On a Moreau Envelope Wirelength Model for Analytical Global Placement

Peiyu Liao[1, 2], Hongduo Liu[2], Yibo Lin[1, 3*], Bei Yu[2*], Martin Wong[2]
[1]Peking University, Beijing, China [2]The Chinese University of Hong Kong, Hong Kong, China
[3]Institute of Electronic Design Automation, Peking University, Wuxi, China

***Abstract*—Analytical placement is proven to be effective in global placement. The differentiability of wirelength models is very critical to gradient-based numerical optimization. Most previous works approximate the non-smooth half-perimeter wirelength (HPWL) model with various differentiable functions. In this paper, we propose a new differentiable wirelength model using the Moreau envelope to approximate HPWL. By combining the state-of-the-art electrostatic-based placement algorithm, the experimental results demonstrate that our proposed algorithm can achieve up to 5.4% HPWL improvement and more than 1% on average compared to the most widely-used nonlinear wirelength model.**

I. Introduction

In order to find suitable locations for circuit components, circuit placement is an essential step in physical design with various objectives and constraints. Different objectives may induce different problem formulations. However, the widely adopted strategy is to perform wirelength-driven global placement first, which directly minimizes the total wirelength of the entire circuit. Analytical placers formulate the global placement problem as mathematical programming with certain constraints on legality [1]–[11].

Thanks to the rapid development of computing power and the popularity of neural networks, an efficient implementation of numerical optimization approaches has gained the great attention of engineers and developers. In order to apply gradient-based numerical optimization approaches, we require the wirelength model to be differentiable everywhere. However, the most popular half-perimeter wirelength model (HPWL) function containing maximum and minimum coordinates is not everywhere differentiable [12]. Therefore, modern placement algorithms heavily rely on differentiable approximations of the HPWL model. There are two main categories, quadratic approximations and nonlinear approximations.

Quadratic models like [3]–[7], [10], [11] approximate every edge cost with the squared length, which gives a strictly convex objective that is very convenient for us to optimize. The general form of quadratic objective has closed-form minimizers which can be found by solving linear systems. However, quadratic models have poor approximation error bounds. To better approximate HPWL with quadratic models, some linearization techniques [13] have been proposed. They integrate certain net weights inversely proportional to net length so that the weighted squared term can be closer to HPWL. The *Bound2Bound* (B2B) model [7], [14] decomposes larger nets by selecting boundary pins and connecting them to each internal node.

Commonly, nonlinear analytical placers use differentiable functions, like the *log-sum-exp* [15] model and the *weighted-average* [16], [17] model, to approximate HPWL to arbitrary precision by controlling some hyperparameters. The state-of-the-art placers [1], [2], [8], [9], [18]–[20] adopt nonlinear analytical models as they approximate HPWL more accurately, have closed-form gradient representations, and can be naturally optimized by gradient descent.

Some works have been seeking new ways to optimize HPWL. The *Bivariate-Gradient-Based* (BiG) model [21], inspired by [22], calculates gradients with respect to coordinates by applying bivariate functions that approximate bivariate maximum or minimum to improve numerical stability and CPU runtime. Subgradient-based approaches [23] directly optimize the non-smooth ℓ_1 wirelength with the B2B model [7], [14], using the Polak-Ribière-Polyak conjugate subgradients method [23], [24].

The WA model [16], [17] has a lower approximation error bound than the LSE model [15], but it still have to face numerical stability and non-convexity issue. In comparison, the BiG model [21] has the advantages of numerical stability and cheap computation, but it relies on a bivariate wirelength model that needs to be specified. Non-smooth optimization approaches do not require any differentiable approximations, but they may encounter an issue of slow and poor convergence.

In this paper, we propose a new differentiable wirelength model using the *Moreau envelope* [25] to approximate HPWL. The proposed model is superior to previous models in numerical stability, convexity, and approximation error. We will also propose algorithms to compute the objective and gradients. To make such computation possible, we build upon the proximal mapping of the HPWL model. The major contributions are summarized as follows.

- We derive the explicit representations of proximal mapping of the non-smooth HPWL function and propose a *water-filling* algorithm similar to that in communication systems [26]–[29] to solve the proximal mapping and Moreau envelope problem.
- We provide theoretical and experimental analysis to compare the proposed wirelength model and the widely-used weighted-average model [16], [17] and verify its feasibility.
- Experimental results show that the proposed wirelength model can achieve up to 5.4% HPWL improvements and over 1% on average for the ISPD2006 contest benchmarks [30]. In addition, we also achieve up to 3.0% HPWL improvements and over 1.5% on average on the recent ISPD2019 contest benchmarks [31].

The rest of the paper is organized as follows. Section II introduces some preliminaries, including foundations of nonlinear placement, wirelength models, and the Moreau envelope. Section III presents the algorithms of the proposed methods. Then, Section IV gives some theoretical properties of our wirelength model. Section V presents experimental results and some related analysis on the adopted benchmarks, followed by the conclusion in Section VI.

II. Preliminaries

A. Analytical Global Placement

Circuit placement usually consists of three steps: global placement, legalization, and detailed placement. At the global placement stage, we aim to find good cell locations with small total wirelength such that

This work is supported by The Research Grants Council of Hong Kong SAR (No. CUHK14209420), National Science Fundation of China (No. 62141404), and The 111 Project (No. B18001).

[*]Corresponding authors

the overlaps are controlled at a low level. A typical nonlinear global placement problem is formulated as

$$\min_{\boldsymbol{x},\boldsymbol{y}} \sum_{e\in E} W_e(\boldsymbol{x},\boldsymbol{y}) + \lambda D(\boldsymbol{x},\boldsymbol{y}), \tag{1}$$

where E is the net set, $W_e(\cdot)$ is the net wirelength function of net $e \in E$, $D(\boldsymbol{x},\boldsymbol{y})$ models the cell density penalty, and λ is the corresponding density weight in the objective.

