Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

An Application of Multi-parametric Programming in Integrated Circuit Automation (Slotting Problem)

Juraj Števek * **Michal Kvasnica** * **Miroslav Fikar** * **Andrej Vrbický** **

* *Slovak University of Technology in Bratislava, Slovakia, (e-mail: {juraj.stevek, michal.kvasnica, miroslav.fikar}@stuba.sk).*
** *ON Semiconductor Bratislava Developing Center, Slovakia, (e-mail: andrej.vrbicky@onsemi.com)*

Abstract: In this paper we address the slotting problem for metal interconnections in integrated circuit automation design for high current conducting metal layers. We show how to design a slotting approach by means of multi-parametric programming. We present a definition of optimization problem for convex shapes of metal layers. Constraints of the problem will reflect restrictions of the production technology and a solution will avoid usage of a commercial optimization solver on a user's side. We will present the approach on an application for generating a power transistor. The approach is applied for specific very-large-scale integration (VLSI) production technology. The solution exhibits simple implementation and gives results in an acceptable response time.

Keywords: slotting, IC automation, multi-parametric programming.

1. INTRODUCTION

In area of integrated circuit (IC) design we distinguish between two different approaches. In the first case ICs are designed by hand, and manually laid out. In the second case a software tool generates desired electronic system while respecting all limitations of a production technology. In this paper we address the latter approach where electronic design automation (EDA) software tools are exploited. A transistor belongs to standard components of the design. In practice, engineers do not design each layer of transistor from scratch but they use a pregenerated transistor from a library of components. The designer just inputs desired parameters of the transistor and an application generates all layers of the component. Such transistor consists of many layers of silicon with various parameters and some metal layers interconnecting parts of the transistor. The transistor application has to respect all limitations of the technology. As mentioned in Kahng et al. (1998), the production technology consists of many manufacturing steps – optical exposure, resist development, electrochemical deposition and chemical-mechanical planarization (CMP). All have different influences on local attributes of the layout. To keep these influences uniform and determinable the layout must posses some kind of uniformity. It is possible to attain this uniformity by inserting (*filling*) or removing (*slotting*) shapes in the layout. Whereas empty surfaces have to be *filled*, very wide shapes have to be *slotted* to prevent defects during the production. From now on the slot will denote a rectangular deletion in top-layer metal shapes.

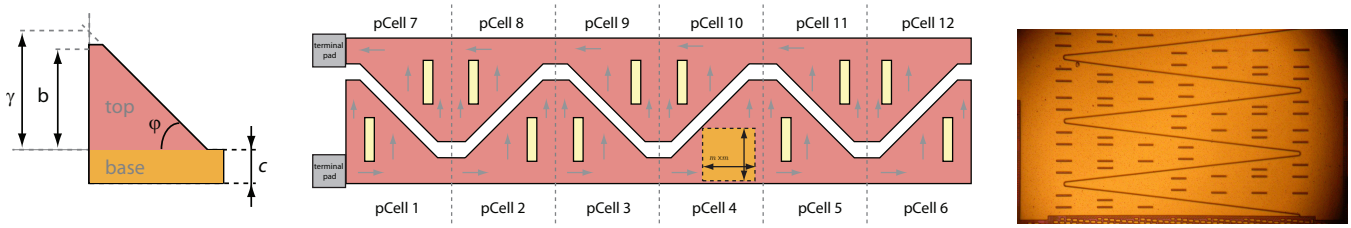
⋆ The authors would like to acknowledge the financial support of the Scientific Grant Agency of the Slovak Republic under the grant 1/0095/11, the contribution of the Slovak Research and Development Agency under the project APVV 0551-11 and the internal grant of the Slovak University of Technology in Bratislava for support of young researchers.

The filling and slotting were first formulated in Kahng et al. (1998). The authors offered a heuristic approach for filling and slotting of rectangular shapes. More detailed study on fill synthesis is given in Kahng et al. (2008) and in Dhumane and Kundu (2012). The high current conducting metal layer (top-layer metal) is mostly designed as a piecewise linear path of certain width. If the path exceeds an allowable width it is needed to insert slots into the path. This problem is usually solved using a wire group or a sloth path (Cadence, 2013; SkillCAD, 2012). If the metal layer takes a different shape than the path then the slotting problem is closely related to rectangle packing problem (Huang and Chen, 2007; Birgin and Lobato, 2010) and floor planning problem (Sherwani, 1999). Mathematical programs that solve latter problems are complex and computationally demanding in general. Results of these programs are given in a response time that can vary from several seconds to tens of minutes depending on a complexity of the problem. However, an acceptable response time of a operation in Computer Aided Design (CAD) systems ranges from miliseconds to several seconds.

