FAST MAP GENERALIZATION HEURISTIC WITH A UNIFORM GRID

Salles V. G. Magalhães
Marcus V. A. Andrade
Universidade Federal de Viçosa, MG, Brazil

W. Randolph Franklin
Wenli Li
Rensselaer Polytechnic Institute, Troy, NY, USA

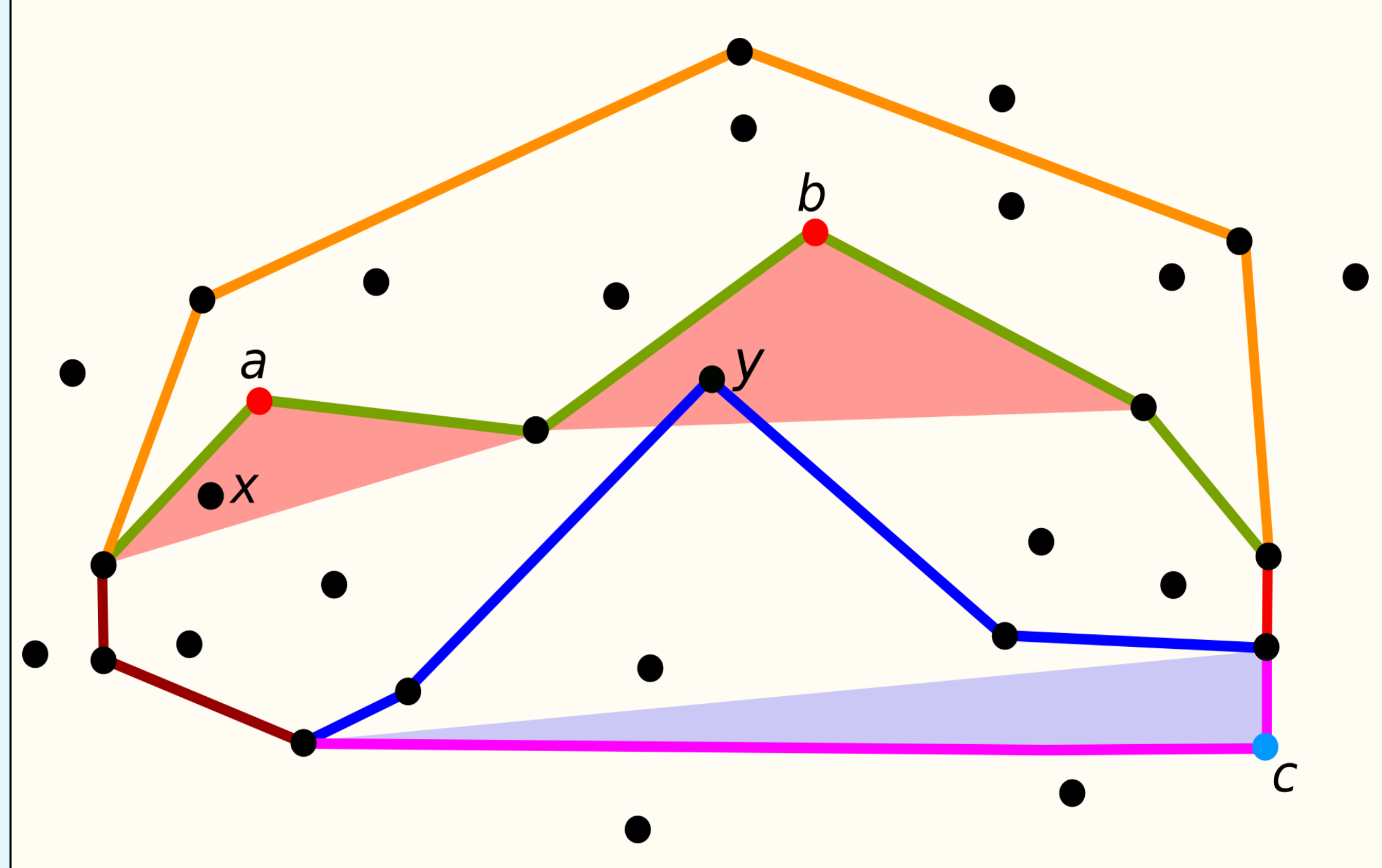
Map generalization

➢Geometry generalization: reduce details in a geometry.