The wirelength model is the detailed representation of the wirelength function $W_e(\boldsymbol{x},\boldsymbol{y})$. For simplicity, we also use $\boldsymbol{x},\boldsymbol{y}$ to represent corresponding pin coordinates ignoring pin offsets in this Section II. Then, the most widely-adopted HPWL model is defined as $W(\boldsymbol{x},\boldsymbol{y}) = \sum_{e\in E} W_e(\boldsymbol{x},\boldsymbol{y})$, where the net HPWL of $e \in E$ is

$$W_e(\boldsymbol{x},\boldsymbol{y}) = \max_{i\in e} x_i - \min_{i\in e} x_i + \max_{i\in e} y_i - \min_{i\in e} y_i. \tag{2}$$

The HPWL function defined in Equation (2) is convex and continuous but not everywhere differentiable. Fortunately, it has a simple and clean formulation. Therefore, many approximation models have been proposed. Here we focus on the *nonlinear* models.

B. Nonlinear Differentiable Models

Since most analytical placement algorithms expect to optimize HPWL, we are going to discuss nonlinear differentiable HPWL-based wirelength models. Consider a net containing n pins with horizontal coordinates $\boldsymbol{x} \in \mathbb{R}^n$. There are two widely-used exponential approximations, the *log-sum-exp* (LSE) model [15] and the *weighted-average* (WA) model [16], [17],

$$W_e(\boldsymbol{x}) \approx W^{\gamma}_{e,\mathrm{LSE}}(\boldsymbol{x}) = \gamma \ln \sum_{i=1}^{n} \exp\left(\frac{x_i}{\gamma}\right) + \gamma \ln \sum_{i=1}^{n} \exp\left(-\frac{x_i}{\gamma}\right),$$
$$W_e(\boldsymbol{x}) \approx W^{\gamma}_{e,\mathrm{WA}}(\boldsymbol{x}) = \frac{\sum_{i=1}^{n} x_i \exp\left(\frac{x_i}{\gamma}\right)}{\sum_{i=1}^{n} \exp\left(\frac{x_i}{\gamma}\right)} - \frac{\sum_{i=1}^{n} x_i \exp\left(-\frac{x_i}{\gamma}\right)}{\sum_{i=1}^{n} \exp\left(-\frac{x_i}{\gamma}\right)}, \tag{3}$$

where γ is a hyperparameter to control the tradeoff between the accuracy and differentiability. When $\gamma \to 0^+$, the approximation will be arbitrarily close to the real HPWL function. Usually at the earlier stages of global placement, γ is set to a large value so that the objective can be very smooth, and decreases as the number of iterations increases.

C. Moreau Envelope

The general Moreau envelope [25] is defined for functions satisfying specific constraints on a real *Hilbert space*. In our placement applications, we only consider lower bounded *closed convex* functions $h(\boldsymbol{x})$ defined in $\mathbb{R}^n$ to simplify the notations. For any $t > 0$, let $H(\boldsymbol{u},\boldsymbol{x}) = h(\boldsymbol{u}) + \frac{1}{2t}\|\boldsymbol{u} - \boldsymbol{x}\|_2^2$, then the Moreau envelope h^t and the proximal operator are defined by

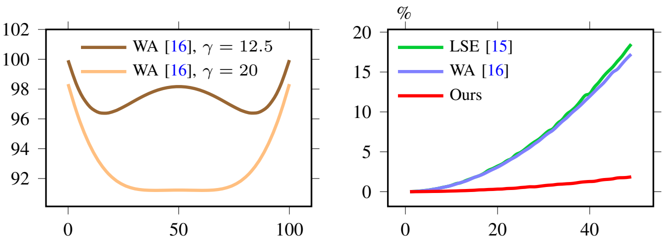
$$h^t(\boldsymbol{x}) = \min_{\boldsymbol{u}\in\mathbb{R}^n} H(\boldsymbol{u},\boldsymbol{x}),\ \mathrm{prox}_{th}(\boldsymbol{x}) = \operatorname*{argmin}_{\boldsymbol{u}\in\mathbb{R}^n} H(\boldsymbol{u},\boldsymbol{x}). \tag{4}$$

Note that the proximal mapping may or may not have a closed-form representation, and it may or may not be easy to calculate, up to the form of function $h(\cdot)$. Also, we do not require function h to be smooth.

Under the constraints of $h(\cdot)$ specified above, the Moreau envelope h^t is a natural differentiable approximation of h, and its gradient is globally Lipschitz continuous. More specifically, the *envelope theorem* [32] states that

$$\nabla h^t(\boldsymbol{x}) = \frac{1}{t}\left(x - \mathrm{prox}_{th}(\boldsymbol{x})\right). \tag{5}$$

Since $\lim_{t\to 0^+} h^t(\boldsymbol{x}) = h(\boldsymbol{x})$ converges pointwise, t is considered to be the approximation precision, similar to γ in LSE [15] and WA [16].



(a) (b)

Fig. 1 (a) The non-convexity of the Weighted-Average (WA) [16] on a simple 3-pin net to approximate $\Delta x = \max\{x_{\min}, x, x_{\max}\} - \min\{x_{\min}, x, x_{\max}\}$. (b) The average approximation error of different models LSE [15], WA [16] and Moreau Envelope against the smoothing parameter γ or t for 4-pin nets, under fixed $\Delta x = 200$.

The interesting point is that, in global placement, we are inspired to consider the net HPWL W_e, which has a clean and straightforward formulation. In this paper, we propose an algorithm to compute its Moreau envelope W_e^t and use $W_e^t + t$ as the approximated wirelength model so that the entire objective is everywhere differentiable.

D. Comparison between WA and the Moreau Envelope models

In this subsection, we will briefly compare WA [16] and the Moreau envelope model. Equation (3) uses exponential terms to assign weights to each coordinate. A larger coordinate x_i has a large weight in the smooth maximum, as $\exp(\frac{x_i}{\gamma})$ would occupy a larger proportion.

*1) **Numerical Stability**:* The exponential terms are sensitive to the coordinate. Note that the Δx will usually be hundreds or even over thousands in actual placement, so the γ should not be very small, as a small γ will be likely to result in a numerical overflow. This phenomenon is also stated in [21]. Nearly all exponential models like the LSE [15] and WA [16] models have to face such an issue.