A challenge is to design a slotting approach that offers solutions in the acceptable response time while satisfying all technology restrictions. Moreover the approach has to respect limits of an implementation scripting language. In terms of a budget it is also preferred to avoid usage of a commercial optimization solver on the user's side.

In this paper we address the slotting problem in IC automation design for metal shapes. The objective is to find an optimal solution in form of an analytical function $z^*(p)$ where p represents parameters of the problem and z^* represents vector of optimal variables. Then calculating $z^*(p)$ for a given value of p reduces to a simple function evaluation.

We show how to design slotting by means of multi-parametric programming (Bank, 1982; Pistikopoulos et al., 2007b,a). We



(a) The top-layer metal *pCell* geometry.

(b) A ridge composition of *pcell*s. Yellow bars represents slots. Orange square represents forbidden area. Grey arrows roughly represent electric current flow.

(c) Top-layer metal sample of power transistor from a testchip.

Fig. 1. Geometry of the basic building block and its composition into metal shapes.

present a definition of an optimization problem for convex shapes of top-layer metals. The constraints of the problem will reflect restrictions of the production technology and the solution will respect the commercial limitation. We will present the approach on an application for generating a power transistor.

The paper is organized as follows. In Section 2, we present geometry and basic rules to compose shape of the top-layer metals of the power transistor. In Section 3 we expose model of the metal shape. We introduce the notations to be used throughout the paper and propose general form of the optimization problem. In Section 4 we detail the solution and discuss important issues. Finally, numerical results are presented and discussed in Section 5, while we draw some conclusions in Section 6.

2. PROBLEM DESCRIPTION

In this section we present basic rules of the slotting problem for the application for generating a power transistor. We touch on important geometry regarding metal shape creation, implementation issues, and limitations of the production process.

The transistor application uses simple parameterized building block (parameterized cell or *pCell*) to generate whole transistor. Figs. 1a, 1b depict top-layer geometry of the *pCell* and a composition of *pCell*s into top-layer metal shapes. Grey arrows represent direction of the electric current. Although, parameterized cell describes many features of the layout in general, *pCell* will refer to the top-layer metal part of the parameterized cell in this paper. Parameters of the *pCell* take values:

$$c \geq c_{\min},\ b \in [b_{\min}, b_{\max}], \gamma \in [\gamma_{\min}, \gamma_{\max}],\ \varphi \in [\varphi_{\min}, \varphi_{\max}] \quad (1)$$

The application for generating the power transistor is prepared in custom IC design scripting language SKILL. SKILL is a scripting language based on Lisp exploited in many EDA software packages from Cadence Design Systems (Barnes, 1990). It is needed to note that SKILL offers only basic procedures for automated IC design and does not offer mathematical optimization solvers, neither allows to link the open source or commercial libraries. An easy implementation of the slotting program is the key restriction in SKILL programming environment.

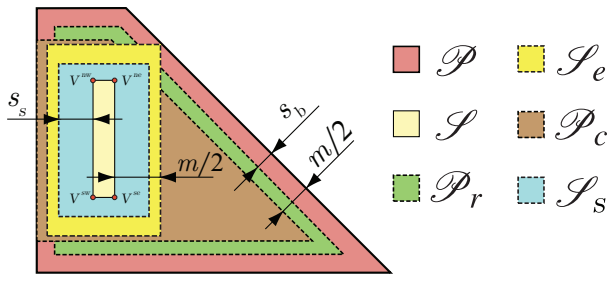


Fig. 2. Single tooth of the ridge composition.

The slotting restriction for a production technology can be interpreted as follows. Each metal shape has to be designed so that no square bigger than $m \times m$ can be contained in the shape of the metal layer, as illustrated in Fig. 1b. However this definition is inexact, therefore it is needed to introduce a design rule check (DRC) principle for wide metal shapes. First we divide the *pCell* into two convex parts: top and base, as shown in Fig. 1a. Since base parts form a simple rectangular supply conductor in the ridge composition we focus only on the top part of the *pCell*, Fig. 2. The yellow area around the slot is created by extending the slot by $m/2$. Similarly the brown area is created by moving selected facets of the shape inward by $m/2$. If the yellow area fully covers the brown area then the technology restriction is satisfied. The technology imposes restriction for the size and position of slots too. Each slot is rectangle of length ℓ and width w with following limitation

$$w_{\min} \leq w \leq w_{\max}, \quad \ell_{\min} \leq \ell \leq \ell_{\max} \quad (2)$$

A spacing between slots in horizontal and vertical directions has to be more than s_s. Distance of any slot from boundary of the metal shape has to be more than s_b.

