Layout Decomposition and Printing Time Optimization for Inkjet-Printed Electronics

YI-TING LIN, National Taiwan University, Taiwan
MENG LIAN, Technical University of Munich, Germany
HU PENG, Technical University of Munich, Germany
BERNHARD WOLFRUM, Technical University of Munich, Germany
TSUN-MING TSENG, Technical University of Munich, Germany
IRIS HUI-RU JIANG, National Taiwan University, Taiwan

Inkjet-printed electronics is a low-cost option for large-scale production. To avoid manufacturing defects, recent research has considered design constraints, such as Laplace and proximity conflicts, decomposed the layouts into different layers, and printed them sequentially. The state-of-the-art work reduced the manufacturing time by optimizing the number of layers and drying time. In this work, we aim to enhance manufacturing efficiency from a new angle, concurrently optimizing the printing time and the layout decomposition of inkjet-printed electronics. We propose an integer linear programming formulation and a dynamic programming algorithm to determine layout decomposition and layer assignment and to estimate the total printing time by carefully considering printing characteristics and design constraints. Experimental results demonstrate significant reductions in overall printing time, leading to improved fabrication efficiency.

CCS Concepts: • **Hardware → Design for manufacturability**; **Flexible and printable circuits**.

Additional Key Words and Phrases: Design for manufacturability, inkjet-printed electronics, layout decomposition

ACM Reference Format:
Yi-Ting Lin, Meng Lian, Hu Peng, Bernhard Wolfrum, Tsun-Ming Tseng, and Iris Hui-Ru Jiang. 2025. Layout Decomposition and Printing Time Optimization for Inkjet-Printed Electronics. In . ACM, New York, NY, USA, 16 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

In recent decades, there has been a growing interest in advancing printing-based manufacturing as an additive fabrication method for electronic devices. Printing-based manufacturing has several advantages that classical fabrication approaches lack, such as lower production costs and higher compatibility with various materials [1, 9, 10]. This enables inkjet-printed electronics to be applied in a variety of fields, such as organic electronics [11], flexible and stretchable electronics [7], RF identification (RFID) and sensing [5].

During the printing process, inkjet printing relies on the generation of droplets of liquid ink ejected from the nozzles of a printing head toward a substrate, where they accumulate and form a given pattern. Due to the characteristics of the ink, incompletely dried ink on the substrate can result in unexpected flow, leading to erroneous printed results. Common failures occur in two scenarios. First, undesired merging features occur when two droplets are too close to each other. Instead of remaining separate, the ink droplets combine, resulting in a different shape or causing a short

Manuscript submitted to ACM

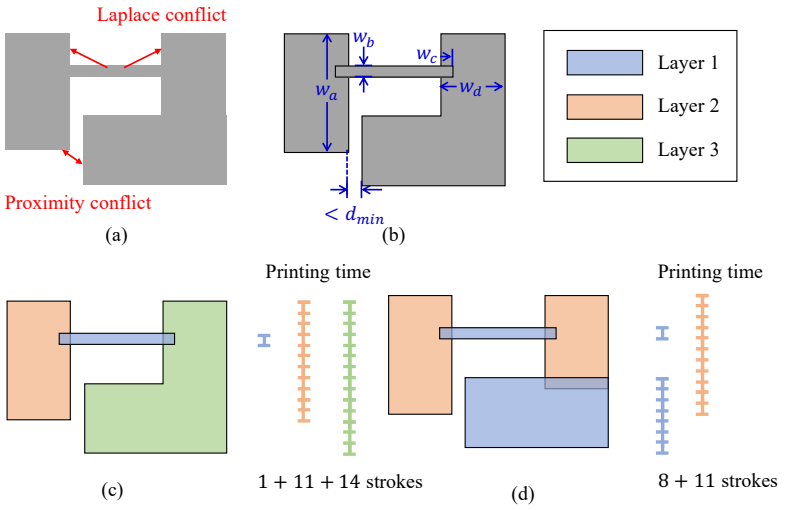

Fig. 1. Illustration of the inkjet-printed layout. (a) The conflicts can be observed in objects. (b) Objects with internal conflicts are decomposed into smaller objects. (c) After the decomposition and layer assignment, the conflicts can be resolved. (d) With a more sophisticated decomposition, a solution requiring fewer layers and less printing time, i.e., better manufacturing efficiency, can be generated.

circuit in the final pattern. Second, accidental ink redistribution occurs when small and large features are printed within the same object. Due to different surface curvatures in the smaller and the larger features, ink within the object flows toward larger features, causing smaller features to become thinner after drying. This may increase resistance in the final circuit, thus degrading timing performance. Rinklin *et al.* in [8] addressed these two non-ideal situations, defining them as the *proximity conflict* and *Laplace conflict*.

To ensure error-free circuit printing, a layering and drying process, similar to but different from the layout decomposition used in multiple patterning lithography [4], is implemented for inkjet-printed electronics. The layout is segmented into individual layers, with each layer being printed in sequence and requiring complete drying before the subsequent layer is printed. However, the repetitive printing and drying stages degrade manufacturing efficiency. Correctly and efficiently printing the circuits presents a significant challenge.

In response to this challenge, existing studies adopt a two-stage methodology: layout decomposition and layer assignment. An algorithm developed in [8] decomposes the entire design, organizes modified objects into different layers, and minimizes the number of layers used. In the first stage, objects with internal conflicts are decomposed into smaller objects, as shown in Figure 1(b). In the second stage, layer assignment dispatches these objects to different layers for printing while minimizing the number of layers used, ensuring that objects within the same layer have no conflicts, as shown in Figure 1(c).

Further, total manufacturing time consists of both printing time and drying time over all layers. Merely minimizing the number of layers used does not necessarily result in the shortest manufacturing time. Therefore, Tseng *et al.* in [12] focus on improving manufacturing efficiency by reducing the total drying time. A Gaussian drying model is employed to evaluate the local evaporation rate in the drying process. The drying time of a given layer is estimated and optimized based on factors such as the number, area, and distribution of objects within the layer.

