Net Separation-Oriented Printed Circuit Board Placement via Margin Maximization

Chung-Kuan Cheng Chia-Tung Ho and Chester Holtz*

{ckcheng, c2ho, chholtz}@eng.ucsd.edu
Department of Computer Science and Engineering
University of California San Diego
La Jolla, CA, 92037

Abstract— Packaging has become a crucial process due to the paradigm shift of More than Moore. Addressing manufacturing and yield issues is a significant challenge for modern layout algorithms.

We propose to use printed circuit board (PCB) placement as a benchmark for the packaging problem. A maximum-margin formulation is devised to improve the separation between nets. Our framework includes seed layout proposals, a coordinate descent-based procedure to optimize routability, and a mixed-integer linear programming method to legalize the layout. We perform an extensive study with 14 PCB designs and an open-source router. We show that the placements produced by NS-place improve routed wirelength by up to 25%, reduce the number of vias by up to 50%, and reduce the number of DRVs by 79% compared to manual and wirelength-minimal placements.

I. INTRODUCTION

Packaging technology continues to advance from Printed Circuit Boards, Flip-Chip, Integrated Fan-Out (InFO), Chip-on-Wafer-on-Substrate (CoWoS) [1], Integrated Fan-Out (InFO) [2], to System-on-Integrated-Circuit (SoIC) [3] for More than Moore. However, the integration of the packages encounters significant manufacturability and yield issues due to components with arbitrary shape, non-Manhattan routing directions, vias of larger than routing track pitches, high resistance, and/or reliability problems. In this paper, we use Printed Circuit Board placement as a test vehicle to improve manufacturability and yield. We separate nets in 2D space to minimize net crossings, and encourage same layer routing. Our goal is to minimize post-route design rule violations, and reduce via usage. We additionally expect a reduction in the routed metal length.

The PCB placement problem exhibits additional challenges compared to Integrated Chip (IC) placement including arbitrary component shapes, board boundaries, multiple layers, and routing constraints. Conventional analytical IC placers [4] suffer from several key issues which prevent convergence to good—or even reasonable—solutions (with respect to wirelength and routability): (1) The vanilla RePlace [4] implementation fails to find good solutions for multi-layer designs with diversely sized components. (2) The rate of convergence of the algorithm is dependent on design-specific parameters—e.g. filler cells and anchor weights.

In this work, we propose and apply a gradient descent-based algorithm to optimize wirelength, component density,

and routability and a mixed-integer linear program (MILP) for local legalization. The computational bottleneck introduced by the MILP is addressed by integrating global relative positioning constraints derived from the gradient-based placement stage. Furthermore, our framework involves very few design-dependent parameters. This allows our framework to be generally applied to a variety of designs without a costly tuning stage.

A. Contribution

Our contributions can be summarized as follows:
1) We propose the NS-Place framework for PCB layout which minimizes net congestion using a support vector machine-like formulation and performs legalization by solving a congestion-aware MILP.
2) We demonstrate that the routed placements produced by our framework have fewer design rule violations and vias, and shorter total metal length compared to manual placements.

In section II, we review previous work. In section III, we describe our routability objective. The NS-Place placement framework, including initialization and the MILP-based legalizer is described in Section IV. In section V, we present experimental results on real PCB testcases. We conclude in section VI.

II. RELATED WORK

In this section we review the previous work with respect to cost and congestion-driven placement and PCB layout.

A. Cost-driven placement

Conventional global placement strategies for ICs seek to minimize wirelength subject to density constraints. Density constraints are typically integrated with the objective to yield an unconstrained relaxation (e.g. as in [4]):

$$\min_{x,y}\left(\sum_{e=(i,j)\in\mathcal{E}} Wl(e;x,y) + \lambda D(x,y)\right) \quad (1)$$

where $Wl(\cdot;\cdot)$ is a function that takes a net instance e as input and returns the cumulative wirelength,, and $D(\cdot)$ is a density penalty. In the context of IC layout, the wirelength of a net is commonly modelled as the half-perimeter wirelength (HPWL) or a smooth alternative and D is a smooth density [5]. Overlap constraints are typically satisfied over the placement process by gradually increasing the weight λ, at the cost of increased wirelength. The current state-of-the-art IC placement algorithms [5], [4] solve Problem 1 in this manner. We adopt a similar formulation for PCB placement.

---

*Corresponding author

B. Congestion-driven Placement

A typical method of estimating routing demand is to consider pin or feasible routed-wire density [6]. Other methods include applying Rent's rule [7], or more sophisticated routing models; for example relying on the construction of rectilinear Steiner trees or the external evaluation of a router [8]. State of the art techniques for congestion-aware placement include mPL [7], a multilevel analytical placer based on non-linear optimization and estimating the routing demand based on a two-pin connection routing model, ROOSTER [8]: a min-cut placer which models nets by Rectilinear Steiner Minimal Trees, and APlace [9], a multilevel analytical placer based on non-linear optimization and stochastic estimates of the routing demand. Similar to our work, [10] propose to minimize a smooth upper bound on the crossing number to reduce edge crossings in the context of graph visualization, but their formulation is incapable of handling multi-pin nets.