3. MINIMIZATION PROBLEM

Since the metal layer will conduct the electric current of several amperes it is desired to minimize the area of all slots. If the area of remaining metal shape would be too small the electric current could burn out the metal. Before we present our solution and discuss implementation issues, let us summarize our goal:

Problem 3.1. Given parameters b, γ, φ of the top part of the *pCell*, as depicted in Fig. 1a, it is needed to minimize overall slots area while satisfying the slotting restriction. The optimized variables are:

- number of slots (R),
- position of center for each slot $(C_{x_1 i}, C_{x_2 i})$,
- size of each slot (ℓ_i, w_i).

where $i = 1, \ldots, R$ denotes the slot's index.

Note that slot denotes rectangular deletion in the metal layer shape. The problem has to be solved in the acceptable response time while respecting all restrictions of the production technology.

3.1 Model of pCell's Top Part

From now on we will consider the shape of the *pCell*'s top part, as shown in Fig. 3, defined in a hyperspace representation of the polytope

$$\mathscr{P} = \{x \in \mathbb{R}^2 \mid Hx \leq K\}. \quad (3)$$

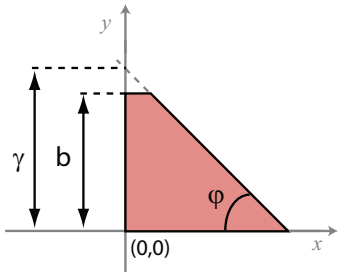


Fig. 3. The model of the tooth (top part of the *pCell*)

where

$$H = \begin{bmatrix} 0 & -1 \\ \alpha & \beta \\ 0 & 1 \\ -1 & 0 \end{bmatrix}, \quad K = \begin{bmatrix} 0 \\ \gamma \\ b \\ 0 \end{bmatrix} \tag{4}$$

where parameters γ, b correspond to geometry depicted in Fig. 3. and $[\alpha, \beta]$ is normalized vector of $[\tan\varphi, 1]$. The slot can be uniquely defined by the set of its vertices $V^{\mathrm{sw}}, V^{\mathrm{se}}, V^{\mathrm{ne}}, V^{\mathrm{nw}}$ (Birgin and Lobato, 2010)

$$V^{\mathrm{sw}} = \begin{bmatrix} C_{x_1} \\ C_{x_2} \end{bmatrix} + 0.5Q(\theta) \begin{bmatrix} -w \\ -\ell \end{bmatrix} \tag{5a}$$

$$V^{\mathrm{se}} = \begin{bmatrix} C_{x_1} \\ C_{x_2} \end{bmatrix} + 0.5Q(\theta) \begin{bmatrix} w \\ -\ell \end{bmatrix} \tag{5b}$$

$$V^{\mathrm{ne}} = \begin{bmatrix} C_{x_1} \\ C_{x_2} \end{bmatrix} + 0.5Q(\theta) \begin{bmatrix} w \\ \ell \end{bmatrix} \tag{5c}$$

$$V^{\mathrm{nw}} = \begin{bmatrix} C_{x_1} \\ C_{x_2} \end{bmatrix} + 0.5Q(\theta) \begin{bmatrix} -w \\ \ell \end{bmatrix} \tag{5d}$$

where $[C_{x_1}, C_{x_2}]^T$ denote coordinates of the slot's center and $Q(\theta)$ denotes the anticlockwise rotation matrix

$$Q(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{6}$$

Then the hyperspace definition of the i-th slot can be expressed as

$$\mathscr{S}_i = Conv(\{V_i^{\mathrm{sw}}, V_i^{\mathrm{se}}, V_i^{\mathrm{ne}}, V_i^{\mathrm{nw}}\}) \tag{7}$$

where *Conv* denotes convex hull operation and i denotes index of the slot. The rotation of each slot should respect the direction of electric current such that slots would not create any unnecessary barrier to an electron flow. If we choose the rotation angle $\theta = 0$, that corresponds to current flow in the top part of the *pCell*, depicted in Fig. 1b, then $Q(\theta) = I$.

Next we state basic notation for shapes illustrated in Fig. 2. When we move each facet of the polytope $\mathscr{P}$ inward by constant s_b, then we consider the polytope that denotes the boarding area for slots

$$\mathscr{P}_r = \{x \in \mathbb{R}^2 \mid Hx \leq K_r\} \tag{8}$$

where H is defined in (4) and K_r is defined as

$$K_r = K - \begin{bmatrix} s_b \\ s_b \\ s_b \\ s_b \end{bmatrix} = \begin{bmatrix} -s_b \\ \gamma - s_b \\ b - s_b \\ -s_b \end{bmatrix} \tag{9}$$

Similarly we define the slot's extended polytopes $\mathscr{S}_e$ by moving each facet of the slot outward by constant $m/2$, yellow shape, the slot's covering polytope $\mathscr{S}_s$ by moving each facet of the slot outward by constant s_s , blue shape and the covered polytope $\mathscr{P}_c$ by moving each facet of the $\mathscr{P}$ inward by constant $m/2$, brown shape, depicted in Fig. 2.

