Train on Pins and Test on Obstacles for Rectilinear Steiner Minimum Tree

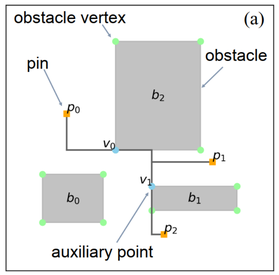
Xingbo Du, Ruizhe Zhong, Junchi Yan

School of Computer Science & School of Artificial Intelligence, Shanghai Jiao Tong University

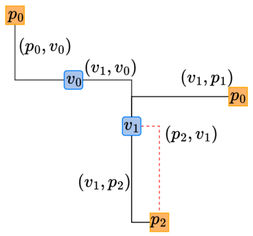
Rectilinear Steiner Minimum Tree (RSMT)

The objective of Obstacle-Avoiding RSMT:

To Connect all given pins using only horizontal and vertical segments with minimum wirelength and avoiding any obstacles.



Components in obstacle-avoiding RSMT problems



Rectilinear Edge Sequence (RES)

Comparisons to Existing RSMT solvers

Method	Representation	End-to-End	Multi-Degree GPU-Parallelization	Obstacle Avoidance
REST [5]	RES (Optimal)	✓	✗	✗
EPST [6]	EPS	✓	✗	✗
Chen et al. [7]	Steiner Points	✗	✓	✓
Lin et al. [8]	OASG + SPF	✗	✓	✓
Chen et al. [9]	Steiner Points	✗	✓	✓
OAREST (ours)	RES (Optimal)	✓	✓	✓

Based on optimal representations, OAREST performs an end-to-end solver, handling multi-degree GPU-parallelization and obstacle avoidance.

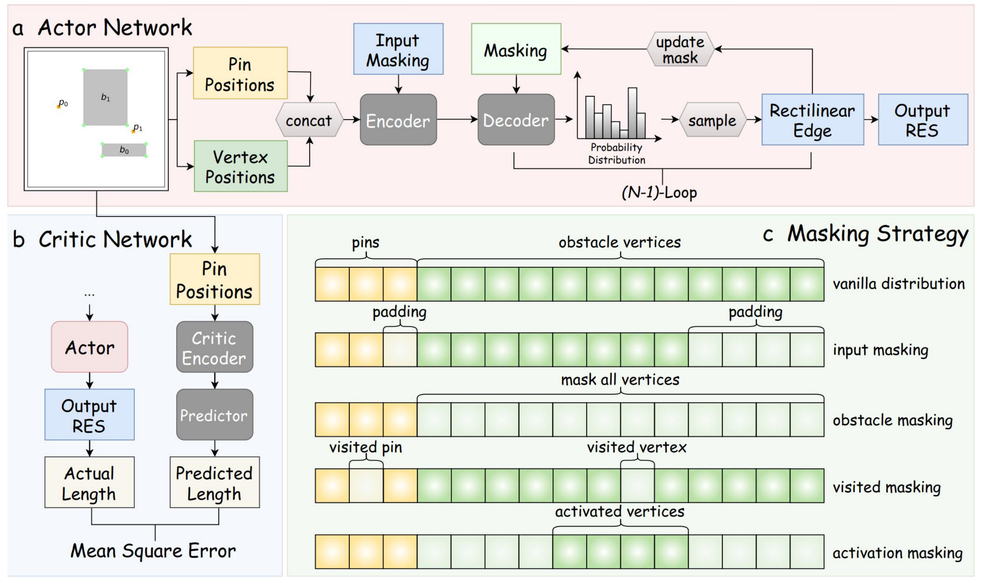
Optimality of RES in OARSMT

Theorem: *For any set of pins $\mathcal{P} = \{p_0, p_1, \cdots, p_{n-1}\}$ and a set of rectangular obstacles $\mathcal{O} = \{o_0, o_1, \cdots, o_{m-1}\}$, an optimal RES of $\mathcal{P} \cup \mathcal{V}'(\mathcal{O})$ can always be found such that its corresponding tree is an optimal OARSMT for $\mathcal{P}$ under obstacles $\mathcal{O}$. Here, $\mathcal{V}'(\mathcal{O}) \subset \mathcal{V}(\mathcal{O})$ is a subset of the corner vertices of all obstacles.*

Propose an RL framework with dynamic masking strategy

Based on the optimal RES representation in OARSMT, we propose an RL-based learning framework following REST [1] and design a dynamic masking strategy to make two main contributions:

- Handle multi-degree GPU-parallelization.
- Avoid obstacles by dynamically activate corner points during inference without training on obstacle-aware instances.