*2) **Non-Convexity**:* There is no theoretical or experimental guarantee of convexity of the WA [16] model. Let us take a 3-pin net as an example and conduct a toy experiment on its horizontal pin coordinate $\boldsymbol{x} \in \mathbb{R}^3$. Assume that we fix the $\Delta x = 100$, *i.e.*, let $\boldsymbol{x} = (0, x, 100)^\top$ without loss of generality where $x \in [0, 100]$. We plot the function curve of $W^{\gamma}_{e,\mathrm{WA}}(\{0, x, 100\})$ for some γ values in Fig. 1(a). As shown in the figure, even for the 3-pin net, the WA model can be non-convex. For high-degree nets, it will get more complicated. Although the optimization process of real placement problems is far more obscure than imagined, a convex wirelength model is usually preferred. As a comparison, the Moreau envelope approximation is always convex [25] for HPWL.

*3) **Approximation Error**:* The net wirelength models can be high-dimensional and thus extremely difficult to analyze. We conduct a toy experiment on the smoothing parameters γ (for LSE [15] and WA [16]) and t. Given a fixed range $\Delta x = x_{\max} - x_{\min} = 200$, we randomly generate horizontal coordinates $\boldsymbol{x} \in \mathbb{R}^4$ for different smoothing parameters 3000 times, calculate the approximated Δx using different wirelength models, take the average, and draw curves in Fig. 1(b).

Although the smoothing parameters here have different mathematical meanings in different wirelength models, the advantage of our proposed model on the approximation error is still well-illustrated in Fig. 1(b). It is worth mentioning that a lower approximation error does not always imply better solution quality after placement, as the optimization procedure is so complicated that it can be easily influenced by various hidden factors. In addition to the approximation

Algorithm 1 GRADIENT ALGORITHM

```
Require: The horizontal (or vertical) pin coordinates $\boldsymbol{x} \in \mathbb{R}^n$, the smoothing parameter $t > 0$.
Sort pin coordinates $\boldsymbol{x}$ such that $x_1 \leq \cdots \leq x_n$;
Apply the WATER-FILLING ALGORITHM described in Algorithm 2 to solve equations
    $$\sum_{i=1}^{n}(x_i - \tau_2)^+ = \sum_{i=1}^{n}(\tau_1 - x_i)^+ = t$$
  to obtain water-filling parameters $\tau_1$ and $\tau_2$;
if $\tau_1 > \tau_2$ then
    Calculate average pin coordinate $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$;
    Assign $\tau_1 \leftarrow \bar{x}$;
    Assign $\tau_2 \leftarrow \bar{x}$;
end if
Given $\tau_1$ and $\tau_2$, calculate gradient $\boldsymbol{g} = \nabla W_e^t(\boldsymbol{x})$ according to Corollary 1;
return the required gradient $\boldsymbol{g}$;
```

error, the gradient properties should also be carefully considered, which will be discussed in Section IV.

III. ALGORITHM

We adopt the Moreau envelope model as a differentiable approximation of net HPWL, which requires the proximal mapping of HPWL. In this paper, we use the *rectified linear unit* (ReLU) activation function $\eta^+ = \max\{\eta, 0\}$ to represent the *positive part* of $\eta \in \mathbb{R}$.

A. The Gradient of the Moreau Envelope

Without loss of generality, consider a single net e connecting n pins $p_1, \cdots, p_n$ with horizontal coordinates $\boldsymbol{x} \in \mathbb{R}^n$. The horizontal part of HPWL function is represented by

$$W_e(\boldsymbol{x}) = \max_{1 \leq i \leq n} x_i - \min_{1 \leq i \leq n} x_i. \tag{6}$$

The horizontal and vertical directions are symmetric, so we only focus on the horizontal direction for analysis.

At each placement iteration, we are required to calculate the gradient of the wirelength model w.r.t. the pin coordinates $\boldsymbol{x}$. The overall algorithm to calculate gradient $\boldsymbol{g}$ of the Moreau envelope $W_e^t(\boldsymbol{x})$ under the smoothing parameter t is decribed in Algorithm 1. The following Theorem 1 states the representation of the proximal mapping $\text{prox}_{tW_e}(\boldsymbol{x})$. Then Corollary 1 extends Theorem 1 and tells us how to calculate gradient $\boldsymbol{g}$ given parameters τ_1 and τ_2. The detailed algorithm of calculating parameters τ_1 and τ_2 is described in Algorithm 2.

Theorem 1. *The proximal mapping of $t \cdot W_e$ for a positive real number $t > 0$ is given by $\text{prox}_{tW_e}(\boldsymbol{x}) = \boldsymbol{u}^*$ where*

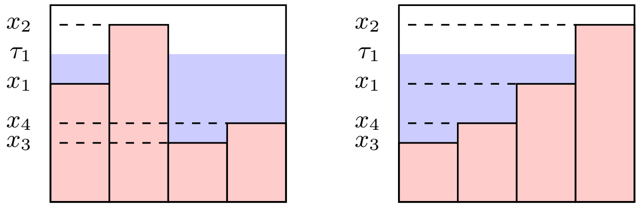
$$u_i^* = \begin{cases} \tau_2, & \textit{if } x_i > \tau_2 \\ x_i, & \textit{if } \tau_1 \leq x_i \leq \tau_2 \\ \tau_1, & \textit{otherwise} \end{cases} \tag{7}$$

is defined for any $i = 1, \cdots, n$, such that

$$\sum_{i=1}^{n}(x_i - \tau_2)^+ = \sum_{i=1}^{n}(\tau_1 - x_i)^+ = t, \tag{8}$$

if the solution τ_1, τ_2 to Equation (8) satisfy $\tau_1 \leq \tau_2$, otherwise $\boldsymbol{u}^*$ is determined by the average coordinate $\boldsymbol{u}_i^* = \frac{1}{n}\boldsymbol{e}^\top \boldsymbol{x}$ for any index $i = 1, \cdots, n$.

