Placement Initialization via Sequential Subspace Optimization with Sphere Constraints

Pengwen Chen, Chung-Kuan Cheng, Albert Chern, **Chester Holtz**, Aoxi Li, and Yucheng Wang

Global Placement Preliminaries

Nonlinear optimization

$$\min_{\mathbf{x},\mathbf{y}} \sum_{e \in E} \mathrm{WL}(e; \mathbf{x}, \mathbf{y})$$
$$\text{s.t.} \quad \mathcal{D}(\mathbf{x}, \mathbf{y}) \leq \hat{\mathcal{D}},$$
$$v_k = (\mathbf{x}_k, \mathbf{y}_k) \in r_k, \quad k = 0, \cdots, K$$

Relax

$$\min_{v} \sum_{e \in E} \mathrm{WL}(e; v) + \langle \lambda, \mathcal{D}(v, r) \rangle$$
$$\lambda = (\lambda_0, \cdots, \lambda_K)$$
$$\mathcal{D}(v, r) = (\mathcal{D}(v_0, r_0), \cdots, \mathcal{D}(v_k, r_k))$$

E-Place & **DREAMPlace** solve this problem using first-order methods

Generally initialize using random or min-WL layout

Initialization plays an important role for NNs and nonlinear optimization

We formulate initialization as a graph embedding problem

Gu et al., DREAMPlace 3.0: Multi-Electrostatics Based Robust VLSI Placement with Region Constraint, 2020