➢Generalization applied to map features → challenge.

➢Simplify polylines: remove interior points and keep topological relations between polylines and control points.

➢Example: simplify counties boundaries but avoid auto-intersection and a city (represented by a point) does not lie in a wrong county.



● May be removed
● May not be removed

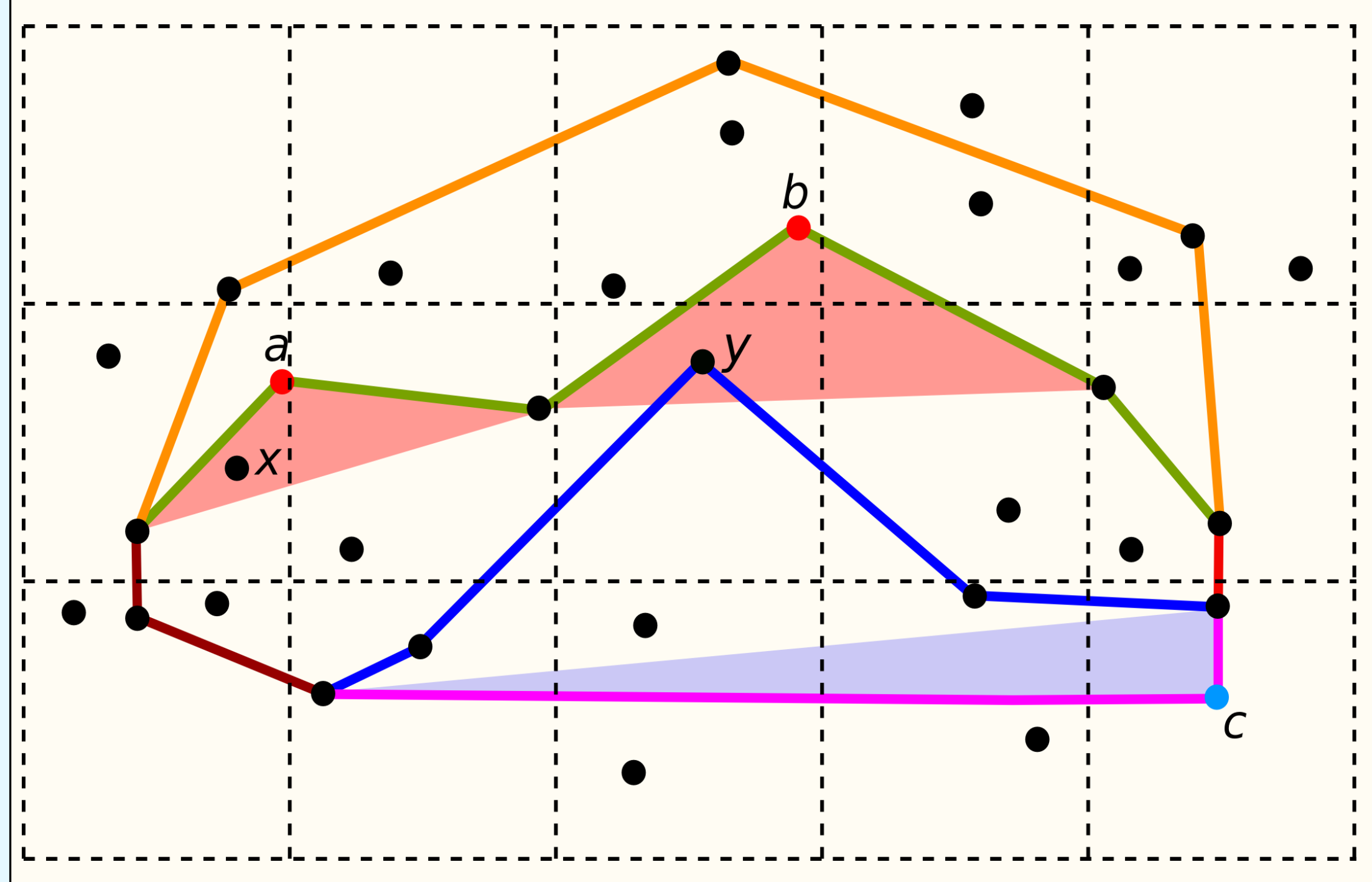
The proposed heuristic

