

Invited Paper

## Essential Issues in Analytical Placement Algorithms

YAO-WEN CHANG,<sup>†1,†2</sup> ZHE-WEI JIANG<sup>†2</sup>  
and TUNG-CHIEH CHEN<sup>†3</sup>

The placement problem is to place objects into a fixed die such that no objects overlap with each other and some cost metric (e.g., wirelength) is optimized. Placement is a major step in physical design that has been studied for several decades. Although it is a classical problem, many modern design challenges have reshaped this problem. As a result, the placement problem has attracted much attention recently, and many new algorithms have been developed to handle the emerging design challenges. Modern placement algorithms can be classified into three major categories: simulated annealing, min-cut, and analytical algorithms. According to the recent literature, analytical algorithms typically achieve the best placement quality for large-scale circuit designs. In this paper, therefore, we shall give a systematic and comprehensive survey on the essential issues in analytical placement. This survey starts by dissecting the basic structure of analytical placement. Then, various techniques applied as components of popular analytical placers are studied, and two leading placers are exemplified to show the composition of these techniques into a complete placer. Finally, we point out some research directions for future analytical placement.

## 1. Introduction

The placement problem is to place objects into a fixed die such that no objects overlap with each other and some cost metric (e.g., wirelength) is optimized. (See Fig. 1 for an illustration.) Placement is a major step in physical design that has been studied for several decades. Although it is a classical problem, many modern design challenges have reshaped this problem. The modern placement problem becomes very tough because we need to handle large-scale designs with billions of transistors (or millions of objects/standard cells). Meanwhile, intellectual property (IP) modules and pre-designed macro blocks (such as embedded memories,

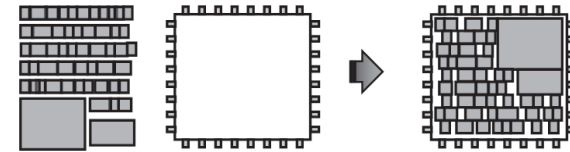


Fig. 1 The placement process.

analog blocks, pre-designed datapaths, etc.) are often reused, making the placement objects very different in their sizes. In addition to wirelength, we also need to consider many other placement constraints such as chip density, routability, timing, etc. As a result, the placement problem has attracted much attention recently, and many new algorithms have been developed to handle the emerging design challenges. To stimulate the placement research, *the ACM International Symposium on Physical Design (ISPD)* even held two placement contests<sup>1),2)</sup> in 2005 and 2006; these contests have successfully driven the placement research forward.

Modern placement algorithms can be classified into three major types: simulated annealing, min-cut, and analytical algorithms. Table 1 summarizes their strengths and weaknesses.

- **Simulated Annealing Based Placement.** This type of placers tries to optimize a placement by perturbing module positions based on simulated annealing. They can thus consider different optimization objectives with little modification due to the generality of simulated annealing. Good placement quality can often be achieved on small designs due to the search on a small solution space. However, the module perturbation may not be trivial with the existence of big macros, and the lack of scalability makes this type of placers not applicable on large-scale circuits. Representative placers of this type are Dragon<sup>54)</sup> and TimberWolf<sup>46)</sup>.
- **Min-Cut Placement.** The min-cut placement recursively partitions the circuit and chip region, and then assign sub-circuits into sub-regions in a top-down fashion. Because of the maturity of the partitioning algorithms, the min-cut placement is usually very efficient and scalable. Besides, since the sub-region of each module is clearly defined during the placement process, the legalization of big macros can be handled pretty well. Nevertheless, the

---

<sup>†1</sup> Department of Electrical Engineering, National Taiwan University

<sup>†2</sup> Graduate Institute of Electronics Engineering, National Taiwan University

<sup>†3</sup> SpringSoft Inc.

**Table 1** Strengths (+) and weaknesses (–) for the three types of placers.

<b>Simulated Annealing Placement</b>
(+) Easier to consider multiple objectives simultaneously
(+) Good quality for small designs
(–) Harder to handle modules of very different sizes
(–) Slower and less scalable for large circuits
<b>Min-Cut Placement</b>
(+) More efficient and scalable, even for large circuits
(+) Good at mixed-size circuit legalization
(–) Harder to handle multiple objectives simultaneously
(–) Harder for whitespace management, especially for designs with low utilization rates
<b>Analytical Placement</b>
(+) More efficient and scalable, even for large circuits
(+) Better quality for large-scale designs
(+) Good at whitespace management, regardless of utilization rates
(+) Easier to handle multiple objectives simultaneously
(–) Harder to legalize large macros
(–) Harder to optimize macro orientations

min-cut partitioning tries to minimize the expected wirelength between subregions by minimizing the number of cuts between sub-circuits, and thus the applicable optimization objectives are limited. It is also harder for placers of this type to handle whitespace in the earlier levels of the top-down process, especially for designs with low utilization rates. Further, the hierarchical approach of solving each subproblem independently might lack the global information for the interaction among different subregions, thus limiting the solution quality. Example min-cut placers are Capo<sup>3)</sup>, FengShui<sup>5)</sup>, and NTUplace<sup>16)</sup>.

- **Analytical Placement.** The analytical placement formulates the placement problem as mathematical programming composed of an objective function and a set of placement constraints, and then optimizes the objective through analytical approaches. It has been shown in the recent literature and the ISPD placement contests that the analytical placement can achieve better placement quality for large-scale circuit designs. In particular, it can consistently achieve high solution quality regardless of different utilization rates. It is also relatively easier to handle multiple objectives simultaneously

than min-cut placement. However, it is harder to optimize macro orientations and legalize big macros, due to the intrinsic limitation of mathematical programming. There are a large number of academic analytical placers, such as Refs. 4), 8), 12), 18), 23), 33), 35), 39), 40), 51), 58)–60), 65).

With the large number of newly developed analytical placers, the reader might be dazzled by the wide variety of those approaches. In this paper, therefore, we shall give a systematic and comprehensive survey on the essential issues in analytical placement. This survey starts by dissecting the basic structure of analytical placement. A modern analytical placement algorithm typically consists of three major stages: global placement, legalization, and detailed placement. We discuss the analytical global placement techniques based on the following four key ingredients: (1) wirelength models, (2) overlap reduction techniques, (3) integration of wirelength models and overlap reduction techniques, and (4) optimization techniques. We then summarize commonly used legalization and detailed placement techniques. Two leading academic placers, NTUplace3<sup>18)</sup> and Kraftwerk2<sup>51)</sup>, are exemplified to show the composition of these techniques into a complete placer. Finally, we point out some research directions for future analytical placement.

The rest of this paper is organized as follows. Section 2 introduces the basic structure of the analytical placement. Sections 3, 4, and 5 survey the techniques applied to global placement, legalization, and detailed placement, respectively. In Section 6, NTUplace3 and Kraftwerk2 are exemplified to show how these techniques can be assembled into a complete placer. Finally, future research directions are discussed in Section 7, and conclusions are given in Section 8.

## 2. Analytical Placement Basics

As mentioned earlier, placement is the process of determining the locations of circuit devices on a fixed die such that no devices overlap with each other and some cost metric (e.g., wirelength) is optimized. Since placement has been proven to be computationally difficult, one way to manage the complexity of placement is to divide it into several easier steps. Most modern analytical placers consist of the following three major steps:

- (1) **Global placement.** Ignoring some placement constraints (e.g., module

overlaps), global placement computes the best position for each module to minimize the predefined cost (e.g., wirelength). Global placement is generally considered the most important step, due to its crucial impact on the overall placement quality.

- (2) **Legalization.** Legalization removes all overlaps among modules.
- (3) **Detailed placement.** Detailed placement further improves the legalized placement solution, typically in an iterative manner by rearranging a small group of modules in a local region while keeping all other modules fixed.

We detail the three steps in the subsequent sections.

### 3. Global Placement

For analytical global placement, a circuit can be modelled by a hypergraph  $H = (V, E)$ . Let vertices  $V = \{v_1, v_2, \dots, v_n\}$  represent cells, and hyperedges  $E = \{e_1, e_2, \dots, e_m\}$  represent nets. Let  $x_i$  and  $y_i$  be the  $x$  and  $y$  coordinates of the center of cell  $v_i$ , respectively. The typical objective of global placement is to minimize its wirelength, and a fundamental constraint is to avoid any cell overlap. The wirelength objective is highly related to the chip performance while the non-overlapping constraint makes the resulting layout manufacturable. Consequently, the global placement problem can be formulated as a constrained minimization problem as follows:

$$\begin{aligned} \min \quad & W(V, E) \\ \text{s.t.} \quad & \text{no overlaps among cells,} \end{aligned} \quad (1)$$

where  $W(V, E)$  is the wirelength function. It can be seen that the minimization problem consists of two ingredients: one is the *wirelength model* for the wirelength estimation, and the other is the *overlap reduction technique* required to keep cells overlap-free. Therefore, we shall start our survey by introducing the wirelength models in Section 3.1, and the overlap reduction techniques in Section 3.2. An important feature distinguishing the analytical placers lies in the way they *unify the wirelength models and overlap reduction techniques*, which will be discussed in Section 3.3. Finally, the *optimization techniques* for the unified global placement problem will be discussed in Section 3.4.

#### 3.1 Wirelength Models

The wirelength of a net  $e \in E$  is usually defined by its total half-perimeter

wirelength (HPWL) as follows:

$$W(V, E) = \sum_e \left( \max_{v_i, v_j \in e} |x_i - x_j| + \max_{v_i, v_j \in e} |y_i - y_j| \right) \quad (2)$$

$$= \sum_e \left( \max_{v_i \in e} x_i - \min_{v_i \in e} x_i + \max_{v_i \in e} y_i - \min_{v_i \in e} y_i \right) \quad (3)$$

$$= \sum_e (L_{e,x} + L_{e,y}). \quad (4)$$

However, since HPWL is not differentiable (although convex), it is hard to find its minimum value. Consequently, it is necessary to use a continuous differentiable function (i.e., “wirelength model”) to approximate the HPWL. We describe popular smooth wirelength approximations (wirelength models) in the following subsections.

##### 3.1.1 Quadratic Model

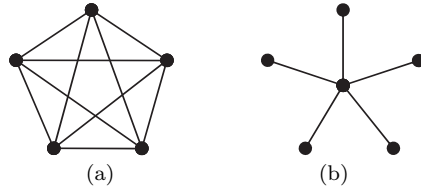
The sum of half of the quadratic Euclidean length of every two-pin connection gives the quadratic wirelength model. As a result, the total wirelength of the circuit can be represented as

$$\sum_{e \in E} \frac{1}{2} \left( \sum_{v_i, v_j \in e, i < j} w_{x,ij} (x_i - x_j)^2 + \sum_{v_i, v_j \in e, i < j} w_{y,ij} (y_i - y_j)^2 \right). \quad (5)$$

Here, half of the total weighted quadratic wirelength is often used to have a simpler derivative form. Since the quadratic model can only handle two-pin connections, multi-pin nets are often modelled by the clique net model or the star net model (see **Fig. 2**). The clique model considers all possible two-pin connections of a net, while the star net model introduces an additional star pin per net and connects each pin of the net to the star pin. With  $P$  representing the number of pins in net  $n$ , the clique model is equivalent to the star net model in the quadratic cost, if the clique cost is scaled with  $1/P^{57}$ . The quadratic cost of the clique net model is

$$L_{e,x} = \frac{1}{2} \sum_{i=1}^P \sum_{j=i+1}^P w_{x,ij} (x_i - x_j)^2. \quad (6)$$

The net weight  $w_{x,ij}$  ( $w_{y,ij}$ ) is used to adjust the quadratic objective to approximate the linear objective (HPWL); for example, Gordian-L<sup>47</sup>) uses the following



**Fig. 2** Two models for a five-pin net. (a) The clique model. (b) The star model.

formula to determine the  $x$ -component weight for the approximation:

$$w_{x,ij}^{GordianL} = \frac{1}{P} \frac{2}{P} \frac{4}{|x_i - x_j|}. \quad (7)$$

The first term  $1/P$  adjusts the clique model to the star net model. The second term  $2/P$  adjusts the number of connections of the clique to the number of connections in the corresponding spanning tree. The third term  $1/|x_i - x_j|$  linearizes the quadratic distance between two pins.  $w_{y,ij}$  is also defined similarly.