These techniques generally suffer from inadequate estimation or prohibitive computational cost. In contrast, the framework proposed in this work is rigorous and does not rely on rerunning the placement algorithm or applying post-placement optimization.

C. PCB Placement

Examples of previous work on the PCB placement problem include [11], [12], [13], [14]. These techniques rely on various meta-heuristics to produce non-overlapping layouts while taking into account various metrics such as thermal and power characteristics of components, timing, and tidiness. In general, these methods suffer from drawbacks—e.g. are computationally expensive, incapable of rotating components, or evaluated on synthetic or toy benchmarks. In contrast, our framework is efficient, capable of rotating modules, extensible, and validated on production PCB designs placed by industry experts. We additionally acknowledge the similarity of the PCB placement problem to macro placement, and point the reader to [15] for a review of relevant techniques.

III. NET SEPARATION-ORIENTED PLACEMENT

A. Preliminaries

net	$e \in \mathcal{E}$
pin matrices	$A_e \in \mathbb{R}^{p \times 2}$
coordinates and orientation	$x, y \in \mathbb{R}_+^n,\ r \in \{0,1\}^n$
density & net separation cost weight	$\lambda_\mathrm{D}, \lambda_\mathrm{NS} \in \mathbb{R}_+$
convex hull coefficients	$u \in \mathbb{R}^2, \gamma \in \mathbb{R}$
wirelength smoothing parameter	$c \in \mathbb{R}_+$

Fig. 1: Notation & key terms

Let $x, y \in \mathbb{R}_+^n$ be vectors corresponding of coordinates of n components such that the i-th component has coordinates encoded in the i-th row of $[x:y]$; $[x:y]_i$. Let $\mathcal{E}$ denote a set of m nets. We aim to assign coordinates so that the resulting layout has small cumulative wirelength, layout density, and routing congestion.

B. NS-Place Objective

Our method may be expressed concisely as the following unconstrained optimization problem given λ:

$$\min_{x,y} \sum_{e \in \mathcal{E}} [Wa(e; x, y) + \lambda_\mathrm{NS} Ns(e; x, y)] + \lambda_\mathrm{D} D(x, y) \quad (2)$$

where Wa and D corresponds to weighted-average wirelength and density terms, and Ns corresponds to the proposed net separation term described in Sec. III.D.

C. Wirelength and density-driven optimization

Many modern techniques for analytic placement rely on quadratic optimization with terms associated with attraction of connected cells, and repulsion of overlapping cells. A typical approach is to represent individual nets as rectangles and to minimize the sum-perimeters over all nets. Repulsion is often applied between overlapping nodes to reduce density. In this work, we adopt the smooth continuous and differentiable weighted-average wirelength (Wa) model [16] for wirelength cost. The horizontal net-wirelength for net e is given by

$$Wa_x^{(e)} = \frac{\sum_{i \in e} x_i \exp\left(\frac{x_i}{c}\right)}{\sum_{i \in e} \exp\left(\frac{x_i}{c}\right)} - \frac{\sum_{i \in e} x_i \exp\left(-\frac{x_i}{c}\right)}{\sum_{i \in e} \exp\left(-\frac{x_i}{c}\right)}$$

where c is a parameter that controls the smoothness and approximation error. We then write the wirelength of e:

$$Wa(e; x, y) = Wa_x^{(e)} + Wa_y^{(e)}$$

The density term corresponds to the mixed-size module bin-based density objective described in [9]. The placement area is divided into B bins, and the placer seeks to equalize the overlap at each bin. For a bin b, let x_b be the x-coordinate of the center and w_b be the width. Then the smoothed overlap $\Theta_x(b, i)$ in the x-direction between bin b and module i with width w_i and height h_i is

$$\Theta_x(b, i) = \begin{cases} 1 - 2d_x^2/w_b^2, & \text{if } 0 \le d_x \le w_b/2 \\ 2(d_x - w_b)^2/w_b^2, & \text{if } w_b/2 \le d_x \le w_b \\ 0 & \text{if } w_b \le d_x \end{cases}$$

where $d_x = |x_i - x_b|$. The overlap in the y-direction is defined similarly. The density function of bin b is then

$$D_b(x, y) = \sum_i C_i \Theta_x(b, i) \Theta_y(b, i) \quad (3)$$

where C_i is a normalization factor such that $\sum_b C_i \Theta_x(b, i) \Theta_y(b, i) = w_i h_i$—the area of module i. Finally, $D(x, y) = \sum_b (D_b(x, y) - \frac{\sum_i (w_i h_i)}{B})^2$.