Theorem 1 discusses the explicit representation of the horizontal part of HPWL w.r.t. pin locations. This is the core of the approximation. However, the proof is complicated and out of scope, and thus not attached due to page limit. Note that it is NOT a complete closed-form representation, as we still need to solve Equation (8) for τ_1 and τ_2. The *water-filling* algorithm to solve these equations will be discussed in Section III-B. Before that, we state the following corollary.



(a) Pin coordinates statistics

(b) Sorted pin coordinates

Fig. 2 The illustration of water-filling to solve τ_1 in Equation (8).

Corollary 1. *Consider the horizontal part of the net HPWL $W_e(\boldsymbol{x}) = \max_{1 \leq i \leq n} x_i - \min_{1 \leq i \leq n} x_i$, its Moreau envelope function W_e^t is everywhere differentiable. The gradient is $\boldsymbol{g} = \nabla W_e^t(\boldsymbol{x})$ where*

$$g_i = \begin{cases} \frac{1}{t}(x_i - \tau_2), & \textit{if } x_i > \tau_2; \\ 0, & \textit{if } \tau_1 \leq x_i \leq \tau_2; \\ \frac{1}{t}(x_i - \tau_1), & \textit{otherwise} \end{cases} \tag{9}$$

is defined for any $i = 1, \cdots, n$, such that

$$\sum_{i=1}^{n}(x_i - \tau_2)^+ = \sum_{i=1}^{n}(\tau_1 - x_i)^+ = t, \tag{10}$$

if the solution τ_1, τ_2 to Equation (10) satisfy $\tau_1 \leq \tau_2$, otherwise $\boldsymbol{g} = \nabla W_e^t(\boldsymbol{x})$ is determined by the average coordinate: $g_i = \frac{1}{t}x_i - \frac{1}{tn}\sum_{k=1}^{n} x_k$ for any index $i = 1, \cdots, n$.

This corollary can be directly verified by the Moreau envelope gradient $\nabla W_e^t(\boldsymbol{x}) = \frac{1}{t}(\boldsymbol{x} - \text{prox}_{tW_e^t}(\boldsymbol{x}))$ according to Equation (5), where the proximal mapping $\text{prox}_{tW_e^t}(\boldsymbol{x})$ is given in Theorem 1.

B. Water-filling Algorithm

Equation (7) and Equation (8) give an explicit representation of the proximal mapping, where the values of τ_1 and τ_2 are not represented in closed-form. Therefore, we must solve equations in Equation (8) first to obtain the exact value of $\text{prox}_{tW_e}(\boldsymbol{x})$.

We take τ_1 as an example, *i.e.*, we are going to solve the equation $\sum_{i=1}^{n}(\tau_1 - x_i)^+ = t$. Suppose we have a 4-pin net with pin coordinate x_1, x_2, x_3, x_4, illustrated in Fig. 2(a). We use a bar graph to illustrate the distribution, with a symbolic horizontal axis. Assume that each bar has width 1 and height x_i for $i = 1, 2, 3, 4$, then we are going to find a value τ_1 such that the blue region has a total area t.

The process for obtaining the exact value of τ_1, similar to that in the communication engineering, is called *water-filling*. One can simply imagine that we have an amount t of water in total, and want to pour it into a reservoir with an uneven bottom [29]. The final level τ_1 the water rise to is the target we desire.

To solve the water-filling, we first sort the pin coordinates. The statistics of sorted pin coordinates is illustrated in Fig. 2(b). On the surface, there seems to be no difference compared to Fig. 2(a). However, once we have the sorted value of pin coordinates, or sorted indices, the process can be solved in $O(n)$ time where n stands for the number of pins connected by this net. The detailed algorithm to solve the equation $\sum_{i=1}^{n}(\tau_1 - x_i)^+ = t$ is described in Algorithm 2.

The process of water-filling in Algorithm 2 is very intuitive. Assume that we have sorted the pin coordinates such that $x_1 \leq x_2 \leq \cdots \leq x_n$. Since we have n pins in this net, we have $n-1$ bottoms $x_1, \cdots, x_{n-1}$

Algorithm 2 WATER-FILLING ALGORITHM

```
Require: The sorted coordinates $\boldsymbol{x} \in \mathbb{R}^n$ such that $x_1 \leq x_2 \leq \cdots x_n$.
Initialize water trial $q = 0$, the bottom index $k = 1$;
while $k < n$ do
    Calculate the amount of water we need to fill in one bottom $q' \leftarrow x_{k+1} - x_k$;
    Accumulate water trial $q \leftarrow q + kq'$ because we need to fill in $k$ bottoms;
    if $q > t$ then
        Break the loop;
    end if
    Proceed to the next bottom $k \leftarrow k + 1$;
end while
if $q < t$ then
    Calculate $\tau_1 \leftarrow x_n + \frac{1}{n}(t - q)$;
else
    Calculate $\tau_1 \leftarrow x_{k+1} - \frac{1}{k}(q - t)$;
end if
return the required level value $\tau_1$;
```

to fill in before we make the bottoms of this "reservoir" even. Therefore, we create a trial and fill in each bottom one by one to check whether the current total trial is larger than the target amount t of water. More specifically, we will find k such that $x_k \leq \tau_1 < x_{k+1}$ in Algorithm 2.

Another interpretation of this algorithm is the *Abel transformation* for discrete sequences [33]:

$$\sum_{i=1}^{k}(\tau_1 - x_i) = k(\tau_1 - x_k) + \sum_{i=1}^{k-1} i(x_{i+1} - x_i), \tag{11}$$

where k is taken such that $x_k \leq \tau_1 < x_{k+1}$, or $k = n$ if $\tau_1 \geq x_n$. Clearly, we have $k(\tau_1 - x_k) \leq k(x_{k+1} - x_k)$ if $\tau_1 < x_n$, indicating

$$\sum_{i=1}^{k-1} i(x_{i+1} - x_i) \leq \sum_{i=1}^{n}(\tau_1 - x_i)^+ < \sum_{i=1}^{k} i(x_{i+1} - x_i), \tag{12}$$

when $\boldsymbol{x}$ are sorted. Therefore, given $t > 0$, we are supposed to find the index k such that either

$$\sum_{i=1}^{k-1} i(x_{i+1} - x_i) \leq t < \sum_{i=1}^{k} i(x_{i+1} - x_i) \tag{13}$$

or $t \geq \sum_{i=1}^{n-1} i(x_{i+1} - x_i)$ is satisfied ($k = n$ for this case). After sorting the pin coordinates, Equation (13) can be solved in $O(n)$ time with a single traversal. The part of solving τ_2 given t is similar.