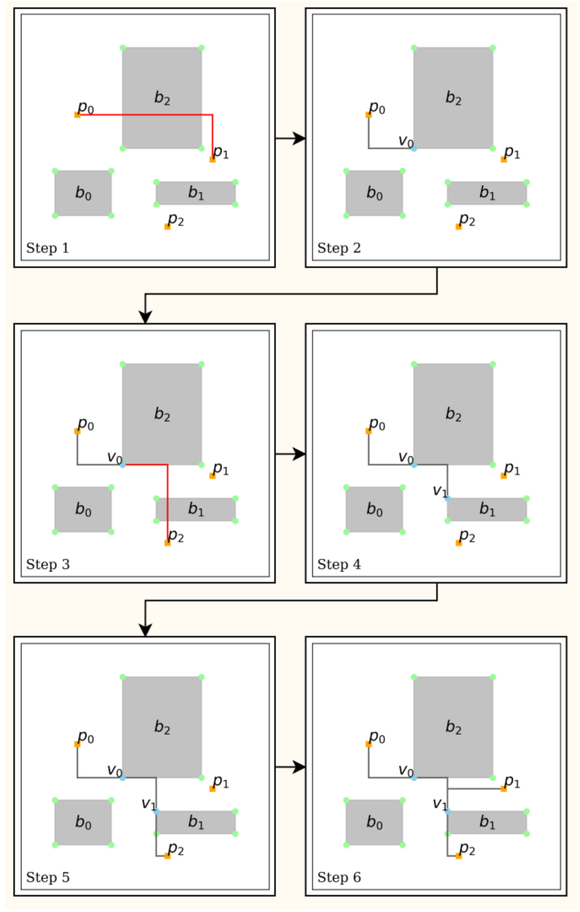
Experiments

Comparable average percentage error (%) to SOTA baselines.

Instances	GeoSteiner [16]	R-MST [17]	BGA [18]	FLUTE [19]		REST [5]		OAREST (ours)
				A = 3*	A = 18*	T = 1*	T = 8*	
R5	0.00	10.91	0.23	**0.00**	**0.00**	0.02	**0.00**	**0.00 ± 0.000**
R10	0.00	11.96	0.48	0.12	0.04	0.23	**0.01**	**0.01 ± 0.000**
R15	0.00	12.19	0.53	0.55	0.06	0.45	**0.03**	**0.03 ± 0.001**
R20	0.00	12.41	0.57	1.03	0.11	0.56	0.07	**0.06 ± 0.001**
R25	0.00	12.47	0.58	1.44	0.18	0.69	0.12	**0.10 ± 0.001**
R30	0.00	12.56	0.60	1.83	0.23	0.77	0.16	**0.15 ± 0.004**
R35	0.00	12.63	0.62	2.13	0.26	0.84	0.21	**0.19 ± 0.002**
R40	0.00	12.65	0.63	1.05	0.29	0.86	0.25	**0.24 ± 0.001**
R45	0.00	12.67	0.63	1.07	**0.30**	0.98	0.32	0.31 ± 0.002
R50	0.00	12.72	0.64	1.12	**0.29**	1.01	0.36	0.36 ± 0.001

* A=3/18 and T=1/8 are different versions of FLUTE and REST, which we illustrate in Appendix B.2.

A toy example



Reference

[1] J. Liu, G. Chen, and E. F. Young. Rest: Constructing rectilinear steiner minimum tree via reinforcement learning. In 2021 58th ACM/IEEE Design Automation Conference (DAC), pages 1135–1140. IEEE, 2021.

Code Available

https://github.com/Thinklab-SJTU/EDA-AI/tree/main/OAREST