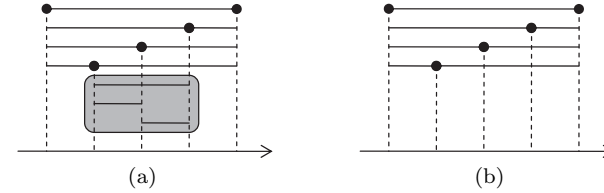
### 3.1.2 Bound2Bound Model

No matter what  $w_{x,ij}$  we choose, the clique net model has a high approximation error between the total length of the clique net and the HPWL. The problem of the clique model is that its inner connections contribute to the clique length but are ignored in the HPWL, since HPWL considers only the distance between the boundary pins. **Figure 3** (a) illustrates the problem with the clique model. In this figure, the boundary pins are those with the smallest/largest coordinates, and other pins are inner pins. There are three connections only connecting to inner pins, but these connections are ignored in the HPWL metric. (The star net model also has the similar situation.)

The Bound2Bound net model removes all inner two-pin connections, as shown in Fig. 3 (b). The net weight  $w_{x,ij}^{B2B}$  of the Bound2Bound net model is defined as follows:

$$w_{x,ij}^{B2B} = \begin{cases} 0, & v_i, v_j \in \text{inner pins} \\ \frac{2}{P-1} \frac{1}{|x_i - x_j|}, & \text{else.} \end{cases} \quad (8)$$

With this connection weight, the quadratic wirelength function in Eq. (5) exactly matches the HPWL<sup>51)</sup>:



**Fig. 3** The clique net model and the Bound2Bound net model<sup>51)</sup>. (a) The clique net model. There are two boundary pins and three inner pins. Three inner pin connections are marked in the shaded region. (b) The Bound2Bound net model. There is no inner pin connections. All nets are connected to boundary pins.

$$L_{e,x} = \frac{1}{2} \sum_{i=1}^P \sum_{j=i+1}^P w_{x,ij}^{B2B} (x_i - x_j)^2 \quad (9)$$

$$= \max_{v_i \in e} x_i - \min_{v_i \in e} x_i. \quad (10)$$

### 3.1.3 LSE Model

To accurately approximate and to smooth the HPWL, logarithm-sum-exponential (LSE) approximation of the max/min function is prevailing in recent placers, such as APlace<sup>32)</sup>, mPL6<sup>12)</sup>, and NTUplace3<sup>18)</sup>. The HPWL of a net  $e \in E$  can be approximated by using LSE as follows:

$$LSE_e = \gamma \left( \log \sum_{v_k \in e} e^{\frac{x_k}{\gamma}} + \log \sum_{v_k \in e} e^{\frac{-x_k}{\gamma}} + \log \sum_{v_k \in e} e^{\frac{y_k}{\gamma}} + \log \sum_{v_k \in e} e^{\frac{-y_k}{\gamma}} \right). \quad (11)$$

When  $\gamma$  approaches zero, the LSE wirelength is close to the HPWL<sup>41)</sup>.

$$\lim_{\gamma \rightarrow 0} LSE_e = HPWL_e \quad (12)$$

However, due to the computer precision, we can only choose a reasonably small  $\gamma$  to avoid any arithmetic overflow during the implementation. In particular,  $LSE_e$  is differentiable, and thus it serves as a good approximation to  $HPWL_e$ , in terms of precision as well as computation.

### 3.1.4 Lp-norm Model

Another good smoothing method of the HPWL is the Lp-norm approximation:

$$Lpnorm_e = \left( \sum_{v_k \in e} x_k^p \right)^{\frac{1}{p}} - \left( \sum_{v_k \in e} x_k^{-p} \right)^{-\frac{1}{p}} + \left( \sum_{v_k \in e} y_k^p \right)^{\frac{1}{p}} - \left( \sum_{v_k \in e} y_k^{-p} \right)^{-\frac{1}{p}}. \quad (13)$$

The use of the parameter  $p$  here is similar to  $\gamma$  for the LSE model. When  $p$  is large, the Lp-norm model gives a very good approximation to the HPWL.

$$\lim_{p \rightarrow \infty} Lpnorm_e = HPWL_e \quad (14)$$

Due to the computer precision, similarly, we can only choose a reasonably large  $p$  to prevent any arithmetic overflow during the implementation. The authors in Ref. 11) compared the LSE and Lp-norm models and concluded that the LSE model usually outperforms the Lp-norm one in terms of HPWL.

### 3.1.5 CHKS Model

Different from the log-sum-exp and Lp-norm wirelength models, an alternative way is to smooth the two-variable max function first, and then the multi-variable max/min function can be computed by the two-variable max function. As mentioned in Ref. 37), the max function has the following properties:

- (1) A multi-variable max function can be obtained by the recursive call of two-variable max functions as follows:

$$\max\{\mathbf{x}\} = \max\{\max\{\mathbf{x}^{(1)}\}, \max\{\mathbf{x}^{(2)}\}\}, \quad (15)$$

where  $\mathbf{x}^{(1)}$  and  $\mathbf{x}^{(2)}$  are two disjoint partitions of  $\mathbf{x}$ .

- (2)  $\min\{\mathbf{x}\}$  can be obtained from the max approximation by

$$\min\{\mathbf{x}\} = -\max\{-\mathbf{x}\}. \quad (16)$$

The CHKS function was proposed to smooth the two-variable max function<sup>13),34),48)</sup>:

$$CHKS(x_1, x_2) = \frac{\sqrt{(x_1 - x_2)^2 + t^2} + x_1 + x_2}{2}, \quad (17)$$

with the smoothing parameter  $t > 0$ . The following definition shows how to construct the smoothed multi-variable max function from two-variable CHKS functions. (Let  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  correspond to the two-variable CHKS function.)

**Definition 1** Define the function  $f_{i,i+1} : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $\forall 1 \leq i \leq n-1$ , by

$$f_{i,i+1}(\mathbf{x}) = f(x_i, x_{i+1}) = CHKS(x_i, x_{i+1}), \quad (18)$$

and the function  $f_{i,i} : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $\forall 1 \leq i \leq n$ , by

$$f_{i,i}(\mathbf{x}) = x_i. \quad (19)$$

Moreover, for  $1 \leq i \leq j \leq n$  and  $j - i + 1 > 2$ , let function  $f_{i,j} : \mathbb{R}^n \rightarrow \mathbb{R}$  be

$$f_{i,j}(\mathbf{x}) = f(f_{i,k}(\mathbf{x}), f_{k+1,j}(\mathbf{x})), \quad (20)$$

where  $k = \lfloor \frac{i+j}{2} \rfloor$ .

Therefore, the multi-variable max function can be smoothed and approximated by  $f_{1,n}(\mathbf{x})$  as defined above, and then the multi-variable min function can be determined through Eq. (16). Smoothing of the HPWL in Eq. (3) can therefore be obtained accordingly. Further, the gradient of the function value  $f_{i,j}(\mathbf{x})$  can be obtained by using the chain rule:

$$\frac{\partial f_{i,j}(\mathbf{x})}{\partial x_l} = \frac{\partial f_{i,j}(\mathbf{x})}{\partial f_{i,k}(\mathbf{x})} \frac{\partial f_{i,k}(\mathbf{x})}{\partial x_l} + \frac{\partial f_{i,j}(\mathbf{x})}{\partial f_{k+1,j}(\mathbf{x})} \frac{\partial f_{k+1,j}(\mathbf{x})}{\partial x_l}, \quad (21)$$

where  $k = \lfloor \frac{i+j}{2} \rfloor$ , and  $i \leq l \leq j$ . If  $i \leq l \leq k$ , and the CHKS smoothing function is used, we have

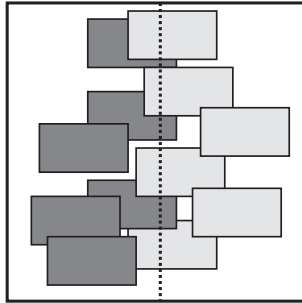
$$\frac{\partial f_{i,j}(\mathbf{x})}{\partial x_l} = \left( \frac{1}{2} + \frac{f_{i,k}(\mathbf{x}) - f_{k+1,j}(\mathbf{x})}{2\sqrt{(f_{i,k}(\mathbf{x}) - f_{k+1,j}(\mathbf{x}))^2 + t^2}} \right) \frac{\partial f_{i,k}(\mathbf{x})}{\partial x_l}. \quad (22)$$

## 3.2 Overlap Reduction Techniques

The second key ingredient for analytical placement is how to reduce overlaps among cells to obtain an evenly distributed placement. Many overlap reduction techniques have been proposed in the literature for analytical placement. These techniques can be classified into six categories: (1) partitioning, (2) cell shifting, (3) assignment, (4) diffusion, (5) density control, and (6) frequency control. The underlying ideas of these techniques are discussed in the following subsections.

### 3.2.1 Partitioning

Partitioning is perhaps the earliest method to reduce overlaps among cells; it decomposes a complex circuit into smaller subcircuits and assigns those partitioned subcircuits to proper sub-regions. The movement of each cell is constrained accordingly within their assigned sub-regions in the later global optimization steps, and thus the amount of overlaps can be reduced. In analytical placement, partitioning-based overlap reduction often consists of two stages, the partitioning stage and the refinement stage. In the partitioning stage, with a given initial placement, the circuit is partitioned and assigned to sub-regions



**Fig. 4** Illustration of physical partitioning. Cells are partitioned into two sets according to their physical positions with respect to the vertical cutline (the dotted line).

while minimizing some cost metric, such as cell displacement. In the refinement stage, heuristics such as the Fiduccia and Mattheyses algorithm<sup>20)</sup> and the window-based repartitioning<sup>24),62)</sup>, can then be applied to further improve the partition quality. In the following, we explain these two popular approaches for partitioning.

For a given initial placement, one most intuitive way to divide the circuit is to order cells according to their physical positions. Such a partitioning manner is referred to as *physical partitioning*. The physical partitioning first decides the cutline position considering the chip boundary and the cell distribution. For a given set  $V$  of cells with known cell positions, considering a vertical cutline  $x = x_{cutline}$ , the cells are then partitioned into two subsets  $V_L$  and  $V_R$  such that

$$\begin{aligned} \forall v_i \in V_L, x_i \leq x_{cutline}, \text{ and} \\ \forall v_i \in V_R, x_i > x_{cutline}. \end{aligned}$$

**Figure 4** gives an example of physical partitioning.

Though the physical partitioning can provide a partitioning solution very quickly, the resulting partitions might not be able to fit the physical sub-regions since the physical partitioning does not consider the capacity of each sub-region, and the partition refinement is thus required to adjust the size of each partition. A transportation problem was formulated in Ref. 10) to overcome this difficulty. The underlying idea of the transportation formulation is to assign cells to sub-regions to minimize displacement such that the capacity constraint is satisfied. However, since the cells usually have different sizes, such an assignment problem

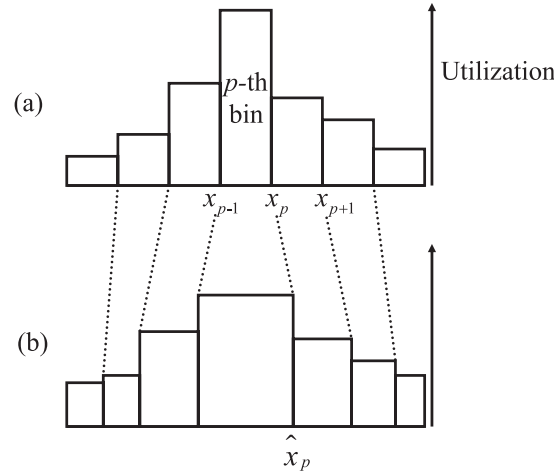
is NP-complete. As a result, Brenner and Struzyna<sup>10)</sup> proposed to relax the assignment problem by allowing to assign cells fractionally to the sub-regions. For a set of cells with given positions, let  $N$  denote the set of nodes that model the cells,  $n_i \in N$  stand for the node representing cell  $v_i$ , and  $R$  denote the set of nodes representing each sub-region. Let  $size(v_i)$  be the size for cell  $v_i$ , and for  $r \in R$ ,  $cap(r)$  be the capacity for sub-region  $r$ . Let  $cost(v_i, r)$  be the cost to move cell  $v_i$  from its initial position to sub-region  $r$ . Then the flow network for the fractional transportation problem can then be constructed as follows:

- (1) Construct the node set  $V_{flow} = N \cup R \cup \{s, t\}$ , where  $s$  and  $t$  are the respective source and sink of the network.
- (2) Construct the edge set  $E_{flow} = (N \times R) \cup (\{s\} \times N) \cup (R \times \{t\})$ . For each edge  $e \in E_{flow}$ ,  $u_e$  denotes the capacity of the edge and  $w_e$  denotes the weight of the edge.
- (3) For each edge  $e = (n_i, r) \in (N \times R)$ ,  $u_e$  is set to  $\infty$ , and  $w_e$  is set to  $cost(v_i, r)$ .
- (4) For each edge  $e = (s, n_i) \in (\{s\} \times N)$ ,  $u_e$  is set to  $size(v_i)$  and  $w_e$  is set to 0.
- (5) For each edge  $e = (r, t) \in (R \times \{t\})$ ,  $u_e$  is set to  $cap(r)$  and  $w_e$  is set to 0.
- (6) The supply of  $s$  is set to  $\sum_{n_i \in N} size(v_i)$  and the demand of  $t$  is set to  $-\sum_{n_i \in N} size(v_i)$ .

Solving the formulated fractional transportation problem can thus obtain the cell assignment to the sub-regions while satisfying the capacity constraints. Though the fractions of some cell might be assigned to different sub-regions, it is mentioned in Refs. 63) and 10) that such a fractional assignment can easily be converted to an integral one. Finally, the cell partitions and the assignment from partitions to sub-regions can both be obtained from the converted integral assignment.

### 3.2.2 Cell Shifting

Another possible method to reduce cell overlaps is to spread cells through *cell shifting*. Such a concept was first proposed by Viswanathan and Chu in Ref. 57). The basic idea is to distribute cells over the placement region with their relative order of an initial placement being retained. To achieve the target, the placement region is divided into equal-sized bins, and each bin accommodates a various number of cells. Then the cell shifting is applied along the  $x$  and  $y$  directions individually. For the case of applying cell shifting along the  $x$  direction, each row



**Fig. 5** (a) Boundaries and utilizations of the initial bin structure before cell shifting. (b) Boundaries and utilizations of the adjusted bin structure after cell shifting.

of the regular bin structure is processed. Cell shifting for each row is composed of two steps. First, the adjusted bin structure is constructed according to the current utilization of each bin in the processing row. Second, every cell is moved along the  $x$  direction based on the linear mapping from the initial bin structure to the adjusted one. Once cell shifting has been applied on each row of the bin structure, each column is then processed and the cells are moved along the  $y$  direction.

**Figure 5** (a) illustrates the changes for the boundaries and utilizations of each bin with cell shifting along the  $x$  direction, considering a particular row in the regular bin structure. Let  $x_p$  indicate the right boundary coordinate of the  $p$ -th bin  $b_p$  in the initial bin structure, and  $\hat{x}_p$  represent the adjusted right boundary coordinates after cell shifting. To even out the utilization among adjacent bins, the following equation was introduced in Ref. 57) to compute the adjusted boundary coordinates:

$$\hat{x}_p = \frac{x_{p-1}(\text{util}(b_{p+1}) + \delta) + x_{p+1}(\text{util}(b_p) + \delta)}{\text{util}(b_p) + \text{util}(b_{p+1}) + 2\delta}, \quad (23)$$

where  $\text{util}(b_p)$  indicates the utilization of bin  $b_p$ , and  $\delta$  is a small constant that

helps the escape from the invalid results at either  $\text{util}(b_p) = 0$  or  $\text{util}(b_{p+1}) = 0$ , where the bin boundaries may cross each other after the bin structure adjustment. Figure 5 (b) gives the adjusted bin structure from the initial bin structure in Fig. 5 (a). It can be seen that the utilization differences become smaller after the bin structure adjustment.

After the construction of the adjusted bin structure for some particular row, the cells within this row is then linearly mapped according to the initial and adjusted bin boundaries. For some cell  $v$  in the bin  $b_p$ , let  $x$  stand for the initial  $x$  coordinate of  $v$  before cell shifting, and  $\hat{x}$  be that after cell shifting.  $\hat{x}$  can then be computed as follows:

$$\hat{x} = \frac{\hat{x}_p(x - x_{p-1}) + \hat{x}_{p-1}(x_p - x)}{x_p - x_{p-1}}. \quad (24)$$

### 3.2.3 Minimum Cost Flow Assignment

In Ref. 4), Angihotri and Madden proposed to spread cells using the minimum cost flow assignment. To reduce the problem size, for a given initial placement, a physical clustering is first performed to cluster nearby cells together. Then the placement region is partitioned into uniform sub-regions, and a minimum cost flow algorithm is used to assign clusters into the corresponding sub-regions. Note that the sizes of clusters are kept as uniform as possible during the physical clustering, and the sub-region area is determined accordingly to maintain a one-to-one correspondence.

After the construction of clusters and sub-regions, a minimum cost flow problem is then formulated to find the best assignment of the clusters to the sub-regions.

The network flow for the minimum cost flow algorithm is constructed as follows:

- (1) Construct the node set  $V_{flow} = C \cup R \cup \{s, t\}$ , where  $C$  is the set of nodes representing the clusters, and  $R$  is the set of nodes representing the sub-regions, and  $s$  and  $t$  are the source and sink of the flow network.
- (2) Construct the edge set  $E_{flow} = (C \times R) \cup (\{s\} \times C) \cup (R \times \{t\})$ . For each edge  $e \in E_{flow}$ ,  $u_e$  denotes the capacity of the edge and  $w_e$  denotes the weight of the edge.
- (3) For each edge  $e = (c, r) \in (C \times R)$ ,  $u_e$  is set to  $\infty$ , and  $w_e$  is set to the HPWL degradation to move all the cells in cluster  $c$  from their original positions to the center of sub-region  $r$ .

(4) For each edge  $e \in (\{s\} \times C) \cup (R \times \{t\})$ ,  $u_e$  is set to 1 and  $w_e$  is set to 0.

(5) The supply of  $s$  is set to  $|C|$  and the demand of  $t$  is set to  $-|C|$ .

It is clear that such a problem is a special case of the minimum cost flow problem, and is known as weighted bipartite matching or the transportation problem<sup>4)</sup>. After finding the minimum cost flow of the constructed flow network, the cells can thus be spread according to the cluster-to-bin assignment. Besides, it can also be seen that the minimum cost flow formulation is very similar to the transportation formulation introduced in Section 3.2.1. The major difference is that in the transportation formulation, the cells with different sizes are directly assigned to sub-regions, and thus the fractional assignment is relaxed to reduce the problem complexity, while in the minimum cost flow formulation, the cluster size is controlled and thus the one-to-one relation from clusters to sub-regions is maintained.

### 3.2.4 Diffusion

The overlap reduction can also be modelled as the physical diffusion process. Such an idea was introduced by Ren, et al. in Ref. 44). The physical diffusion process is driven by the gradient of concentration. Mathematically, the relationship between material concentration, time, and space can be written as the following equation<sup>44)</sup>:

$$\frac{\partial d_{x,y}(t)}{\partial t} = \bar{D} \nabla_{x,y}^2(t), \quad (25)$$

where  $d_{x,y}(t)$  is material concentration at point  $(x,y)$  at time  $t$ , and  $\bar{D}$  is the diffusivity which determines the speed of diffusion. For easier presentation,  $\bar{D}$  is assumed to be 1 in the following discussions. For a fixed diffusion region, the boundary conditions are defined as  $\nabla d_{x_b,y_b}(t) = 0$  for coordinates  $(x_b, y_b)$  on the region boundary.

To determine the route of a cell from its initial position to the final equilibrium position, a velocity function is required to obtain the velocity of the cell at every location for a given time  $t$ . The velocity is determined by the amount of density and the local density gradient. Let vector  $\mathbf{v}_{x,y} = (v_{x,y}^H, v_{x,y}^V)$  represent the velocity at position  $(x,y)$ . The velocity function can then be written as the following functions<sup>44)</sup>:

$$\begin{aligned} v_{x,y}^H(t) &= -\frac{\partial d_{x,y}(t)}{\partial x} / \partial d_{x,y}(t), \\ v_{x,y}^V(t) &= -\frac{\partial d_{x,y}(t)}{\partial y} / \partial d_{x,y}(t). \end{aligned} \quad (26)$$

After obtaining the velocity function, the cell position can be computed easily. Given a starting position  $(x(0), y(0))$  for a cell, its new position  $(x(t), y(t))$  at time  $t$  can be determined by integrating the velocity field:

$$\begin{aligned} x(t) &= x(0) + \int_0^t v_{x(t'),y(t')}^H(t') dt', \\ y(t) &= y(0) + \int_0^t v_{x(t'),y(t')}^V(t') dt'. \end{aligned} \quad (27)$$

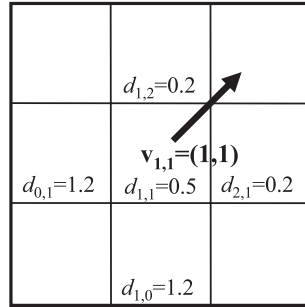
Equations (25), (26), and (27) are sufficient to simulate the diffusion process at any position at time  $t$ . However, to apply diffusion on placement problems, the challenge is to translate those equations from a continuous domain to a discrete one. The same as the other overlap reduction techniques, the placement region is again divided into equal-sized bins indexed by  $(i,j)$ . Then in the discrete domain,  $d_{i,j}$  now stands for the density of bin  $b_{i,j}$  with respect to the material concentration in the continuous domain. The density for each bin can easily be computed by the accumulation of the overlapping area between each cell and the bin divided by the bin area. Assuming that the density  $d_{i,j}(n)$  for all bins has been computed for time step  $n$ , to compute the density for the next time step  $n+1$ , Ren, et al.<sup>44)</sup> proposed to discretize Eq. (25) by the forward-time-centered-space (FTCS)<sup>43)</sup> scheme. The new density can then be written as:

$$\begin{aligned} d_{i,j}(n+1) &= d_{i,j}(n) + \frac{\Delta t}{2} (d_{i+1,j}(n) + d_{i-1,j}(n) - 2d_{i,j}(n)) \\ &\quad + \frac{\Delta t}{2} (d_{i,j+1}(n) + d_{i,j-1}(n) - 2d_{i,j}(n)). \end{aligned} \quad (28)$$

It can be seen that the density of a bin at time step  $n+1$  is determined by its density and the densities of its four neighboring bins at time step  $n$ .