On the other hand, to the best of the authors' knowledge, there is currently no related research focusing on printing time optimization. Therefore, in addition to drying time optimization, there is still significant potential for improvement in the printing time required for circuit fabrication. The printing time is mainly determined by the projected length of objects in the y-axis within each layer. (A more detailed explanation will be provided in the next section.) As shown in Figure 1(c), the corresponding printing time of the three layers is 1, 11, and 14 units, respectively. However, the solution quality is somewhat limited by the layout decomposition result of the existing two-stage methodology. If we simultaneously consider layout decomposition, layer assignment, and the required printing time, as shown in Figure 1(d), the object can be printed in two layers, whose printing time is 8 and 11 units, respectively. The total printing time is reduced from 26 units to 19 units, and total number of layers is reduced from 3 to 2, significantly reducing the manufacturing time.

In this work, we enhance manufacturing efficiency from a new angle. We develop a novel layout decomposition approach to directly minimize the printing time. Layer assignment in our work is used to resolve conflicts. The main contributions of our work are summarized as follows:

- This is the first work that simultaneously optimizes the layout decomposition and the printing time for inkjet-printed electronics.
- An integer linear programming (ILP) formulation and an efficient dynamic programming (DP)-based algorithm are developed. Both of the two methods minimize the printing time while considering the design constraints induced by the inkjet-printing process.

Experimental results show that our approach significantly reduces the printing time while accurately estimating the time required for printing each layer.

The remainder of this paper is organized as follows. In Section 2, we introduce the design constraints for the inkjet-printed electronics and give the problem formulation. Section 3 details our algorithm, including ILP-based and DP-based methods. Section 4 reports experimental results. Finally, Section 5 concludes this work.

2 PROBLEM FORMULATION

In this section, we introduce the design constraints and formulate the research problem addressed in this work.

To avoid undesired manufacturing failures, [8] defines two design constraints: **Laplace conflict** constrains the geometry of objects with significantly different dimensions shown in Figure 2(a). Specifically, the two contacting objects P_a and P_b have a Laplace conflict if the length ratio of their contacting edges (w_b/w_a) is smaller than a constant m_1 and their area-related ratio is smaller than a constant m_2. m_1 and m_2 are process-dependent parameters. In this condition, the smaller object, P_b, must be printed before the larger object, P_a, to prevent the redistribution of ink. **Proximity conflict** may happen between two separate objects when they are placed in proximity. As illustrated in Figure 2(b), if the distance between P_a and P_b is less than the process-allowed minimum spacing d_{min}, one of the objects should be printed after the other is fully dried. Otherwise, a short circuit may occur due to unexpected ink merging.

In the manufacturing process, the circuit layout will be divided into multiple layers and printed separately. Therefore, an **introduced overlap** is needed to ensure proper contact between two objects that originally belonged to the same object. As indicated in Figure 2(c), P_c and P_d are the two contacting objects, and P_c' is P_c after introducing the overlap. The extended length, referred to as w_c, is defined as the product of a constant p and the depth perpendicular to the contacting edge in the unchanged object, referred to as w_d.

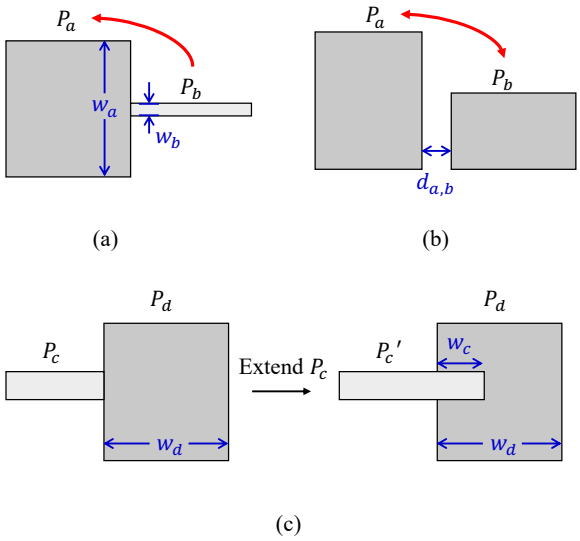

Fig. 2. Design constraints. (a) A Laplace conflict between small and large objects. (b) A proximity conflict between objects closely placed. (c) An introduced overlap for contacted objects.

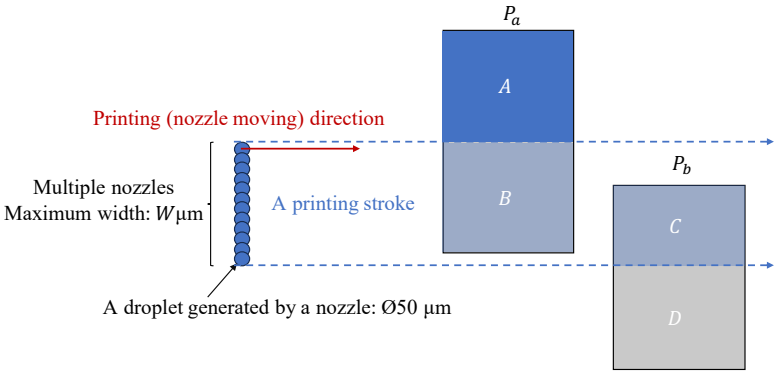

Fig. 3. Printer's printing characteristic.

Different from prior work, which indirectly reduces the printing time by minimizing the number of layers used (i.e., by optimizing the layer assignment), this work aims to reduce the printing time by optimizing the layout decomposition. The printing characteristics, particularly the time it takes and the printing limitations of inkjet printers, will affect how we design the decomposition algorithm. During the printing process, objects will be printed droplet by droplet. Each nozzle can generate one droplet at a time, with each droplet having a constant diameter (e.g., 50μm). A printer has multiple nozzles arranged in the y-direction when printing, allowing produced droplets to have 0-100% overlap. Typically, we set the y-direction distance between two adjacent nozzles (e.g., 35-40μm) to achieve a 20-30% overlap between droplets, ensuring their connection. Each time, these nozzles will move to a specific y-coordinate and travel along the x-direction, generating droplets at positions corresponding to the presence of an object. We refer to a unit of

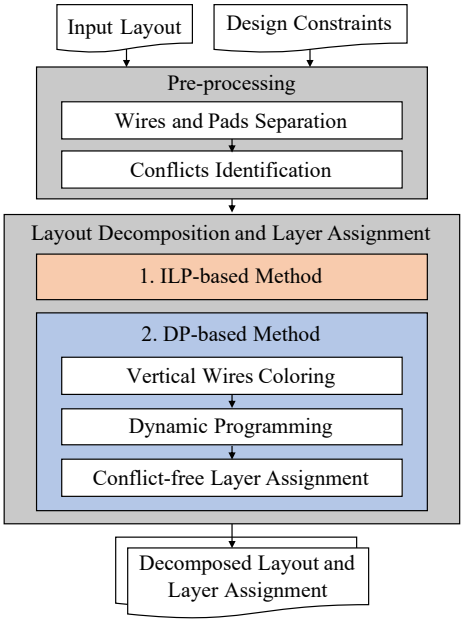

Fig. 4. Overall flow.