A straightforward way to express the optimization problem 3.1 is

$$\min_{R,z} \sum_{i=1}^{R} S_i \tag{10a}$$

$$\mathscr{S}_i \subseteq \mathscr{P}_r \tag{10b}$$

$$\mathscr{P}_c \subseteq \bigcup_{i=1}^{R} \mathscr{S}_{ei} \tag{10c}$$

$$\mathrm{int}\,\mathscr{S}_i \cap \mathrm{int}\,\mathscr{S}_{sj} = \emptyset, \ \ \forall i \neq j \tag{10d}$$

$$\ell_{\min} \leq \ell_i \leq \ell_{\max} \tag{10e}$$

$$w_{\min} \leq w_i \leq w_{\max} \tag{10f}$$

where $z = [z_1, z_2, ..., z_i, ..., z_R]^T$ is vector of optimized variables where $z_i = [C_{x_1 i}, C_{x_2 i}, \ell_i, w_i]^T$. $S_i = w_i \ell_i$ is area of i-th slot, $\mathscr{S}_i$ denotes i-th slot defined by (7), $\mathscr{S}_{ei}$ denotes i-th extended polytope (the slot with extended facets by $m/2$) and $\mathscr{S}_{si}$ denotes i-th covering polytope (the slot with extended facets by s_s). Note that $\mathscr{P}_r$ is a nonlinear function of *pCell*'s parameters φ, γ, b. R denotes number of slots used in the problem. Since the optimal number of slots is unknown the problem (10) is a nonlinear combinatorial optimization problem in the parameters φ, γ, b. Mathematical programs that solve the problem (10) are complex and computationally demanding in general (Birgin and Lobato, 2010).

The condition (10b) forces slots to lie inside the reduced polytope $\mathscr{P}_r$ such that distance of any slot from boundary of the metal shape is more than s_b. The condition (10c) expresses the DRC principle discussed above. The union of all the slots's extended polytopes has to cover polytope $\mathscr{P}_c$. The constraint for the distance between slots is denoted by (10d). Constraints (10e), (10f) are technology restrictions for slots width and length (2).

Remark 1. It is sufficient to replace the nonlinear function of the rectangle area $w_i \ell_i$ by the linear function of the rectangle perimeter $w_i + \ell_i$ in the objective (10a).

Here we summarize the notation used to this point

$x \in \mathbb{R}^2$	the coordinate vector
$p \in \mathbb{R}^4$	the parameter vector $p = [\alpha, \beta, \gamma, b]^T$,see Fig. 3
$z_i \in \mathbb{R}^4$	optimized variables for i-th slot
$z \in \mathbb{R}^{4R}$	the vector of optimized variables $z = [z_1, ..., z_R]^T$
R	the number of slots in the problem
w, ℓ	width and height of the slot
$V^{(i)}$	vertices of the slot $i \in \{sw, se, nw, ne\}$
$\mathscr{S}$	the halfspace definition of the slot
$\mathscr{P}$	the halfspace definition of the pCell's top-part
H, K	matrices of the halfspace definition of $\mathscr{P}$
s_b	the minimum allowed distance of the slot from boundaries of $\mathscr{P}$
s_s	the minimum allowed distance between slots
$\mathscr{P}_r$	the halfspace definition of the reduced pCell's top-part
$\mathscr{P}_c$	the halfspace definition of the subset of pCell's top-part that needs to be covered
$\mathscr{S}_c$	the halfspace definition of the slot's covering area
$\mathscr{S}_e$	the halfspace definition of the slot's extended area

3.2 Multi-parametric Problem

Multiparametric programming is a method for solving optimization problems with varying parameters (Pistikopoulos

et al., 2007b). An output of the multiparametric program is an analytic function $z^*(p)$ which gives optimal values of an optimized variables as a function of varying parameters p (Bank, 1982). It was shown (Bemporad et al., 2002; Borrelli, 2003) that the solution to the multi-parametric (mixed-integer) linear program is a piecewise affine function defined over polytopic regions in the space of the parameters.

Remark 2. The computational complexity of the multi-parametric program rapidly increases with the number of the constraints in the problem (Borrelli, 2003).