D. Net-separation optimization via margin maximization

As mentioned in Sec. II, typical approaches to model routing congestion rely on estimating or expressing the routed wire density as a function of pin-density and the feasible routing-area (e.g. the pin-bounding box [6]). The goal is then to minimize this notion of wire-density. In this work, we model the feasible routing region as the convex-hull of the net-pins, and our goal is to separate routing regions. This prevents over-estimation of the routing density as shown in Fig. 2. The method consists of two steps:
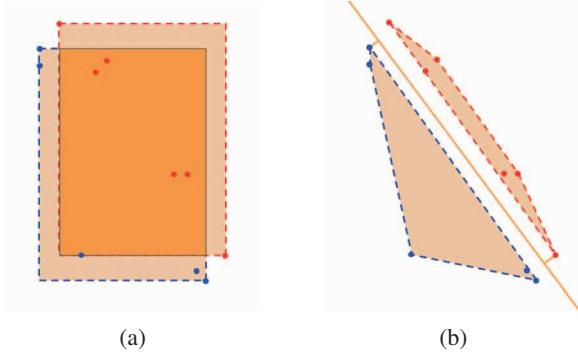
1) Given two nets, find the max-margin separator h.

(a)   (b)

Fig. 2: Congestion between nets denoted by red and blue pins. **(a)** Rectilinear density metrics are pessimistic. **(b)** An optimistic model of routability. The margin between the convex hulls of nets.

2) For all movable components, find directions (gradients) that maximize the margin with respect to h.

Consider a k-pin net e defined by pins with coordinates $e_1 = [e_{1x}, e_{1y}], e_2 = [e_{2x}, e_{2y}], \ldots, e_k = [e_{kx}, e_{ky}] \in \mathbb{R}^2$. Note that the coordinates of *any* point lying in the convex hull of the net-area may be written as a convex combination of these pin coordinates. Let $A_e^\top = \begin{bmatrix} e_{1x} & e_{2x} & \cdots & e_{kx} \\ e_{1y} & e_{2y} & \cdots & e_{ky} \end{bmatrix}$ be the *pin-matrix* associated with net e. Now consider a k'-pin net e' with associated pin matrix $A_{e'}$. For the convex hulls characterized by the pins to *not* intersect, there necessarily must be a $u \neq 0$ and γ such that $x^\top u - \gamma$ is nonpositive for all x lying in one net, and nonnegative for all x lying in the other net. We denote the hyperplane defined by $\{x | x \in \mathbb{R}^2, x^\top u = \gamma\}$ the *separating hyperplane*, and we want to introduce a regularizer which encourages e and e' to lie in different half-spaces with sufficient margin. Note that e and e' do not intersect if there is no shared point in the interior of the convex hulls characterized by the coordinates of pins in e and e'—i.e. if the following has *no solution*.

$$\exists \delta_{A_e} \in \mathbb{R}^k, \delta_{A_{e'}} \in \mathbb{R}^{k'}$$
$$\text{such that } A_e^\top \delta_{A_e} = A_{e'}^\top \delta_{A_{e'}}, \quad \mathbf{1}^\top \delta_{A_e}, \mathbf{1}^\top \delta_{A_{e'}} = 1,$$
$$\delta_{A_e}, \delta_{A_{e'}} \geq 0$$

Namely, duality & Farkas' Lemma provide the conditions that must be satisfied if the convex hulls defined by the pins *do not* intersect:

$$A_e u \geq \alpha \mathbf{1} \quad A_{e'} u \leq \beta \mathbf{1} \quad \alpha - \beta > 0$$
$$\implies A_e u - \gamma \mathbf{1} \geq 1, \quad A_{e'} u - \gamma \mathbf{1} \leq -1$$

This formulation naturally implies the solution to the following minimization problem. Note that one might alternatively aim to find the *maximum-margin* separator. Due to the equivalence with the SVM optimization problem [17], [?], efficient solvers may be [17]employed to recover γ and u.

$$0 = \min_{u,\gamma} f(e, e', u, \gamma)$$
$$= \min_{u,\gamma} ||(-A_e u + (\gamma + 1)\mathbf{1})_+||_2 + \quad (4)$$
$$||(A_{e'} u - (\gamma - 1)\mathbf{1})_+||_2$$

Let A_e be the pin matrix corresponding to net e. We define the net-separation regularizer:

$$Ns(\cdot) = \frac{1}{|\mathcal{M}|} \sum_{e' \in \mathcal{E}} \min_{u,\gamma} f(e, e', u, \gamma) \quad (5)$$

The gradient of Ns can then be recovered with respect to the i-th row (pin coordinates) of A_e:

$$\frac{\partial Ns}{\partial (A_e)_i} = \mathbb{1}_{(A_e)_i \cdot u \leq \gamma+1} 2\langle u, (\gamma + 1)\mathbf{1} - A_e u\rangle$$

