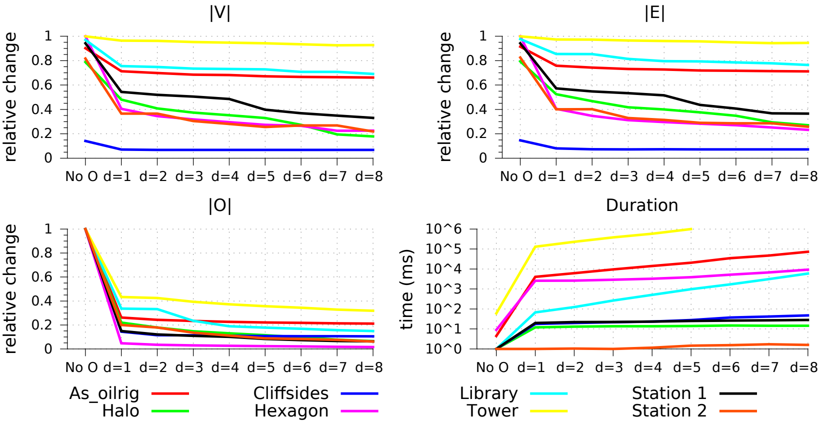
- Look at all sub-paths of length d originating in v
- For each sub-path register the overlaps
- Perform BFS from the end of each sub-path and search for w

Environments

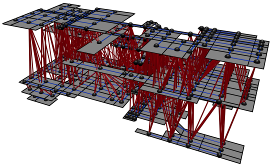




Environment	**Type**	**Tri.**	$\vert V\vert$	$\vert E\vert$	$\vert O\vert$
As_oilrig	V	✓	2077	2399	10717
Halo	V	✓	179	184	346
Cliffsides	V	✓	748	764	162
Hexagon	V	✓	2368	2419	20207
Library	R	✗	298	420	775
Tower	R	✗	5932	8033	116983
Station 1	R	✓	206	209	1026
Station 2	R	✓	82	86	115

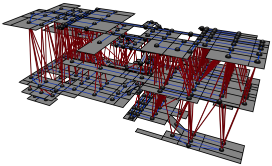
Results


|V|
relative change
1
0.8
0.6
0.4
0.2
0
No O d=1 d=2 d=3 d=4 d=5 d=6 d=7 d=8
|E|
relative change
1
0.8
0.6
0.4
0.2
0
No O d=1 d=2 d=3 d=4 d=5 d=6 d=7 d=8
|O|
relative change
1
0.8
0.6
0.4
0.2
0
No O d=1 d=2 d=3 d=4 d=5 d=6 d=7 d=8
Duration
time (ms)
10^6
10^5
10^4
10^3
10^2
10^1
10^0
No O d=1 d=2 d=3 d=4 d=5 d=6 d=7 d=8
As_oilrig
Halo
Cliffsides
Hexagon
Library
Tower
Station 1
Station 2

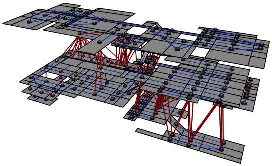
Results



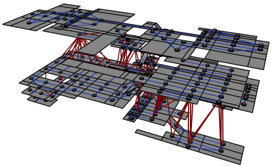
Original WEG



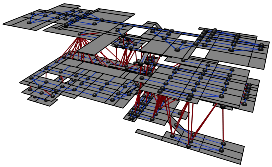
$d = 1$



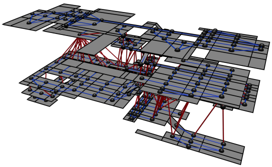
$d = 2$



$d = 3$



$d = 4$



$d = 5$

Conclusion

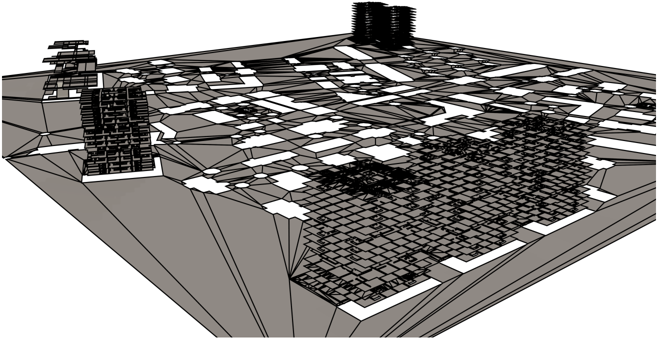
We have:

- Identified a common (sub-)problem: finding a MICLE;
- Proven this problem to be NP-Hard;
- Implemented and tested algorithms to shrink the problem size.

In the future:

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

Thanks!



A.Hillebrand@uu.nl
http://www.cs.uu.nl/staff/hillebra.html