IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 18 March 2023, accepted 3 April 2023, date of publication 7 April 2023, date of current version 13 April 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3265481

SURVEY

Analog Integrated Circuit Routing Techniques: An Extensive Review

RICARDO M. F. MARTINS[1,2], (Member, IEEE), AND NUNO C. C. LOURENÇO[1], (Senior Member, IEEE)

[1]Instituto de Telecomunicações, 1049-001 Lisbon, Portugal
[2]Instituto Superior Técnico, Universidade de Lisboa, 1049-001 Lisbon, Portugal

Corresponding author: Ricardo M. F. Martins (ricmartins@lx.it.pt)

This work was supported by the Fundação para a Ciência e a Tecnologia (FCT)/MCTES through national funds and when applicable co-funded European Union (EU) funds, through the Internal Research Project LAY(RF)^2/X-0002-LX-20 and HAICAS/X-0009-LX-20, under Project UIDB/50008/2020.

ABSTRACT Routing techniques for analog and radio-frequency (A/RF) integrated circuit (IC) design automation have been proposed in the literature for over three decades. On those, an extensive set of geometric constraints have already been covered as surrogates for routing quality, but also, performance-related criteria were progressively included. However, as A/RF design moved into advanced integration technology nodes, the increasing number of design rules/constraints, wire resistance, congestion, and interwire parasitic growth is constantly challenging existing automatic routing techniques and keeping pressure on their improvement. Fortunately, recent developments in modern workstations' capabilities allowed the growth of sophisticated routing processes, including some assisted by the latest machine and deep learning methods, offering unprecedented solutions for the automation of this task. Still, as the correlation between routing-induced parasitic structures and the circuit's functional behavior is far from simple, computational-intensive parasitic-inclusive and layout-aware synthesis techniques have also been proposed, where automatic routing techniques play a decisive role. This paper conducts an extensive review of A/RF IC routing techniques, from the digitally-inspired earliest approaches to state-of-the-art developments, providing a complete and comprehensive guide for circuit designers and design automation developers while defining research lines to facilitate more activities within this field.

INDEX TERMS Analog and radio-frequency, automatic routing, layout-aware synthesis, machine learning, parasitic-inclusive synthesis, path-finding algorithm, physical design.

I. INTRODUCTION

As complementary metal–oxide–semiconductor (CMOS) technologies were scaled down, an increase in complexity was observed in all fields related to the semiconductor industry. To name a few, the characterization of sub-nanometer CMOS devices has led to a gigantic increase in the number of parameters of BSIM models [1], widely used for simulation. Also, the characterization processes and devices became increasingly more challenging, requiring the application of sophisticated techniques [2], [3], [4], which are still open research topics. Additionally, to meet the stringent fabrication requirements, the average number of layout design rules contained in process design kits (PDKs) has escalated from below 1000 rules in 180/130-nanometer nodes to more than 10 times in 22-nanometer and below [5]. Consequently, this generalized increase in complexity has also impacted integrated circuit (IC) layout design.

In particular, routing is the task during IC layout design where the precise conductor paths necessary to electrically connect all parts of the circuit/system are formally defined. In the analog domain, it is widely acknowledged that circuits/systems' performance is critically dependent on the parasitic structures induced by the layout. Thus, analog routing has historically been a handcrafting process, requiring a different level of attention and detail than its digital counterpart. When comparing both, analog has a relatively low number of nets to be routed, but each is usually carefully drawn due to

The associate editor coordinating the review of this manuscript and approving it for publication was Yuh-Shyan Hwang.

the constraints/requirements imposed on it. For that reason, tools for its automation have been intensively proposed in the literature for over three decades. Since it is one of the final steps in the design flow, the attainable routing quality is strongly dependent on the previous layout design tasks, especially placement. To clarify, several design techniques can reduce the necessary wiring during device-or-block-level placement, such as interdigitated and common-centroid structures or device merging. Moreover, the location on the floorplan of devices and terminals determines the routing task's success. Automatic placement techniques have been exhaustively studied by the electronic design automation (EDA) community, and the proposed approaches have been progressively enhanced to cope with several layout constraints and design challenges. Extensive reviews on automatic analog and radio-frequency (A/RF) placement tools are remitted to [6], [7], and [8], and the latest efforts on the application of machine and deep learning (M/DL) techniques to [9]. Therefore, this paper conducts an extensive review of analog IC routing techniques. It covers from the earliest approaches, such as procedural generators or digitally-inspired channel routers, to the latest sophisticated routing processes, including those assisted by M/DL methods that exploit existent legacy data.

Geometric constraints, generally derived from manual layout heuristics, have ever driven the analog routing techniques, and while they are primarily used as surrogates for routing quality, they also help to reduce the degrees of freedom in an automatic routing mechanism. The reduced dimensions explored in CMOS technologies led to an increase in wire resistance, congestion, and interwire capacitances growth, turning cross-coupling, and noise-dominant effects in fully scaled designs. However, A/RF systems do not scale so aggressively as digital. Several works report the effective use of highly scaled nodes for A/RF, e.g, in [10] the authors show comparisons between 22nm FDSOI, 14nm FinFET, and 28/45nm bulk CMOS. They show that on the technology side, solutions for A/RF are being provided to the designers. For instance, passive device structures, e.g. transformers/inductors, that are built on the thick top metals do not change much between nodes, even with the scaling on the deep metals of the technology. Even inductor-less designs, like the one reported in [11], can perform with the same or better noise in 16 FinFET as in 40, 65, or 90nm, this can be achieved with proper design and by being mindful during the layout. Therefore, since the earliest approaches, analog routing techniques have also attempted to introduce performance-related criteria either by separating noise and sensitive nets on the layout or minimizing coupling parasitics among different wires of the different nets. Even in those cases, the correlation between minimized parasitic structures and the circuit's functional behavior is far from simple. Ultimately, the circuit's functional constraints can only be verified through post-layout circuit/system simulations, and the performance degradation (and consequently routing quality) can only be quantified at this stage. Consequently, parasitic-inclusive synthesis (PiS) and layout-aware synthesis (LaS) techniques become an important trend in A/RF IC design [12]. In those, the gap between sizing and layout design steps is shortened by including parasitic-related data into an automatic sizing process or closed by fully generating the layout inside it, as shown in Fig. 1. In both PiS and LaS techniques, a vast number of layouts have to be automatically generated without user's intervention, where, automatic routing it determinant. These techniques are broadly covered within this review as well.

The remainder of this manuscript is organized as follows. Section II summarizes the topics addressed in this review. Section III overviews the routing generation by a set of operations described in a procedural language, while Section IV describes the encoding routing guidelines in a graphical template description. Section V details the sequential application of path-finding algorithms over certain grid representations, and Section VI presents routing techniques that simultaneously handle all nets of a circuit using integer linear programming (ILP) and optimization. Section VII outlines the most recent routing techniques enhanced by M/DL methods. Finally, Section VIII concludes this review while discussing the main challenges and future research directions for EDA researchers and developers.