Generally speaking, Algorithm 2 can be super fast as it visits n pins at most. The bottleneck is the sorting before each water-filling. Since we may have millions of pins in a design, an efficient sorting algorithm is critical.

C. Nonlinear Optimization

The update schemes of the precision γ in WA [16] and t in ours are also essential to the solution quality. The ePlace [18] algorithm and related placers [20] use a form of $\gamma(\phi) = \gamma_0 \left(w_{\text{bin}}^{(x)} + w_{\text{bin}}^{(y)}\right) \cdot 10^{k\phi + b}$ to update the parameter γ according to overflow ϕ. Higher overflow requires a large γ to sacrifice precision for better differentiability. To better adapt to the proposed wirelength model, we use the following tangent-based update scheme

$$t(\phi) = \frac{t_0}{2}\left(w_{\text{bin}}^{(x)} + w_{\text{bin}}^{(y)}\right) \tan\left(\frac{\pi}{2}\phi - \delta\right), \tag{14}$$

where t_0 is the initial value, ϕ is the density overflow, $w_{\text{bin}}^{(x)}$ and $w_{\text{bin}}^{(y)}$ stand for the horizontal and vertical bin size, respectively, and δ is a small positive number to avoid numerical overflow. For example, a configuration that $\delta = 10^{-4}$ and $t_0 = 4$ will normally give a good result for most cases.

We follow a moderate scheme similar to DREAMPlace3.0 [34] and elfPlace [35] to update density weight λ in Equation (1) iteratively:

$$\begin{aligned} \lambda_{k+1} &= \lambda_k + \alpha_k, \\ \alpha_k &= \left(\alpha_{\text{H}} - \frac{\alpha_{\text{H}} - \alpha_{\text{L}}}{1 + \ln(1 + \frac{\beta D_k}{D_0})}\right)\alpha_{k-1}, \end{aligned} \tag{15}$$

where D_k is the density at the k-th iteration. The parameters $\alpha_{\text{H}} \geq \alpha_{\text{L}} > 1$ depict how fast the density weight increases. The parameter pair $(\alpha_{\text{L}}, \alpha_{\text{H}})$ is set to $(1.01, 1.02)$ by default. The hyper-parameter β is set to 2000 in our experiments. Note that we do not use the quadratic penalty in our formulation, so β is simply a tunable parameter to adjust the step size α_k of density weight without any physical meaning. Here α_0 is $(\alpha_{\text{L}} - 1)\lambda_0$, and λ_0 is determined according to ePlace [18].

IV. WIRELENGTH MODEL ANALYSIS

The HPWL model has specific properties. By controlling the precision, the approximation models should also preserve similar properties.

A. Approximation Bound

Theorem 2. *Assume that there are $n_{\max} \geq 1$ pins sharing the maximum coordinates, and $n_{\min} \geq 1$ pins sharing the minimum coordinates. Then, we have*

$$-\frac{t}{2}\left(\frac{1}{n_{\max}} + \frac{1}{n_{\min}}\right) \leq W_e^t(\boldsymbol{x}) - W_e(\boldsymbol{x}) \leq 0, \tag{16}$$

for any positive parameter $t > 0$.

Theorem 2 gives a lower bound that describes how well the Moreau envelope function generally approximates the original one. Similar to LSE [15] and WA [16], the Moreau envelope we use here can also approximate HPWL in arbitrary precision. The proof is complicated and out of scope, and thus not attached due to page limit.

B. Gradient Properties

Ideally, when the smoothing parameter tends to zero, the gradient of any differentiable model should tend to a subgradient of W_e.

Theorem 3. *Consider the weighted-average model $W_{e,\mathit{WA}}^{\gamma}(\boldsymbol{x})$ defined by Equation (3) of a net with $x_{\max} > x_{\min}$. Assume that there are $n_{\max}$ pins sharing the maximum coordinates, and $n_{\min}$ pins sharing the minimum. Then, the gradient limit $\boldsymbol{g} = \lim_{\gamma \to 0^+} \nabla W_{e,\mathit{WA}}^{\gamma}$ exists and it is determined by*

$$g_i = \begin{cases} \frac{1}{n_{\max}}, & \textit{if } x_i = x_{\max}; \\ 0, & \textit{if } x_{\min} < x_i < x_{\max}; \\ -\frac{1}{n_{\min}}, & \textit{if } x_i = x_{\min}. \end{cases} \tag{17}$$

Besides, $g(\boldsymbol{x}) \in \partial W_e(\boldsymbol{x})$ is a subgradient of W_e.

Theorem 4. *Consider the Moreau envelope $W_e^t(\boldsymbol{x})$. Then, we always have $\nabla W_e^t = \boldsymbol{g}$ when t is small enough, where $\boldsymbol{g}$ is determined by Equation (17).*

If we only consider the maximum part, the subgradients of $\max(\boldsymbol{x})$ form a convex hull: $\operatorname{conv}\{\boldsymbol{e}_i : x_i = x_{\max}\}$, where $\boldsymbol{e}_i$ is the unit vector with the i-th entry being one. The components of any subgradient should sum to 1. The differentiable approximation models should also satisfy such a property.

Theorem 5. *The smooth maximum in the weighted-average model has a gradient whose components sum to 1.*

TABLE I The Statistics of the ISPD2006 [30] and ISPD2019 [31] Contest Benchmark Suites.