this printing cycle as a "**printing stroke**." As shown in Figure 3, the printer has multiple nozzles so that each printing stroke can cover a maximum width in the range of $W\mu m$. For larger objects, we require multiple printing strokes to complete the printing. Each printing stroke can cover a part of the area. For example, in Figure 3, the two regions of object P_a, namely A and B, will be completed by different printing strokes. Furthermore, when different objects overlap in the y-direction, as in the case of region B of P_a and region C of P_b, they can be printed using a single printing stroke. Each nozzle can be disabled and enabled independently to print B and C. Preliminary experiments have shown that regardless of the number of objects printed in the x-direction, the time required for a single printing stroke remains approximately constant. This indicates that the printing time is directly proportional to the number of printing strokes required.

Having the above knowledge, we formulate the problem of simultaneous layout decomposition and layer assignment for inkjet-printed electronics as follows:

Problem: Given an inkjet-printed circuit layout, the printer's settings, and design constraints, including m_1 and m_2 for Laplace conflict, the minimum spacing $d_{\min}$ for proximity conflict and the overlap ratio p for introduced overlap, the goal is to find a layout decomposition and layer assignment solution such that the number of printing strokes used is minimized, thus minimizing the required printing time, while all design constraints are satisfied.

3 ALGORITHM

In this section, we propose an ILP formulation and a DP-based algorithm. Figure 4 shows the overall flow of the proposed algorithm, which is composed of the following main stages: (1) Pre-processing, involving wires and pads separation and conflicts identification; (2) Concurrent layout decomposition and layer assignment, including the ILP-based method and DP-based method. Each step is detailed in the following subsections.

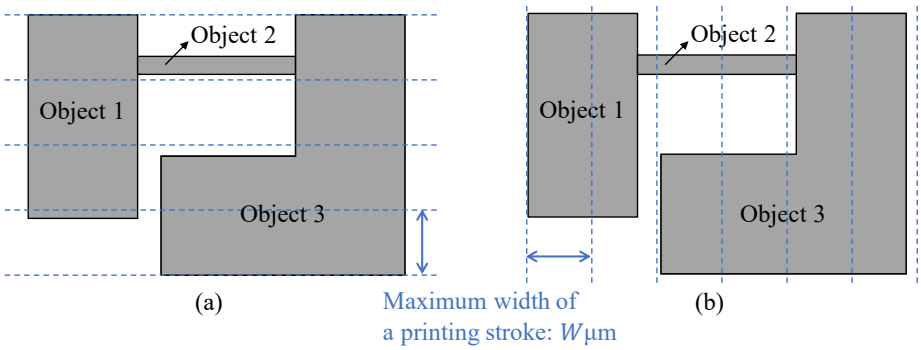

Fig. 5. ILP based algorithm. (a) Cutting the objects into smaller polygons, which can be printed by a single printing stroke. (b) Cutting into vertically-printed polygons.

3.1 Pre-processing

First, we perform an initial layout decomposition and classify objects into vertical wires, horizontal wires, and pads based on their shapes. We define an object as a wire when one of its dimensions is less than a threshold c_1, and the aspect ratio between the two dimensions (shorter over longer) is less than a constant c_2. If the length in the x-direction is longer, it is classified as a horizontal wire; otherwise, it is a vertical wire. All other objects are defined as pads. This step has two primary purposes. Firstly, conflicts within the same object cannot be resolved through layer assignment. Following a strategy similar to [8], we eliminate conflicts within an object, including Laplace conflicts and proximity conflicts, by cutting at concave endpoints. This segmentation process divides the object into multiple parts, ensuring that conflicts only exist between different objects. Subsequently, we can resolve these conflicts by determining a proper layer assignment. The second reason for this step is that if the printing direction of an object can be freely chosen, it can significantly reduce the printing time, especially for long and narrow objects like wires. The different printing directions can be achieved by rotating the substrate plate by 90 degrees. The printing time of a single object is proportional to its length perpendicular to the selected printing direction. For instance, when printing in the x-direction, it is proportional to the length projected onto the y-axis, and vice versa in the y-direction. Therefore, for vertical wires, we prefer printing along the y-direction, while for horizontal wires, we use the x-direction.

Next, based on the criteria mentioned in Section 2, we can identify all conflict relationships between objects. Meanwhile, overlaps will be introduced to ensure the proper connection of objects after the initial decomposition.

3.2 ILP-Based Layout Decomposition and Layer Assignment

Given the notations listed in Table 1, we propose an ILP formulation to resolve the layout decomposition and layer assignment problem.

To consider the printing time during layer assignment, we treat a layout as numerous tracks, and the height of each track is equal to the maximum width W of a single printing stroke. As shown in Figure 5(a), the layout is regarded as 4 disjoint tracks. For a single layer, if any part of an object falls within a track, one printing stroke is required to print it. As a result, the printing time for a layer is approximately proportional to the number of non-empty tracks within the layer. This serves as a criterion in our algorithm to evaluate the quality of a solution. To consider different printing directions, we can also regard the layout as multiple vertical tracks, as in Figure 5(b).

Table 1. Notations of ILP Formulation.

Given (Input)	
L	The number of available layers for inkjet printing
H_i	The set of horizontally-printed polygons of object i
V_i	The set of vertically-printed polygons of object i
Binary variable (Output)	
l_k	$l_k = 1$ if layer k is printed vertically
d_i	$d_i = 1$ if object i is printed vertically
$a_{j,k}$	$a_{j,k} = 1$ if polygon j is assigned to layer k
$t^h_{i,k}$	$t^h_{i,k} = 1$ if the i-th track of horizontal layer k is non-empty
$t^v_{i,k}$	$t^v_{i,k} = 1$ if the i-th track of vertical layer k is non-empty
$o_{i,j,k}$	$o_{i,j,k} = 1$ if p_i and p_j need introduced overlap

We first divide the objects into smaller polygons in both horizontal and vertical directions, as in Figure 5(a) and 5(b). As a result, each object i can be divided into horizontally-printed polygons H_i or vertically-printed polygons V_i, allowing us to consider different printing orientations.

