

SDCEL - Scalable overlay operations over DCEL polygon layers

① What is a DCEL?

Doubly Connected Edge List

An edge-based spatial data structure...

collects info: topological, geometric

about: Vertices (Points), Edges (Lines), Faces (Polygons)

Table 1: Vertex records.

vertex	coordinates	incident edge
a	(0,2)	\overrightarrow{ba}
b	(2,0)	\overrightarrow{ab}
c	(2,4)	\overrightarrow{dc}
...

Table 2: Face records.

face	boundary edge	hole list
f_1	\overrightarrow{ab}	nil
f_2	\overrightarrow{fe}	nil
f_3	nil	nil

Table 3: Half-edge records.

half-edge	origin	face	twin	next	prev
\overrightarrow{fe}	f	f_2	\overrightarrow{ef}	\overrightarrow{ec}	\overrightarrow{df}
\overrightarrow{ec}	c	f_3	\overrightarrow{ce}	\overrightarrow{cb}	\overrightarrow{dc}
\overrightarrow{cb}	b	f_1	\overrightarrow{bc}	\overrightarrow{ba}	\overrightarrow{fb}
...

Example

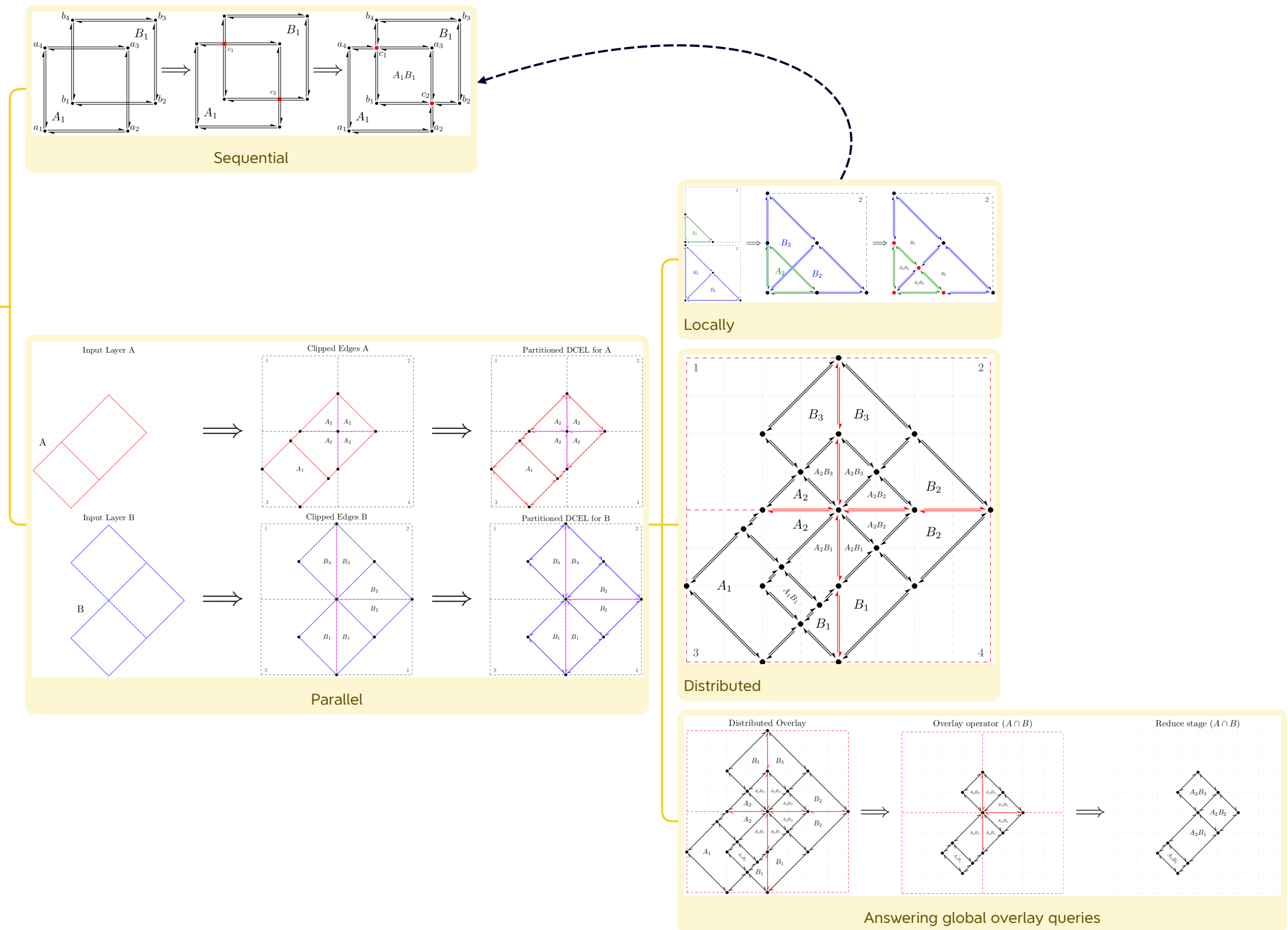
Applications: Polygon triangulation, Surveillance (art gallery problem), Circuit board printing, Robot motion planing

Overlay maps and operators — once built overlay operators are linear

② Challenges and contributions

- Multiple operation over the same DCEL (chaining)
- Just sequential implementation for now
- Unabel to deal with LARGE datasets — i.e. US Census tracks at national level
- A Scalable and distributed approach to compute overlay between two DCEL Layers**
- Addresses issues related to holes and large empty areas.
- Offers optimizations during the overlay computation.