➢Polylines are simplified individually.

➢Heuristic: pass (more than once) through the polylines removing points.

➢Satisfying topological constraints: point p_i is removed ↔ triangle (p_{i-1} , p_i , p_{i+1}) doesn't contain any control/polyline point.

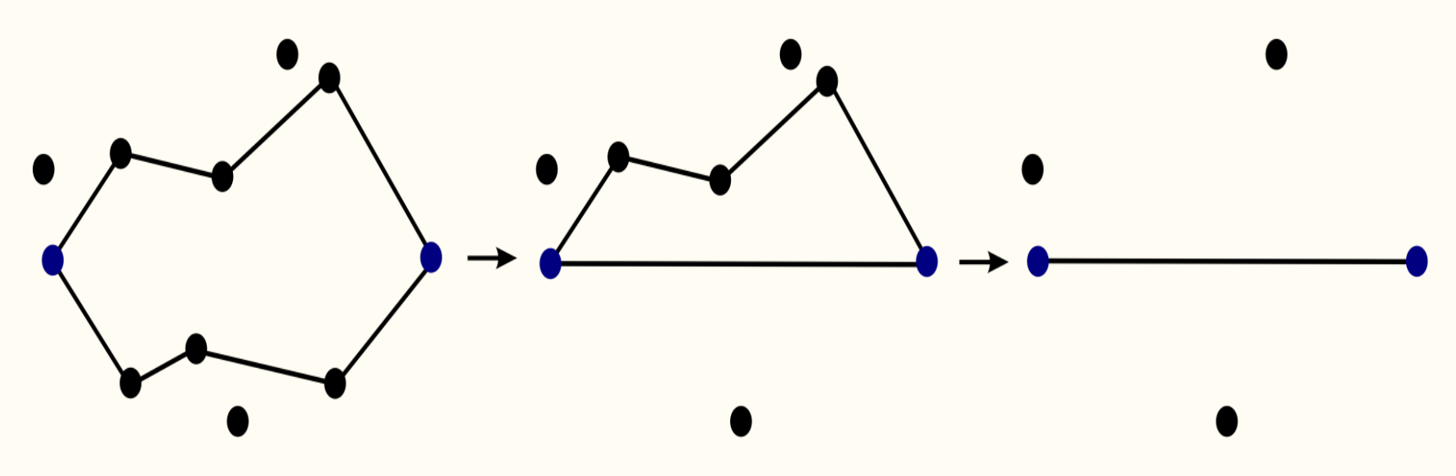
➢How to accelerate point in triangle detection? Uniform grid with control points + polyline points.