To optimize the printing process, we use binary variables l_k and $a_{j,k}$ to decide the printing orientation of each layer and to determine which polygons to be printed in each layer. Variable l_k represents the printing direction of layer k. l_k is set to 1 if layer k is assigned to be printed vertically; otherwise, l_k is set to 0 if layer k is assigned to be printed horizontally. Variable $a_{j,k}$ represents whether polygon p_j is assigned to layer k.

The printer restricts all polygons assigned in the same layer to be printed in the same direction (i.e., cannot print horizontal polygon in vertical layers). For each polygon $p_j \in V_i$, p_j can only be assigned to layer k (i.e, $a_{j,k} = 1$) when layer k is assigned to be printed vertically (i.e, $l_k = 1$). We have the following constraint.

$$a_{j,k} \leq l_k, \quad \forall p_j \in V_i. \tag{1}$$

Similarly, for each polygon $p_j \in H_i$, p_j can only be assigned to a layer that is printed horizontally.

$$a_{j,k} \leq 1 - l_k, \quad \forall p_j \in H_i. \tag{2}$$

To simplify the problem, we assume all polygons of an object are printed in the same direction. Variable d_i represents the printing direction of object i. If object i is printed vertically, d_i is 1, and all polygons in the vertically-printed set V_i should be assigned to one of the vertically-printed layers.

$$\sum_{k=1}^{L} a_{j,k} = d_i, \forall p_j \in V_i. \tag{3}$$

If object i is printed horizontally, d_i is 0, and all polygons in the horizontally-printed set H_i should be assigned to one of the horizontally-printed layers.

$$\sum_{k=1}^{L} a_{j,k} = 1 - d_i, \forall p_j \in H_i. \tag{4}$$

To avoid Laplace conflicts and proximity conflicts, mentioned in Section 2, we introduce the following constraints. First, for each pair of objects with a Laplace conflict, the smaller object must be printed earlier than the bigger object.

For polygon p_i in the smaller object and polygon p_j in the bigger object,

$$a_{i,k} + \sum_{k'=1}^{k} a_{j,k'} \leq 1, \forall 1 \leq k \leq L. \tag{5}$$

Second, if two polygons from different objects, p_i and p_j, have a proximity conflict, they must be assigned to different layers.

$$a_{i,k} + a_{j,k} \leq 1, \forall 1 \leq k \leq L. \tag{6}$$

Variables $t^h_{i,k}$ and $t^v_{i,k}$ represent whether track i of layer k is non-empty. If any of polygon j in track i is assigned to layer k, $t^*_{i,k}$ is 1.

$$\sum_{p_j \in \text{track } i} a_{j,k} - M \cdot t^*_{i,k} \leq 0, \tag{7}$$

where $*$ is h or v depends on the direction of the track i, and M is a large constant. The large constant M is used to eliminate the constraint on $\sum_{p_j \in \text{track } i} a_{j,k}$ when $t^*_{i,k} = 1$. When $t^*_{i,k} = 1$, $a_{j,k}$ can be any value.

Additionally, we include the overlap costs to estimate the impact caused by the introduced overlap. When we need an overlap to ensure the connection of polygons, we print the overlapped area twice. We model the additional printing time as the overlap cost. Variable $o_{i,j,k}$ represents whether two adjacent polygons, p_i and p_j, need an overlap. If p_i and p_j are assigned to different layers, an overlap should be introduced.

$$a_{i,k} - a_{j,k} - o_{i,j,k} \leq 0. \tag{8}$$

Finally, the objective function is set to minimize the total number of non-empty tracks and overlap costs. As previously mentioned, this equates to minimizing the total printing time.

$$\min \ \alpha \cdot (\sum t^h_{i,k} + \sum t^v_{i,k}) + \beta \cdot \sum o_{i,j,k}, \tag{9}$$

where α is the printing stroke cost and β is the overlap cost.

3.3 Dynamic Programming Based Layout Decomposition and Layer Assignment

When the number of objects in the layout is increasing, the ILP-based method might suffer from an extremely long runtime. Therefore, we develop an efficient heuristic algorithm that leverages the advantages of dynamic programming.

In this method, we use different layer assignment strategies for different types of objects. Firstly, for vertical and horizontal wires, as mentioned in the Section 3.1, we prefer printing them along the vertical and horizontal directions, respectively. Additionally, since Laplace conflicts usually occur between wires and pads, these wires must be printed before their connecting pads. This leads us to choose a strategy as follows:

First, we print vertical wires along the y-direction. If there are proximity conflicts between the vertical wires, we apply *vertical wires coloring* to solve it. Second, we handle the horizontal wires and all pads with the *dynamic programming based algorithm.* In this step, we assign the layer of each object while considering the stroke-based printing characteristic. Finally, we adopt the *conflict-free layer assignment* to ensure the solution is feasible.

Vertical Wires Coloring. We first solve the layer assignment problem for vertical wires. Because Laplace conflicts between vertical wires and pads will be resolved by printing the wires first and there exist no Laplace conflicts between vertical wires, only proximity conflicts are considered. Furthermore, since the x-projection of a vertical wire is small (less than c_1), the printing time required for each vertical wire is not substantial. Even without considering the printing time here, it will not significantly increase the overall printing time for the entire layout. Therefore, to simplify the

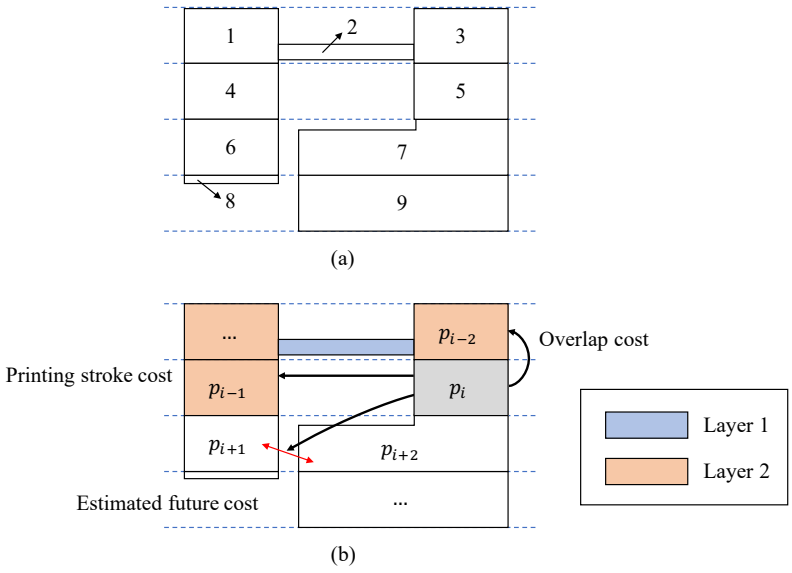

Fig. 6. Dynamic programming based algorithm. (a) The order of assigning the layer. (b) Neighboring relations while assigning polygon p_i corresponds to different types of cost.

problem, we can temporarily ignore the printing time of vertical wires and focus on printing more vertical wires. By doing so, we have fewer instances to consider later when resolving proximity/Laplace conflicts between horizontal wires or pads. Thus, the problem can be reduced to a graph coloring problem, and we use a simple coloring heuristic.