The velocity Eq. (26) can also be discretized by the FTCS scheme. Assuming that the cells within bin  $b_{i,j}$  are assigned to the same velocity, the velocity vector  $\mathbf{v}_{i,j}$  is discretized as follows:





**Fig. 6** An example of the bin velocity computation. The velocity of bin  $b_{1,1}$  is determined by the densities of bins  $b_{0,1}$ ,  $b_{1,0}$ ,  $b_{1,1}$ ,  $b_{1,2}$ , and  $b_{2,1}$ .

$$v_{i,j}^H(n) = -\frac{d_{i+1,j}(n) - d_{i-1,j}(n)}{2d_{i,j}(n)},$$

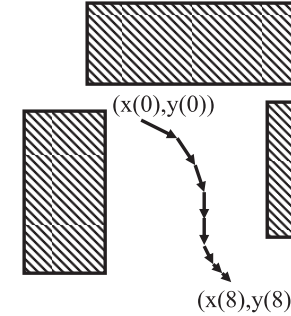
$$v_{i,j}^V(n) = -\frac{d_{i,j+1}(n) - d_{i,j-1}(n)}{2d_{i,j}(n)}. \quad (29)$$

**Figure 6** shows an example of the bin velocity computation. However, assigning cells within a bin with the same velocity loses the differences between the cells in different positions. One easy way to solve this problem is to interpolate the cell velocity from the closest bin velocities, and in Ref. 44) the bilinear interpolation is applied. Assuming that for the cell locating at point  $(x, y)$ , the four closest bins are bins  $b_{p,q}$ ,  $b_{p+1,q}$ ,  $b_{p,q+1}$ , and  $b_{p+1,q+1}$ . Let  $(x_p, x_q)$  denote the center position of bin  $b_{p,q}$ . Define the horizontal distance ratio  $\alpha$  for the cell to bin  $b_{p,q}$  as  $(x - x_p)/(x_{p+1} - x_p)$ , and the vertical distance ratio  $\beta$  as  $(y - y_q)/(y_{q+1} - y_q)$ . Now the cell velocity  $\mathbf{v}_{x,y}$  at point  $(x, y)$  can be computed by

$$v_{x,y}^H = v_{p,q}^H + \alpha(v_{p+1,q}^H - v_{p,q}^H) + \beta(v_{p,q+1}^H - v_{p,q}^H) + \alpha\beta(v_{p,q}^H + v_{p+1,q+1}^H - v_{p+1,q}^H - v_{p,q+1}^H),$$

$$v_{x,y}^V = v_{p,q}^V + \alpha(v_{p+1,q}^V - v_{p,q}^V) + \beta(v_{p,q+1}^V - v_{p,q}^V) + \alpha\beta(v_{p,q}^V + v_{p+1,q+1}^V - v_{p+1,q}^V - v_{p,q+1}^V). \quad (30)$$

Finally, the cell positions at each time step can be derived from their corresponding velocity. The cell position function at time step  $n + 1$  can be written in the recursive form as



**Fig. 7** An example of the diffusion process.

$$x(n+1) = x(n) + v_{x(n),y(n)}^H \cdot \Delta t,$$

$$y(n+1) = y(n) + v_{x(n),y(n)}^V \cdot \Delta t. \quad (31)$$

**Figure 7** gives an example of the diffusion process. It can be seen that the cell moves from higher density locations to the lower ones, and the movement becomes smaller toward the end of the path.

### 3.2.5 Density Control

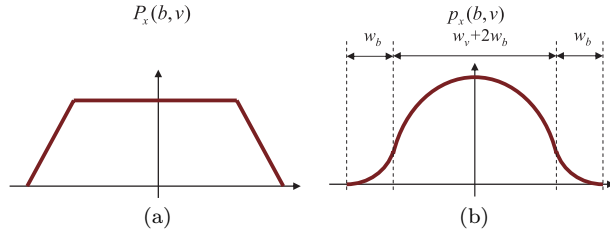
One most popular method to spread cells evenly in analytical placement is working through the density domain, which is adopted by various famous academic placers, such as APlace<sup>32)</sup>, FDP<sup>60)</sup>, Kraftwerk<sup>51)</sup>, mFAR<sup>23)</sup>, mPL6<sup>12)</sup>, and NTUplace3<sup>18)</sup>. At first, to compute the density induced by the cells, the placement region is divided into uniform non-overlapping bin grids. The density function for bin  $b$  can be expressed as

$$D_b(\mathbf{x}, \mathbf{y}) = \sum_{v \in V} P_x(b, v) P_y(b, v), \quad (32)$$

where  $P_x$  and  $P_y$  are the overlap functions of bin  $b$  and block  $v$  along the  $x$  and  $y$  directions. Then the cell spreading can be transformed into the following constraint:

$$D_b(\mathbf{x}, \mathbf{y}) \leq M_b \text{ for each bin } b, \quad (33)$$

where  $M_b$  is the maximum allowable area of movable cells in bin  $b$ . However, since density  $D_b(\mathbf{x}, \mathbf{y})$  is neither smooth nor differentiable, it is hard to optimize it directly. Therefore, many smoothing techniques have been proposed to solve this



**Fig. 8** (a) The overlap function  $P_x(b, v)$ . (b) The smoothed overlap function  $p_x(b, v)$ .

problem. Three popular smoothing techniques, (1) bell-shaped smoothing, (2) Helmholtz smoothing, and (3) Poisson smoothing, are explained in the following.

**Bell-Shaped Smoothing:** APlace<sup>32)</sup> and NTUPlace<sup>318)</sup> adopt the bell-shaped function  $p_x$  to smooth  $P_x$ .  $p_x$  is defined by

$$p_x(b, v) = \begin{cases} 1 - ad_x^2, & 0 \leq d_x \leq \frac{w_v}{2} + w_b \\ b(d_x - \frac{w_v}{2} - 2w_b)^2, & \frac{w_v}{2} + w_b \leq d_x \leq \frac{w_v}{2} + 2w_b \\ 0, & \frac{w_v}{2} + 2w_b \leq d_x, \end{cases} \quad (34)$$

where

$$\begin{aligned} a &= \frac{4}{(w_v + 2w_b)(w_v + 4w_b)} \\ b &= \frac{2}{w_b(w_v + 4w_b)}, \end{aligned} \quad (35)$$

$w_b$  is the bin width,  $w_v$  is the cell width, and  $d_x$  is the center-to-center distance of the cell  $v$  and the bin  $b$  in the  $x$  direction. **Figure 8** (a) and Fig. 8 (b) show the original and the smoothed overlap functions, respectively. The range of cell's potential is  $w_v + 4w_b$  in the  $x$  direction. The smooth  $y$ -potential function  $p_y(b, v)$  can be defined in a similar way, and the range of cell's potential is  $h_v + 4h_b$  in the  $y$  direction. By doing so, the non-smooth function  $D_b(\mathbf{x}, \mathbf{y})$  can be replaced by a smooth one,

$$\hat{D}_b(\mathbf{x}, \mathbf{y}) = \sum_{v \in V} c_v p_x(b, v) p_y(b, v), \quad (36)$$

where  $c_v$  is a normalization factor so that the total potential of a cell equals its area.

**Helmholtz Smoothing:** mPL6<sup>12)</sup> approximates the smoothed density

$\hat{D}_b(\mathbf{x}, \mathbf{y})$  by the solution to the Helmholtz equation with zero-derivative boundary conditions:

$$\Delta \hat{D}_b(\mathbf{x}, \mathbf{y}) - \epsilon \hat{D}_b(\mathbf{x}, \mathbf{y}) = -D_b(\mathbf{x}, \mathbf{y}), \quad (37)$$

where  $\epsilon$  is a smoothing parameter,  $\epsilon > 0$ , and  $\Delta$  is the Laplacian operator ( $\Delta \equiv \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ ).

**Poisson Smoothing:** In FDP<sup>60)</sup>, Kraftwerk<sup>51)</sup>, and mFAR<sup>23)</sup>, the density is treated as the electrostatic potential. Therefore, the smoothed density  $\hat{D}_b(\mathbf{x}, \mathbf{y})$  is approximated by the solution of the Poisson equation:

$$\Delta \hat{D}_b(\mathbf{x}, \mathbf{y}) = -D_b(\mathbf{x}, \mathbf{y}). \quad (38)$$

### 3.2.6 Frequency Control

Instead of working in the density (spatial) domain, Yao, et al.<sup>65)</sup> proposed to quantitatively evaluate the cell distribution in the frequency domain, and then the evaluation can be optimized to reduce the overlaps during global placement. First of all, the densities on a pre-partitioned bin structure of a given placement still need to be computed. Assuming that the placement region is partitioned into  $N \times N$  equal-sized bins, let  $\mathbf{D} = \{d_{i,j}\}$  represent the density matrix. Then the density matrix  $\mathbf{D}$  can be interpreted to the frequency domain by the two dimensional *Discrete Cosine Transformation (DCT)*. Let  $\mathbf{F} = \{f_{i,j}\}$  denote an  $N \times N$  frequency distribution matrix. The DCT is defined as follows:

$$f_{i,j} = \frac{2}{N} C(i) C(j) \cdot \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \left( d_{x,y} \cos\left(\frac{(2x+1)i\pi}{2N}\right) \cos\left(\frac{(2y+1)j\pi}{2N}\right) \right), \quad (39)$$

where  $C(i)$  is the coefficient defined by  $C(0) = 1/\sqrt{2}$ , and  $C(i) = 1$  for  $1 \leq i \leq N-1$ . Through such a transformation, in the frequency matrix  $\mathbf{F}$ , each entry actually represents the different frequency distribution of the density matrix. **Figure 9** gives the distribution patterns on a  $4 \times 4$  bin structure. Besides, the density energy remains the same before and after the transformation (in other words,  $\sum_{i,j} d_{i,j}^2 = \sum_{i,j} f_{i,j}^2$ ). To make the cells spread evenly (and thus the overlaps are reduced), the density distribution should concentrate on those

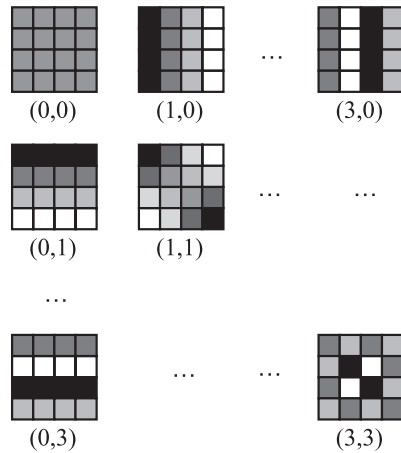


Fig. 9 Distribution patterns of different frequencies in the  $\mathbf{F}$  matrix on a  $4 \times 4$  bin structure.

frequencies with better evenness. Therefore, Yao, et al. defined the distribution cost  $DIST$  as follows:

$$DIST = \sum_{i,j} (w_{i,j} \cdot f_{i,j}^2), \tag{40}$$

where  $w_{i,j}$  is the weight of the distribution at frequency  $(i, j)$ ,  $w_{i,j}$  is defined by  $w_{0,0} = 0$ , and  $w_{i,j} = 1/(i+j)$  otherwise. Consequently, the  $DIST$  is approximated as a convex quadratic function of cell positions  $(\mathbf{x}, \mathbf{y})$ , and thus can be integrated into the analytical objective function of the global placement.

### 3.3 Integration of the Wirelength Models and Overlap Reduction Techniques

In Sections 3.1 and 3.2, the wirelength models and overlap reduction techniques have been studied. Now the problem is to integrate the wirelength model and the overlap reduction technique into one unified global placement algorithm. Since the wirelength optimization tends to pull cells together, this objective is contradictory to the overlap reduction, which pushes cells away from each other. Therefore, the integration must consider both objectives carefully to avoid the results from biasing to one of the two objectives. In this section, we classify the integration of wirelength models and overlap reduction techniques into three

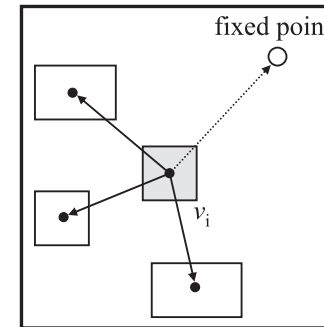


Fig. 10 Illustration of the fixed point method. For cell  $v_i$ , the fixed point is added on the top-right corner to make cells distributed more evenly.

types: (1) the fixed point method, (2) the penalty method, and (3) the region constraint method. The details are given in the following.