Note that due to the reliance of the above gradient on γ, and the reliance of γ on pin coordinates A_e, we adopt an alternating minimization method described in Sec. III.B.
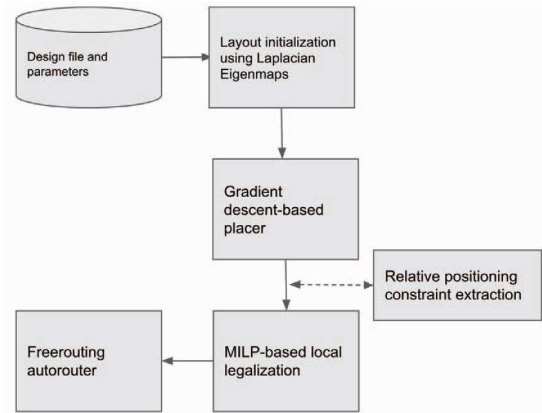
IV. NS-PLACE PLACEMENT FLOW

Fig. 3: Placement procedure with Laplacian Eigenvector initialization, net-separation minimization, and MILP-based legalization with relative positioning constraints.

In this section, we describe the overall flow of our method. The high-level flow is described in Figure 3.

A. Initialization with Laplacian eigenvectors

First-order optimization algorithms are notoriously sensitive to initialization when applied to nonlinear problems. We address this by first collapsing the netlist hypergraph to a component graph via the clique model. We then construct a matrix-representation of the graph connectivity—the *graph Laplacian*. The solution to the associated eigenvalue problem approximates the solution to the sparsest cut problem [18], and clusters arising out of the vertex-projection into the space spanned by the first nontrivial eigenvalues correspond highly connected components of the graph. We use these coordinates to initialize global placement. More concretely, we solve the following problem, where x and y are coordinates of components, c_i are constants, v is a vector of component areas, $V = \text{diag}(v)$, and L is the normalized graph Laplacian; $L = R^{-\frac{1}{2}} A R^{-\frac{1}{2}}$, where A is an adjacency matrix and R is the diagonal degree matrix.

$$\min \ x^\top L x + y^\top L y$$
$$\text{s.t. } v^\top x = 0, \qquad v^\top y = 0, \qquad x, y \neq 0 \quad (6)$$
$$x^\top V x = c_1, \qquad y^\top V y = c_2, \qquad x^\top V y = c_3$$

Intuitively, the objective is to minimize the weighted squared wire length. The linear and non-equality constraints concentrate the layout about the origin and prevent the trivial solution. The quadratic constraints spread the placement over the x and y axes.

The above problem is unaware of component orientations or fixed position constraints. To resolve this, we generate candidate initializations by considering all possible relative component orientations for a given solution to Problem 6. The solution with minimal cost with respect to Eq. 2 is used to seed the global placer.

B. Global placement using coordinate descent

In Alg. 1, we present the detailed steps of our iterative method to reduce congestion.

Algorithm 1 Net crossing minimization

Input: Initial placement $[x:y]$, pin matrices A_e, $A_{e'}$, regularization parameter λ , learning rate α, budget n
Output: Placement $[x:y]$

1: **function** NSOPT $([x:y], A_e, A_{e'}; \lambda, \alpha)$
2:     **while** Eq. 2 not converged **do**
3:         Fix A_e, $A_{e'}$, compute u, γ by Eq. 4   ▷ separator
4:         Fix u, γ, compute $\nabla_{p_i} F$, P
5:         $g \leftarrow \lambda_D \nabla_{[x:y]} D + P$ ▷ compute component update
6:         $[x:y]_{t+1} \leftarrow [x:y]_t + \alpha \cdot g$ ▷ update components
7:         update A_e, $A_{e'}$ according to $[x:y]_{t+1}$
8:     **end while**
9:     **return** $[x:y]_n$
10: **end function**

Recall the proposed global placement optimization problem described in Eq. 2. For brevity, we refer to the cumulative objective as F. Note that the above problem may be solved exactly via quadratic programming. We propose to solve the problem approximately by applying first-order methods in an alternating minimization framework by iteratively solving for u and γ while keeping x and y fixed and visa-versa. Given a layout, we first compute the separating hyperplane characterized by u and γ in Eq. 4 (line 4 of Alg. 1). Given u and γ, we can derive the net separation and wirelength gradients associated with individual pins (line 4). To recover the component position update, we introduce the auxiliary variable $P \in \mathbb{R}^{n \times 2}$ corresponding to component update derived from pin-gradients where the i-th row of P is defined to be the average of the ith component's pin gradients—i.e. $P_i = \frac{1}{|\mathcal{P}(i)|} \sum_{p \in \mathcal{P}(i)} \nabla_p F$. The gradients associated with the density and wirelength terms can then be computed, and component positions updated (lines 5-6). Given these new positions, the pin matrices A_e and $A_{e'}$ can then be updated (line 7).