The varying parameters in the slotting problem are parameters α, β, γ, b of the model (3) with matrices (4). However, the constraints that force slots to lie inside $\mathscr{P}_r$ (13b) bring bilinearity into the problem. To overcome this obstacle we propose to quantize φ angle such that the $\alpha = const, \beta = const$ and generate set of parametric solutions for all discrete values of the angle. We can say that the geometry of *pCell* will be "quantized". In spite of the reduction of the range of the angle we are able to proceed without major consequences as it will be shown in the numerical example. The desired geometry is derived from requirements on a power and layout properties. A slight change in the geometry of the *pCell* will not cause any discomfort to the circuit designer.

4. PROPOSED SOLUTION

To solve the problem discussed in Section 3, we propose to use a quantization of the angle $\varphi \in \{\varphi_1, ..., \varphi_i, ..., \varphi_K\}$ such that the $\alpha = const, \beta = const$ for each value of the angle and the optimization problem (10) transforms to mixed-integer linear problem (MILP). We propose to obtain the parametric solution by means of the multi-parametric programming. It was shown in (Borrelli, 2003; Pistikopoulos et al., 2007b) that the solution of the MILP is a piecewise affine function (PWA)

$$z^*(p) = \begin{cases} F_1 p + G_1 & \text{if } p \in \mathscr{R}_1 \\ \vdots & \\ F_L p + G_L & \text{if } p \in \mathscr{R}_L, \end{cases} \tag{11}$$

where $\{\mathscr{R}_j\}_{j=1}^{L}$ is the partition of the polyhedron $\Omega \subseteq \mathbb{R}^4$, $\mathscr{R}_j \subseteq \mathbb{R}^4$ are polyhedra, $F_j \in \mathbb{R}^{4\times 4R}$, $K_j \in \mathbb{R}^{4R}$, $j = 1, \ldots, L$, and $z = [z_1, z_2, ..., z_i, ..., z_R]^T$ is vector of optimized variables where $z_i = [C_{x_1 i}, C_{x_2 i}, \ell_i, w_i]^T$.

The parametric solution for the slotting problem will consist of the family of the piecewise affine (PWA) functions $\{z_1, z_2, ..., z_R\}$ that i-th function will define position and size of i-th slot:

$$y_i = z_i(p) \tag{12}$$

where $y_i = [C_{x_1 i}, C_{x_2 i}, w_i, \ell_i]^T$ and $p = [\alpha, \beta, \gamma, b]^T$ are parameters expressing the "quantized" geometry of *pCell*.

Since every PWA function is associated with a partitioning of the parameter domain into L polyhedral regions $\mathscr{R}_j = \{p \in \mathbb{R}^4 | H_j p \leq K_j\}$, $j = 1, ..., L$, an implementation of the solution for i-th slot will involve only two steps:

(1) Identification of the region $\mathscr{R}_j$ that contains p.
(2) Evaluation of the affine function $F_j^{(i)} p + G_j^{(i)}$.

The first step is often referred to as the *point location* problem (Kvasnica, 2009).

4.1 Single Slot Problem

In this part we formulate optimization problem for obtaining optimal position and size of single slot in the *pCell*'s top part. The slot's volume in the objective function (10) is replaced by the sum of width and length of the slot.

$$\min_{C_{x_1}, C_{x_2}, \ell, w} \quad (w + \ell) \tag{13a}$$

$$HV^{(i)} \leq K_r, \quad \forall i \in \{\text{sw}, \text{se}, \text{ne}, \text{nw}\} \tag{13b}$$

$$V_e^{\text{se}} \geq \begin{bmatrix} K_{c_1} \\ K_{c_2}/\alpha - 7.5/\tan(\varphi) \end{bmatrix} \tag{13c}$$

$$H_{c_3} V_e^{\text{nw}} \geq K_{c_3} \tag{13d}$$

$$\ell_{\min} \leq \ell \leq \ell_{\max} \tag{13e}$$

$$w_{\min} \leq w \leq w_{\max} \tag{13f}$$

where notation H_{c_j} and K_{c_j} represents j-th row of matrices H_c and K_c of the covered polytope $\mathscr{P}_c$. V_e^{se} and V_e^{nw} represent south-east and north-west vertex of the extended polytope $\mathscr{S}_e$, respectively. Note that vertices $V^{(i)}$ (5) of the slot are linear functions of the slot's center $[C_{x_1}, C_{x_2}]^T$, width w and length ℓ. The first set of constraints (13b) forces the slot to lie inside the polytope $\mathscr{P}_r$. Otherwise, constraints (13c), (13d) push the extended polytope $\mathscr{S}_e$ to cover $\mathscr{P}_c$.

For the unique parameter vector $[\alpha, \beta, \gamma, b]^T$ the problem (13) is a linear optimization problem (Boyd and Vandenberghe, 2004) that can be easily formulated using Yalmip (Löfberg, 2004) and solved with a variety of solvers, e.g. (ILOG, 2008).