First, we construct a conflict graph, where each vertex represents a vertical wire, and each edge denotes a conflict between two vertical wires. Two adjacent vertices need to be colored differently. Also, each connected component in the conflict graph can be colored independently. For each connected component, we follow the Breadth-First Search (BFS) order for coloring. When visiting a vertex, we use the available layer (color) with the smallest index. This method cannot guarantee the minimal use of layers. Because an excessive number of layers may lengthen the manufacturing process time, we select only k layers (k is small) with most vertical wires from each connected component for vertical printing. The objects in the unselected layers will be handled later with horizontal wires and pads. Fewer objects in the unselected layers lead to less printing time and less impact on the subsequent horizontally printed layers.

Dynamic Programming Based Algorithm. For the remaining objects, finding a layout decomposition and layer assignment that results in the optimal printing time is a highly challenging problem, which is NP-hard. Hence, we simplify the problem and leverage dynamic programming to find a good solution efficiently.

Similar to the object dividing in the ILP-based method, we first divide the objects into smaller polygons, but only consider horizontally-printed polygons. Next, we apply dynamic programming to compute the cost of assigning each polygon to different layers from the top-left to the bottom-right, as shown in Figure 6(a). When we assign two objects in the same track to the same layer, it means these two objects share a printing stroke.

We define $T(i, j)$ to represent the minimum total cost of the first i polygons, and the i-th polygon is assigned to the j-th layer. A lower cost implies lower printing time. The objective function of the layout decomposition and layer

Algorithm 1: Dynamic programming for layer assignment

Input: $\{p_i\}, N, H$
Output: $\min_{1\le j\le H} T(N, j)$
Initialization;
for $j \leftarrow 1$ **to** H **do**
  $T(1, j)$ = one printing stroke cost;
end
for $i \leftarrow 2$ **to** N **do**
  for $j \leftarrow 1$ **to** H **do**
    $T(i, j) = \min_{1\le j'\le H}\{T(i-1, j') + C_{j'}(i, j)\}$;
    Record the best assignment for $T(i, j)$
    $\pi(i, j) = \text{argmin}_{1\le j'\le H}\{T(i-1, j') + C_{j'}(i, j)\}$;
  end
end
Output the decomposition and layer assignments;

assignment problem is

$$\min_{1\le j\le H} T(N, j), \tag{10}$$

where N is the total number of polygons, and H is the number of available layers that will be printed horizontally.

To compute $T(i, j)$, we show that it can be derived from $T(i-1, j')$ and the additional cost $C_{j'}(i, j)$ induced by printing the i-th polygon. If the previous polygon, the $(i-1)$-th, is assigned to the j'-th layer, the total cost of the first $i-1$ polygons is $T(i-1, j')$, and $C_{j'}(i, j)$ is the additional cost incurred by assigning the i-th polygon to the j-th layer. The minimum cost $T(i, j)$ considering all possible assignments of the $(i-1)$-th polygon can be derived.

$$T(i, j) = \min_{1\le j'\le H}\{T(i-1, j') + C_{j'}(i, j)\}. \tag{11}$$

$C_{j'}(i, j)$ is composed of four components: the printing stroke cost, the overlap cost, the conflict cost, and the estimated future cost. Now, we detail the four components of $C_{j'}(i, j)$ using the instance in Figure 6(b), where the previous polygon, p_{i-1}, is assigned to layer 2, i.e., $j' = 2$. We illustrate the possible scenarios for different j.

First, the printing stroke cost represents the time needed to print p_i. If any polygon in the same track is assigned to layer j, it means that p_i can share the printing stroke with the previously assigned polygon. (As mentioned in Section 2, the printing time of a single stroke is independent of the number of polygons printed together.) Therefore, no additional time is needed to print p_i in this case, and the cost is set to 0. Conversely, a new printing stroke is required to print p_i. The cost is set to the printing time of a single stroke.

The second component is the overlap cost, representing the printing time required to produce an overlap when two adjacent polygons originally belonging to the same object are assigned to different layers. When j is 1 in Figure 6(b), p_{i-2} and p_i are assigned to different layers. Since p_{i-2} and p_i both belong to object 3, an overlap is introduced, incurring additional printing time. Conversely, when j is 2, no overlap is introduced.

The third component is the conflict cost. A conflict cost is incurred if assigning p_i to layer j results in a proximity or Laplace conflict. This cost is set to a very high value, ensuring that the solver minimizes the conflicts or generates a conflict-free solution.

The fourth component is the estimated future cost. Assigning p_i without knowledge of the actual assignment of subsequent polygons may lead the solver to finding suboptimal results. To address this, we consider costs that are certain to occur due to the current decision to avoid underestimating the cost caused by p_i. This future cost is similar to

the heuristic function in A* search [3]. First, suppose p_i shares a printing stroke with another polygon, and there is a proximity conflict between the objects to which these two polygons originally belonged. In this case, an overlap cost will occur in the future. For instance, if p_i and p_{i-1} share a printing stroke, at least one overlap will be introduced due to the proximity conflict between the original objects, object 1 and object 3 in Figure 5. Another case to incur the future conflict cost is when p_i is associated with a Laplace conflict. If p_i is the smaller polygon in a Laplace conflict, it must be printed before another one. Therefore, it cannot be assigned to the last layer. Conversely, if p_i is larger, it cannot be assigned to the first layer.

