LRSDP: Low-Rank SDP for Triple Patterning Lithography Layout Decomposition

Yu Zhang[1,2†], Yifan Chen[1†], Zhonglin Xie[1], Hong Xu[2], Zaiwen Wen[1], Yibo Lin[1,3*], Bei Yu[2*]

[1]Peking University, Beijing, China, [2]Chinese University of Hong Kong, Hong Kong, China
[3]Institute of Electronic Design Automation, Peking University, Wuxi, China

Highlights

- We propose **LRSDP**, a scalable low-rank SDP solver for the MPL decomposition problem on large graphs.
- We leverage the decomposition-based augmented Lagrangian method (ALM) to solve the large-scale SDP problem from MPL.
- We propose to exploit the problem structure by restricting the searching space on a smooth manifold (unit sphere).
- Our LRSDP method can achieve $186\times$, $25\times$, and $12\times$ speedup with better solution quality than the state-of-the-art decomposition approaches with highly competitive solution quality.

Background

Triple Patterning Lithography Layout Decomposition Due to the availability issue of EUV (Extreme Ultra-Violet) lithography, triple patterning layout (TPL) decomposition is widely adopted in advanced technology nodes. It imposes a layout decomposition step in the design flow that layout features close to each other are assigned to different masks to prevent coloring conflicts.

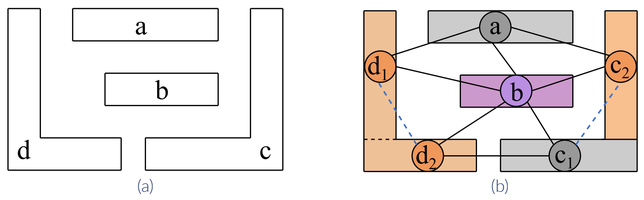

Figure 1. (a) Input features. (b) The final decomposed layout with three colors, where the conflict edges are marked as black edges and stitch edges are marked as blue dashed edges.

SDP-Based Layout Decomposition Among current TPL decomposition methods, ILP can provide exact solutions at the cost of exponential runtime complexity, while SDP can provide high-quality approximation with affordable runtime. TPL decomposition problem can be relaxed as the following semidefinite program [4]:

$$\begin{aligned} \min_{X\in\mathbb{R}^{n\times n}} \quad & \langle C, X\rangle \\ \text{s.t.} \quad & x_{ii} = 1, \quad \forall i \in V \\ & x_{ij} \geq -\frac{1}{2}, \quad \forall e_{ij} \in \text{CE} \\ & X \geq 0, \end{aligned}$$

where $\langle C, X\rangle = tr(C^\top X)$. The symmetric matrix X should be positive semidefinite. Therefore, we can always find a matrix $R \in \mathbb{R}^{p\times n}$ such that $X = R^\top R$.

Motivation

Existing Problems Existing TPL decomposition studies follow a two-step procedure [3]: 1) graph simplification; 2) subgraph decomposition (ILP, EC, SDP, etc.). However, they usually assume the subgraphs after the simplification step are small, i.e., fewer than 100 vertices. When the design complexity increases, the sizes of subgraphs also boost.

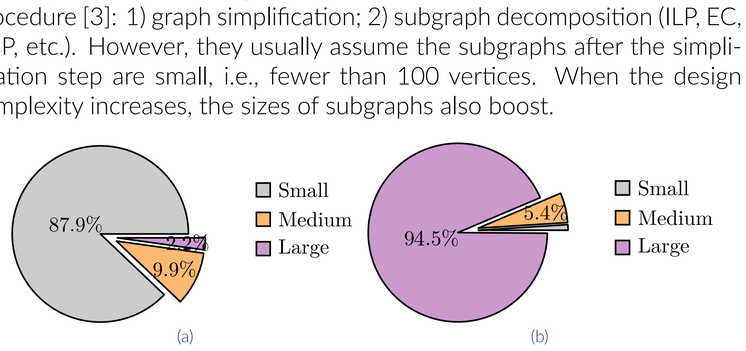

Figure 2. (a) The proportion of small, medium, and large subgraphs after graph simplification. (b) The time ratio spent on solving the TPL decomposition for these subgraphs.

The time spent on solving large subgraphs is the runtime bottleneck of current TPL decomposition methods.

Observations

- The solution to the SDP program generated by the relaxation of combinatorial optimization tasks is often low-rank [1], i.e., $p << n$.
- R lies in a smooth manifold: $\mathcal{M} = \{R \in \mathbb{R}^{p\times n} | R = [r_1, \cdots, r_n], \|r_i\|_2 = 1\}$, which is exactly a unit sphere.

Therefore, to achieve efficient yet high-quality decomposition, we propose a scalable SDP-solving algorithm leveraging the low-rank property and exploiting the problem structure.