4.2 Multiple Slots Problem

In the previous section we showed how to formulate the single slot placement problem as a linear program. Now we want to extent the formulation for R slots. A *floor planning problem* (Sherwani, 1999) can be considered an extension of the placement problem (Boyd and Vandenberghe, 2004). The non-overlap constraints (10d) make the floor planning problem a complicated combinatorial optimization problem or a rectangle packing problem (Huang and Chen, 2007; Birgin and Lobato, 2010). However, if the *relative positioning* of the rectangles is

(a) A two slot floor plan.

(b) A three slot floor plan.

Fig. 4. The figure shows two *flooring scenarios* and their horizontal and vertical graphs $\mathscr{H}$ and $\mathscr{V}$ that specify the relative positioning of the slots. If there is a path from node i to node j in $\mathscr{H}$, then slot i has to be placed left of the slot j. If there is a path from node i to node j in $\mathscr{V}$, then slot i has to be placed below slot j.

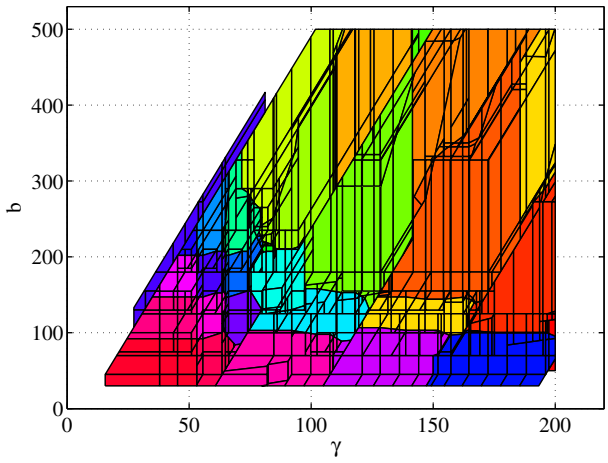


Fig. 5. Domain partitioning of the parametric solution for $\varphi = 1.3802$. Different colors correspond to different flooring scenarios.

specified, it is possible to formulate the problem as a convex optimization problem with minimum number of constraints (Boyd and Vandenberghe, 2004). It is also desired to keep the number of constraints minimal w.r.t. remark 2. We consider a system of minimal number of relative positioning constraints as was defined in Chapter 8.8.1 of Boyd and Vandenberghe (2004). The system uses two graphs (horizontal and vertical graph) that represent slots position relations in horizontal and vertical direction. A simple example is shown in Fig. 4. We will refer to single floor plan represented by unique pair of graphs $\mathscr{H}$ and $\mathscr{V}$ as a *flooring scenario* and its model expressed as

$$A_j z \leq B_j \; ; \; j = 1,...,N \tag{14}$$

where A_j, B_j are matrices representing relative positioning of j-th scenario including (10e), (10f) and $z = [z_1, z_2, ..., z_i, ..., z_{R_j}]^T$ is vector of optimized variables where $z_i = [C_{x_1 i}, C_{x_2 i}, \ell_i, w_i]^T$.

Fig. 4 shows two different scenarios. Each scenario is defined by minimal set of constraints that create the placement model. For example, the model for the single slot scenario is formed by constraints (13b), (13c), (13d). For N different scenarios we define mixed-integer linear optimization problem

$$\min_{z,\delta} \; \sum_{i=1}^{R} (w_i + \ell_i) \tag{15a}$$

$$(\delta_j = 1) \Rightarrow \; (A_j z \leq B_j), \;\; \forall j = 1,...,N \tag{15b}$$

$$\sum_{j=1}^{N} \delta_j = 1 \tag{15c}$$

where $z = [z_1, z_2, ..., z_i, ..., z_{R_j}]^T, z_i = [C_{x_1 i}, C_{x_2 i}, \ell_i, w_i]^T$, $\delta = [\delta_1, ..., \delta_N]^N, \delta_j \in \{0,1\}, \forall j = 1,...,N$. The IF-THEN logic statement in (15b) needs to be converted into an equivalent mathematical representation. It can be achieved by exploiting the big-M technique suggested in Williams (1999).

5. NUMERICAL RESULTS

In this section we present numerical results for the optimization problem defined in previous section and implementation issues. Since the proposed heuristic approach for the base part of *pCell* is quite straightforward we will discuss only results for the top part of *pCell*.