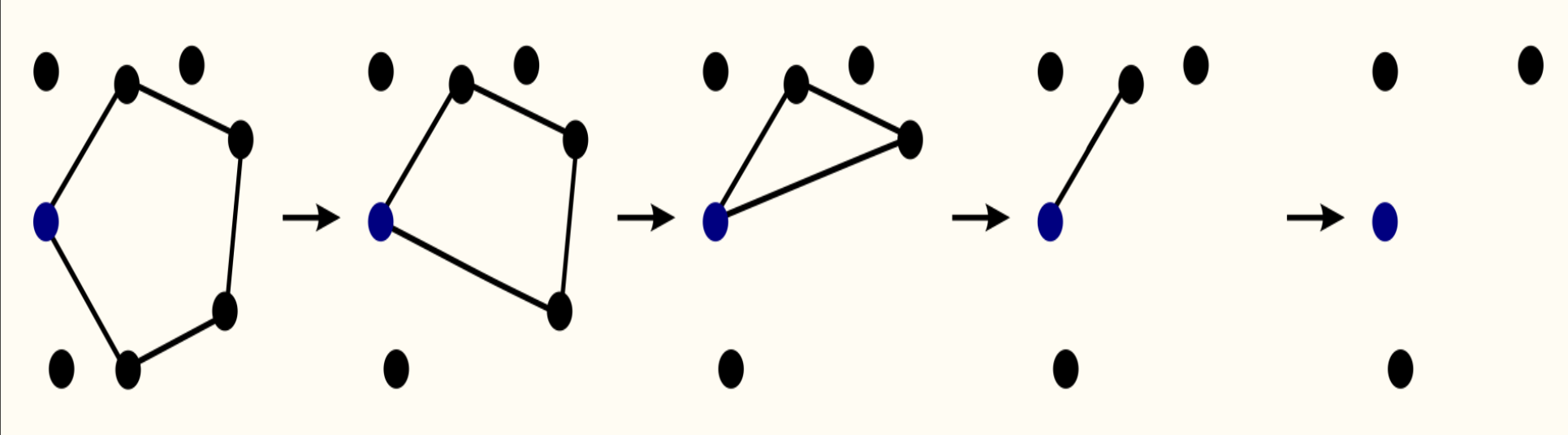
● May be removed
● May not be removed

Drawbacks and solutions

➢Two polylines with the same endpoints → simplified to two identical line segments:



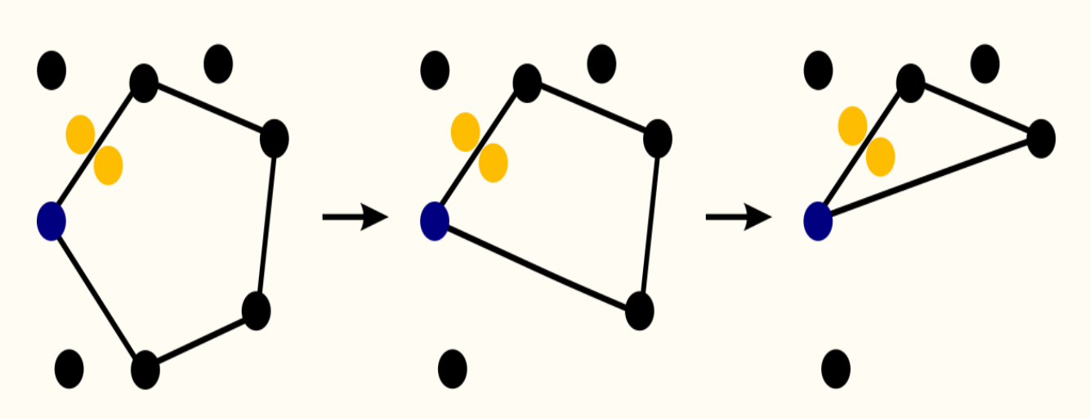
➢A polygon with both endpoints equal → simplified to a point:



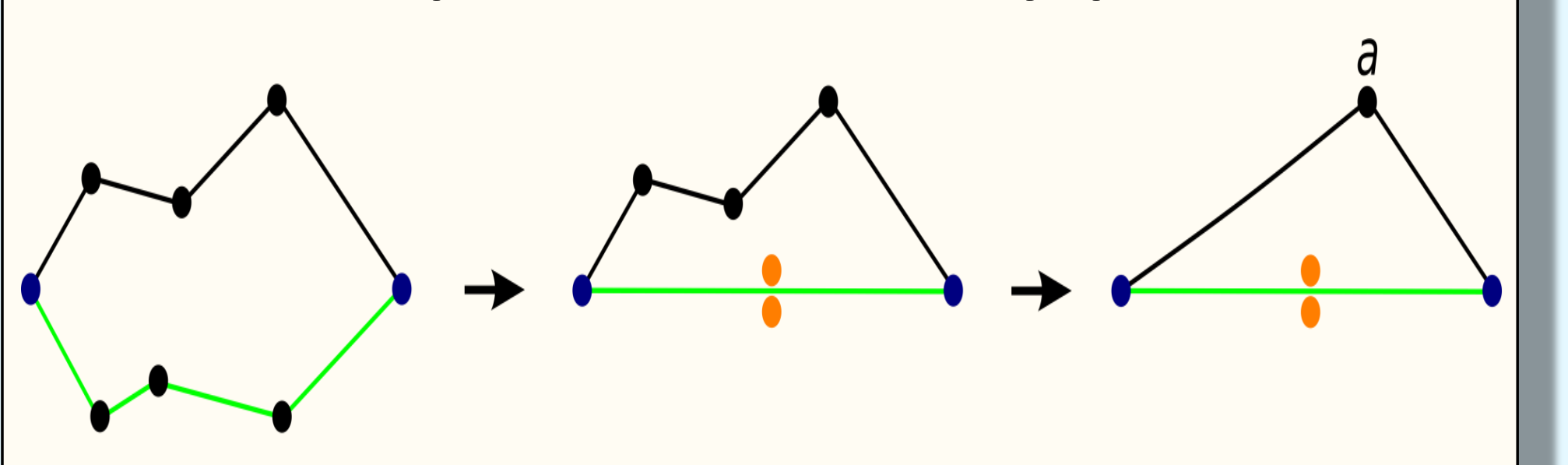
Solutions

➢Add *dummy* points → the heuristic does not need to be changed!

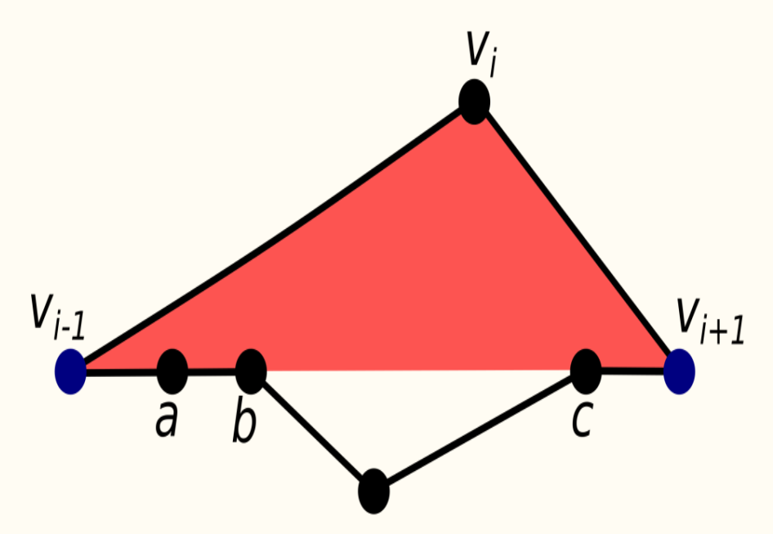
➢Polyline with no interior point (from input or created during simplification) → add a pair of dummy points (in and outside):