Finally, we can solve the layout decomposition and layer assignment problem by a dynamic programming-based algorithm as listed in Algorithm 1. The first for loop is used to initialize the boundary conditions, and the second for loop determines the bottom-up computation order for the dynamic programming (DP). In the second for loop, $T(i, j)$ is computed based on Equation (11), and we record the previous polygon assignment, $\pi(i, j)$, in each step so that we can follow the record and assign the layer to each polygon after we obtain the final solution. Meanwhile, for adjacent polygons that originally belonged to the same object, we merge them together or introduce an overlap if they use the same layer or not, respectively.

Conflict-free Layer Assignment. Sometimes, the algorithm in the previous section may generate a layer assignment with design constraint violations. Inspired by the stitch insertion adopted by multiple patterning lithography [6], we use the projection method to insert stitches and reassign the layer for the illegal objects.

Considering the projection by $d_{\min}$ from surrounding objects, the target polygon can be divided into segments, e.g., 4 segments in Figure 7. Subsequently, proximity conflicts can be resolved with respect to segments. We find a legal assignment for these segments, as well as a stitch location of the legal assignment of the target polygon. As shown in Figure 7, the leftmost segment, constrained by a Laplace conflict, uses layer 1, while the right segment is assigned layer 3. After cutting the target polygon at the stitch location, an overlap is also introduced to prevent the disconnection of two segments.

Finally, although we have initially assumed a horizontal printing direction for horizontal wires and pads, we can still choose a more suitable printing direction for each layer to minimize the printing time. After determining the layer assignment for all objects, we can estimate the time required for each layer when being printed in either the vertical or horizontal direction and choose the direction that requires less time for printing.

4 EXPERIMENTAL RESULTS

We implemented our approach in C++ programming language and conducted experiments on a Linux workstation with a 2.6 GHz CPU and 197 GB memory. Seven benchmark designs and the decomposition results for fair comparison are provided by the authors of the state-of-the-art work [12] as listed in Table 2. Cases 1–3 are three basic functional designs, case 4 is a microheater, case 5 is an experimental design proposed in [8], case 6 is a 75mm×50mm digital microfluidic device from [2], and case 7 is a microfluidic design provided by the authors of [12]. In the experiments, the design was printed on a 125μm thick polyethylene naphthalate (PEN) film (as a substrate) (Teonex Q65HA, DuPont Teijin Films, Wilton, UK) using a silver nanoparticle ink (Silverjet DGP 40LT-15C, Sigma-Aldrich, St. Louis, MO, USA) with an inkjet printer (CeraPrinter F-Series, Ceradrop, Limoges, France). Prior to printing, the silver nanoparticle ink was sonicated for approximately 30 minutes using a Bransonic ultrasonic cleaner (5510E-MTH, Branson Ultrasonics, Danbury, CT, USA), filtered through a polyvinylidene fluoride (PVDF) filter (GD/X, Whatman, Maidstone, UK; pore size: 0.45μm), and loaded into a disposable 2.4 pL cartridge (Samba, Fujifilm Dimatix, Santa Clara, CA, USA). Each printer

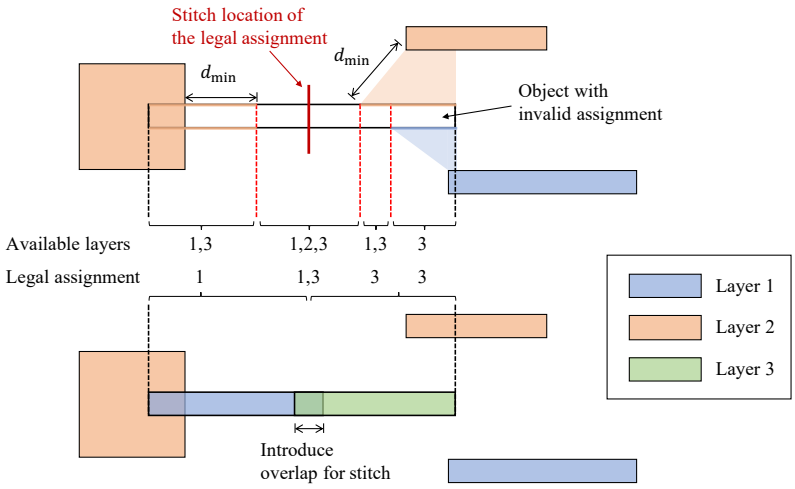

Fig. 7. Stitch location finding by the projection method.

Table 2. Layout Decomposition and Layer Assignment Results and Comparison.

Testcase	#Obj	MILP with drying time minimization [12]						Ours: ILP-based Method							Ours: DP-based Method						
		#L	#PS	PT(s)	DScore	DT(s)	MT(s)	#L	#PS	PT(s)	DScore	DT(s)	MT(s)	RT(s)	#L	#PS	PT(s)	DScore	DT(s)	MT(s)	RT(s)
case 1	18	4	46	274	2.997	338	612	4	39	242	2.944	332	564	3.998	4	39	242	2.944	332	564	0.007
case 2	12	4	28	167	1.793	202	369	4	28	167	1.793	202	369	3.661	4	28	167	1.793	202	369	0.009
case 3	16	4	39	232	2.472	279	511	4	39	232	2.472	279	511	3.948	4	39	232	2.472	279	511	0.008
case 4	52	3	188	1120	9.502	1070	2190	-	-	-	-	-	-	timeout	3	188	1120	9.502	1070	2190	0.018
case 5	22	3	29	173	0.653	73	246	3	29	173	0.653	73	246	5.968	3	29	173	0.653	73	246	0.004
case 6	710	6	1389	8272	17.808	2006	10278	-	-	-	-	-	-	timeout	6	1002	5928	25.751	2898	8826	0.454
case 7	349	6	1199	7140	13.042	1469	8609	-	-	-	-	-	-	timeout	5	616	3668	14.683	1654	5322	0.080
Ratio 1				1.031		1.003	1.015			1.000		1.000	1.000	780.8			1.000		1.000	1.000	1.000
Ratio 2				1.000		1.000	1.000										0.615		1.285	0.738	

#L: the number of used layers. #PS: the number of used printing strokes. DScore: the drying score computed by the Gaussian model [12].
PT: the printing time. DT: the estimated drying time. MT: the total manufacturing time, which includes printing and drying time. RT: the runtime.
The derived value of $d_{\min}$ in cases 1–7 was 0.468, 0.468, 0.468, 0.463, 0.036, 1.125, and 1.698, respectively.

nozzle generates ink droplets with a size of 50μm. Five nozzles operate simultaneously with a spacing of 30μm between two adjacent nozzles, meaning W, the maximum width for each printing stroke, is 170μm.