C. Legalization via mixed integer linear programming

In this section, we introduce a standard MILP-based layout formulation for placing rectilinear components subject to overlap and boundary constraints. We note that this formulation shares similarities with previous work, e.g. [19] who discuss the optimality of MILP for wirelength-minimal block placements. Additionally, we integrate relative positioning constraints derived from the global placement procedure to preserve net separation while improving scalability.

MILP-based wirelength-minimal layout The objective and constraints are described in the following set of equations. We introduce a second term corresponding to wirelength variance, which we find improves the routability of designs further.

$$\min_{x,y,r} \left[\sum_{i \in |\mathcal{E}|} \text{hpwl}(i) + \left[\max_{i \in |\mathcal{E}|} \text{hpwl}(i) - \min_{i \in |\mathcal{E}|} \text{hpwl}(i) \right] \right]$$
$$\text{hpwl}(i) = (U_x^{(i)} - L_x^{(i)}) + (U_y^{(i)} - L_y^{(i)})$$

where the hpwl term (given for the x-direction only) is

$$U_x^{(i)} \geq p_j^{(i)}(x), \quad L_x^{(i)} \leq p_j^{(i)}(x) \quad \forall j \in \mathcal{E}_i$$

and the solution is subject to non-overlapping constraints (for brevity, boundary constraints are not included):

$$x_i + r_i h_i + (1 - r_i)w_i \leq x_j + W(p_{ij} + q_{ij}) \qquad \text{i-left-j}$$
$$y_i + r_i w_i + (1 - r_i)h_i \leq y_j + H(1 + p_{ij} - q_{ij}) \qquad \text{i-under-j}$$
$$x_i - r_j h_j - (1 - r_j)w_j \geq x_j - W(1 - p_{ij} + q_{ij}) \qquad \text{i-right-j}$$
$$y_i - r_j w_j - (1 - r_j)h_h \geq y_j - H(2 - p_{ij} - q_{ij}) \qquad \text{i-over-j}$$
$$x_i, y_i \geq 0, \quad r_i, q_{ij}, p_{ij} \in \{0, 1\} \qquad \text{variables}$$

Where r_i, q_{ij}, p_{ij} are variables representing orientation and relative positions between modules i and j.

Relative positioning constraints Given a global placement solution, we derive relative position constraints for pairs of components. By doing so, we preserve the global structure of the global placement solution while allowing the MILP placer to make local adjustments.

For each pair of modules, we consider the minimum horizontal and vertical distances between modules (i.e. from boundary to boundary). We then check if the maximum of the horizontal and vertical distances exceeds a pre-defined threshold k which trades off a preference for HPWL-minimal solutions or routability and runtime. A horizontal or vertical relative position constraint is introduced depending on which distance is greater and the associated binary decision variables (q_{ij} or p_{ij}) are removed and the corresponding overlap constraints are simplified or pruned entirely. By integrating these constraints, we preserve the global structure of the analytical solution while eliminating a nontrivial number of decision variables.

V. EXPERIMENTS

A. Experiment setup

We applied our method to 14 PCB benchmarks [20]. Details are provided in Table I. We solve Eq. 5 via gradient descent with momentum with $\alpha = 1e^{-3}$, $\lambda_{\text{NS}} = 1$, and $\lambda_{\text{D}} = 1$. GDMILP corresponds to running the NS-Place with $\lambda_{\text{NS}} = 0$, and MILP corresponds to solving the MILP without relative-position constraints. Layout quality is evaluated using the open-source FreeRouting router [21]. The routed wirelength, number of design rule violations, and the number of vias are reported. Experiments are performed with an 3.4GHz Intel i7-4770 CPU and 31GB RAM. If no optimal solution is found after 4 hours, we report results for the best-so-far (with respect to the MILP objective) feasible solution. If no feasible solution is found, associated entries are marked with "-".

TABLE I: Design characteristics. locked are fixed components. layers are layers available for routing.