#### 3.3.1 Fixed Point Method

One most popular method to integrate the wirelength model and the overlap reduction technique is called the *fixed point method*. It can be briefly summarized as feeding the placement obtained from overlap reduction techniques back to the placement problem by adding fixed points and pseudo connections into the original netlist. Then the placement problem is again solved on the modified netlist to find an equilibrium between the wirelength minimization and overlap reduction objectives. There is still slight difference between the fixed point methods applied on different placers. For example, the most common way is to create one fixed point at the target position obtained from overlap reduction techniques for each cell, and make a pseudo connection between them. This is adopted by DPlace<sup>39)</sup>, FDP<sup>60)</sup>, Kraftwerk2<sup>51)</sup>, mFAR<sup>23)</sup>, and RQL<sup>58)</sup>. **Figure 10** shows an illustration for this idea. However, for some specific overlap reduction techniques (such as the minimum cost flow assignment<sup>4)</sup> introduced in Section 3.2.3), only a rough position guide (the bin assignment) is obtained for a group of cells. Therefore, those cells will share the same fixed point (located at the center of the assigned bin in Ref. 4)) instead of creating one fixed point for each cell. In addition to directly putting the fixed points at the target position obtained from the overlap reduction techniques, some other placers (such as FastPlace<sup>59)</sup>) might only take

the direction and distance from the original position of a cell to its target position as reference. Then the fixed points are put on the chip boundary in the same direction to the target positions with adjusted weights of pseudo connections.

### 3.3.2 Penalty Method

Adopting the density control method introduced in Section 3.2.5, the global placement problem in Eq. (1) can be transformed as follows:

$$\begin{aligned} \min \quad & W(\mathbf{x}, \mathbf{y}) \\ \text{s.t.} \quad & D_b(\mathbf{x}, \mathbf{y}) \leq M_b, \quad \text{for each bin } b, \end{aligned} \quad (41)$$

where  $W(\mathbf{x}, \mathbf{y})$  is the wirelength function,  $D_b(\mathbf{x}, \mathbf{y})$  is the density function of movable blocks of bin  $b$  on the pre-partitioned bin structure, and  $M_b$  is the maximum allowable area of movable blocks in bin  $b$ . It should be noted that the density constraints of the optimization problem increase the difficulty of solving the problem. Therefore, the *quadratic penalty method*, adopted by APlace<sup>33)</sup> and NTUPlace<sup>34)</sup>, is often used to solve Eq. (41), which implies to solve a sequence of unconstrained minimization problems of the form:

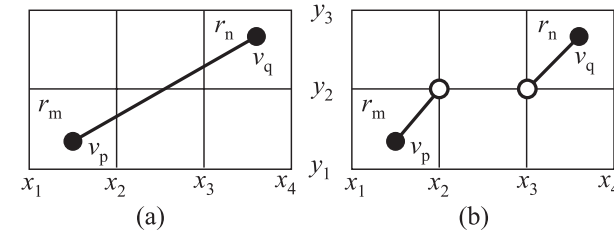
$$\min \quad W(\mathbf{x}, \mathbf{y}) + \lambda \sum_b (\hat{D}_b(\mathbf{x}, \mathbf{y}) - M_b)^2, \quad (42)$$

where  $\lambda$  is the normalizing factor to balance the wirelength and density values, and can also be changed to vary the weighting between wirelength and density. It should also be noted that the smoothing techniques introduced in Section 3.2.5 are still required for the unconstrained minimization problem, since a smoothed density usually eases the search of the density optimization directions, which helps minimize the density part of Eq. (42).

Similarly, for the frequency control method introduced in Section 3.2.6, the density penalty is added into the objective function by computing the weighted square sum of all entries in the frequency matrix  $\mathbf{F}$ . Since such a method works on the frequency domain directly, no smoothing technique is required. Instead, the density penalty is approximated by a convex quadratic function of cell positions  $(\mathbf{x}, \mathbf{y})$ , which also helps the search of the optimization directions<sup>65)</sup>.

### 3.3.3 Region Constraint

For the partitioning-based overlap reduction techniques introduced in Section 3.2.1, the cells are assigned to sub-regions instead of specific positions. Such



**Fig. 11** An example of the net splitting. (a) The connection before net splitting. (b) An abstract view of the connection after applying net splitting.

assignment is still required to be linked back to the original placement problem for the later optimization. The most intuitive way is to add inequalities for each cell to constrain the cell positions within the assigned sub-region, but this will increase the difficulty of solving the placement problem too much. One alternative is to add equalities to make the gravity center of cells assigned to the same sub-region fixed at the center of the sub-region. Besides, the *net splitting* proposed by Vygen<sup>62)</sup> is also helpful to control the cell positions locating within the assigned sub-regions. Consider the net splitting on a given two-pin connection shown in **Fig. 11** (a). The pins  $v_p$  and  $v_q$  are assigned to sub-regions  $r_m$  and  $r_n$ , respectively. The  $x_i$ 's and  $y_i$ 's give the boundary coordinates of all bins, and  $(x_p, y_p)$  and  $(x_q, y_q)$  stand for the respective coordinates of  $v_p$  and  $v_q$ . Then, to constrain  $v_p$  and  $v_q$  locating within  $r_p$  and  $r_q$  respectively during the global placement process, the net splitting will break the connection into two pieces by modifying the wirelength objective of this connection to

$$|x_p - x_2| + |y_p - y_2| + |x_q - x_3| + |y_q - y_2|. \quad (43)$$

As shown in **Fig. 11** (b), this operation is equivalent to breaking the connection on the corner of the assigned regions of the pins. Therefore, the global optimization process will try to move  $v_p$  and  $v_q$  to the other end to minimize wirelength without exceeding the boundary of  $r_m$  and  $r_n$ . Consequently, the partitioning assignment is successfully linked back to the global placement problem.

### 3.4 Optimization Techniques

**Table 2** lists the state-of-the-art placers and their wirelength models, overlap reductions, integration approaches, and optimization techniques. Most analytical

**Table 2** Comparisons of wirelength models, overlap reductions, integrations approaches, and optimization techniques among popular analytical placers.

Placer	Wirelength Model	Overlap Reduction	Integration	Optimization
APlace <sup>33)</sup>	LSE	Density (Bell-Shaped)	Penalty Method	Nonlinear
BonnPlace <sup>8)</sup>	Quadratic	Partitioning	Region Constraint	Quadratic
DPlace <sup>39)</sup>	Quadratic	Diffusion	Fixed Point	Quadratic
FastPlace <sup>59)</sup>	Quadratic	Cell Shifting	Fixed Point	Quadratic
FDP <sup>60)</sup>	Quadratic	Density (Poisson)	Fixed Point	Quadratic
Gordian <sup>35)</sup>	Quadratic	Partitioning	Region Constraint	Quadratic
hATP <sup>40)</sup>	Quadratic	Partitioning	Region Constraint	Quadratic
Kraftwerk2 <sup>51)</sup>	Bound2Bound	Density (Poisson)	Fixed Point	Quadratic
mFAR <sup>23)</sup>	Quadratic	Density (Poisson)	Fixed Point	Quadratic
mPL6 <sup>12)</sup>	LSE	Density (Helmholtz)	Penalty Method	Nonlinear
NTUplace3 <sup>18)</sup>	LSE	Density (Bell-Shaped)	Penalty Method	Nonlinear
RQL <sup>58)</sup>	Quadratic	Cell Shifting	Fixed Point	Quadratic
UPlace <sup>65)</sup>	Quadratic	Frequency	Penalty Method	Quadratic
Vaastu <sup>4)</sup>	LSE	Assignment	Fixed Point	Nonlinear

placers can be classified into two categories based on the type of the mathematical optimization technique: (1) quadratic programming and (2) non-linear (non-quadratic) programming.

### 3.4.1 Quadratic Programming

The quadratic programming is one of the most common approaches to the placement problem. With the quadratic wirelength model, it can be solved in the quadratic optimization problem (for the  $x$  direction) given by

$$\min_x \sum_{i,j} w_{x,ij} (x_i - x_j)^2 = \min_x \frac{1}{2} \mathbf{x}^T \mathbf{Q}_x \mathbf{x} + \mathbf{c}_x^T \mathbf{x} + \mathbf{d}_x, \quad (44)$$

where  $w_{x,ij}$  represents the weight of the edge connecting cells  $i$  and  $j$ . The matrix  $\mathbf{Q}_x$  is the Hessian which represents the hyperedge connectivity. Assuming that some cells are fixed, the Hessian is a symmetric, positive-definite matrix. The vector  $\mathbf{c}_x$  represents fixed-cell-to-movable-cell connections, and the vector  $\mathbf{d}_x$  represents fixed-cell-to-fixed-cell connections. The optimization problem is strictly convex and has a unique minimizer given by the solution of a single, positive-definite system of linear equations,  $\mathbf{Q}_x \mathbf{x} + \mathbf{c}_x = 0$ . The wirelength along the  $y$  direction can also be solved by the same approach. Since the formulation optimizes quadratic wirelength, some other wirelength models, such as

the Bound2Bound wirelength model in Kraftwerk<sup>51)</sup>, is proposed to modify the netlist graph and weighting to fix the quadratic wirelength to linear wirelength. Further, overlap reduction techniques are usually adopted by either fixed point or partitioning. In the fixed point approach, the cell overlaps are gradually reduced by adding a fixed point that modified the matrix  $\mathbf{Q}_x$  or the vector  $\mathbf{c}_x$ . The partitioning approach either physically partitions the placement region (such as BonnPlace<sup>8)</sup> and hATP<sup>40)</sup>) or adds linear constraints to change the center-of-gravity of the cells<sup>35)</sup>. For both approaches, the optimization problem is always in the quadratic form, which can be solved efficiently.

### 3.4.2 Nonlinear Programming

The general nonlinear optimization problem for placement is usually solved by the penalty method. Solving the nonlinear problem is usually very time-consuming, and therefore the multilevel approach is often used. APlace<sup>33)</sup>, mPL6<sup>12)</sup>, NTUplace3<sup>18)</sup>, and Vaastu<sup>4)</sup> belong to this type. These four placers all use the LSE wirelength model. Among these placers, APlace and NTUplace3 both use the bell-shaped density model, and mPL6 uses the Helmholtz smoothed density model to reduce overlaps. Since both the wirelength model and overlap reduction technique are modelled in an analytical way, it is easy to apply the penalty method to optimize the nonlinear programming. Vaastu uses the linear

assignment method to find the target fixed-point for each cell, and add a pseudo net to connect the fixed point and the corresponding cell. The modified netlist is solved again, and the resulting placement spreads cells more. The fixed points and net weights are changed iteratively until all cells are spread enough.

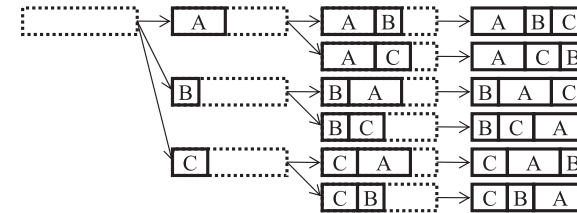
#### 4. Legalization

The legalization stage tries to remove all overlaps with minimum wirelength or total displacement while the relative cell order of the global placement is kept. The *Tetris-like greedy legalization method*<sup>22)</sup> is perhaps the most popular approach. Cells are first sorted according to their  $x$  coordinates, and then cells are placed at the closet available positions with minimal costs in left-to-right/right-to-left order. This greedy algorithm is very fast with negligible running time compared to that of the global placement. Other modifications has been applied to increase the success rate and the quality of the legalized placement, such as squeezing the cell position to the left/right side<sup>31)</sup>, or using higher priority for large cells/blocks<sup>18)</sup>.

Another popular legalization method is called *single-row placement*. This method concerns about the optimal positions for the cells to be placed within the same row at one time, while their relative ordering is kept. The single-row placement with respect to HPWL or total linear displacement minimization is solved by linear programming by Vygen<sup>61)</sup>. Then, Kahng, et al.<sup>30)</sup> proposed the *clumping algorithm* to solve such a problem more efficiently. Following this work, the clumping algorithm is further sped up with a specific data structure proposed by Brenner and Vygen<sup>9)</sup>. The single-row placement with respect to quadratic displacement minimization is formulated as a quadratic problem by Spindler, et al.<sup>50)</sup>, and is solved by dynamic programming. They also proposed to integrate the single-row placement with the Tetris-like method to obtain a better balance between the solution quality and running time.