For fair comparison, we adopt the same design constraints as [12]: The threshold ratios for Laplace conflicts, m_1 and m_2, are set to $\sqrt{0.2}$ and 0.2, respectively. For the proximity conflicts, the minimum spacing $d_{\min}$ is set to min $(0.25D, 3\delta)$, where D is the perimeter of the chip frame, and δ is the minimum distance between any pair of objects in the design. For the introduced overlaps, the constant p is set to 0.2 or 0.4 depending on whether the depth of the corresponding unchanged object, which is w_d in Figure 2(c), is larger or smaller than $0.0125D$. In pre-processing, the threshold values c_1, c_2 are set to 0.6 and 0.5 to classify wires and pads. In our ILP formulation, the printing stroke cost α is set to 1, representing 1 unit of printing time. The overlap cost β is set to 3, representing the potential extra printing time cost by the introduced overlap area. The large constant M is set to 1000. In our dynamic programming algorithm, the printing stroke cost and overlap cost are set to 1 and 3 as same as those in the ILP formulation. The conflict cost is set to 1000. The estimated future cost is set to 3 or 1000, depending on whether it corresponds to an overlap cost or a conflict cost.

In Table 2, we compare the results of our layout decomposition and layer assignment with [12]. "#Obj" lists the number of total objects in each case. "#L" is the number of used layers. "#PS" is the total sum of printing strokes used across all layers, and "PT" is the total printing time. "DScore" is the drying score, a metric computed by the Gaussian drying model proposed by [12]. "DT" is the total drying time estimated by the drying score. "MT" is the total manufacturing time, which includes printing time and drying time. Also, "RT" lists the execution time of our layout decomposition and layer assignment for each case. For each case, the maximum execution time is set to 2 hours. Because the printing direction is not specified in [12], we choose the printing direction that requires fewer printing strokes for each layer for them. For our results, the printing direction is determined by our algorithm.

"Ratio 1" compares the first 5 cases, which are small-scale designs. Both [12] and our method generate the same results except for case 1. In case 1, our methods use slightly less printing time and drying time than [12]. "Ratio 2" lists the comparison of the last two cases, which are large-scale designs. On average, we achieved a 38.5% reduction in printing time. Because [12] focuses on drying time optimization, which we do not explicitly address, we have a longer drying time compared to the results of [12]. However, we still achieved a 26.2% reduction in overall manufacturing time due to the advantage of optimized printing time. In particular, for case 7, our layer assignment requires 49% less printing time compared to [12], resulting in a significant 39% reduction in manufacturing time. The results show that optimizing printing time can effectively reduce overall manufacturing time. Comparing the results of the ILP-based method and the DP-based method, the ILP-based method considers all conflicts simultaneously and minimizes the number of printing strokes, making it suitable for handling cases where conflicts between objects are dense and complex. For such cases, the ILP-based method can produce optimal solutions. However, for larger cases, the inherent scalability issue of ILP becomes evident, and its execution time exceeds the 2-hour limit. In contrast, the DP-based method focuses on resolving conflicts between nearby polygons in a bottom-up manner. Although the DP-based method does not guarantee a globally optimal solution as ILP does, our experiments show that it achieves the same performance as the ILP-based method for practical cases in a much shorter time. Additionally, the DP-based method can handle larger cases, demonstrating its good scalability.

In the following, we detail the experiment results for case 6. First, the layout decomposition and the layer assignment results are shown in Figure 8. The first layer, Figure 8(a), is printed vertically, and the others, Figure 8(b)–(f), are printed horizontally. Figure 8(g) is an image of the printed design with a grid size of 5mm×5mm. Here, we zoom in to take a closer look at different regions of the layout. It can be observed that closely spaced fine lines are printed well and no undesired ink redistribution occurs between wires and pads. Additionally, some subtle horizontal stripes can be observed on the pads, indicating that the printer behaves as expected, printing horizontal strokes. Table 3 lists the optical measurement results collected by a confocal laser microscope for ink film thickness at different regions in Figure 8(g).[1] Table 4 lists the detailed manufacturing time of each layer of case 6. "#Printing Strokes" is the number of used printing strokes for each layer, while "Printing Time" lists the actual printing time of each layer. For our results, "V1" is printed vertically, and "H1–H5" are printed horizontally, as shown in Figure 8.

Figure 9 shows the correlation between the number of printing strokes and actual printing time to validate our observations of the printing characteristic and back up the basis for the cost settings used in our method. We selected the data from each layer of case 6 design for detailed analysis. For both sets of data, our layer assignments and [12]'s layer assignments, a high correlation is observed between the number of printing strokes and printing time, with a

[1] As reported in [8], the ink film thickness of a thin wire may drop over 60% compared to the connected pad when a Laplace conflict occurs, while two nearby lines may merge when a proximity conflict occurs. The local thickness difference is around or below 10%, considered successful. The measured thickness data and the image show the absence of these two types of conflicts.