Benchmark		#Movable	#Fixed	#Nets	#Pins
ISPD2006 [30]	adaptec5	842482	646	867798	3433359
	newblue1	330137	337	338901	1223165
	newblue2	440239	1277	465219	1761069
	newblue3	482833	11178	552199	1881267
	newblue4	642717	3422	637051	2455617
	newblue5	1228177	4881	1284251	4849194
	newblue6	1248150	6889	1288443	5200208
	newblue7	2481372	26582	2636820	9971913
ISPD2019 [31]	ispd_test1	8879	0	3153	17203
	ispd_test2	72090	4	72410	318245
	ispd_test3	8208	75	8953	30271
	ispd_test4	146435	7	151612	436707
	ispd_test5	28914	8	29416	80757
	ispd_test6	179865	16	179863	793289
	ispd_test7	359730	16	358720	1584844
	ispd_test8	539595	16	537577	2376399
	ispd_test9	899325	16	895253	3957481
	ispd_test10	899325	79	895253	3957499

Corollary 2. *The weighted-average wirelength $W^{\gamma}_{e,WA}(\boldsymbol{x})$ defined by Equation (3) has a gradient whose components sum to 0.*

Theorem 6. *Let $\boldsymbol{g}$ be gradient of the Moreau envelop $\nabla W^t_e(\boldsymbol{x})$, then we have $\sum_{i:x_i \geq \tau_2} g_i = 1$ and $\sum_{i:x_i \leq \tau_1} g_i = -1$.*

Corollary 3. *The Moreau envelope model $W^t_e(\boldsymbol{x})$ has a gradient whose components sum to 0.*

Corollary 3 indicates that $W^t_e(\boldsymbol{x})$ preserves the gradient properties like Corollary 2. No matter what wirelength model is adopted, the total gradients w.r.t. all pins should sum to zero.

V. Experimental Results

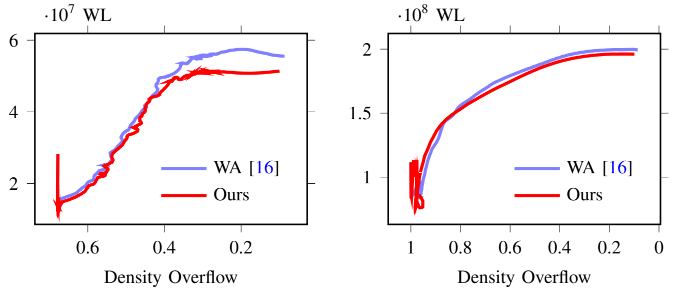
We use benchmarks from the ISPD2006 [30] and ISPD2019 [31] contests. The circuit statistics are shown in TABLE I. Compared to the ISPD2006 contest [30] targeting at wirelength only, recent contest benchmarks may focus more on other objectives like routability, timing, and region-constrained.

We implemented our model in C++/CUDA based on the open-source analytical placer DREAMPlace [20]. The following experiments are conducted on a 64-bit Linux workstation with Intel Xeon 2.90GHz CPUs and an NVIDIA GeForce RTX 3090 GPU.

We compare wirelength and runtime results using BiG_CHKS [21], LSE [15], and WA [16] on ISPD2006 benchmarks [30] in TABLE II and on ISPD2019 benchmarks [31] in TABLE III.

In our experiments, we use the ePlace [18] algorithm to perform analytical placement and DREAMPlace [20] as the placer because it has enabled high-performance GPU-accelerated techniques to obtain high-quality results extremely fast. For a fair comparison on wirelength optimization, we follow the settings of [21] on ISPD2006 benchmarks [30]. Considering that the reported wirelength and runtime results of BiG_CHKS and BiG_WA in [21] are roughly equal, we re-implement the BiG model proposed in [21] with CHKS bivariate function [36] in DREAMPlace [20]. The CHKS function [36] is more representative of bivariate functions.

After global placement, legalization [37], and detailed placement, we evaluate the results and list them in TABLE II. We incorporate ABCDPlace [38] as our detailed placement engine to fully leverage the GPU resources. Since the original binary of BiG in [21] is unavailable, as notified by the author, so we cite the performance directly from [21], listed in the column named "BiG_CHKS Reported in [21]" in TABLE II. There exists a quality gap between their reported results and ours, so we also list the results of executing the binary of NTUPlace3 [8] on our machine for reference. The wirelength results in TABLE II are evaluated by NTUPlace3 [8] for a fair comparison. TABLE II shows that we can achieve more than 1% improvements on ISPD2006 [30] after detailed placement. It is worth mentioning that the maximum improvement is over **5%** on newblue1 which has large movable macros. TABLE II indicates that the wirelength improvements are preserved after detailed placement.



(a) ISPD2006 newblue1 (b) ISPD2019 ispd19_test10

Fig. 3 (a) The wirelength curve against density overflow during global placement for ISPD2006 newblue1. (b) The wirelength curve against density overflow during global placement for ISPD2019 ispd19_test10. The overflow decreases as the nodes spread out.

We also test our wirelength model on the recent ISPD2019 benchmarks [31]. Since NTUPlace3 [8] and NTUplace4dr [9] binary executable files currently do not support ISPD2019 [31], we only compare different models incorporated in DREAMPlace [20] in TABLE III. As shown in the table, the achieved improvement is more than 1.5% and almost 2% on the recent ISPD2019 benchmarks [31].

To visualize the trend of wirelength during global placement, we take newblue1 in ISPD2006 [30] and ispd19_test10 in ISPD2019 [31] as examples and plot the curves in Fig. 3. We can achieve approximately 5.4% and 1.6% improvement, respectively, compared to WA [16] after detailed placement.

From Fig. 3, we can observe that the wirelength result obtained with our model is better at the same density overflow during global placement on the adopted cases. The density overflow roughly reflects the overall cell overlap. Therefore, a smaller wirelength at the same density overflow implies better placement quality.

VI. Conclusion

In this paper, we propose a novel HPWL-based differentiable wirelength model. We have made theoretical and experimental comparisons of the widely-used WA model [16] and our Moreau envelope model. It has been shown that our model has the advantage of numerical stability and convexity, which is preferred in numerical optimization. Moreover, our model has a better approximation error bound. Experimental results show that our model can rapidly produce better placement solutions achieving up to 5.4% HPWL improvement and more than 1% improvement on average compared to the most widely-used nonlinear wirelength models with GPU acceleration. Since further improvement on HPWL is challenging, our future work shall focus on novel optimizers to generally improve the analytical placement quality.