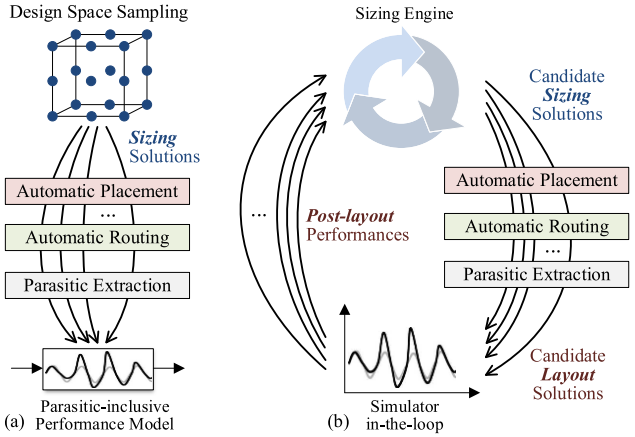

FIGURE 1. (a) Parasitic-inclusive synthesis (PiS): parasitic extraction is conducted for each point of the design space sampled and used to build parasitic-inclusive performance models for further resizing iterations/optimizations; (b) Layout-aware synthesis (LaS): explicit layout generation is performed in-the-loop for each candidate sizing solution, and the post-layout performances used to guide the sizing engine.

II. SUMMARIZED TOPICS

As stated, this paper conducts an extensive review of analog IC routing techniques. It starts by addressing the approaches that generate the routing by coding it entirely in procedural scripts, i.e., procedural routing generators (PRGs). In these, different executions for the same set of inputs always produce the same output, and while their ability to generalize to new designs is limited, they are useful when complete control over

the routing generation is desired. They are time-consuming to develop and maintain but very fast at producing the routing.

Subsequently, template-based routing generators (TbRGs) are covered, where a specification and technology-independent template acts as an alternative representation, generating the target routing when a new set of devices' sizes and technology information is provided. These are particularly useful when legacy layouts must be migrated to other technology nodes, or changes in the design are needed. Following, the digitally-inspired routers, which later led to the appearance of sequential path-finding generators, where an algorithm is applied to draw a wire over a grid representation, connecting two different terminals of a net in the presence of obstacles. While these successfully encoded a set of routing techniques as constraints, they are highly dependent on techniques to deal with the conflicts among the different nets or even wires within the same net.

More recently, different techniques attempted to handle all the wires of all the nets at once, i.e., concurrent routing, via integer linear programming or evolutionary computation. Afterwards, routing processes that learn from existing legacy data via M/DL have been proposed. Although these methods are still taking the first steps, given the recent advances in M/DL fields, many expectations have been made in this research community around them. To complete the analysis, the roles of all these routing methodologies that were covered are discussed within the context of PiS and LaS, which became an important trend in A/RF IC design.

Finally, future research directions are highlighted to facilitate more research activities within this field. A final note is left for simultaneous place and route [13], [14] and routing of array-type analog blocks [15], [16], e.g., large-scale field-programable analog arrays, flash type or folding/interpolating A/D converters, current-steering D/A converters, or cellular neural networks, whose discussion, despite valuable, is left out of the scope of this review.

III. PROCEDURAL SCRIPTS

The earliest approaches to generating very large-scale integration layouts coded the entire layout of a circuit in procedural scripts, often designated by procedural layout generators, such as the earlier proposals of ALI [17], SILT [18], ADL [19] or Layla [20]. These scripts would then generate the target layout when a new set of devices' sizes (or, in some cases, technology information) is provided. Different tool executions for the same set of inputs always produce the same output, behaving similarly to a parametric cell (PCell) but for the complete circuit/system to be generated. These have a low level of automation compared with more sophisticated routers (as will be overviewed in the next Sections), but they are relevant for circuit designers/design houses that want complete control over the routing generation process and always obtain a meaningful layout solution. These designers are usually skeptical about the success/reliability of other routing tools with a higher level of automation, as they usually involve non-systematic/stochastic processes. Thus, it is likely that the produced solutions will not be the same for every run of the generator.

Consequently, it is impossible to guarantee that the produced solutions match designers' expectations. In procedural generators, designers use the software tool as an assistant, coding the interconnects instead of drawing them in the layout editor. Therefore, future iterations/ modifications in the design, or even technology migrations, represent changes in the code instead of changes in the layout itself. The procedural routing generators, PRGs, that will be overviewed within this Section are summarized in Table 1.

TABLE 1. Summary of the procedural routing generators (PRGs).

Tool	Year	Key Routing Spec.	Tech.	Base Code
OAC [34]	1990	Symmetry support	µm-range	C & Fortran
BALLISTIC [21]	1995	Total control over shapes, layers and widths	1.2-0.6µm	Lx
CAIRO [36]	2000	DC currents used to adjust interconnect widths	0.6µm	C
MSL [22]	2003	Reproduces channel routing style	0.18µm	Custom
BAG [23]	2013	Predeveloped helper classes with design routines	65-40nm	Python
MAGIC [38]	2018	Macros for the open-source Magic VSLI tool [40]	0.35-0.13µm	Python
BAG2 [25]	2018	Grid systems for multi-patterning design rules	45-16nm	Python
LAYGO [26]	2021		28-16nm	Python

A. ROUTING GENERATED BY PROCEDURES

As stated, PRGs work with a set of operations described in some procedural language. However, since the circuit designers fully specify such PRGs, they are responsible for ensuring the correctness of the layout. BALLISTIC [21] was one of the first languages specifically developed for analog layout. The language was inherently technology-independent, and the code written in it could be translated into the native layout code of a commercial layout environment, where the graphical interface, design verification tools, and extractors could be exploited. The interconnect commands controlled paths by defining break-points, layers, and widths. Additionally, different wire shielding and bus structures were also provided. Reference [22] proposed a high-level module specification language (MSL, 2003). The routing procedures were achieved in MSL by using routing boxes for each net of the circuit, which are constituted by a horizontal wire and several vertical wires, as shown in Fig. 2, where T_1, T_2, and T_3 are the three terminals to be connected. For each floorplan provided, the relative positions of T_1 with respect to T_2 and T_3 are different, and the leftmost and rightmost terminal positions determine the routing box module. A horizontal wire is then drawn spanning the leftmost and rightmost terminals, and the connectivity is ensured by simply dropping a vertical connection from each terminal to the horizontal wire.

In the BAG (Berkeley Analog Generator, 2013) framework [23], coded in Python programming language, a designer-oriented platform to develop procedural scripts was proposed. BAG's foundations rely on the freely available Synopsys' PyCell API [24], which allows process-independent PCells. However, it was later replaced by BAG2 [25], offering a so called XBase and Laygo engines, that enable the development of process-portable layout generators in order to cover state-of-the-art technologies, such as 28-nm or 22-nm (and not broadly supported on Synopsys' PyCell). BAG2 adopts regular routing grids, which are preferable to deal with the complex and un-intuitive multi-patterning design rules of advanced technologies nodes. Therefore, the width and spacing of the wires are quantized in this grid system, and wires on the same layer must travel in the same direction (with adjacent layers having alternating directions). Design-rule correctness and process portability are guaranteed by adjusting the grid parameters. More recently, from the same research team, LAYGO (LAYout with Gridded Objects 2021) engine [26] offered similar multi-patterning routing concepts for advanced FinFET technologies. Using integer-based abstract coordinates, as illustrated in Fig. 3, the generator alleviates designers from directly dealing with the design rules and physical coordinates. Moreover, port names can be used to identify the coordinate values, enhancing the programmability and portability of the scripts. LAYGO's scripts can also be optimized for the target process technology/design, as the grid can be non-uniform.