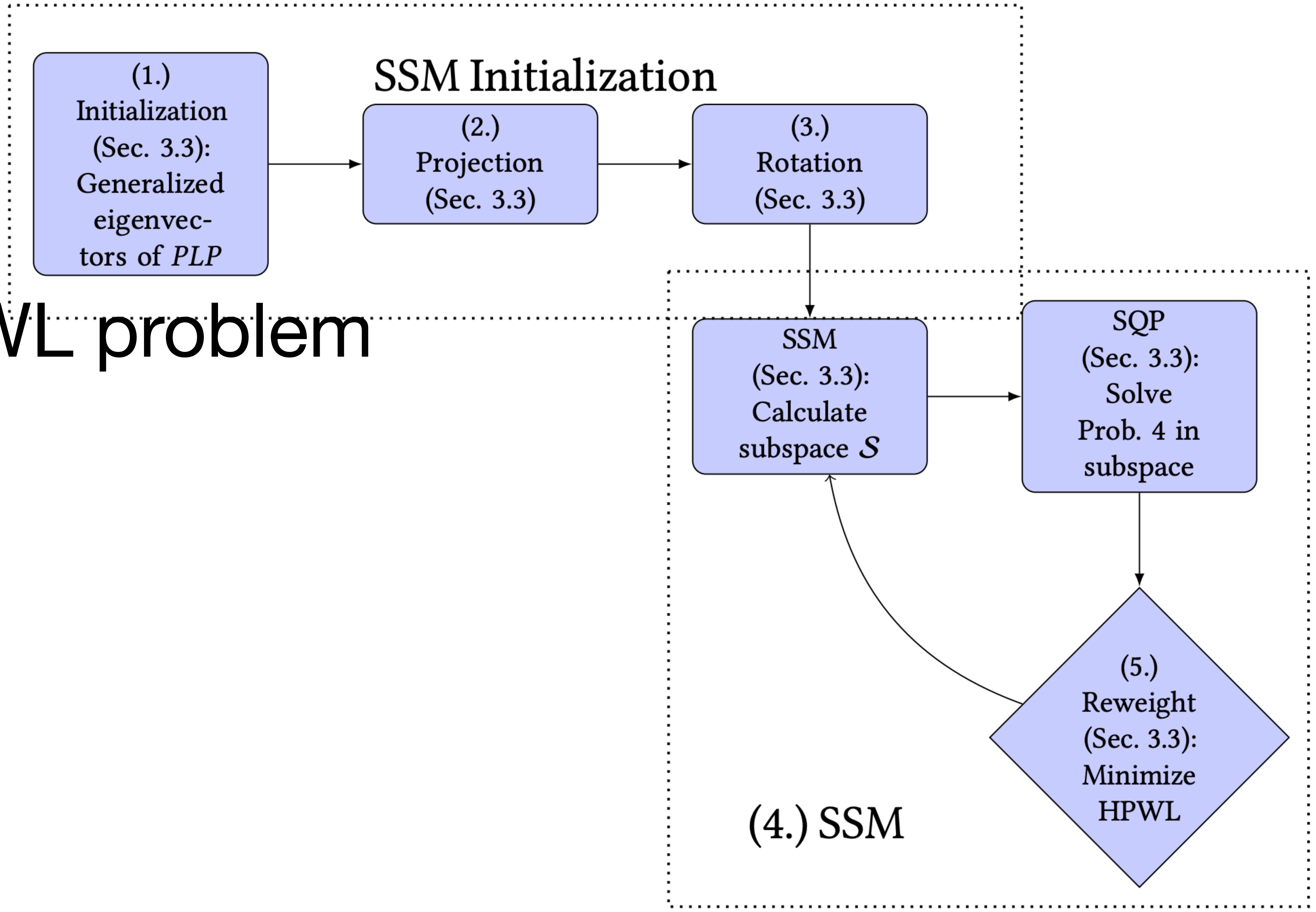
Overview

Chen et al., Placement Initialization via Sequential Subspace Optimization with Sphere Constraints, 2022

1. Compute eigenvectors of laplacian. Align with fixed nodes

2. Find block-global solution to the min-HPWL problem

3. Perform global and detailed placement using 3. as a seed



Lu et al., ePlace: Electrostatics-Based Placement Using Fast Fourier Transform and Nesterov's Method, 2015
Lin et al., DREAMPlace: Deep Learning Toolkit-Enabled GPU Acceleration for Modern VLSI Placement, 2019

Spectral embeddings

A generalized eigenvalue problem and geometric interpretation

Squared distance

Cell positions

Normalized node volumes

Assume $v = 1$

Origin constraints

$$\min_{x,y} \; \boxed{x^\top L x + y^\top L y} \quad \boxed{\text{s.t. } v^\top x = 0, \qquad v^\top y = 0,}$$

$$\boxed{x^\top G x = c_1, \qquad y^\top G y = c_2, \qquad x^\top G y = c_3}$$

Cell positions

Congestion constraints

Any psd matrix.

E.g. assume diag(v)