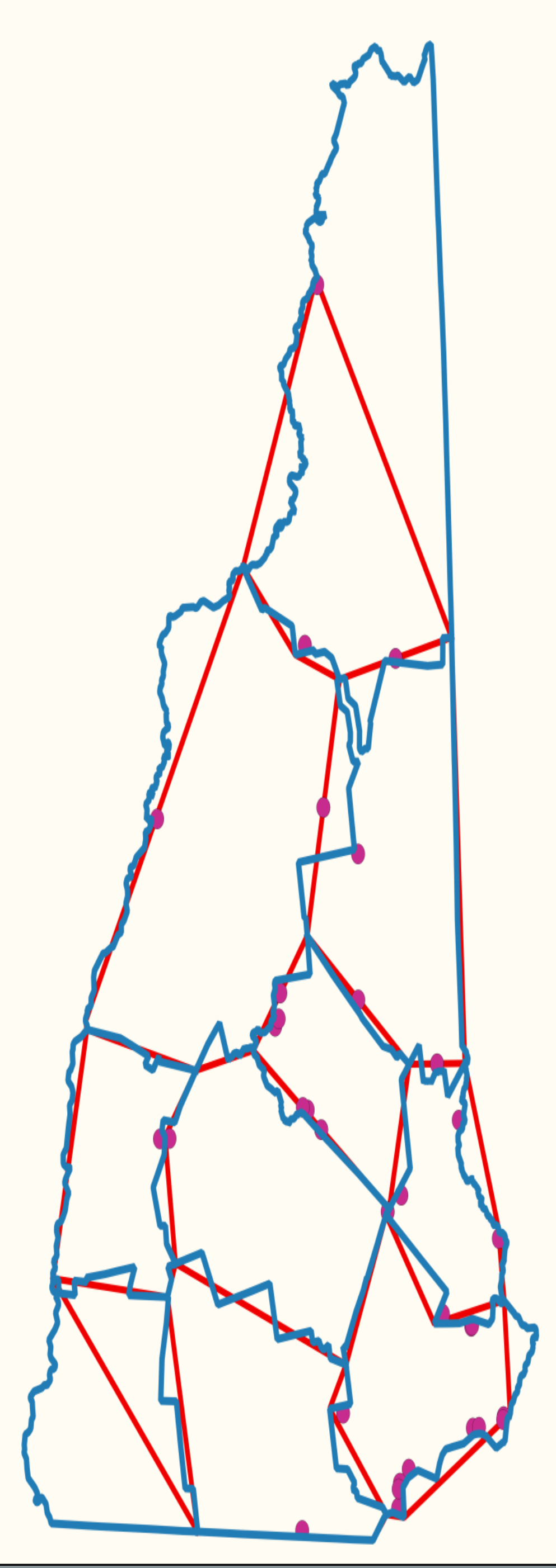
➢Polyline with equal endpoints → add a pair of dummy points:



➢Points in triangle border → consider they inside the triangle.



Example of simplified map



Results

➢Tests in a Lenovo T430s laptop with a 3.6 GHz i7 processor and Samsung EVO 840 SSD.

✓ Datasets 2-5 are from GISCUP
✓ Datasets 6/7: Brazil/USA counties

Dataset	2	3	4	5	6	7
Control pts	127	151	256	1607	10000	10000000
Polyline pts	1564	8531	28014	28323	342738	3645559
Pts removed	1435	7545	25212	23411	308992	3613026

Grid size	Dataset					
	2	3	4	5	6	7
125	0	2	11	10	333	163875
250	2	3	9	8	170	48940
500	5	6	12	10	134	22529
1000	17	15	19	17	132	14307
2000	65	51	50	45	203	10708
4000	269	171	168	149	376	9172

Processing time (ms) excluding I/O

Dataset	Grid size	I/O	Init.	Simpl.
2	125	2	0	0
3	125	11	0	2
4	250	27	1	8
5	250	35	1	6
6	1000	315	25	107
7	4000	37726	1567	7605

Processing times (ms) for the best grid sizes

Conclusions

➢Bottleneck: I/O (tests performed in a SSD!).

➢No topological change.

Future work

➢Compare heuristic with other methods.

➢Automatically choose grid size.

➢Improve "similarity" of the output with input.

➢3D version of the problem.

Acknowledgements

This work was partially supported by Capes (Ciência Sem Fronteiras), FAPEMIG and NSF grant IIS-1117277.