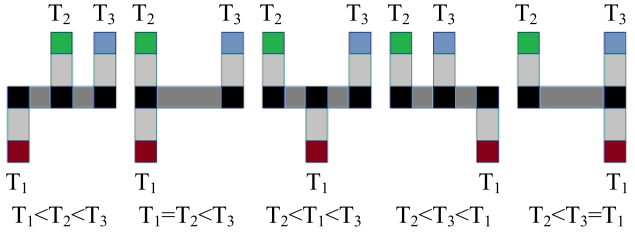

FIGURE 2. Routing box instances of the high-level MSL. Adapted from [29].

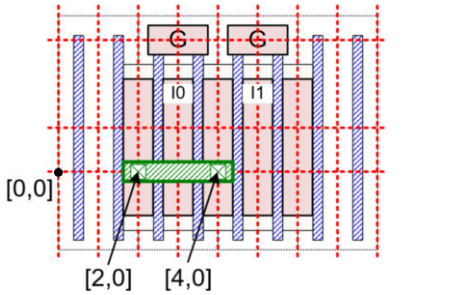

FIGURE 3. LAYGO' s routing coordinates are abstracted by integerbased abstract coordinates. Adapted from [26].

Even though the interconnects are generated quickly within PRGs, since they are highly customized, they lack the flexibility to easily accommodate broad changes in the layout, i.e., if the relative positioning between the cells' terminals to be routed changes, the procedural routing scripts may no longer be valid and have to be partially/fully re-coded. Similarly, the cost of introducing new functionalities to accommodate different design choices is usually high. At this point, it becomes easier just to repurpose a copy of a previously existing generator to the new design choice. This solves the individual complexity of the generator and allows fast development, but has as a disadvantage the proliferation of similar but different generators that all need maintenance. Libraries of sub-components/structures that are systematically reused whenever a full custom layout is not required, facilitate updates. But it is important to be careful when updating changes in the fundamental blocks upstream though, as they might break some of the generators using them. A solid version control system and dependence system, such as the ones used in software [27], [28] can be of much help. Another drawback of these methodologies is that since designer-defined scripts are followed, no parasitic impact due to the routing is accounted for during generation. The latter aspect is somewhat bypassed in the PiS and LaS, as described in the following sub-sections.

B. LAYOUT SAMPLING FOR PARASITIC-INCLUSIVE SYNTHESIS

Since PRGs produce the output quickly, they became an interesting solution to route multiple floorplan solutions for the layout sampling step of PiS approaches, as shown in Fig. 1(a). In [29], the previously presented MSL was used to route several floorplan solutions, and then parasitic extraction was conducted on each. For the interconnects, area and coupling parasitics are used to build macro-models that linearly correlate performances and parasitics. These macro-models are then used to re-optimize the circuit. In [30], layout sampling was also made using MSL, but instead of building macro-models, the capacitive and resistive parasitic impact is embedded into the circuit's symbolic performance models (SPMs). These SPMs are then used to estimate the circuit performances, replacing the simulator. Later, a multiobjective optimizer was applied over the SPMs, and the parasitic-inclusive Pareto-optimal performance surface of the circuit was determined [31]. Pareto optimality is the state at which resources in a given system are optimized in a way that one target cannot improve without worsening another competing target.

These approaches were applied in analog blocks, such as single-ended/differential amplifiers, regenerative comparators, and bandpass filters, for a 0.18-μm technology node. On the same technology node but focused on RF design, in [32], parasitic-inclusive SPMs were created for a low noise amplifier (LNA) routed by MSL scripts. There, rule-based extraction was used to calculate interconnects' parasitic capacitances, while an off-the-shelf quasi-static extractor was used for interconnects' parasitic resistances and inductances.

C. FAST GENERATION INSIDE LAYOUT-AWARE SYNTHESIS

Similar to the layout sampling of PiS, the LaS methodologies, previously shown in Fig. 1(b), also favor fast routing approaches, as the layout is generated numerous times during the automatic sizing procedure. The first report of a PRG embedded into a LaS flow was made in the C programming language-based OAC (Operational Amplifier Compiler, 1990) framework [33], [34]. Extraction functions for the interconnect parasitic capacitances to the substrate were also specified as procedures in the layout environment. Later, in [35] and [36], a dedicated layout description language composed of a superset of C programming language functions, CAIRO, is used to describe the routing inside a knowledge-based circuit sizing tool, i.e., a hard-coded procedure to determine the devices' sizes. The routing functions are based on predefined reference points on the layout, and each wire's width is determined according to the assigned metal layer and the current of the devices' terminals connected to it. All parasitic estimations are done using geometrical methods and tested on an operational transconductance amplifier (OTA) for a 0.6-μm technology. The high-level MSL was used on the RF domain to route single and differential LNAs inside an optimization-based sizing loop [37]. There, as performed in [32], mixed standard rule-based and quasi-static extraction methods were applied to compute the interconnect parasitics. More recently, MAGIC [38], [39] proposed configurable layouts described using macros specially designed for an open-source layout editor [40], and these were integrated into a LaS flow. Finally, BagNet [41] used the BAG2 generator and a deep neural network to select which candidate solutions are sent to the simulator.

IV. TEMPLATE-BASED DESCRIPTIONS

Template-based routing generators, TbRGs, are somewhat less restrictive than PRGs because only the relative location of interconnects between cells is defined, i.e., routing guidelines. The template acts as an alternative specification and technology-independent graphical layout representation, which generates the target layout when a new set of devices' sizes and technology information is provided. The template guidelines always produce expectable layout solutions for the circuit designer (even though some may apply stochastic processes during generation). They are particularly useful when a validated previously designed layout, i.e., a legacy layout, has to be migrated to other(s) close/similar technology node(s) or changes in the design are needed. Instead of drawing the new interconnects from scratch, the designer focuses on developing a reusable template description for it or applying a methodology that automatically extracts the legacy information into a template representation. The most relevant TbRGs that will be discussed within this Section are summarized in Table 2.

TABLE 2. Summary of the template-based routing generators (TbRGs).