Pipeline

The overall **LRSDP** framework is shown in Fig. 3.

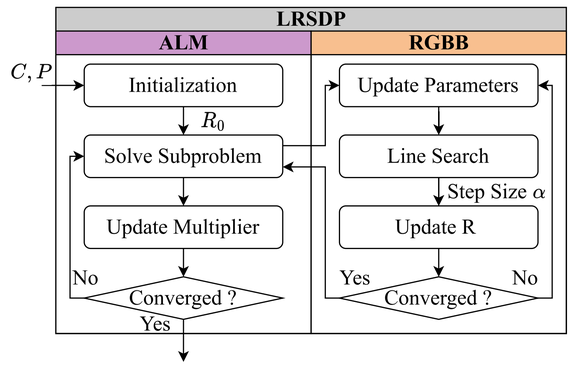

Figure 3. The algorithm flow of **LRSDP**.

Methodology

Low-Rank Factorization leads to many fewer variables in the problem of interest and naturally guarantees the satisfiability of semidefinite constraint. More importantly, the global optimality of the solution to the factorized surrogate can still be ensured by properly choosing p [1].

Augmented Lagrangian Method The major framework of **LRSDP** is based on the augmented lagrangian method, which includes an additional term to penalize infeasible points.

The ALM function is denoted by:

$$L_\sigma(R, W, y, \sigma) = \langle C, R^\top R\rangle + h(W) - \langle y, P\odot R^\top R - W\rangle + \frac{\sigma}{2}\|P\odot R^\top R - W\|_F^2, \tag{1}$$

where $y \in \mathbb{R}^{n\times n}$, and $\sigma > 0$ are parameters for ALM. The subproblem in k-th iteration of the augmented Lagrangian framework is an unconstrained Riemannian manifold optimization problem, denoted as:

$$\min_{R\in\mathcal{M}} \Phi_k(R) = \langle C, R^\top R\rangle + h(P\odot R^\top R - y^k/\sigma_k - T(R)) + \frac{\sigma_k}{2}\|T(R)\|_F^2. \tag{2}$$

Riemannian gradient descent is chosen as the subproblem solver. The gradient-type method in Riemannian space performs:

$$x_{k+1} = R_{x_k}(-\alpha_k \text{grad} f(x_k)), \tag{3}$$

where R_{x_k} is a retraction from the tangent space $T_{x_k}\mathcal{M}$ at x_k to the manifold $\mathcal{M}$. We can compare Euclidean optimization with Riemannian optimization:

Euclidean optimization:

1. Find a descent direction $g_k = -\nabla f(x_k)$;
2. Update $x_{k+1} = x_k + \alpha_k g_k$.

Riemannian optimization:

1. Find a descent direction $g_k = -\text{grad} f(x_k) \in T_{x_k}\mathcal{M}$;
2. Update $x_{k+1} = R_{x_k}(\alpha_k g_k)$.

Barzilai–Borwein (BB) method is employed as the step size selection strategy for Riemannian gradient descent. The basic idea of the Barzilai–Borwein (BB) method [2] is to approximate the computationally expensive Hessian matrix. When $s_k^\top y_k > 0$, the BB step-length is

$$\alpha_k^{BB} = \frac{s_k^\top s_k}{s_k^\top y_k}. \tag{4}$$

with $s_k := x_k - x_{k-1}$ and $y_k := \nabla f(x_k) - \nabla f(x_{k-1})$. Then, the BB method performs the iteration: $x_{k+1} = x_k - \alpha_k^{BB}\nabla f(x_k)$.

Riemannian gradient descent with Barzilai–Borwein Method (RGBB) is performed here to find the optimal solution for the subproblem. The RGBB at k-th step is illustrated as follows:

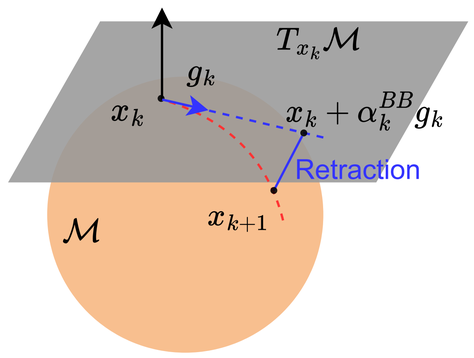

Figure 4. A single iteration of RGBB.

Evaluation Results

Table 1. Experiments on different decomposition algorithms. The cases can be solved by all the 4 decomposers.