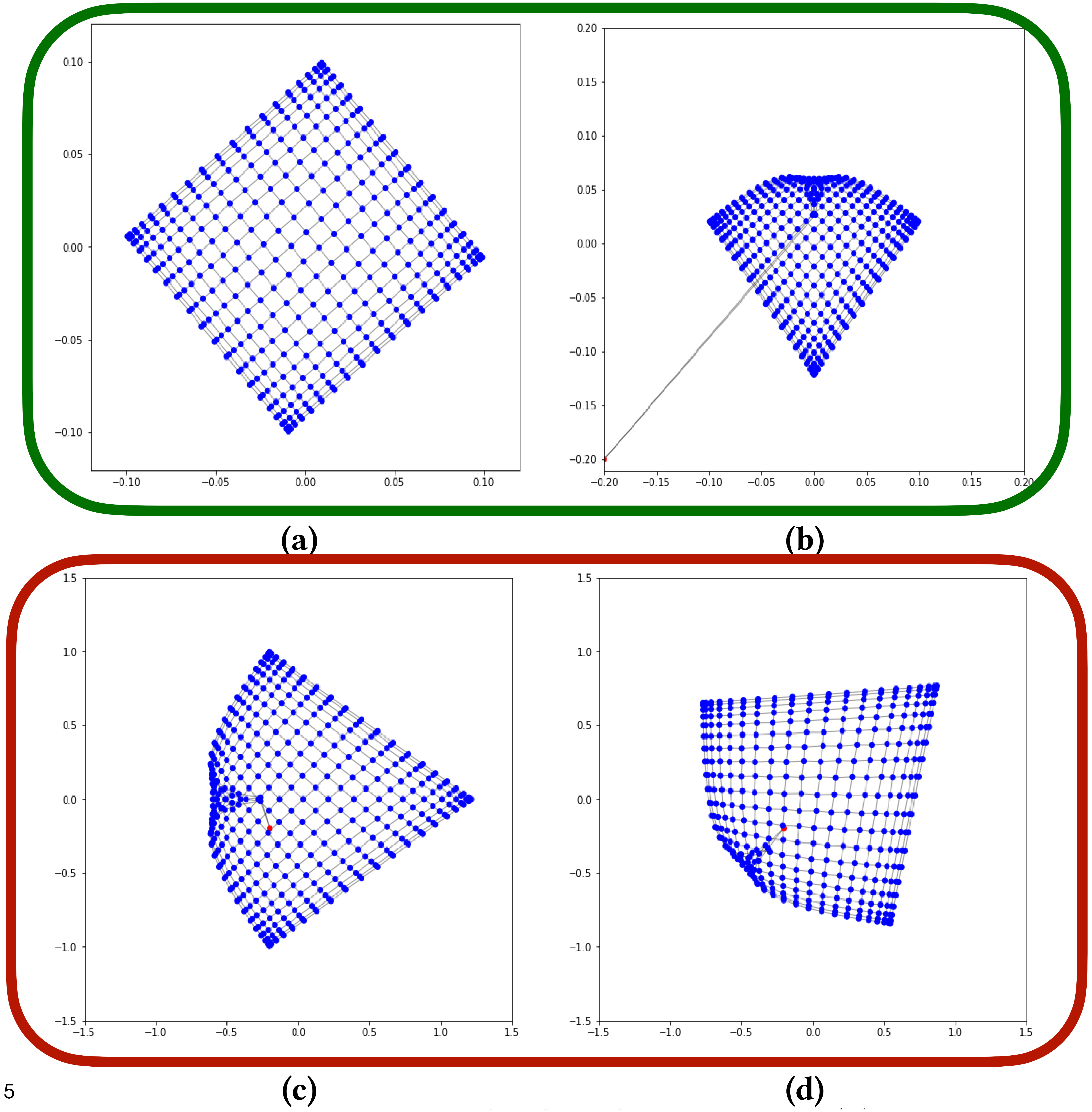
Initialization: two-stage eigenvector projection

Chen et al., LBR: Placement Initialization via a Projected Eigenvector Algorithm, 2022

1. Solving the EV problem

$$\min_{X:X^\top X=I} \ \text{trace}(X^\top, PLPX)$$

2. Transform to (a.) resolve $X^\top X = C$ (b.) align X with fixed nodes



(a) **(b)**

(c) **(d)**

Iterative refinement with SSM

- Need to invert L each time we update X (Newtons Method) - Expensive!
- Can show that a solution lies in a low-dim subspace spanned by 8 vectors:

Small eigenvector of L
Current iterate Newton Gradient
Fixed pins

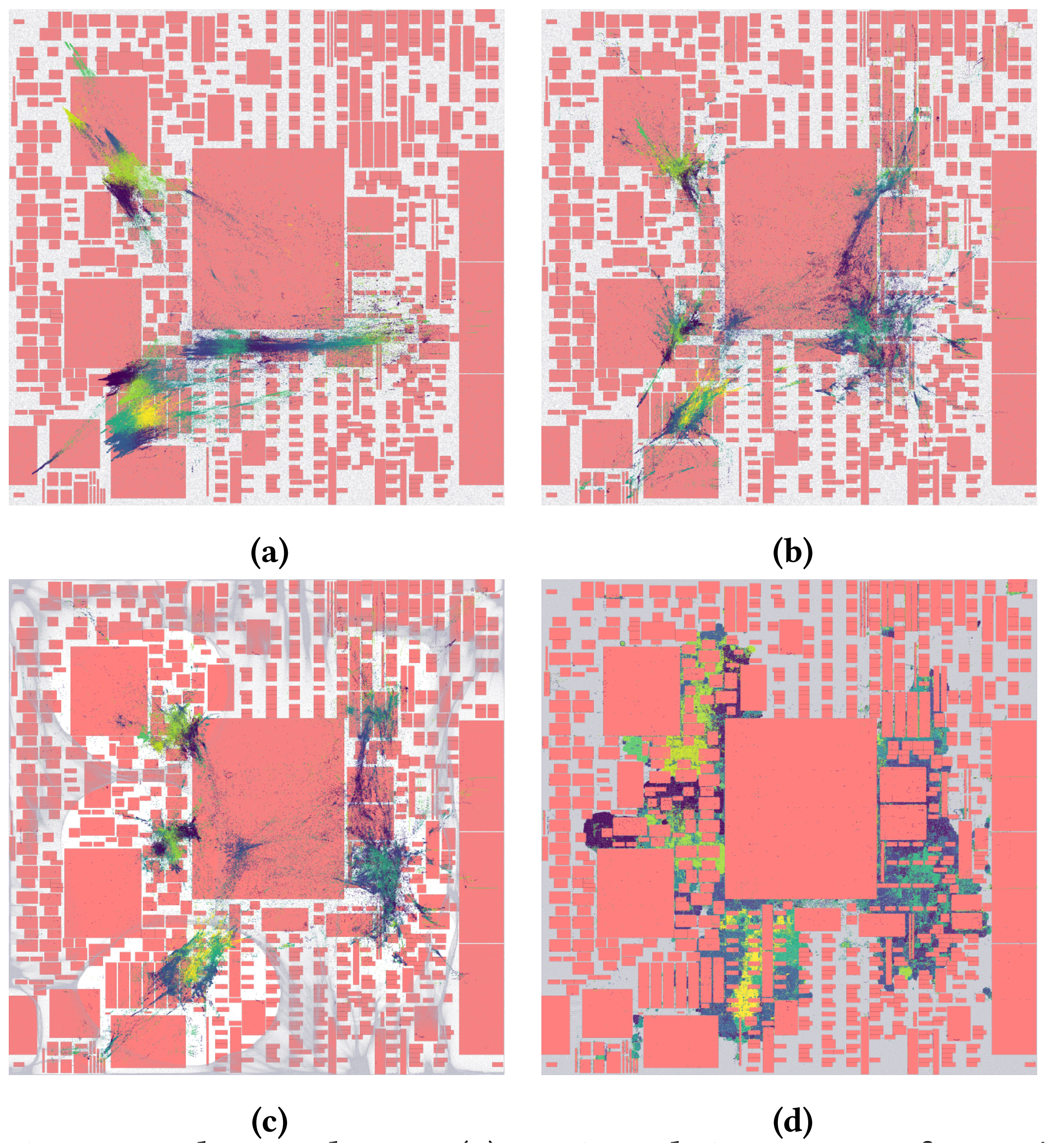
  - $\mathcal{S} = \text{span}(PX_t, Z_t, v, LPX_t + E_0)$

- Can generate a sequence of candidate solutions by iteratively solving the graph embedding problem in this (tiny) subspace. Very efficient!

  - $L_{\mathcal{S}} \to V^\top LV, V = \text{col}(\mathcal{S})$

Hager, Minimizing a Quadratic over a Sphere, 2001

Results



Design	Squared-wirelength		Direct HPWL	
	HPWL	z	HPWL	z
adaptec1	70.34	0.131 ± 0.046	70.07	0.139 ± 0.052
adaptec2	81.21	0.069 ± 0.031	81.13	0.073 ± 0.038
adaptec3	187.95	0.072 ± 0.041	187.27	0.076 ± 0.043
adaptec4	171.62	0.126 ± 0.057	170.39	0.131 ± 0.061
bigblue1	87.04	0.063 ± 0.039	85.9	0.067 ± 0.041
bigblue2	131.37	0.079 ± 0.037	130.04	0.081 ± 0.044
bigblue3	297.31	0.074 ± 0.041	296.11	0.074 ± 0.043
bigblue4	724.78	0.081 ± 0.053	724.39	0.081 ± 0.054

- Significant improvements up to (4%) on adaptec1 in post-detailed placement wire length
- runtime details
- Structure of the solution to direct HPWL-minimization problem is "preserved" best