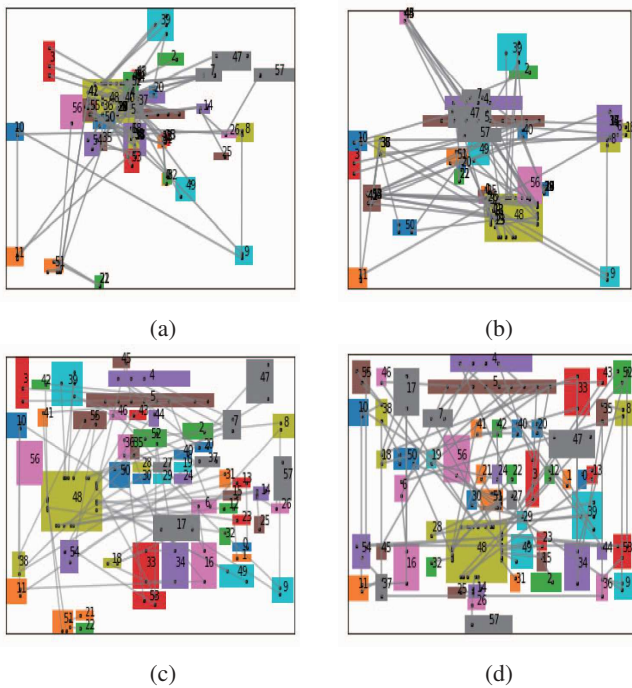| design	$W \times H(mm^2)$	#comp	#locked.	util.	#nets	#pins	#layers
PCB1	21×14	8	1	0.59	15	40	1
PCB2	51×23	18	5	0.79	34	77	2
PCB3	55×28	34	2	0.44	38	138	2
PCB4	23×60	28	6	0.67	52	140	2
PCB5	41×22	48	2	0.40	54	163	2
PCB6	62×57	48	2	0.17	64	190	2
PCB7	51×23	46	2	0.55	69	211	2
PCB8	57×87	36	2	0.62	70	188	2
PCB9	44×36	58	2	0.60	80	229	2
PCB10	102×54	57	18	0.21	99	319	2
PCB11	89×58	64	2	0.10	134	401	4
PCB12	58×60	58	4	0.31	35	233	4
PCB13	86×72	61	4	0.51	63	314	4
PCB14	86×54	1570	947	0.64	386	1638	4

B. Main experiments

Fig. 4: **PCB12 layouts**. (**a**): Seed placement produced from Laplacian eigenvectors. (**b**) Global placement to minimize net crossings. (**c**) MILP-based legalization. (**d**) Manual layout.

PCB placement metric comparison We provide manual, MILP, GDMILP, and NS-Place placement results in table II. We show that NS-Place reduces the net separation cost by 77% and 41% on average compared to manual and GDMILP. This implies that although the MILP-based fine-tuning step does not optimize net separation explicitly, satisfaction of relative position constraints results in preservation of the global structure produced by the net separation step. Furthermore, the inclusion of the net separation term does not result in a large increase in HPWL. Although the HPWL of NS-Place solutions increase 9% on average compared to GDMILP, NS-Place still achieves 20% lower HPWL compared to manual layouts.

We report the cumulative runtime to complete a placement in Table II. Note that methods employing relative positioning constraints (i.e., GDMILP, NS-Place) strictly improve over

vanilla MILP. We see that NS-Place closely matches the performance of the GDMILP flow with only around 4% increase on average runtime. This implies that the net separation regularizer imposes low overhead. Fig. 4 (a), (b), and (c) show the PCB placement results of PCB12 of each stage of automatic placement flow. Fig. 4 (d) is the manual layout of PCB12. Compared to Fig. 4 (d), we observe that NS-Place produces placement with fewer net crossings as shown in Fig. 4 (c).

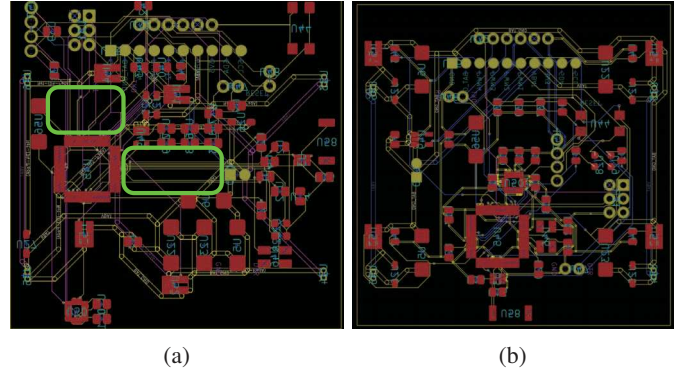**PCB routing metric comparison** We evaluate routability

Fig. 5: **PCB12 P&R result in KiCAD**. (**a**) Solution produced by our method. Routable regions are emphasized with green rectangles. (**b**) Routed manual placement.

in Table III by reporting the routed wirelength, the number of DRVs and vias, and the number of unrouted nets using FreeRouting and Kicad. Compared with manual, MILP, and GDMILP, NS-Place reduces the #DRVs and #unrouted nets by roughly 80%, 70%, and 75% on average.

NS-Place additionally reduces the routed wirelength by 10% and 6% on average compared to manual and GDMILP respectively and reduces the #Vias by 18% and 39%. Moreover, for PCBs 7, 10, 11, and 13, which have #components larger than 60 and #pins larger than 300, NS-Place achieves 34% #Vias reduction on average compared to manual. An example routed result is given in Fig. 5. Compared to the manual layout, NS-Place successfully improves routability via net separation (i.e., the routable regions near the center of the board). Note that NS-Place fails to produce a superior solution for PCB2. We attribute this to PCB2's high utilization, with only about 20% of the board area available to the global placer as whitespace. In summary, we demonstrate that NS-Place significantly reduces #DRVs, #unrouted nets, and #Vias for PCB reliability compared to manual, MILP, and GDMILP with an extensive study on 14 PCBs.