#### 5. Detailed Placement

In the detailed placement stage, the standard cell positions are further optimized to improve the placement quality. The objective of the detailed placement algorithm is to find a better position for each standard cell in the available free

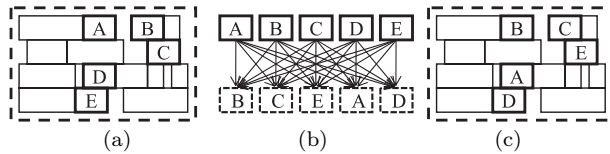


**Fig. 12** An illustration of all possible orders for three cells. The branch-and-bound method can be used to find the cell order with smallest wirelength.

spaces. We introduce three popular approaches, cell order polishing, cell matching, and global swapping/moving in the following.

**Cell order polishing** permutes a small window of cells each time to find the best ordering by enumerating all possible orderings using the branch-and-bound method. The number of cells contained by the window is an important factor to control the tradeoff between the running time and solution quality. **Figure 12** gives a cell order polishing example for a window containing three cells. This technique is widely used in the state-of-the-art placers<sup>6),16)–18),29),42),45),51)</sup>.

**Cell matching** was first proposed by Chen, et al. in Ref. 18); it is an efficient technique that can optimize more cells at the same time. The cell matching algorithm finds a group of exchangeable cells inside a given window, and formulates a bipartite matching problem by assigning the cells to available slots in the window. To keep the legality of the placement solution, for each slot, only the assigning relations for cells with widths less than or equal to the slot width are constructed. The assignment cost is given by the HPWL difference of placing a cell in different slots. Then, the shortest augmenting path algorithm<sup>27)</sup> is applied to solve the bipartite matching problem. Though the bipartite matching problem can be solved optimally in polynomial time, the optimal assignment cannot guarantee the optimal HPWL result, because the HPWL of a cell connected to each empty slot depends on the positions of other connected cells. The cell matching algorithm<sup>18)</sup> remedies this drawback by selecting *independent cells* at one time to perform bipartite matching. Here by independent cells, it means that there is no common net between any pair of the selected cells. The bipartite matching problem can be solved very quickly when the number of cells is smaller than 100.



**Fig. 13** An illustration of cell matching. (a) Select exchangeable cells. (b) Create the bipartite matching problem. (c) Update the placement using the matching result.

Compared with other detailed placement algorithms, cell matching can optimize the placement result more globally. **Figure 13** illustrates the cell matching.

**Global moving/swapping**<sup>29),42)</sup> moves each cell to the optimal location among available whitespaces without changing the positions of other cells. This technique is especially useful when the design utilization is low. When design utilization is high, it may not be easy to find a whitespace to place the cell. In this case, this technique tries to swap the cell with a cell within the optimal region to see if a better result can be obtained.

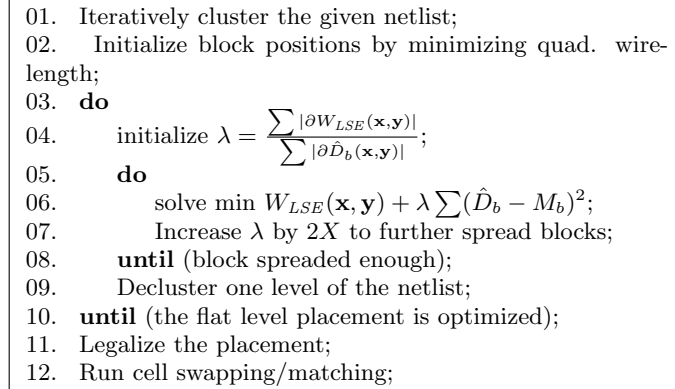
## 6. Example Placers

In this section, we take the two leading academic placers, NTUplace3<sup>18)</sup> and Kraftwerk2<sup>51)</sup>, as example placers to explain how to unify the aforementioned ingredients into complete analytical placers.

### 6.1 NTUplace3

NTUplace3<sup>18)</sup> is an analytical placer based on the LSE wirelength model and the bell-shaped potential smoothing for overlap reduction. **Figure 14** summarizes the NTUplace3 algorithm. The multilevel framework is adopted to increase the scalability of the placer. During the coarsening stage, NTUplace3 clusters blocks to reduce the number of movable blocks. The hierarchy of clusters is built by the first-choice (FC) clustering algorithm<sup>11)</sup>. The area of a clustered block is controlled so that it does not exceed 1.5 times of the average area of clustered blocks. The clustering process continues until the number of blocks is reduced by 5 times, and then a level of clustered circuit is obtained. The FC clustering algorithm is applied several times until the block number in the resulting clustered circuit is less than a user-specified number (6000 by default).

After clustering, the initial placement for the coarsest level is generated by min-



**Fig. 14** The NTUplace3 algorithm.

imizing the quadratic wirelength using the conjugate gradient method. Then, the placement problem is solved from the coarsest level to the finest level. The horizontal/vertical placement grid numbers are set to the square root of the number of clusters in the current level, and the maximum area of movable blocks  $M_b$  for each bin is calculated. Also, the value of  $\lambda$  for Eq. (42) is initialized according to the strength of wirelength and density gradients,

$$\lambda = \frac{\sum |\partial W_{LSE}(\mathbf{x}, \mathbf{y})|}{\sum |\partial \hat{D}_b(\mathbf{x}, \mathbf{y})|}, \quad (45)$$

where  $W_{LSE}$  is the LSE wirelength function, and  $\hat{D}_b$  is the bell-shaped potential function, and the value of  $\lambda$  is increased by  $2X$  for each iteration. A conjugate gradient solver with dynamic step-size control is used to solve the nonlinear optimization problem in Eq. (42) (nonlinear programming with the quadratic penalty method).

During uncoarsening, all blocks inside a cluster inherit the center position of the original cluster. Then, blocks are declustered, providing the initial placement for the next level.

To measure the evenness of the block distribution, NTUplace3 adopts the *overflow ratio*. The overflow ratio is defined as the total overflow area in all bins over the area of total movable blocks. The global placement stage stops when the

overflow ratio is less than or equal to a user-specified target value, which is 0 by default.

The Tetris standard-cell legalization method is extended to solve the mixed-size legalization problem. The legalization order of blocks are determined by their  $x$  coordinates, widths, and heights. The legalization priority of a block  $v_i$  is given by

$$\text{priority}(v_i) = k_1x_i + k_2w_i + k_3h_i, \quad (46)$$

where  $k_1$ ,  $k_2$ , and  $k_3$  are user-specified weights for each term. (By default,  $k_1 = 1000$  and  $k_2 = k_3 = 1$ .) As a result, large blocks are legalized earlier than small blocks when they have the same  $x$  coordinate to achieve higher success rates.

In the detailed placement stage, cell swapping (cell-order polishing) and cell matching are used to further reduce the wirelength.

## 6.2 Kraftwerk2

**Figure 15** summarizes the Kraftwerk2 algorithm<sup>51</sup>). At first, an initial placement is computed by minimizing the quadratic cost function over a few iterations. In each iteration, the Bound2Bound wirelength model is applied to adjust the two-pin connection weights. The initial placement has a minimal netlength. However, the cells are concentrated somewhere on the chip (mostly at the center), and there may be significant overlaps.

In global placement, the cells are spread iteratively over the chip. Each placement iteration starts by determining the supply (free space) and demand (cell density) system and by computing the smoothed potential using the Poisson equation in Eq. (38). Then, the Bound2Bound net model is applied to determine the weights of the two-pin connections. Once the two-pin connection weights in the Bound2Bound wirelength model are determined, they remain constant for the rest of the placement iteration. Kraftwerk2 is based on quadratic programming. Since the objective function is convex, the minimum value is obtained by setting its derivative to zero. Therefore, solving  $\mathbf{Q}_\mathbf{x}\mathbf{x} + \mathbf{c}_\mathbf{x} = 0$ . In quadratic placement, each two-pin connection can be viewed as an elastic spring, the cost function represents the total energy of the spring system, and the derivative of an energy is a force. Therefore, the wire force  $\mathbf{F}^{net}$  between the pins is given by

$$\mathbf{F}^{net} = \mathbf{Q}_\mathbf{x}\mathbf{x} + \mathbf{c}_\mathbf{x}. \quad (47)$$

- |     |  |
|-----|--|
| 01. | Initialize placement by min. Bound2Bound wirelength;                                 |
| 02. | <b>while</b> (cell overlap > 20%)  |
| 03. | Create the demand-and-supply system;   |
| 04. | Calculate the potential by solving Equation (38);                                    |
| 05. | Apply the Bound2Bound wirelength model;  |
| 06. | <b>for</b> $x$ -direction and $y$ -direction   |
| 07. | Create $\mathbf{Q}_\mathbf{x}$ , $\mathbf{Q}_\mathbf{x}$ , $\mathbf{D}_\mathbf{x}$ ; |
| 08. | Solve Equation (52) w.r.t. $\Delta\mathbf{x}$ ;                                      |
| 09. | Update cell position $x$ by $\Delta\mathbf{x}$ ;                                     |
| 10. | Control the quality;   |
| 11. | Legalize the placement;  |
| 12. | Run cell flipping/swapping;  |

**Fig. 15** The Kraftwerk2 algorithm.

There are two additional forces in Kraftwerk2, the hold force and the move force. The hold force provides forces to keep cells at current position. Hence, the hold force  $\mathbf{F}^{hold}$  equals the negative wire force

$$\mathbf{F}^{hold} = -(\mathbf{Q}_\mathbf{x}\mathbf{x}' + \mathbf{c}_\mathbf{x}), \quad (48)$$

where  $\mathbf{x}'$  is the current  $x$  coordinates of all cells. The move force moves the cells to reduce the cell overlaps. The target fixed point  $\hat{x}_i$  of each cell  $v_i$  is given by

$$\hat{x}_i = x'_i - \frac{\partial}{\partial x} \hat{D}(x, y) \Big|_{(x_i, y_i)}, \quad (49)$$

or

$$\hat{\mathbf{x}} = \mathbf{x}' - \hat{\mathbf{D}}_\mathbf{x}, \quad (50)$$

where  $\hat{D}(x, y)$  is the Poisson smoothed density function. Then, the move force is defined as

$$\mathbf{F}^{move} = \dot{\mathbf{Q}}_\mathbf{x}(\mathbf{x} - \hat{\mathbf{x}}). \quad (51)$$

Setting the sum of the wire force, the hold force, and the move force to zero, the following linear system can be obtained:



$$(\mathbf{Q}_x + \dot{\mathbf{Q}}_x)\Delta\mathbf{x} = -\dot{\mathbf{Q}}_x\hat{\mathbf{D}}_x, \quad (52)$$

where  $\Delta\mathbf{x} = \mathbf{x} - \mathbf{x}'$ .

The next step in each placement iteration is to solve the above system of linear equations for the  $x$ -direction and a similar one for the  $y$ -direction. Then, the cell positions are updated. Solving the systems of linear equations and updating the cell positions are performed once per placement iteration. At the end of each placement iteration, a quality-control procedure is called to adjust the weights of the move force. The global placement is stopped if the *cell overlap*  $\Omega$  is below a certain limit, e.g., below 20%. The definition of  $\Omega$  is given by

$$\Omega = 1 - \frac{\text{union of cell areas}}{\text{sum of cell areas}}. \quad (53)$$

When there is no overlap, the union of all cell areas is the same as the summation of cell areas, and  $\Omega = 0$ .

After global placement, the cells are legalized first, i.e., the remaining overlap is removed, and the cells are aligned to rows if necessary. Kraftwerk2 utilizes an approach similar to Tetris to legalize standard cells. After legalization, a simple greedy detailed placement method is applied to improve the legal placement: Single cells are flipped, or pairs of neighboring cells are exchanged.

## 7. Future Research Directions

Although recent analytical placement techniques have made significant progress for the wirelength-driven placement problem, there are still many emerging challenges arising from advanced VLSI process technologies and thus increasing design complexity. In this section, we present some potential research directions for modern VLSI placement with these challenges.