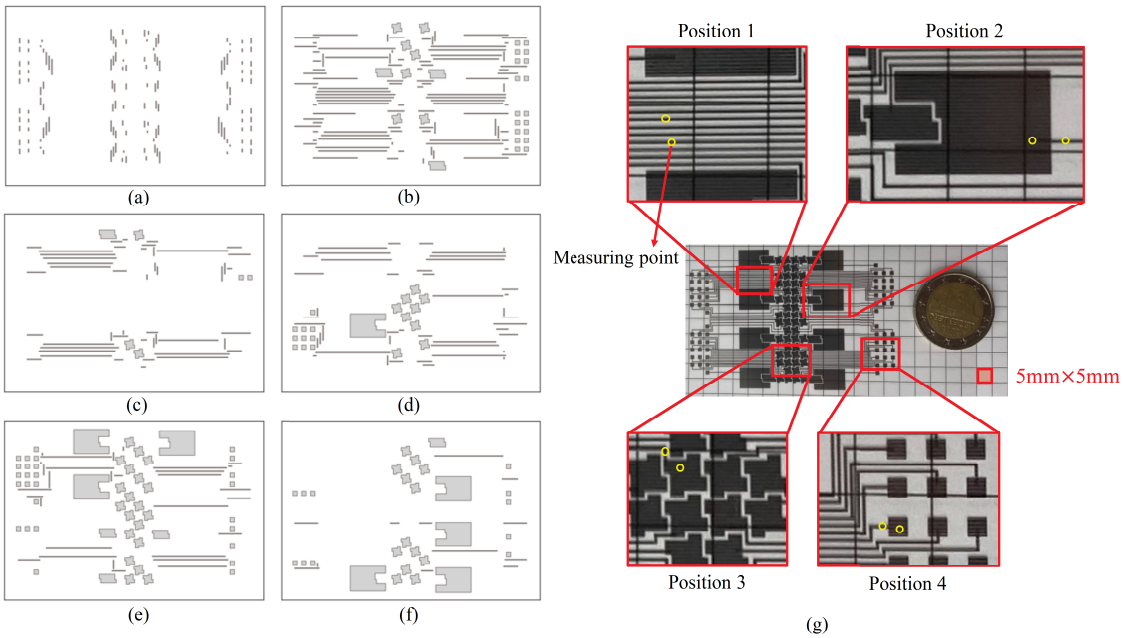

Fig. 8. The layout decomposition and the layer assignments results of case 6 design. (a) Vertically printed layer. (b)–(f) Horizontally printed layers. (g) Picture of the printed design.

Table 3. Optical Measurement Results.

Object Type	Position	Measuring Point	Ink Film Thickness (μm)
Lines	1	line1	1.178
		line2	1.290
Pad-wire	2	wire	1.353
		pad	1.508
	3	wire	1.481
		pad	1.567
	4	wire	1.613
		pad	1.662

coefficient of determination R^2 values of 0.9985 and 0.9976, respectively. This suggests that the printing stroke-like behavior closely approximates the actual printing process, allowing for the estimation of printing time based on the number of used printing strokes.

5 CONCLUSION

In this paper, we formulate a simultaneous layout decomposition and layer assignment problem to enhance the manufacturing efficiency of inkjet-printed electronics, specifically to reduce the printing time from the viewpoint of printing strokes. An ILP formulation and a dynamic programming algorithm are proposed to determine each object's decomposition and layer assignment while minimizing the printing time. Experimental results show that our method

Table 4. Layout Decomposition and Layer Assignment Results and Manufacturing Time of Case 6 Design.

Layer	#Printing Strokes	Printing Time (s)	*Drying Score	Drying Time (s)	Manufacturing Time (s)
V1	41	254	0.4409	113	367
H1	214	1269	2.7424	580	1849
H2	122	693	1.8596	449	1142
H3	158	919	6.7209	495	1414
H4	244	1455	7.0507	436	1891
H5	223	1338	6.9368	825	2163
Sum	**1002**	**5928**	25.7512	2898	8826

*The drying score is simulated by the Gaussian drying model [12].

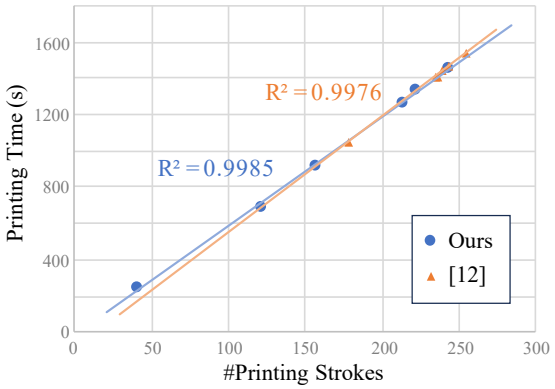

Fig. 9. Scatter plot of the number of printing strokes versus actual printing time.

can significantly reduce the manufacturing cycle-time. Future work includes the co-optimization of printing and drying times and the reliability analysis of inkjet-printed electronics.

REFERENCES

[1] Vimanyu Beedasy and Patrick J Smith. 2020. Printed electronics as prepared by inkjet printing. *Materials* 13, 3 (2020), 704.

[2] Christopher Dixon, Alphonsus HC Ng, Ryan Fobel, Mark B Miltenburg, and Aaron R Wheeler. 2016. An inkjet printed, roll-coated digital microfluidic device for inexpensive, miniaturized diagnostic assays. *Lab on a Chip* 16, 23 (2016), 4560–4568.

[3] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. 1968. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics* 4, 2 (1968), 100–107.

[4] Andrew B Kahng, Chul-Hong Park, Xu Xu, and Hailong Yao. 2008. Layout decomposition for double patterning lithography. In *International Conference on Computer-Aided Design*. IEEE, 465–472.

[5] Sangkil Kim. 2020. Inkjet-printed electronics on paper for RF identification (RFID) and sensing. *Electronics* 9, 10 (2020), 1636.

[6] Jian Kuang and Evangeline F. Y. Young. 2013. An efficient layout decomposition approach for Triple Patterning Lithography. In *50th Design Automation Conference*. 1–6.

[7] Nanshu Lu and Dae-Hyeong Kim. 2014. Flexible and stretchable electronics paving the way for soft robotics. *Soft Robotics* 1, 1 (2014), 53–62.

[8] Philipp Rinklin, Tsun-Ming Tseng, Cai Liu, Mengchu Li, Korkut Terkan, Leroy Grob, Nouran Adly, Sabine Zips, Lennart Weiß, Ulf Schlichtmann, et al. 2019. Electronic design automation for increased robustness in inkjet-printed electronics. *Flexible and Printed Electronics* 4, 4 (2019), 045002.

[9] John A. Rogers, Takao Someya, and Yonggang Huang. 2010. Materials and mechanics for stretchable electronics. *Science* 327, 5973 (26 March 2010), 1603–1607.

[10] Vivek Subramanian, Josephine B Chang, Alejandro de la Fuente Vornbrock, Daniel C Huang, Lakshmi Jagannathan, Frank Liao, Brian Mattis, Steven Molesa, David R Redinger, Daniel Soltman, et al. 2008. Printed electronics for low-cost electronic systems: Technology status and application

development. In *34th European Solid-State Circuits Conference*. 17–24.

[11] Anke Teichler, Jolke Perelaer, and Ulrich S Schubert. 2013. Inkjet printing of organic electronics–comparison of deposition techniques and state-of-the-art developments. *Journal of Materials Chemistry C* 1, 10 (2013), 1910–1925.

[12] Tsun-Ming Tseng, Meng Lian, Mengchu Li, Philipp Rinklin, Leroy Grob, Bernhard Wolfrum, and Ulf Schlichtmann. 2021. Manufacturing Cycle-Time Optimization Using Gaussian Drying Model for Inkjet-Printed Electronics. In *International Conference On Computer Aided Design*. 1–8.

Received 30 August 2024; revised 30 December 2024; accepted 17 February 2025