Tool	Year	Key Routing Spec.	Tech.	Base Code
Castro-López [46]	2002	Parameterized templates for Cadence's Virtuoso platform	0.35µm	SKILL
IPRAIL [59]	2004	Direct wire conversion from legacy design	0.25-0.18µm	C/C++
LAYGEN [51]	2006	Genetic algorithm reduces interconnect length	0.35µm	Java
LDS [53]	2011	Paths drawn by A* algorithm	0.35-0.13µm	Java
Zhang [77]	2011	Generation bounded by previous sensitivity analysis	0.25-0.18µm	C++
Pan [63]	2015	Correlated w/placement by a planar straight line graph	90-65nm	C++
LASER [83]	2015	Parameterized templates for Synopsys' Laker	0.18µm	Tcl/Tk & C++
Chi [67]	2021	Correlated w/placement by a Cartesian detection line	90-65nm	C++

A. ROUTING GUIDELINES ENCODED IN TEMPLATE

The first report of encoding routing guidelines in a graphical template description was made by Conway and Schrooten [42], even though some preliminary ideas can be analogously drawn from earlier efforts from Chowdhury et al. [43], [44]. The fundamental principle is that when a new set of devices' sizes are provided, and the devices replaced to accommodate those changes, the interconnects described in the routing template are stretched or compacted according to the new terminals' locations. Later, Castro-López et al. [45], [46] built the parameterized layout templates in SKILL language to be tailored for the commercial Cadence Virtuoso platform [47]. The particular focus was on creating retargetable analog intellectual property (IP) blocks. Layout retargeting is the process of generating a layout from a legacy design. Its main target is maintaining most of the design choices and knowledge of the source while migrating it to another technology node, updating specifications, or attempting to re-optimize the legacy design. Similar routing concepts are followed in Chameleon ART [48] and, more recently, by Naguib et al. [49].

In LAYGEN [50], [51], the knowledge of the designer in terms of interconnect structure is encoded in extensible markup language (XML) template descriptions but without forcing an implicit definition during generation. For that purpose, LAYGEN uses a two-stage generation process. First, it adjusts the routing template to the floorplan provided, and then, the routing solution is optimized. In the adjustment procedure, the template paths are scaled and moved to the wire's source point, and after, the paths are corrected (by adding new wire segments) to connect to the wire's sink point, ensuring connectivity, as illustrated in Fig. 4. In the optimization stage, a genetic algorithm encodes the routing information by assigning one gene to each adapted net [52]. The crossover operator generates offspring that presents a combination of their parents' nets, and mutation operators, mostly based on sliding and moving the source/sink (or both) point of a wire inside the terminals' shapes, are used. A cost function is minimized to eliminate remaining design rule

violations, reduce wire length, and increase the distance between the noisy and sensitive nets identified by the circuit designer.

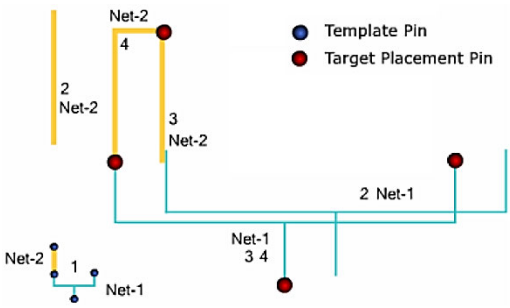

FIGURE 4. LAYGEN's adjustment procedure: (1) Routing template; (2) After scale; (3) After moving; and (4) Target routing (after adding new wire segment). Adapted from [50].

In [53] and [54], a specific layout description script (LDS, 2011–14) [55] is used to generate the interconnects from simple declarative statements using linear programming. In Fig. 5(b), a path coded with LDS is shown, and the resulting path and script in Fig. 5(c) and (a), respectively. The routing statements are classified into seven groups: group 1 statements assign widths to the paths; groups 2 and 5 abut the paths to the related ports; groups 3 and 4 attach the paths; group 6 is required to ensure design-rule correctness of the layout; and finally, group 7 adds statements the increase the distance to the surrounding paths, attempting to reduce parasitic coupling. The interconnect abstract representation in LDS requires a path-finding algorithm to draw them, in this case, an A* algorithm [56] operating over a Hanan grid [57]. In LDS, the estimate of the minimum cost from a wire's intermediate point to the wire's sink point of A* is the wire resistance of the Manhattan segments that constitute it.

B. EXTRACTION OF TEMPLATES FROM LEGACY DATA

In the previous sub-Section, it was assumed that routing templates had to be explicitly defined by circuit designers. However, methodologies that automatically extract information from legacy layouts into template representations exist and are interesting options for layout retargeting. Initially, a C/C++ language tool, IPRAIL (Intellectual Property Reuse-based Analog IC Layout, 2003–04) [58], [59], attempted to maintain most of the design choices and knowledge of the source design. It first converted the legacy layout into a resizable symbolic template, and when the new set of design rules and devices' sizes were provided, it generated the retargeted layout with adjusted interconnects. The approach was applied to RF design in [60]. However, no significant details about the legacy layouts' interconnects are kept in the conversion process, as only simple recursive algorithms are used to detect the nets of the original design. LDS Analyser [61], an add-on to the previously presented LDS, aims to convert a legacy layout via an interactive graphical interface into LDS templates. Interconnects are captured by selecting path segments and assigning them a net name and are represented in data structures that assign points for the vias and line segments for the paths. The designer is also requested to input a routing style that mainly describes the path's effective width. The extracted LDS scripts are specification-independent and can be reused for retargeting directly, or additional constraints may be added manually to the extracted scripts.

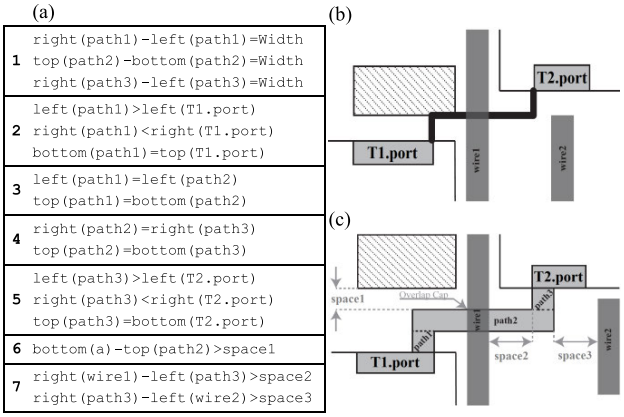

FIGURE 5. (a) LDS statements for the path in (b); (b) Candidate path; and (c) Realization of the candidate path. Adapted from [53].

Chin et al. [62], [63] intended to fully preserve the routing behavior of the legacy design by using a constrained Delaunay triangulation (CDT) algorithm [64], which represents the correlation between routing and placement into a planar straight line graph. First, placement and routing paths are decomposed with a set of triangles, as illustrated in Fig. 6(b) for a layout diagram with two blocks and one routing path in Fig. 6(a). Afterward, the points where the CDT edges cross the routing path are marked (Figs. 6(c)). These are then used to save the path behavior as multiple routing paths. Routing is ultimately migrated hierarchically, following a bottom-up fashion in agreement with the hierarchical placement algorithm. However, since the routing is adapted straightforwardly, significant variations in the device sizes and related distance on the retargeted layout cause a gap in the approach, and thus, some reference lines in CDT models may overlap with some blocks, as shown in Fig. 6(d). Those lines must be removed from the CDT model and routed manually after migration. Therefore, Chi et al. [65], [66], [67] proposed an enhanced model, i.e., cartesian detection line (CDL), to represent the correlation between routing and placement. Instead of using the triangles, the CDLs extended from each block boundary to help to capture the routing paths completely, as illustrated in Fig. 6(e)-(g), significantly increasing the number of nets that can be routed automatically based on the original layout style. A similar idea of reusing legacy knowledge is taken by Wu et al. [68], [69]. However, instead of a one-to-one retargeting process, the target topology sub-circuits are compared with those found on a library of legacy topologies/layouts. Each sub-circuit pattern is used as a guideline for the new design, but the routing between

sub-circuit patterns still has to be performed manually or using another automatic routing methodology.