The parameters for a mosfet technology were applied when $w_{\min} = 2, w_{\max} = 10, \ell_{\min} = 20, \ell_{\max} = 250, s_s = 10, s_b = 10, m = 35$ (μm). The optimization problem (15) was defined in Matlab environment using Yalmip and Multi-parametric toolbox (Löfberg, 2004; Kvasnica et al., 2004) as the parametric mixed integer linear program. Twelve slots were used in 35 flooring scenarios and introduced into the problem. The program was evaluated for sixty discrete values of φ angle in the range $[0.3; \pi/2]$ and set of the parametric solutions was generated. Each set contains twelve PWA slot functions (12). The parametric solution uses a subset of flooring scenarios. In Fig. 5 a domain $[\gamma, b]$ of the parametric solution for $\varphi = 1.3802$ rad is depicted. The solution uses 27 of 35 flooring scenarios. Different colors represent different scenarios in Fig. 5.

Table 1 shows numerical properties of the parametric program for ten values of angle φ. Columns of the table denote, respectively, value of φ, the total number of regions L of the parametric solution, the number of used scenarios (maximum is 35) in the solution, the off-line calculation time of the parametric solution (in seconds) and the average evaluation time of the parametric solution (in miliseconds). Note that the off-line calculation times correspond to generating the parametric solution and it is performed only once. The user evaluates the solution in miliseconds with no need of commercial optimization solvers.

Table 1. Results for ten values of angle φ.

φ (rad)	L	Used scenarios	Off-line calculation (s)	Average evaluation time (ms)
π	280	14	1222	< 10
1.5496	838	24	1370	< 10
2.5284	914	25	2172	< 10
1.5073	919	24	2114	< 10
1.4861	917	25	2157	< 10
1.4649	967	27	2230	< 10
1.4437	1020	27	2075	< 10
1.4225	1123	27	2086	< 10
1.4014	1117	27	2129	< 10
1.3802	1184	27	2084	< 10
Binary variables = 35x1, Real variables = 48x1				

Computation was performed on a 2.5 GHz Core i5 CPU with 4GB of RAM using MATLAB 7.8 and MPT 2.6.3.

5.1 Implementation

Each slot function returns the optimal values of $[C_{x_1}, C_{x_2}, \ell, w]^T$ which denote, respectively, x_1, x_2 coordinates of the slot's center, slot's width and slot's length. The parametric solution for single value of φ contains twelve slot functions defined over L polyhedral regions. If we consider sixty values of φ angle then we can express memory requirements as

$$\sum_{j=1}^{60} 48 L_j \tag{16}$$

cells of a table where each cell contains parameters F_j, G_j of the parametric solution (11). In spite of, the memory requirement (16) can be a huge number $> 10^6$, the evaluation of the solution will mostly depend on the number of regions L_j and effectiveness of the *point location* algorithm. The *point location* can be implemented by checking whether $p \in \mathscr{R}_j$ holds sequentially for $j = 1,...,L_j$, which has a runtime complexity of $\mathscr{O}(L_j)$ (Kvasnica, 2009) or exploiting binary search tree (Tondel et al., 2003) which has a logarithmic $\mathscr{O}(\log L_j)$ runtime complexity.

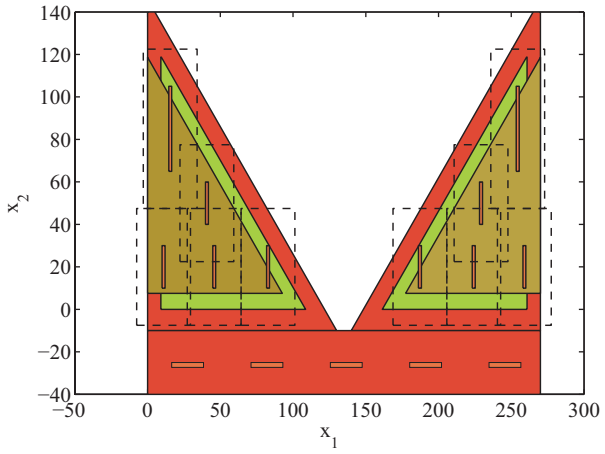


Fig. 6. Slotting for composition of two *pCell*s. Green area denotes $\mathscr{P}_r$, dashed areas denote $\mathscr{S}_e$, and brown area denotes $\mathscr{P}_c$.

5.2 Example

At the end, a circuit designer inputs desired parameters of the power transistor (power, amperage, layout orientation) and the application creates all layers of the component. Fig. 6 shows composition of two *pCell*s and its slotting for desired parameters $\varphi = 0.8761, b = 150, \gamma = 156$. The "quantized" *pCell* differs from desired one in $\Delta\varphi = 0.0042 rad$ that cause no discomfort for the designer. The sequential search was applied for evaluating the parametric solution and evaluating times range in fractions of seconds.