### 7.1 Large Macro Placement

Modern VLSI designs tend to have thousands of macros due to the use of IP modules, and these macros significantly differ in both sizes and shapes. However, traditional analytical placers cannot handle the macro orientation problem well, so the resulting placement quality may substantially degrade. When the macros are very large, the resulting placement may even contain significant overlaps and dead spaces. Thus, it is desirable for a modern placer to handle macro orien-

tations and legalize large macros. Recently, the two-stage approach of placing large macros first and then small standard cells has shown better results than the traditional one-stage approach of placing large macros and standard cells simultaneously<sup>14),15)</sup>. However, the two-stage approach has the intrinsic limitation to optimize the macro placement globally. Further, pre-designed macros, such as embedded memories and analog blocks, may preserve three or four metal layers for interior routing, and those regions will become routing blockages during the routing stage. Consequently, macros have a significant impact on chip routability. The most popular method to enhance the chip routability of mixed-size designs is to preserve free space around macros. However, the way to determine the amount of preserved free space is yet an unsolved issue. To facilitate chip placement and routing, it is desired to investigate a routing resource allocation method within macro placement.

### 7.2 Routability-Driven Placement

Traditional placement focuses on total wirelength minimization to obtain better circuit performance and a smaller layout area. Despite the pervasive use of the HPWL objective, there is a mismatch between the wirelength and congestion objectives in placement. Although congestion is widely addressed in routing algorithms, in most cases, some routing violations cannot be removed with cell locations being fixed. Hence, it is of particular importance to consider routability in the placement stage. Traditional congestion-aware placement algorithms<sup>32),38),49),64)</sup> allocate whitespaces to congestive regions for better routability. However, preserving whitespaces might not solve the congestion problem effectively. Therefore, new ideas for routability optimization, such as *net overlapping removal*<sup>26)</sup> and the *routed wirelength modelling*<sup>55)</sup> into the analytical objective, are proposed recently, which shows the research potential in routability-driven placement. Another issue is that, due to the lack of interaction between the placement and the routing stages, a routers may not honor the resource allocation obtained from a placer. Thus, a possible research direction is to develop a fast and accurate routing demand estimation method and integrate it into the placement stage.

### 7.3 Timing-Driven Placement

In high-speed circuits, a large portion of timing optimization is performed in

the placement stage. Conventional placement algorithms usually achieve the timing goal via wirelength minimization. Nevertheless, there is a gap between wirelength and actual delay, so many methods have been proposed recently to overcome this challenge. Those proposed timing-driven placement methods can be classified into two major categories: (1) path-based and (2) net-based methods. The path-based methods<sup>21),25),52),53)</sup> try to control critical path delays directly, but they are not suitable for modern circuits due to their exponentially growing number of timing paths. The net-based method<sup>31)</sup> transfers the timing constraint of each path into net weights. However, since the net-based method ignores paths individuality, the placement result will be barely controllable. Due to these drawbacks in existing timing-driven placement algorithms, it is worthy to further study the timing optimization techniques with lower complexity and higher controllability. Further, the existence of large macros imposes more difficulties for the timing-driven placement.

#### 7.4 Power-Aware Placement

With the pervasive use of hand-held devices and the reliability/thermal issues in modern chips, power consumption has become the first-order cost metric in modern VLSI designs. Previous works, like<sup>19)</sup> by Cheon, et al., have been proposed to reduce the power consumption during the placement stage. To further reduce the power consumption, the multiple supply voltage<sup>56)</sup> has been widely applied in advanced low-power designs, and brought new issues to physical design. Recent research<sup>36)</sup> has investigated a voltage assignment method integrated with the floorplanning stage. If we can honor such voltage assignment during the placement stage, we can have higher chances to further reduce the power consumption.

#### 7.5 Thermal Placement

As the process technology advances, the feature size keeps shrinking and thus the integration density keeps increasing while the clock frequency keeps rising. As a result, the increased power density significantly increases the chip temperature. However, reducing the power consumption alone is not sufficient to reduce the chip temperature, since the power density is also a dominant factor<sup>7)</sup>. Therefore, it is desirable to develop placement techniques that can spread blocks/cells over the whole placement region to lower the chip temperature variation. Kahng,

et al. proposed an analytical placement algorithm to minimize the maximum temperature and improve the chip reliability<sup>28)</sup>. In addition to the maximum temperature, the distribution of hot blocks/cells and thermal gradient are also important and should be considered to reduce the on-chip performance variations.

## 8. Conclusions

In this paper, we have surveyed essential techniques and algorithms for modern analytical placement. Unlike the previous articles that survey existing placement algorithms one by one, we start by dissecting the basic structure of analytical placement, then discuss the techniques applied to recent analytical placers, and exemplify the two leading placers, NTUplace3 and Kraftwerk2, for the composition of these techniques into a complete placer. Although significant progress has been made in recent analytical placement research, modern circuit designs have induced many more challenges and thus opportunities for future research on large macro placement and routability-, timing-, power-, and/or thermal-driven optimization of the placement problem.

**Acknowledgments** This work was supported in part by ITRI, SpringSoft, Synopsys Inc., TSMC, and National Science Council of Taiwan under Grant No's. NSC 96-2752-E-002-008-PAE, NSC 96-2628-E-002-248-MY3, NSC 96-2628-E-002-249-MY3, and NSC 96-2221-E-002-245.

## References

- 1) *ISPD 2005 Placement Contest*. <http://www.sigda.org/ispd2005/contest.htm>
- 2) *ISPD 2006 Placement Contest*. <http://www.sigda.org/ispd2006/contest.html>
- 3) Adya, S.N., Chaturvedi, S., Roy, J.A., Papa, D.A. and Markov, I.L.: Unification of partitioning, placement and floorplanning, *Proc. IEEE/ACM International Conference on Computer-Aided Design*, San Jose, CA, pp.550–557 (2004).
- 4) Agnihotri, A.R. and Madden, P.H.: Fast Analytic Placement using Minimum Cost Flow, *Proc. IEEE/ACM Asia South Pacific Design Automation Conference*, Yokohama, Japan, pp.128–134 (2007).
- 5) Agnihotri, A.R., Ono, S. and Madden, P.H.: Recursive Bisection Placement: Feng Shui 5.0 Implementation Details, *Proc. ACM International Symposium on Physical Design*, San Francisco, CA, pp.230–232 (2005).
- 6) Agnihotri, A.R., Ono, S., Li, C., Yildiz, M.C., Khatkhate, A., Koh, C.-K. and Madden, P.H.: Mixed Block Placement via Fractional Cut Recursive Bisection,

- IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol.24, No.5, pp.748–761 (2005).
- 7) Banerjee, K., Pedram, M. and Ajami, A.H.: Analysis and Optimization of Thermal Issues in High-Performance VLSI, *Proc. ACM International Symposium on Physical Design*, pp.230–237 (2001).
  - 8) Brenner, U., Struzyna, M. and Vygen, J.: BonnPlace: Placement of Leading-Edge Chips by Advanced Combinatorial Algorithms, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol.27, No.9, pp.1607–1620 (2008).
  - 9) Brenner, U. and Vygen, J.: Faster Optimal Single-Row Placement with Fixed Ordering, *Proc. IEEE/ACM Design, Automation and Test in Europe Conference*, Paris, France, pp.117–121 (2000).
  - 10) Brenner, U. and Struzyna, M.: Faster and better global placement by a new transportation algorithm, *Proc. ACM/IEEE Design Automation Conference*, pp.591–596 (2005).
  - 11) Chan, T., Cong, J. and Sze, K.: Multilevel Generalized Force-directed Method for Circuit Placement, *Proc. ACM International Symposium on Physical Design*, San Francisco, CA, pp.185–192 (2005).
  - 12) Chan, T., Cong, J., Shinnerl, J., Sze, K. and Xie, M.: mPL6: Enhanced Multilevel Mixed-size Placement, *Proc. ACM International Symposium on Physical Design*, San Jose, CA, pp.212–214 (2006).
  - 13) Chen, B. and Harker, P.T.: A non-interior-point continuation method for linear complementarity problems, *SIAM Journal on Matrix Analysis and Applications*, Vol.14, pp.1168–1190 (1993).
  - 14) Chen, H.-C., Chuang, Y.-L., Chang, Y.-W. and Chang, Y.-C.: Constraint graph-based macro placement for mixed-size circuit designs, *Proc. IEEE/ACM International Conference on Computer-Aided Design*, San Jose, CA, pp.218–223 (2008).
  - 15) Chen, T.-C., Yuh, P.-H., Chang, Y.-W., Liu, F.-J. and Liu, D.: MP-trees: A packing-based macro placement algorithm for modern mixed-size designs, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol.27, No.9, pp.1621–1634 (2008).
  - 16) Chen, T.-C., Hsu, T.-C., Jiang, Z.-W. and Chang, Y.-W.: NTUplace: A ratio partitioning based placement algorithm for large-scale mixed-size designs, *Proc. ACM International Symposium on Physical Design*, San Francisco, CA, pp.236–238 (2005).
  - 17) Chen, T.-C., Jiang, Z.-W., Hsu, T.-C. and Chang, Y.-W.: A High-Quality Mixed-Size Analytical Placer Considering Preplaced Blocks and Density Constraints, *Proc. IEEE/ACM International Conference on Computer-Aided Design*, San Jose, CA (2006).
  - 18) Chen, T.-C., Jiang, Z.-W., Hsu, T.-C., Chen, H.-C. and Chang, Y.-W.: NTUplace3: An analytical placer for large-scale mixed-size designs with preplaced blocks and density constraints, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol.27, No.7, pp.1228–1240 (2008).
  - 19) Cheon, Y., Ho, P.-H., Kahng, A.B., Reda, S. and Wang, Q.: Power-Aware Placement, *Proc. ACM/IEEE Design Automation Conference*, pp.795–800 (2005).
  - 20) Fiduccia, C.M. and Mattheyses, R.M.: A linear-time heuristic for improving network partitions, *Proc. ACM/IEEE Design Automation Conference*, pp.175–181 (1982).
  - 21) Hamada, T., Cheng, C.K. and Chau, P.M.: Prime: A Placement Tool Using a Piece Wise Linear Resistive Network Approach, *Proc. ACM/IEEE Design Automation Conference*, pp.531–536 (1993).
  - 22) Hill, D.: US patent 6,370,673: Method and system for high speed detailed placement of cells within an intergrated circuit design (2002).
  - 23) Hu, B., Zeng, Y. and Marek-Sadowska, M.: mFAR: fixed-points-addition-based VLSI placement algorithm, *Proc. ACM International Symposium on Physical Design*, San Francisco, CA, pp.239–241 (2005).
  - 24) Huang, D.J.-H. and Kahng, A.B.: Partitioning-based standard-cell global placement with an exact objective, *Proc. ACM International Symposium on Physical Design*, pp.18–25 (1997).
  - 25) Jackson, M. and Kuh, E.S.: Performance-Driven Placement of Cell based IC's, *Proc. ACM/IEEE Design Automation Conference*, pp.370–375 (1989).
  - 26) Jiang, Z.-W., Su, B.-Y. and Chang, Y.-W.: Routability-driven analytical placement by net overlapping removal for large-scale mixed-size designs, *Proc. ACM/IEEE Design Automation Conference*, Anaheim, CA, pp.167–172 (2008).
  - 27) Jonker, R. and Volgenant, A.: A shortest augmenting path algorithm for dense and sparse linear assignment problems, *Computing*, Vol.38, No.4, pp.325–340 (1987).
  - 28) Kahng, A.B., Kang, S.-M., Li, W. and Liu, B.: Analytical Thermal Placement for VLSI Lifetime Improvement and Minimum Performance Variation, *Proc. IEEE International Conference on Computer Design*, Glasgow, Scotland, pp.71–77 (2007).
  - 29) Kahng, A.B., Reda, S. and Wang, Q.: Architecture and Details of a High Quality, Large-Scale Analytical Placer, *Proc. IEEE/ACM International Conference on Computer-Aided Design*, San Jose, CA, pp.890–897 (2005).
  - 30) Kahng, A.B., Tucker, P. and Zelikovsky, A.: Optimization of Linear Placements for Wirelength Minimization with Free Sites, *Proc. IEEE/ACM Asia South Pacific Design Automation Conference*, pp.241–244 (1999).
  - 31) Kahng, A.B. and Wang, Q.: An Analytic Placer for Mixed-Size Placement and Timing-Driven Placement, *Proc. IEEE/ACM International Conference on Computer-Aided Design*, San Jose, CA, pp.565–572 (2004).
  - 32) Kahng, A.B. and Wang, Q.: Implementation and extensibility of an analytic placer, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol.24, No.5 (2005).
  - 33) Kahng, A.B. and Wang, Q.: A Faster Implementation of APlace, *Proc. ACM International Symposium on Physical Design*, San Jose, CA, pp.218–220 (2006).