By abstractly representing the paths structure and/or terminals locations, TbRGs present higher flexibility than PRGs. Even so, if the relative positioning between the cells' terminals to be routed changes significantly, the template descriptions may have to be re-adjusted. It is also worth noting that the detailed implementation of the connections, where the intricacies of the design rules are handled, must account for all rules required to create DRC correct connections. Therefore, at the detailed level, TbRGs must be updated to be able to handle new design rules or limiting factors due to evolving technology features. Another drawback of these methodologies is the fact that since designer-defined templates or knowledge extracted from legacy designs are followed, usually, no parasitic impact considerations due to the routing is accounted for during generation, with some minor exceptions such as LAYGEN, that attempt to minimize the interconnect ground capacitances (computed with linear equations) during the optimization stage. The lack of parasitic impact considerations is dealt with in the following sub-Sections.

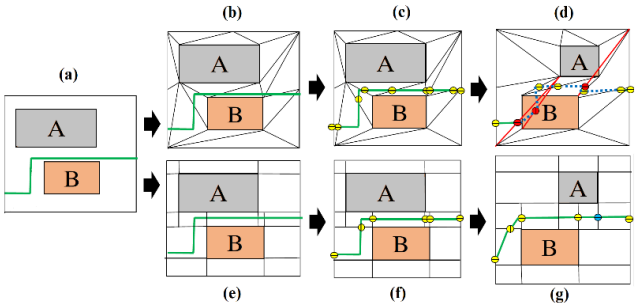

FIGURE 6. (a) Generic legacy design with one path. (b)-(d) Routing preservation by CDT, where (c) illustrates the crossing points of the CDT graph with the path, and (d) the overlap of the path with cell B after displacement. (e)-(g) Routing preservation by CDL, where (f) shows the crossing points of the CDL graph with the path, and in (g), the original shape of the legacy design is kept after displacement. Adapted from [65].

C. BOUNDED BY PERFORMANCE SENSITIVITY

Until this point, the overviewed standalone routing tools cannot guarantee to meet the circuit/system functional behavior or even detect the causes of its infeasibility as soon as possible. For that reason, constraint-driven layout tools were proposed based on a pre-layout sensitivity analysis of the circuit performances [70], [71]. These performance sensitivities are used to generate a set of routing bounds on the critical parasitics, in theory, driving the router to ensure the satisfaction of the functional constraints. These concepts were used by Bhattacharya et al. [72], [73], [74] to enhance IPRAIL, where analytical models are created for parasitic resistances and parallel capacitances and embedded into the underlying reduced-template-graph extracted from the legacy layout. Afterward, linear and nonlinear optimization methods solve the graph and retarget the layout constrained by the parasitic bounds imposed. The resistance and capacitance parasitic bounds can be described with the following inequalities:

$$b_n \leq \sum_i \rho_{sh} \frac{len}{wid} + R_{cont} \leq b_x \tag{1}$$

$$b_n \leq \sum_i c_a.len.wid + \sum_i c_{sw}.2.len \leq b_x \tag{2}$$

where b_n and b_n are the minimum and maximum parasitic resistance/capacitance bounds obtained from the sensitive analysis [70], [71]. ρ_{sh} is the sheet resistance per unit length, R_{cont} is the net resistance contributed by contact rows, c_a the substrate capacitance per unit area, and c_{sw} the sidewall substrate capacitance per unit length of each rectangle with a width *wid* and length *len*. While this approach generates the layout towards functional constraint fulfillment, parasitic-related geometry limits may over-constrain the problem, and possibly no routing solution may be found since the parasitic bounds are generated before optimization. Therefore, Zhang and Liu [75], [76], [77] featured the modeling of functional constraints directly in the nonlinear optimization method.

D. PARASITIC-INCLUSIVE AND LAYOUT-AWARE SYNTHESIS

Similar to the PRGs, TbRGs also suit PiS and LaS methodologies' needs, as they provide a fast layout generation. For PiS, in [78], the previously introduced SKILL-based parameterized templates of Castro-López et al. [45], [46] are used to route several floorplan solutions. The interconnect parasitics of each sample are then extracted using an off-the-shelf tool and stored in a lookup table. This lookup table correlates the sizing of the circuit with the values of its interconnect parasitics, and, at each iteration of the sizing algorithm, the lookup table is used to retrieve an approximate value of the parasitics for the current sizing solution. Later in [79] and [80], Toro-Frías et al. work was enhanced to a LaS flow, where a sizing optimization based on a multiobjective algorithm performed explicit routing generation and extraction using Cadence's Diva in-the-loop. The tool ultimately provided a layout-aware Pareto front at the output instead of a single sizing/layout solution. The methodologies were tested on 2-stage amplifiers for a 0.35-μm technology node. Likewise, in [81], LDS [55] was used for the LaS of a folded cascode OTA on a 0.13-μm technology node, where a commercial extractor also computed the interconnect parasitics. A final note is left on LASER (Layout-aware Analog Synthesis Environment on Laker, 2013–15) [82], [83], which provides a user-friendly graphical user interface (GUI) to assist the designer in iterative steps required on a LiS flow. On LASER, predefined templates for different circuit topologies can be easily converted into Synopsys' Laker [84] scripts.

V. SEQUENTIAL PATH-FINDING

Sections III and IV presented PRGs and TbRGs that inherently follow designer-defined scripts, templates, or even knowledge extracted from legacy designs. This Section starts

by presenting digitally-inspired earlier strategies that preceded a different set of approaches more tailored for A/RF design, henceforward designated sequential path-finding generators, SPfG, which generate a routing solution with minor inputs from the designer, and, therefore, present a higher automation level and flexibility to deal with new design problems.