Design	Random		Min-wirelength		Projected Eigenvectors			Projected Eigenvectors + SSM		
	HPWL	GP runtime (s)	HPWL	GP runtime (s)	HPWL	GP runtime (s)	Init. runtime (m)	HPWL	GP runtime (s)	Init. runtime (m)
adaptec1	73.24	84.39	73.23	74.31	70.36 (3.9%)	63.86	1.56	**70.34** (3.96%)	**62.42**	43.92
adaptec2	82.51	189.46	82.24	172.91	81.68 (1.0%)	164.37	1.47	**81.21** (1.58%)	**162.49**	37.61
adaptec3	194.12	314.54	193.87	**309.88**	189.13 (2.5%)	**313.29**	3.02	**187.95** (3.18%)	314.01	96.31
adaptec4	174.43	371.72	174.16	**354.16**	171.73 (1.5%)	372.14	2.81	**171.62** (1.61%)	361.37	79.91
bigblue1	89.43	112.64	89.43	107.56	87.32 (2.3%)	**94.11**	2.07	**87.04** (2.67%)	94.23	76.19
bigblue2	136.69	387.94	136.69	361.75	132.49 (3.0%)	327.14	2.51	**131.37** (3.89%)	**321.86**	89.27
bigblue3	303.99	1064.63	303.99	1047.66	298.47 (1.8%)	**847.03**	6.15	**297.31** (2.20%)	849.23	184.39
bigblue4	743.75	1534.11	743.75	1500.70	726.71 (2.2%)	1372.49	25.66	**724.78** (2.55%)	**1293.10**	537.21

Lin et al., DREAMPlace: Deep Learning Toolkit-Enabled GPU Acceleration for Modern VLSI Placement, 2019

Conclusions and future work

- Initialization can significantly help placement engines based on first-order methods
- Scalability and implementation is critical for huge cases
- More practical considerations for chip placement applications in the wild
  - Timing via reweighing
  - Local congestion and “over clustering”
- Slow convergence of alternating minimization (reweighing for HPWL minimization)

Appendix
Reweighting for HPWL minimization

Addressing "HPWL-minimization"

- VLSI engineers: HPWL as a surrogate for wire length
- HPWL = reformulate as a linear ℓ_1 minimization problem

$$\min \sum_{i,j \in \mathscr{E}} 2(|x_i - x_j| + |y_i - y_j|) \leq \min_{u_{i,j}>0} \max_{v_{i,j}>0} \left\{ \sum_{i,j \in \mathscr{E}} (u_{i,j}|x_i - x_j|^2 + \frac{1}{u_{i,j}} + v_{i,j}|y_i - y_j|^2 + \frac{1}{v_{i,j}}) \right\}$$

- The solution to this problem can be recovered by solving a sequence of *reweighted* quadratic programs
    1. For each $u, v \geq 0$, solve inner problem with respect to (x, y).
    2. For each (x, y), solve outer problem with respect to $u, v : u_{i,j} = |x_i - x_j|^{-1}$ (likewise for v).

Alpert et al., Faster minimization of linear wirelength for global placement, 1998

Appendix

Computation of the SQP descent directions and eigenvectors of PAP

PROPOSITION 3 (NEWTON DIRECTION). *Assume Λ symmetric and $PZ = Z$, i.e., $v^\top Z = v^\top X = [0, 0]$. The solution (Z, Δ) is*

$$(\Delta)_j = (X^T(L + W_j I)^{-1} X)^{-1} X^T (L + W_j I)^{-1} E_j \tag{10}$$

$$Z_j = P(L + W_j I)^{-1} P\{-X(\Delta)_j + E_j\} \tag{11}$$

where W is given by the eigenvector decomposition of Λ: $\Lambda = UWU^{-1}$, $W = diag(W_1, W_2)$ for two eigenvalues W_1, W_2 of Λ.

Proposition 2.1. *Let $P = I - vv^\top$ with some unit vector v. Suppose $A - sI$ is invertible. Let $u' := (A - sI)^{-1} u_{k-1}$ and $v_1 := (A - sI)^{-1} v$. Let $y_k = \{P(A - sI)^{-1} P\}^\dagger u_{k-1}$. Then $v^\top v_1 \neq 0$*

$$Py_k = \{P(A - sI)^{-1} P\}^\dagger u_{k-1} = (I - \frac{v_1 v^\top}{v^\top v_1}) u'. \tag{6}$$