- 34) Kanzows, C.: Some tools allowing interior-point methods to become noninterior, *Technical Report, Institute of Applied Mathematics, University of Hamburg, Germany* (1994).
- 35) Kleinhans, M., Sigl, G., Johannes, F.M. and Antreich, K.J.: GORDIAN: VLSI placement by quadratic programming and slicing optimization, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol.10, No.3, pp.356–365 (1991).
- 36) Lee, W.-P., Liu, H.-Y. and Chang, Y.-W.: Voltage Island Aware Floorplanning for Power and Timing Optimization, *Proc. IEEE/ACM International Conference on Computer-Aided Design*, pp.389–394 (2006).
- 37) Li, C. and Koh, C.-K.: Recursive Function Smoothing of Half-Perimeter Wirelength for Analytical Placement, *Proc. IEEE/ACM International Symposium on Quality of Electronic Design*, San Jose, CA, pp.26–28 (2007).
- 38) Li, C., Xie, M., Koh, C.-K., Cong, J. and Madden, P.H.: Routability-Driven Placement and White Space Allocation, *Proc. IEEE/ACM International Conference on Computer-Aided Design*, San Jose, CA, pp.394–401 (2004).
- 39) Luo, T. and Pan, D.Z.: DPlace2.0: A stable and efficient analytical placement based on diffusion, *Proc. IEEE/ACM Asia South Pacific Design Automation Conference*, pp.346–351 (2008).
- 40) Nam, G.-J., Reda, S., Alpert, C.J., Villarrubia, P.G. and Kahng, A.B.: A fast hierarchical quadratic placement algorithm, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol.25, No.4, pp.678–691 (2006).
- 41) Naylor, W.C., Donnelly, R. and Sha, L.: US patent 6,301,693: Non-Linear Optimization System and Method for Wire Length and Delay Optimization for an Automatic Electric Circuit Placer (2001).
- 42) Pan, M., Viswanathan, N. and Chu, C.: An Efficient and Effective Detailed Placement Algorithm, *Proc. IEEE/ACM International Conference on Computer-Aided Design*, pp.48–55 (2005).
- 43) Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P.: *Numerical Recipes in C++*, Cambridge University Press, 2nd edition (2002).
- 44) Ren, H., Pan, D.Z., Alpert, C.J., Villarrubia, P.G. and Nam, G.-J.: Diffusion-Based Placement Migration With Application on Legalization, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol.26, No.12, pp.2158–2172 (2007).
- 45) Roy, J., Papa, D., Ng, A. and Markov, I.: Satisfying Whitespace Requirements in Top-down Placement, *Proc. ACM International Symposium on Physical Design*, San Jose, CA, pp.206–208 (2006).
- 46) Sechen, C. and Sangiovanni-Vincentelli, A.: The TimberWolf Placement and Routing Package, *IEEE Journal of Solid-State Circuits*, Vol.SC-20, No.2, pp.510–522 (1985).
- 47) Sigl, G., Doll, K. and Johannes, F.M.: Analytical placement: A linear or a quadratic objective function?, *Proc. ACM/IEEE Design Automation Conference*, pp.427–432 (1991).
- 48) Smale, S.: Algorithms for solving equations, *Proc. Int. Congress of Mathematicians*, pp.172–195 (1987).
- 49) Spindler, P. and Johannes, F.M.: Fast and Accurate Routing Demand Estimation for Efficient Routability-driven Placement, *Proc. IEEE/ACM Design, Automation and Test in Europe Conference*, Acropolis, Nice, France, pp.1–6 (2007).
- 50) Spindler, P., Schlichtmann, U. and Johannes, F.M.: Abacus: Fast Legalization of Standard Cell Circuits with Minimal Movement, *Proc. ACM International Symposium on Physical Design*, Portland, OR, pp.47–53 (2008).
- 51) Spindler, P., Schlichtmann, U. and Johannes, F.M.: Kraftwerk2 — A Fast Force-Directed Quadratic Placement Approach Using an Accurate Net Model, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol.27, No.8, pp.1398–1411 (2008).
- 52) Srinivasan, A., Chaudhary, K. and Kuh, E.S.: RITUAL: Performance Driven Placement Algorithm for Small Cell ICs, *Proc. IEEE/ACM International Conference on Computer-Aided Design*, San Jose, CA, pp.48–51 (1991).
- 53) Swartz, W. and Sechen, C.: Timing Driven Placement for Large Standard Cell Circuits, *Proc. ACM/IEEE Design Automation Conference*, pp.211–215 (1995).
- 54) Taghavi, T., Yang, X. and Choi, B.-K.: Dragon2005: Large-Scale Mixed-size Placement Tool, *Proc. ACM International Symposium on Physical Design*, San Francisco, CA, pp.245–247 (2005).
- 55) Tsota, K., Koh, C.-K. and Balakrishnan, V.: Guiding global placement with wire density, *Proc. IEEE/ACM International Conference on Computer-Aided Design*, San Jose, CA, pp.212–217 (2008).
- 56) Usami, K. and Horowitz, M.: Clustered Voltage Scaling Technique for Low-Power Design, *Proc. IEEE/ACM International Symposium on Quality of Electronic Design*, pp.3–8 (1995).
- 57) Viswanathan, N. and Chu, C.C.-N.: Fastplace: Efficient analytical placement using cell shifting, iterative local refinement and a hybrid net model, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol.24, No.5, pp.722–733 (2005).
- 58) Viswanathan, N., Nam, G.-J., Alpert, C., Villarrubia, P., Ren, H. and Chu, C.: RQL: Global Placement via Relaxed Quadratic Spreading and Linearization, *Proc. ACM/IEEE Design Automation Conference*, pp.453–458 (2007).
- 59) Viswanathan, N., Pan, M. and Chu, C.: FastPlace 3.0: A Fast Multilevel Quadratic Placement Algorithm with Placement Congestion Control, *Proc. IEEE/ACM Asia South Pacific Design Automation Conference*, pp.135–140 (2007).
- 60) Vorwerk, K., Kennings, A. and Vannelli, A.: Engineering Details of a Stable Force-Directed Placer, *Proc. IEEE/ACM International Conference on Computer-Aided Design*, San Jose, CA (2004).

- 61) Vygen, J.: Algorithms for detailed placement of standard cells, *Proc. IEEE/ACM Design, Automation and Test in Europe Conference*, Paris, France, pp.321–324 (1998).
- 62) Vygen, J.: Algorithms for large-scale flat placement, *Proc. ACM/IEEE Design Automation Conference*, pp.746–751 (1997).
- 63) Vygen, J.: Platzierung im VLSI-Design und ein zweidimensionales Zerlegungsproblem, PhD Thesis, University of Bonn (1997).
- 64) Yang, X., Choi, B.-K. and Sarrafzadeh, M.: Routability-Driven White Space Allocation for Fixed-Die Standard-Cell Placement, *Proc. ACM International Symposium on Physical Design*, Del Mar, CA, pp.42–47 (2002).
- 65) Yao, B., Chen, H., Cheng, C.-K., Chou, N.-C., Liu, L.-T. and Suaris, P.: Unified Quadratic Programming Approach for Mixed Mode Placement, *Proc. ACM International Symposium on Physical Design*, San Francisco, CA, pp.193–199 (2005).

(Received March 3, 2009)

(Released August 14, 2009)

(Invited by Editor-in-Chief: *Hidetoshi Onodera*)



**Yao-Wen Chang** received the B.S. degree from National Taiwan University (NTU), Taipei, Taiwan, in 1988, and the M.S. and Ph.D. degrees from the University of Texas at Austin in 1993 and 1996, respectively, all in computer science. He is a Professor in the Department of Electrical Engineering and the Graduate Institute of Electronics Engineering, NTU. He is currently also a Visiting Professor at Waseda University, Kitakyushu, Japan. He was with National Chiao Tung University (NCTU), Hsinchu, Taiwan from 1996 to 2001 and IBM T.J. Watson Research Center in the summer of 1994. His current research interests lie in VLSI physical design, design for manufacturability/reliability, and design automation for biochips. He has been working closely with industry in these areas. He has co-edited one textbook on EDA and coauthored one book on routing and over 150 ACM/IEEE conference/journal papers in these areas. Dr. Chang was a winner of the 2009 ACM ISPD Clock Network Synthesis Contest, the 2008 ACM ISPD Global Routing Contest, and the 2006 ACM ISPD Placement Contest. He was a recipient of Best Paper Awards at the 1995 IEEE ICCD and the 2007 and 2008 VLSI Design/CAD Symposia and 12 Best Paper Award Nominations from DAC (four times), ICCAD (twice), ISPD (three times), ACM TO-DAES, ASP-DAC, and ICCD in the past eight years. He has received many research awards, such as the 2007 Outstanding Research Award, the inaugural 2005 First-Class Principal Investigator Award, and the 2004 Dr. Wu Ta You Memorial Award, all from National Science Council of Taiwan, and the 2004 MXIC Young Chair Professorship from the MXIC Corp, and excellent teaching awards from NTU (five times) and NCTU. He is currently an associate editor of IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD) and an editor of the Journal of Information Science and Engineering (JISE) and the Journal of Electrical and Computer Engineering (JECE). He has served on the ICCAD Executive Committee, the ASP-DAC Steering Committee, the ACM/SIGDA Physical Design Technical Committee, the ACM ISPD and IEEE FPT Organizing Committees, and the Technical Program Committees of ASP-DAC, DAC, DATE, FPL, FPT, GLSVLSI, ICCAD, ICCD, IECON, ISPD, SOCC, TENCON, and VLSI-DAT. He is currently an independent board director of Genesys Logic, Inc, a technical consultant of RealTek Semiconductor Corp., and a member of board of governors of Taiwan IC Design Society.



**Zhe-Wei Jiang** received the B.S. degree in electronics engineering from the National Chiao Tung University, Hsinchu, Taiwan, in 2003. He is currently working toward the Ph.D. degree at the Graduate Institute of Electronics Engineering, National Taiwan University, Taiwan. His current research interests focus on large-scale mixed-size placement and design for manufacturability. He has received an Outstanding Research Award from GIEE, NTU in 2008, the 1st place at the ACM/SIGDA CADathlon ICCAD Contest in 2007, and the 3rd place at the ACM ISPD Placement Contest in 2006.



**Tung-Chieh Chen** received the B.S. degree in electrical engineering and the Ph.D. degree in electronics engineering from National Taiwan University, Taipei, Taiwan, in 2003 and 2008, respectively. He is currently an engineer with SpringSoft Inc., Hsinchu, Taiwan. He was a visiting scholar with the University of Texas, Austin, in 2007. He has coauthored four book chapters, seven IEEE journal papers, and 11 ACM/IEEE conference papers, all on floorplanning and placement. He has received the Best Dissertation Award from Graduate Institute of Electronic Engineering (GIEE), National Taiwan University (NTU) in 2008, the Outstanding Research Award from GIEE, NTU in 2007 and 2008, the 1st place in ACM SIGDA CADathlon in 2007, and the 3rd place in ACM ISPD Placement Contest in 2006.