A. PRELIMINARIES: DIGITALLY-INSPIRED STRATEGIES

When the necessity to automate analog IC design arose due to the need to design circuits with higher complexity in a shorter time, the EDA research community was inspired by the experience and development path already paved for digital, which had been fundamental for the IC industry's growth. Regarding routing, and due to the computational limitations of workstations at that period, a transversal characteristic was the use of channel routing [85] or its variations, such as Glitter (1986) [86], a grid-less and variable-width implementation. In channel routing, it is assumed that the wiring area, i.e., the channel, is rectangular with fixed terminals at the top and bottom and no terminals at the right or left. Moreover, designer knowledge was incorporated into most generation processes. In MIGHTY [87], [88], the routing tool used in the OPASYN layout generation framework [89], [90], particular focus is given on pre-assigning the nets of the circuit into different categories, defining the order in which they are routed. These are then routed one at a time sequentially, and when a particular net is blocking another net, rip-up, and rerouting techniques are used. The rip-up and rerouting terms (often called strong modification) refer to the action of removing parts of already routed nets to reclaim their space and allow the routing of blocked connections. MIGHTY-type routing concepts were also employed in SLAM (Smart Layout tool for Analog Modules, 1989–92) [91], [92], [93] and by Gyurcsik and Jeen [94]. Later, MIGHTY was enhanced with symmetrical routing capabilities in the channels to support the design of differential circuits, resulting in the ART tool [95], [96].

A concept also adopted from digital is the usage of a two-step routing process, i.e., global routing, and only after, detailed routing. In ILAC (Interactive Layout of Analog CMOS Circuits, 1988–89) [97], [98], the router from the IDAC framework [99], nets are classified into different categories: sensitive nets (for high-impedance nodes), noisy nets (output nodes and high swing nodes), noncritical signal nets, and power supply. A channel router in the detailed phase is used to route net-by-net in the following priority order: power nets, sensitive nets (to ensure the shortest path), noncritical, and, finally, noisy nets. Noncritical and power are used as shields between the sensitive and noisy. A semi-custom digitally inspired routing process is also taken in LADIES [100], where routing areas are extracted and decomposed into rectangles and used to form an adjacency graph. Global routing is achieved by finding the shortest path on the adjacency graph and the detailed routing using the river routing algorithm [101]. Similar concepts are applied in SALIM (System for Analog Layout generation of Integrated Microcircuits, 1988) [102], [103], where a directed graph of the routing channels is obtained from the slicing tree used to encode the floorplan. During detailed routing, different constraints are considered when materializing the path on each channel, such as: priority to sensitive nets, low resistance for power supply lines, or even geometrical symmetry to equalize signals' propagation delays. These constraints are provided as rules by the user. Later, a gridless version of SALIM was proposed in [104], while in [105], hierarchical channel graphs were reported for global routing only. A channel-style global router is used in ALADIN [106], [107], [108], [109], where a Dijkstra shortest path algorithm is applied to solve the minimum-Steiner-tree problem. The minimum-Steiner-tree method used for net-length estimation considers the net sensitivity and channel congestion. A channel-inspired detailed router is also employed in ALG [110], [111]. Quite recently, and as a final note, Wei and Murmann discussed the application of digital place-and-route tools to the design of analog and mixed-signal blocks on a 16-nm FinFET CMOS technology [112]. While such an approach can be used in some generic medium-performance blocks, it is not well suited for leading-edge analog blocks.

B. ANALOG SPECIFICS

The driving factor of digitally-inspired strategies was to augment established digital techniques with additional constraints, primarily focused on minimizing crosstalk between nets identified by the designer. This approach is fair as the order in which nets are routed (in a sequential process) dramatically impacts the overall routing solution. Routing critical nets first allows them to occupy more privileged locations in the layout, minimizing their length and using preferable conductors, and later these nets will then act as obstacles to less critical nets, whose paths may be less efficient.

However, in an analog router, there was a severe need to consider analog-specific routing constraints, which resulted in cost-minimization routers with constraints, often referred to as area routers. In such routers, the path-finding algorithm is applied to draw a wire over a particular grid representation, connecting two different terminals of a net in the presence of obstacles, i.e., devices placed on the floorplan and previously routed wires, where a set of constraints enforces analog routing design strategies.

Analog routing constraints, as shown in Fig. 7, are formulated to match the parasitic among different nets. In both symmetry and common-centroid constraints, the wire lengths, widths, layers, and the number of vias must be the same for both routing paths, symmetric with respect to some common axis or center. Topology-matching, such as exact matching [128], [129], [130], [131], is usually imposed on the critical yet asymmetry nets, with the same number of bends, vias and wirelength. For less critical nets, length-matching constraints are imposed to route them with equivalent wirelength [132].

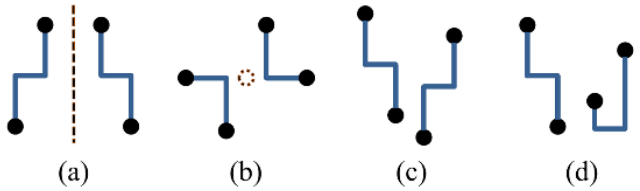

FIGURE 7. (a) symmetry, (b) common-centroid, (c) topology-matching, (d) length-matching constraints. Adapted from [148].

Two of the most widely studied non-idealities in analog routing are electromigration, which refers to the material migration in the power networks and signal wires that are stressed with high current-densities, and IR-Drop, which consists of a fluctuation of the net voltage due to the interconnect resistances, affecting circuit behavior. Such non-idealities can be ultimately mitigated by using routing algorithms with variable wire widths [133]. These, along with Steiner routing, have been broadly covered in the literature [134]. These routers account for the design rules during the search by expanding the grid representation with the minimum space requirements. Moreover, grid routers tend to degrade space efficiency as the grid resolution increases. On the other, the grid's size (number of points) also increases as the resolution is improved, leading to increased computation time. In order to bypass the problems of a fixed grid, a shift factors technique [106], which can control the slight shift of each unit rectangle, could be used. Many approaches use a global coarse grid for planning and an exemplary local grid for detailed routing [109] to balance the grid resolution and size. In some approaches, the grid is uniform and complete [114], while others use non-uniform [153] or tile-based [118] representing possible connections between the connection points and obstacles. The most relevant SPfGs that will be discussed within this Section are outlined in Table 3.

C. ANALOG MAZE ROUTERS

ROAD [113] used a modified version of Lee's maze router, where a cost function weighted the metal resistivity, ground capacitances, proximity to wires already routed, and congestion, the latest based on both the real wire crowding of the surrounding area and, on an estimate of its final crowding using a fast first-attempt path-search for each of the remaining wires. A scheduler produced a priority list. Still, rip-up and reroute phases were mandatory. A maze-style router is also used in AAR [114], ALSYN's analog router, which supports individually selected wire width, and noisy and sensitive nets can be marked to avoid crosstalk. In the previously presented ILAC [97], [98], a maze algorithm is used for global routing only. Maze routing is also used in the detailed routing phase of ALADIN [106], [107], [108], [109], where a ring router [106] is firstly used to create dense wiring around the module boundary. The size of the ring depends on the number of layers used and the overlapping definition of different nets. Xiao et al. in [115] support symmetrical routing where one path is routed, and then the route is mirrored to produce the routing of its counterpart. Obstacles, i.e., the active areas of devices, are mirrored to produce the first path. The symmetrical nets are routed first, and only then the nets with smaller bounding box areas and finally the nets with the smaller number of pins. Crosstalk is considered during layer assignment only, which is made after drawing the path.

TABLE 3. Summary of the sequential path-finding generators (SPfGs).