References

[1] T. F. Chan, K. Sze, J. R. Shinnerl, and M. Xie, "MPL6: Enhanced multilevel mixed-size placement with congestion control," in *Modern Circuit Placement*. Springer, 2007, pp. 247–288.

[2] A. B. Kahng and Q. Wang, "A faster implementation of APlace," in *Proc. ISPD*, 2006.

[3] M.-C. Kim, D.-J. Lee, and I. L. Markov, "SimPL: An effective placement algorithm," *IEEE TCAD*, 2011.

TABLE II The HPWL and runtime comparison on the ISPD2006 contest benchmarks [30]. **LGWL** (10^6) indicates the HPWL results after legalization. **DPWL** (10^6) indicates the HPWL results after detailed placement. **RT** (s) indicates the total runtime results including global placement, legalization, and detailed placement.

Benchmark	NTUPlace3 [8]			BiG_CHKS Reported in [21]			DREAMPlace [20] + BiG_CHKS [21]			DREAMPlace [20] + LSE [15]			DREAMPlace [20] + WA [16]			Ours		
	LGWL	DPWL	RT	LGWL	DPWL	RT	LGWL	DPWL	RT	LGWL	DPWL	RT	LGWL	DPWL	RT	LGWL	DPWL	RT
adaptec5	360.62	344.57	1249	281.23	270.82	1830	302.00	297.55	56.19	302.51	297.87	43.77	304.38	299.28	46.05	302.25	297.26	76.45
newblue1	62.770	60.152	319	88.350	84.150	642	61.516	60.328	35.33	61.710	60.548	29.08	60.354	58.784	30.08	57.047	55.590	56.17
newblue2	205.65	191.66	647	185.98	174.73	696	180.23	176.96	32.64	180.94	177.77	25.42	181.07	177.57	25.92	180.07	176.73	53.18
newblue3	290.59	275.91	583	278.43	258.71	1351	262.59	257.89	45.89	258.64	257.66	32.13	258.14	257.48	34.43	259.34	256.28	60.80
newblue4	260.08	247.59	916	219.10	212.20	1040	220.01	214.99	40.92	220.69	215.30	37.74	221.07	216.20	36.13	221.16	215.71	54.16
newblue5	447.30	426.91	2077	403.03	387.15	3290	388.39	381.19	81.14	390.02	382.37	64.86	389.26	382.05	65.27	387.77	380.18	118.13
newblue6	521.82	498.16	2048	433.55	417.42	2928	446.93	441.76	87.66	445.61	440.63	69.53	446.85	441.14	68.07	445.33	439.83	118.55
newblue7	1160.1	1100.2	4119	990.40	957.06	7289	944.51	931.32	148.53	942.77	929.64	131.83	945.07	931.60	128.36	941.34	926.99	193.13
Avg. Ratio	1.161	1.124	14.61	1.080	1.051	22.52	1.012	1.013	0.71	1.011	1.014	0.58	1.010	1.011	0.58	1.000	1.000	1.00

TABLE III The HPWL and runtime comparison after legalization and detailed placement among the DREAMPlace [20] + LSE [15], DREAMPlace [20] + WA [16], DREAMPlace [20] + BiG_CHKS [21], and ours on the ISPD2019 contest benchmarks [31]. **LGWL** (10^6) indicates the HPWL results after legalization. **DPWL** (10^6) indicates the HPWL results after detailed placement. **RT** (s) indicates the total runtime results including global placement, legalization, and detailed placement.

Benchmark	DREAMPlace [20] + BiG_CHKS [21]			DREAMPlace [20] + LSE [15]			DREAMPlace [20] + WA [16]			Ours		
	LGWL	DPWL	RT	LGWL	DPWL	RT	LGWL	DPWL	RT	LGWL	DPWL	RT
ispd19_test1	0.41688	0.40778	7.06	0.41341	0.40561	7.09	0.41856	0.40887	5.33	0.41036	0.40225	14.32
ispd19_test2	18.211	18.027	11.74	17.858	17.705	10.96	18.047	17.890	10.83	17.500	17.345	15.97
ispd19_test3	0.75403	0.71541	7.24	0.73048	0.70335	7.60	0.76892	0.72112	7.17	0.74330	0.70771	14.42
ispd19_test4	17.182	17.021	15.23	17.153	16.983	13.37	17.189	17.027	12.78	17.087	16.903	20.68
ispd19_test5	3.5963	3.3161	11.83	3.6446	3.3097	14.30	3.5786	3.2727	14.77	3.5066	3.2281	15.78
ispd19_test6	45.497	45.097	15.28	45.402	44.987	14.27	45.138	44.755	14.68	44.707	44.351	23.33
ispd19_test7	89.457	88.373	25.79	90.731	89.496	23.52	90.105	88.831	23.87	88.526	87.561	37.74
ispd19_test8	135.21	134.06	32.99	134.60	133.57	29.92	135.05	133.89	30.43	133.04	131.89	45.66
ispd19_test9	212.99	208.71	47.53	213.40	209.51	45.01	210.98	207.16	43.79	208.83	205.14	66.53
ispd19_test10	211.68	209.46	47.78	211.21	209.04	45.81	211.99	209.73	44.46	208.48	206.32	64.59
Avg. Ratio	1.018	1.017	0.67	1.014	1.014	0.65	1.018	1.015	0.64	1.000	1.000	1.00

[4] N. Viswanathan, M. Pan, and C. Chu, “FastPlace 3.0: A fast multilevel quadratic placement algorithm with placement congestion control,” in *Proc. ASPDAC*, 2007.

[5] U. Brenner, M. Struzyna, and J. Vygen, “BonnPlace: Placement of leading-edge chips by advanced combinatorial algorithms,” *IEEE TCAD*, 2008.