	ILP			CSDP			Ours		
test case	conflict	stitch	time/s	conflict	stitch	time/s	conflict	stitch	time/s
test1 100	241	299	88.9	269	287	4.5	262	285	2.8
test1 101	78	138	3739.1	94	134	34.1	98	141	4.8
test1 102	1	1	2.2	1	1	0.1	1	1	0.1
test2 100	5046	8934	22120.4	6439	8179	330.1	6456	8202	101.2
test2 102	213	502	12243.6	479	475	579.8	297	486	28.6
test3 100	680	757	24566.2	1058	1109	13577.3	911	733	168.3
test3 101	130	270	10422.4	196	276	854.4	194	266	30.7
test3 102	2	1	0.1	2	1	0.0	2	1	0.0
test5 100	354	330	5523.0	396	329	43.9	402	321	6.8
test5 101	467	232	113.1	527	228	9.9	496	229	3.8
test5 102	197	174	7225.2	262	151	1526.5	238	144	40.8
test6 102	115	482	451.1	144	477	65.0	150	479	10.8
test7 100	8424	9740	36696.6	9020	9509	2936.4	9089	9490	698.6
test8 100	5683	4606	158.2	5750	4547	47.2	5752	4549	38.0
test8 101	6199	13139	52660.6	7275	12741	7466.4	7235	12840	820.7
test9 100	8739	6969	249.3	8842	6880	73.1	8841	6879	60.3
test10 100	9775	9580	409.3	9963	9457	115.9	9964	9457	94.7
ratio	0.87	1.03	186.62	1.05	1.03	12.48	1.00	1.00	1.00

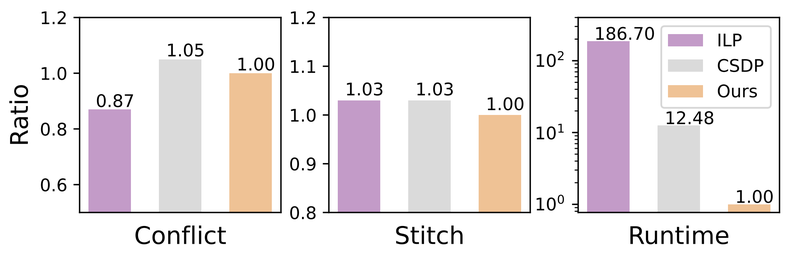

Figure 5. Comparison of ILP, CSDP, and our LRSDP.

Table 2. Experiments on different decomposition algorithms on test cases with large subgraphs. Some algorithms crash ('Failed') or exceed the time limit of 16 hours ('TLE').

	ILP			CSDP			Ours		
test case	conflict	stitch	time/s	conflict	stitch	time/s	conflict	stitch	time/s
test2 101	TLE	TLE	TLE	4553	5026	12327.0	3837	5124	489.5
test4 100	TLE	TLE	TLE	Failed	Failed	Failed	16377	10559	18357.1
test4 101	18012	6250	30439.1	TLE	TLE	TLE	12041	7238	41421.6
test6 100	14954	23427	11318.6	17657	22134	407.9	17596	22215	213.8
test6 101	TLE	TLE	TLE	Failed	Failed	Failed	8851	12238	7469.2
test7 101	TLE	TLE	TLE	Failed	Failed	Failed	13831	18247	23700.9
test7 102	TLE	TLE	TLE	TLE	TLE	TLE	6480	6192	1880.9
test8 102	TLE	TLE	TLE	Failed	Failed	Failed	97885	8937	16016.8
test9 101	TLE	TLE	TLE	24475	21418	11375.8	12031	21909	2471.8
test9 102	TLE	TLE	TLE	Failed	Failed	Failed	10015	17668	33270.6
test10 101	TLE	TLE	TLE	Failed	Failed	Failed	18480	28608	9748.4

Conclusion

- Among two SDP-based approaches, our method is $12.48\times$ faster than CSDP on average with 5% lower cost.
- Our method is $186.62\times$ faster than ILP and only increases about 11% cost, which makes a better trade-off between performance and efficiency.
- Our method is able to deal with fairly large cases within the time limit, whereas CSDP is prone to fail on these large layouts.

References

[1] Nicolas Boumal, Vlad Voroninski, and Afonso Bandeira. The non-convex burer-monteiro approach works on smooth semidefinite programs. *nips*, 29, 2016.

[2] Bruno Iannazzo and Margherita Porcelli. The riemannian barzilai–borwein method with nonmonotone line search and the matrix geometric mean computation. *IMA Journal of Numerical Analysis*, 38(1):495–517, 2018.

[3] Wei Li, Yuzhe Ma, Qi Sun, Lu Zhang, Yibo Lin, Iris Hui-Ru Jiang, Bei Yu, and David Z Pan. Openmpl: An open-source layout decomposer. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 40(11):2331–2344, 2020.

[4] Bei Yu, Kun Yuan, Boyang Zhang, Duo Ding, and David Z. Pan. Layout decomposition for triple patterning lithography. pages 1–8, 2011.