Tool	Year	Key Routing Spec.	Tech.	Base Code
ANAGRAM [117]	1988	Line-expansion: coupling between wires quantified	within µm-range	C
ILAC [98]	1989	Maze & Channel: different global and detailed phases		Pascal
ANAGRAM II [118]	1991	Line-expansion: tile-plane grid		C
ROAD [113]	1993	Maze: generation bounded by previous sensitivity analysis	within .µm-range	C
AAR [114]	1993	Maze: considerations for sensitive nets		C
Malavasi [139]	1996	Channel: generation bounded by previous sensitivity analysis		C
ALADIN [109]	2006	Channel: Minimum-Steiner-tree problem solved by Dijkstra	0.8µm	C++, Skill & Tcl/Tk
Xiao [115]	2010	Maze: Symmetry support	0.13µm	C++
Ozdal [131]	2012	Maze: Exact matching support	n/s	C++
MAGICAL [173]	2021	Line-expansion: global and detailed phases	40nm	Python, C/C++

D. LINE-EXPANSION ROUTERS

Line-expansion-style routers combine the flexibility of a maze with computational efficiency [116]. ANAGRAM [117], developed in *C* programming language, instead of simply departing the noisy/sensitive nets, a line-expansion style router operating over a course-grid is adopted to balance crosstalk interactions in a quantifiable manner. ANAGRAM routes wires sequentially by expanding paths from the source terminal to the sink terminal (this is achieved by popping the most promising partial path from the heap), and during this process, the cost of each wire/segment is given by the sum of a cost proportional to its length (including vias and design rule violations) and the cost of crosstalk. The cost of crosstalk models the "amount" of coupling with other previously routed segments. Even though no exact quantification of capacitance coupling is actually made, this qualitative measure is made assuming that nets can be neutral, noisy, or sensitive, resulting in different weights for the cost of crosstalk component. The next partial path to be extended (by a segment) is the one with the lowest cost so far. Finally, multi-port terminals are introduced, i.e., each terminal can be modeled as collections of segments free to be connected instead of individual points. A rip-up and reroute phase

randomly eliminates violations caused by blocked connections that result from the net ordering.

An improved version operating over a tile-plane grid and supporting wiring symmetry was proposed in ANAGRAM II [118], and included in the ACACIA framework [119]. In a tile-plane grid, there is no difference between unused space, wires, and device geometry, and the router can "see" the internal details of devices. This has several advantages: over-the-device wiring incurs no overhead; pieces of devices can now themselves be used for wiring paths; electrical terminals can appear as arbitrary collections of geometry; and the same crosstalk penalties that apply to wire segments can be applied to pieces of placed devices. Additionally, tile-based grid-less routers [120], [121] or even mixed A* and maze [122] were also reported. A* (and related) became a popular approach, and several works used it to deal with more routing challenges, such as minimum area [123] and step [124] constraints, wire load reduction [125], [126], [127], net shielding [122], parallel run and end-of-line spacings [124], etc. Recently, in MAGICAL [172], [173], A* search is used at both global and detailed routing phases. In the first, the placement is divided into a limited number of grid cells of unified size and the rough path are drawn after transforming the multi-pin nets into two-pin nets, and only after, the global paths are re-drawn while guaranteeing the alignment to the manufacturing grid.

E. BOUNDED BY PERFORMANCE SENSITIVITY

As explained in Section IV-C, performance sensitivity can be used to bound the routing generation. An application of these constraints were implemented over a digitally-inspired channel router in [136], [137], [138], and [139]. In SPfGs, since each net is routed one after another, the only option is to weight each edge during path-finding according to a set of sensitivity constraints. In ROAD [113] the contribution of a specific wire's parasitic to performance degradation is proportional to its sensitivity and inversely proportional to the maximum variation range allowed to that performance. An optimization step was also used to reroute nets while minimizing a performance impact factor [140], [141].

F. PARASITIC-INCLUSIVE AND LAYOUT-AWARE SYNTHESIS

The SPfG bounded by performance sensitive of [140] and [141] was used to route RF circuit inside a LaS flow, with either 1/2-D analytical extraction of wire parasitic capacitances [144] or electromagnetic simulation carried for the passive devices [146]. In [142] a customizable industrial routing tool [143], whose underling algorithm is based on simulated annealing, was used inside LaS Geometric routing constraints are set to meet DRC rules, as well as specify minimum wire widths, maximum wire lengths, the number of vias and corners allowed along a connection, maximum parallel and tandem wire separation and length, and so on. The maximum resistance and load capacitance of each connection between devices is also specified, as is the maximum coupling capacitance between critical nodes. For symmetrical devices, a preference for symmetrical signal routing paths is registered. Two issues associated with placement are pin assignment and the margins between devices.

VI. HANDLING ALL NETS SIMULTANEOUSLY

The SPfGs of Section V successfully encoded a set of routing techniques as constraints. However, since an A/RF circuit/system contains a considerable number of nets to be routed, SPfGs struggle to deal with the conflicts among the different nets, or even wires within the same net, as the routing of a single net may inadvertently hinder the subsequent routing, and, therefore, heuristics for net rip-up and reroute are mandatory to attain valid solutions. Additionally, the different routing constraints have to be prioritized, and this ordered routing style might result in congested regions, and consequently, inefficient paths and performance degradation. Therefore, during the last decade, and due to computational power provided by modern workstations, different techniques attempted to handle all wires of all the nets at once (A@O). The A@O generators that will be overviewed within this Section are condensed in Table 4.

TABLE 4. Summary of the integer linear programming (ILP) and evolutionary routing generators.

Tool	Year	Key Routing Spec.	Tech.	Base Code
Ou [148]	2014	ILP: Octagonal grid model	0.18µm-90nm	C++
LAYGEN II [154]	2014	Multiobjective NSGA-II fully optimizes interconnects	0.35-0.13µm	Java
Wu [152]	2015	ILP: A priori high-quality routes for individual nets	0.18µm	C++
Torabi [149]	2016	ILP: mitigates electromigration	90-65nm	C++
AIDA-L [158]	2017	Wiring/symmetry planner added before optimization	0.35µm-65nm	Java
Chiang [175]	2023	ILP: constrains design space to reduce generation time	90-65nm	C++

A. INTEGER LINEAR PROGRAMMING

Ou et al. [147], [148] took a drastic solution to simultaneously handle of all nets of a circuit by solving the routing as an ILP problem. There, a constructive ILP operated over an octagonal grid model, illustrated on Fig. 8, allowed to consider both 45 and 90-degree routing styles in the same grid. Additionally, the four different types of routing constraints simultaneously, i.e., wiring symmetry, common-centroid, topology-matching and length-matching, are supported within this representation. The ILP problem is set to minimize the weighted cost-function:

$$min \sum_{n_k} (\alpha O_w(n_k) + \beta O_b(n_k) + \gamma O_v(n_k) + \delta O_c(n_k)) \quad (3)$$

where the functions $O_w(n_k)$, $O_b(n_k)$, $O_v(n_k)$ and $O_c(n_k)$ correspond to the total wirelength, bend numbers, via counts and coupling noise of net n_k, respectively. α, β, λ and δ weights are user-specified constants. Coupling noise is generically computed as a function of the distance between two parallel wire segments, and by minimizing the sum of all coupling components, it moves towards a global minimum, unlike PfSGs, where the parasitic coupling is minimized case-by-case as a wire is being drawn.