VI. CONCLUSION AND FUTURE WORK

We have presented a novel algorithm which encourages routability of multi-pin nets in PCB designs. In an extensive study on 14 PCB designs, we have demonstrated that NS-Place achieves 80.98%, 70.00%, and 74.68% reduction on average #DRVs and #unrouted nets for routability and 34.36% fewer #Vias on average. We plan to extend our technical and empirical analysis to integrated circuit (IC) placement and explore congestion minimization via nonlinear separators and plan to investigate methods to optimize component orientation

TABLE II: Pre-route metrics for PCB designs. We report the cumulative HPWL and the net separation cost. The top performing result is **bolded**. The "-" represents that MILP cannot produce a feasible placement in 4 hours.

Design	HPWL (mm)				Net Separation Obj.				Runtime (s)		
	manual	MILP	GDMILP	NS-Place	manual	MILP	GDMILP	NS-Place	MILP	GDMILP	NS-Place
PCB1	110.22	**64.44 (41.50)**	68.90 (37.4)	72.10 (34.60)	192.13	192.11	192.11	**187.40**	2.70	4.90	5.10
PCB2	362.70	**291.10 (19.70)**	294.32 (18.80)	296.18 (18.30)	497.50	497.40	499.30	**493.10**	6.30	6.20	6.80
PCB3	312.60	**212.70 (32.0)**	252.40 (19.30)	271.30 (13.20)	1427.10	2124.30	1426.40	**1411.40**	14400.00	3793.00	3917.60
PCB4	603.30	-	536.48 (11.10)	**531.19 (12.00)**	3750.20		3753.10	**3746.13**	-	9.40	9.90
PCB5	654.10	-	**621.00 (5.10)**	637.00 (2.60)	2400.00		2400.00	**2300.00**	-	4231.20	4461.50
PCB6	771.80	-	**649.40 (15.90)**	708.60 (8.40)	1930.00		1940.00	**1740.00**	-	4893.10	5072.40
PCB7	2987.90	-	**563.10 (88.50)**	1601.40 (46.40)	1241.34		1239.70	**1017.16**	-	5927.30	6008.20
PCB8	766.10	**731.30 (4.60)**	748.40 (2.30)	756.40 (1.30)	98500.00	99500.00	99400.00	**96400.00**	14400.00	4360.90	4732.40
PCB9	714.90	-	677.60 (5.20)	**662.50 (7.30)**	2600.00		2600.00	**2400.00**	-	5327.30	5719.80
PCB10	4355.61	-	3317.94 (23.80)	**3315.46 (23.90)**	2117.80		2124.10	**2103.60**	-	5314.70	5417.20
PCB11	2941.90	-	**2573.20 (12.50)**	2618.10 (11.00)	9210.00		9210.00	**9070.00**	-	4651.90	4782.40
PCB12	972.50	929.30 (4.40)	**932.60 (4.10)**	941.30 (3.20)	10.73	11.31	11.47	**36.92**	14400.00	4619.10	4752.30
PCB13	2644.97	-	**2126.13 (19.60)**	2151.77 (18.60)	2400.00		2400.00	**2300.00**	-	3278.60	3416.30
PCB14	7069.26	-	**6432.19 (9.01)**	6691.34 (5.35)	29400		29870	**11186**	-	14400.00	14400.00

TABLE III: Post-route metrics for PCB designs. We report routed wirelength using FreeRouting, the number of vias, and the number of DRVs as reported by KiCAD. We report the percent improvement in parenthesis. The best result is **bolded**.

Design	Routed Wirelength (mm)				#Vias				#DRVs + #unrouted nets			
	manual	MILP	GDMILP	NS-Place	manual	MILP	GDMILP	NS-Place	manual	MILP	GDMILP	NS-Place
PCB1	129.00	123.00 (4.70)	194.00 (-50.4)	**121.00 (6.20)**	**0**	10	9	4	6	**0**	**0**	**0**
PCB2	**354.00**	632 (-78.50)	507.00 (-43.20)	421.00 (-18.90)	**3**	6	8	5	13	26	24	23
PCB3	638.00	682.00 (-6.90)	771.00 (-20.80)	**616.00 (3.40)**	17	21	24	**15**	11	2	3	**0**
PCB4	809.00	-	857.00 (-5.90)	**806.00 (0.40)**	31	-	59	54	17	-	4	**0**
PCB5	558.00	-	**538.00 (3.60)**	541.00 (3.00)	**32**	-	84	49	7	-	5	**3**
PCB6	1007.00	-	**854.00 (15.90)**	906.00 (10.00)	23	-	47	**19**	34	-	29	**0**
PCB7	3735.00	-	3140.00 (15.90)	**2949.00 (21.00)**	161	-	141	**93**	23	-	**0**	**0**
PCB8	913.00	854.30 (6.30)	713.40 (21.90)	**711.90 (22.00)**	57	73	89	**43**	27	6	5	**0**
PCB9	1069.00	-	**749.00 (29.90)**	772.00 (27.80)	**38**	-	87	64	17	-	7	**0**
PCB10	5043.00	-	5294.00 (-1.00)	**4914.00 (2.60)**	129	-	136	**97**	12	-	3	**0**
PCB11	3460.00	-	3271.90 (5.40)	**3107.80 (10.20)**	131	-	286	**109**	23	-	57	13
PCB12	1790.00	1960.00 (-9.40)	1930.00 (7.80)	**1720.00 (3.90)**	49	62	54	**45**	4	9	8	**0**
PCB13	3150.00	-	2903.00 (7.80)	**2897.00 (8.00)**	161	-	98	**83**	11	-	9	**0**
PCB14	9017.00	-	9643.00 (-6.94)	**8873.00 (1.60)**	976	-	1007	**942**	104	-	139	92