[6] T. Luo and D. Z. Pan, “DPlace2.0: A stable and efficient analytical placement based on diffusion,” in *Proc. ASPDAC*, 2008.

[7] P. Spindler, U. Schlichtmann, and F. M. Johannes, “Kraftwerk2—A fast force-directed quadratic placement approach using an accurate net model,” *IEEE TCAD*, 2008.

[8] T.-C. Chen, Z.-W. Jiang, T.-C. Hsu, H.-C. Chen, and Y.-W. Chang, “NTUplace3: An analytical placer for large-scale mixed-size designs with preplaced blocks and density constraints,” *IEEE TCAD*, 2008.

[9] C.-C. Huang, H.-Y. Lee, B.-Q. Lin, S.-W. Yang, C.-H. Chang, S.-T. Chen, Y.-W. Chang, T.-C. Chen, and I. Bustany, “NTUplace4dr: A detailed-routing-driven placer for mixed-size circuit designs with technology and region constraints,” *IEEE TCAD*, 2017.

[10] M.-C. Kim and I. L. Markov, “ComPLx: A competitive primal-dual Lagrange optimization for global placement,” in *Proc. DAC*, 2012.

[11] M.-C. Kim, N. Viswanathan, C. J. Alpert, I. L. Markov, and S. Ramji, “MAPLE: Multilevel adaptive placement for mixed-size designs,” in *Proc. ISPD*, 2012.

[12] I. L. Markov, J. Hu, and M.-C. Kim, “Progress and challenges in VLSI placement research,” *Proceedings of the IEEE*, 2015.

[13] G. Sigl, K. Doll, and F. M. Johannes, “Analytical placement: A linear or a quadratic objective function?” in *Proc. DAC*, 1991.

[14] H. Eisenmann and F. M. Johannes, “Generic global placement and floorplanning,” in *Proc. DAC*, 1998.

[15] W. C. Naylor, R. Donelly, and L. Sha, “Non-linear optimization system and method for wire length and delay optimization for an automatic electric circuit placer,” Oct. 9 2001, US Patent 6,301,693.

[16] M.-K. Hsu, Y.-W. Chang, and V. Balabanov, “TSV-aware analytical placement for 3D IC designs,” in *Proc. DAC*, 2011.

[17] M.-K. Hsu, V. Balabanov, and Y.-W. Chang, “TSV-aware analytical placement for 3-D IC designs based on a novel weighted-average wirelength model,” *IEEE TCAD*, 2013.

[18] 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 TODAES*, 2015.

[19] C.-K. Cheng, A. B. Kahng, I. Kang, and L. Wang, “RePlAce: Advancing solution quality and routability validation in global placement,” *IEEE TCAD*, 2018.

[20] Y. Lin, S. Dhar, W. Li, H. Ren, B. Khailany, and D. Z. Pan, “DREAMPlace: Deep learning toolkit-enabled GPU acceleration for modern VLSI placement,” in *Proc. DAC*, 2019.

[21] F.-K. Sun and Y.-W. Chang, “BiG: A bivariate gradient-based wirelength model for analytical circuit placement,” in *Proc. DAC*, 2019.

[22] C. Li and C.-K. Koh, “Recursive function smoothing of half-perimeter wirelength for analytical placement,” in *Proc. ISQED*, 2007.

[23] W. Zhu, J. Chen, Z. Peng, and G. Fan, “Nonsmooth optimization method for VLSI global placement,” *IEEE TCAD*, 2015.

[24] E. Polak and G. Ribiere, “Note sur la convergence de méthodes de directions conjuguées,” *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 1969.

[25] J.-J. Moreau, “Proximité et dualité dans un espace hilbertien,” *Bulletin de la Société mathématique de France*, 1965.

[26] C. E. Shannon, “A mathematical theory of communication,” *Bell System Technical Journal*, 1948.

[27] A. D. Wyner, “The capacity of the band-limited gaussian channel,” *Bell System Technical Journal*, 1966.

[28] W. Yu and J. M. Cioffi, “On constant power water-filling,” in *Proc. ICC*, vol. 6, 2001.

[29] R. W. Yeung, *Information theory and network coding*. Springer Science & Business Media, 2008.

[30] G.-J. Nam, “ISPD 2006 placement contest: Benchmark suite and results,” in *Proc. ISPD*, 2006.

[31] W.-H. Liu, S. Mantik, W.-K. Chow, Y. Ding, A. Farshidi, and G. Posser, “ISPD 2019 initial detailed routing contest and benchmark with advanced routing rules,” in *Proc. ISPD*, 2019.

[32] S. Afriat, “Theory of maxima and the method of Lagrange,” *SIAM Journal on Applied Mathematics*, 1971.

[33] N. H. Abel, “Untersuchungen über die reihe: $1+\frac{m}{1}x+\frac{m\cdot(m-1)}{1\cdot 2}\cdot x^2+\frac{m\cdot(m-1)\cdot(m-2)}{1\cdot 2\cdot 3}\cdot x^3+\cdots$ u.s.w.” *J. Reine Angew. Math*, 1826.

[34] J. Gu, Z. Jiang, Y. Lin, and D. Z. Pan, “DREAMPlace 3.0: Multi-electrostatics based robust VLSI placement with region constraints,” in *Proc. ICCAD*, 2020.

[35] W. Li, Y. Lin, and D. Z. Pan, “elfPlace: Electrostatics-based placement for large-scale heterogeneous FPGAs,” in *Proc. ICCAD*, 2019.

[36] B. Chen and P. T. Harker, “A non-interior-point continuation method for linear complementarity problems,” *SIAM Journal on Matrix Analysis and Applications*, 1993.

[37] P. Spindler, U. Schlichtmann, and F. M. Johannes, “Abacus: Fast legalization of standard cell circuits with minimal movement,” in *Proc. ISPD*, 2008.

[38] Y. Lin, W. Li, J. Gu, H. Ren, B. Khailany, and D. Z. Pan, “ABCDPlace: Accelerated batch-based concurrent detailed placement on multithreaded cpus and GPUs,” *IEEE TCAD*, 2020.