6. CONCLUSIONS AND FUTURE WORK

The presented approach solves the slotting problem in a way which is new in the area. We formulate the problem as a convex programming problem and obtain the solution by means of the parametric programming. The result is stored as a set of piecewise affine functions that are easy to evaluate in short times (miliseconds). The usage of any commercial optimization solvers is avoided on the user's side. We demonstrated the approach on the problem of automatic generation of the power transistor in specific VLSI technology.

Our future work will focus on an optimization of memory requirements of the parametric solution. Since each analytical slot function (12) consists of piecewise continuous parts, the approach based on min-max algebraic functions (Wen et al., 2009) can be appealing.

The optimal solution should use flooring scenario with slots size close to the minimum slot size ($w_{\min} \times \ell_{\max}$). To achieve this property it is needed to introduce all possible scenarios into the parametric problem. An automatic generation of *flooring models* will make the usage of the proposed methodology much simpler.

REFERENCES

Bank, B. (1982). *Non-linear parametric optimization*. Mathematische Lehrbücher und Monographien: Mathematische Monographien. Birkhäuser Verlag.

Barnes, T.J. (1990). SKILL: a CAD system extension language. In *Proceedings of the 27th ACM/IEEE Design Automation Conference*, 266–271. doi:10.1109/DAC.1990.114865.

Bemporad, A., Morari, M., Dua, V., and Pistikopoulos, E.N. (2002). The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(2), 3–20.

Birgin, E.G. and Lobato, R.D. (2010). Orthogonal packing of identical rectangles within isotropic convex regions. *Computer*, 59(4), 595–602. doi:10.1016/j.cie.2010.07.004.

Borrelli, F. (2003). *Constrained Optimal Control of Linear and Hybrid Systems*, volume 50. Springer. doi: 10.1109/TAC.2005.851468.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press, New York, NY, USA.

Cadence (2013). Cadence Design System. URL `www.cadence.com`.

Dhumane, N. and Kundu, S. (2012). Critical area driven dummy fill insertion to improve manufacturing yield. In *13th International Symposium on Quality Electronic Design (ISQED)*, 334–341. doi:10.1109/ISQED.2012.6187514.

Huang, W. and Chen, D. (2007). An efficient heuristic algorithm for rectangle-packing problem. *Simulation Modelling Practice and Theory*, 15(10), 1356–1365. doi: 10.1016/j.simpat.2007.09.004.

ILOG, I. (2008). ILOG CPLEX: High-performance software for mathematical programming and optimization. URL `http://www.ilog.com/products/cplex/`.

Kahng, A.B., Member, S., Samadi, K., and Member, S. (2008). CMP Fill Synthesis: A Survey of Recent Studies. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 27(1), 3–19. doi: 10.1109/TCAD.2007.907061.

Kahng, A.B., Robins, G., Singh, A., Wang, H., and Zelikovsky, A. (1998). Filling and slotting: analysis and algorithms. In *ISPD*, 95–102.

Kvasnica, M., Grieder, P., and Baoticc, M. (2004). Multi-Parametric Toolbox (MPT). URL `http://control.ee.ethz.ch/ mpt/`.

Kvasnica, M. (2009). *Real-Time Model Predictive Control via Multi-Parametric Programming: Theory and Tools*. VDM Verlag, Saarbruecken.

Löfberg, J. (2004). YALMIP : A Toolbox for Modeling and Optimization in MATLAB. In *Proceedings of the CACSD Conference*, 284–289. IEEE, IEEE. doi: 10.1109/CACSD.2004.1393890.

Pistikopoulos, E., Galindo, A., and Dua, V. (2007a). *Process Systems Engineering, Volume 2, Multi-Parametric Model-Based Control*. Process Systems Engineering. Wiley.

Pistikopoulos, E., Georgiadis, M., Dua, V., Papageorgiou, L., and Kikkinides, E.S. (2007b). *Process Systems Engineering: Volume 1: Multi-Parametric Programming*. Wiley, London, UK.

Sherwani, N.A. (1999). *Algorithms for VLSI physical design automation*. Kluwer Academic Publishers.

SkillCAD (2012). SkillCAD - Suite. URL `www.skillcad.com`.

Tondel, P., Johansen, T.A., and Bemporad, A. (2003). Evaluation of Piecewise Affine Control via Binary Search Tree. *Automatica*, 39(5), 945–950. doi:10.1016/S0005-1098(02)00308-4.

Wen, C., Ma, X., and Ydstie, B.E. (2009). Analytical expression of explicit MPC solution via lattice piecewise-affine function. *Automatica*, 45(4), 910–917.

Williams, H.P. (1999). *Model Building in Mathematical Programming*, volume 4 of *A Wiley - Interscience Publication*. John Wiley & Sons, 3rd edition.