during the net separation minimization procedure. Finally, while first-order methods are efficient in practice, due to the nonlinearity of our objective, optimal solutions are not guaranteed and initialization remains an open problem.

VII. Acknowledgements

References

[1] C. Douglas, "Wafer level system integration for sip," in *2014 IEEE Int. Electron Devices Meeting*. IEEE, 2014.

[2] ——, "Advanced packaging with greater simplicity," in *2017 IEEE Int. Electron Devices Meeting*. IEEE, 2017.

[3] M.-F. Chen, F.-C. Chen, W.-C. Chiou, and C. Doug, "System on integrated chips for 3d heterogeneous integration," in *69th Electronic Components and Technology Conf.* IEEE, 2019.

[4] C. Cheng, A. B. Kahng, I. Kang, and L. Wang, "Replace: Advancing solution quality and routability validation in global placement," *IEEE Trans. on Comp.-Aided Design of Integrated Circuits and Systems*, 2018.

[5] J. Lu, P. Chen, C.-C. Chang, L. Sha, D. J.-H. Huang, C.-C. Teng, and C.-K. Cheng, "eplace: Electrostatics-based placement using fast fourier transform and nesterov's method," *ACM Trans. Des. Autom. Electron. Sys.*, vol. 20, 2015.

[6] P. Spindler and F. M. Johannes, "Fast and accurate routing demand estimation for efficient routability-driven placement," in *2007 Design, Automation Test in Europe Conf.*, 2007.

[7] C. Li, M. Xie, C.-K. Koh, J. Cong, and P. Madden, "Routability-driven placement and white space allocation," in *IEEE/ACM Int. Conf. on Computer Aided Des.*, 2004.

[8] J. A. Roy and I. L. Markov, "Seeing the forest and the trees: Steiner wirelength optimization in placement," *IEEE Trans. on Comp.-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 4, 2007.

[9] A. B. Kahng, S. Reda, and Q. Wang, "Aplace: A general analytic placement framework," in *Proc. of the Int. Sym. on Phys. Des.*, 2005.

[10] A. Shabbeer, C. Ozcaglar, and K. P. Bennett, "Crossing minimization within graph embeddings," 2012.

[11] S. Jain and H. C. Gea, "PCB Layout Design Using a Genetic Algorithm," *J. of Electronic Packaging*, vol. 118, no. 1, 1996.

[12] F. S. Ismail, R. Yusof, and M. Khalid, "Optimization of electronics component placement design on pcb using self organizing genetic algorithm," *J. of Intell. Manuf.*, vol. 21, 2012.

[13] T. Badriyah, F. Setyorini, and N. Yuliawan, "The implementation of genetic algorithm and routing lee for pcb design optimization," in *Int. Conf. on Informatics and Computing*, 2016.

[14] A. Alexandridis, E. Paizis, E. Chondrodima, and M. Stogiannos, "A particle swarm optimization approach in printed circuit board thermal design," *Integ. Comp.-Aided Eng.*, vol. 24, 2017.

[15] S. N. Adya and I. L. Markov, "Combinatorial techniques for mixed-size placement," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 10, 2005.

[16] M. Hsu, Y. Chang, and V. Balabanov, "Tsv-aware analytical placement for 3d ic designs," in *ACM Trans. Des. Autom. Conf.*, vol. 48, 2011.

[17] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, 1995.

[18] K. M. Hall, "An r-dimensional quadratic placement algorithm," *Management Science*, vol. 17, no. 3, Nov. 1970.

[19] J. Funke, S. Hougardy, and J. Schneider, "An exact algorithm for wirelength optimal placements in VLSI design," *Integration*, vol. 52, 2016.

[20] T.-C. Lin, D. Merrill, Y.-Y. Wu, C. Holtz, and C.-K. Cheng, "A unified printed circuit board routing algorithm with complicated constraints and differential pairs," in *Proc. of the 26th Asia and South Pacific Design Automation Conf.*, 2021.

[21] "Freerouting." [Online]. Available: https://freerouting.org/