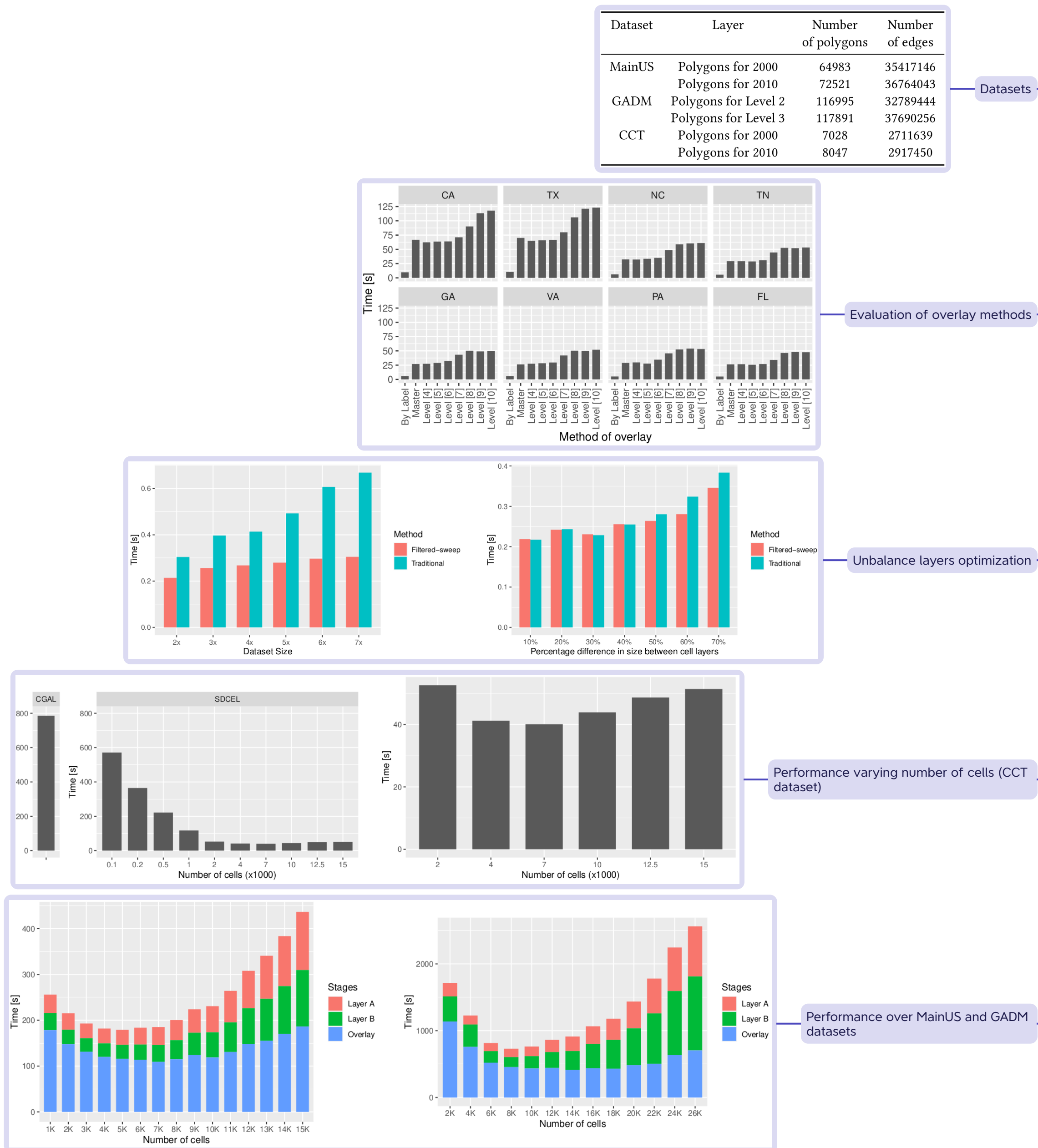
③ Sequential vs Parallel



Conclusions

- A scalable approach to compute the overlay operation among two DCEL layers has been presented.
- We introduced a partition strategy which make possible the construction of a distributed and scalable DCEL.
- We fix challenges of the parallel approximation and proposed several optimization to improve performance.
- Our experiment in real datasets show very good performance. Very large layers are able to be overlayed in few minutes.

④ Experiments



⑤ Optimizations

- Naive approach: send all to master node
- Intermediate reducers: user provides a quadtree level to partially process
- Repartition by face's label: avoid reduce phase but imply an additional shuffle
- Finding intersections is critical during the overlay computation
- Sweep-line algorithm runs over all the edges
- We scan the larger dataset for x-intervals which intersect the smaller dataset
- Find intersections just for those x-intervals

⑥ The orphan cell problem