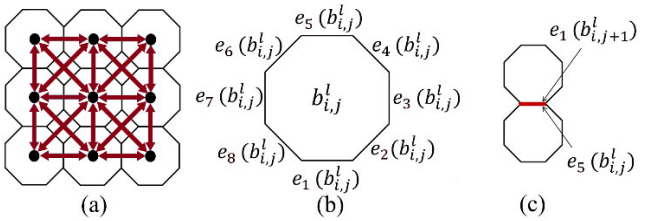

FIGURE 8. (a) Octagonal grid model with 9 bins (central dot of an octagon); (b) Detail of a bin $b^l_{i,j}$, with $0 \le i \le w$, $0 \le j \le h$ and $l \in L$, where w and h are the number of grids in the width and height of the chip, and L the number of metal layers available for routing. Additionally, each bin is defined by 8 edges $e_{1,\ldots,8}(b^l_{i,j})$; (c) To satisfy the design rules, each edge allows only one net to pass, therefore, the situation of overlapped edges $e_5(b^l_{i,j})$ and $e_1(b^l_{i,j+1})$ between bins $b^l_{i,j}$ and $b^l_{i,j+1}$ cannot occur [148].

However, even though certain types of constraints are prioritized to reduce the number of variables for problem-solving, ILP-based routers have a severe scalability problem as the number of octagonal grids increases with the width and height of the routable area, often requiring prohibitive memory allocations. Torabi and Zhang took a similar approach [149] and later enhanced it with a post-processing step attempting to mitigate electromigration effects [150], [151]. Wu et al. [152] simplified the problem by constructing a priori high-quality routes for individual nets using a SPfG. Afterward, the ILP is used to commit each net to only one of its candidate routes, designated by the pre-assisted ILP approach. While this greatly reduced the complexity of the problem, the number of candidate routes selected for each wire is still a delicate tradeoff to be decided a priori. Recently, Chiang et al. [174], [175] used an ILP routing technique to automate finger-capacitor array synthesis for custom successive-approximation register ADCs, where the problem was severely constrained in order to reduce the generation time.

B. EVOLUTIONARY

Another possibility to handle interactions among nets is to pursue complete routing optimization. In LAYGEN II [153], [154], an enhanced version of the TbRG LAYGEN, fully developed in Java language, is proposed. LAYGEN II discards the interconnect structure contained in the template description and uses only the terminal-to-terminal connectivity and constraints information. The core algorithm is the multiobjective NSGA-II (Non-dominated Sorting Genetic Algorithm II) [155], and the chromosome encodes each wire of each net as a different gene, as shown in Fig. 9. Different heuristics are used for the initialization of the genes. Then the optimization process thoroughly performs structural and layer changes (using mutation and crossover operators) in a population of different and independent routing solutions, i.e., chromosomes. Wiring symmetry is forced directly on the genes. This way all wires of all nets can be optimized simultaneously. Therefore, the huge search space is explored by a number of optimization variables equal to:

$$x^d = \sum_{i}^{N} \left[\sum_{j}^{n_i} \left(\sum_{k}^{K} \left(lenght_{ijk} + layer_{ijk} \right) + SourcePort + SinkPort \right) + \sum_{j}^{s_i} \left(x_j + y_j \right) \right] \tag{4}$$

where, N is the number of nets, n_i the number of wires in the net i, and K stands for the fixed number of segments in wire j. $length_{ijk}/layer_{ijk}$ is the length/layer of the segment k of wire j of net i. Finally, s_i is the number of Steiner points in the net i. Wires are freely drawn in the technology grid, and the different routing solutions are evaluated by built-in design-rule and layout-versus-schematic-check algorithms that provide the constraints at each generation of the optimization problem. Additionally, different objectives were set, such as minimizing interconnect length and number of vias, and maximize the distance between noisy and sensitive nets. The number of short circuits, design-rule errors, crossings between noisy and sensitive nets, and noisy/sensitive nets running on top of devices must be driven to zero.

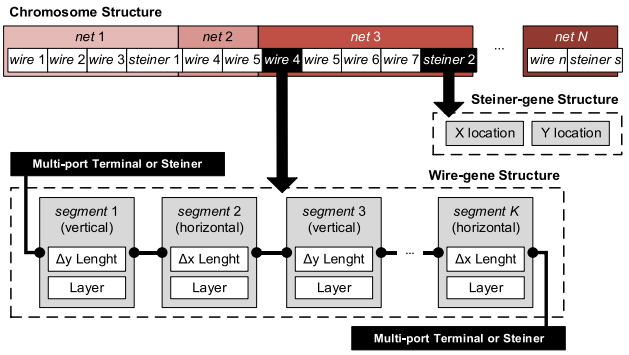

FIGURE 9. Chromosome structure with n wires and s Steiner points distributed by N different nets; Wire-gene structure composed of K-even linked segments; and Steiner-gene structure [8].

C. BOUNDED BY PERFORMANCE SENSITIVITY

For the ILP routing approaches, as the problem being solved deals with all the nets simultaneously, in [147], the summation of all estimates of the capacitive coupling noise between two parallel wire segments is weighted in the single-objective cost function being minimized by the ILP. While the previous work only considers mutual capacitances, in [151], only self parasitics are accounted. There, each interconnect

is replaced by a π model in the pre-layout netlist, and then, its parasitic resistance and capacitance values are varied and simulated. This sensitivity analysis step sets a minimum and maximum parasitic resistance and capacitance values for each wire that still provide acceptable performances. The parasitic structures considered are modeled as a function of the wire's length and width and then used as constraints during problem-solving. Nonetheless, not only the relations between parasitic structures (self and mutual) and post-layout performance degradation are nonlinear and complex, it is not even known which nodes will be connected by parasitic devices until the routing is complete, turning it extremely difficult to predict the performance before having the routing.

D. LAYOUT-AWARE SYNTHESIS

Despite the inherent high computational time required, the evolutionary routing technique's flexibility proved useful to incorporate LaS methodologies. LAYGEN II's optimization flow was preceded by an electromigration-aware wiring planner, symmetry planner, and A*-based operating over a course grid global router, resulting in the AIDA-L (Analog IC Design Automation – Layout, 2014-2017) [8], [156], [157], [158], and embedded into a complete framework for A/RF IC synthesis. The LaS flow was based on a multiobjective algorithm performing explicit routing generation and extraction using Mentor Graphics' Calibre in-the-loop, and applied on several analog and RF case studies, e.g., voltage-controlled oscillators (VCOs) [159], [160], LNAs & mixers [161] and amplifiers [162] for 0.35-μm down to 65-nm nodes.

VII. ASSISTED BY MACHINE/DEEP LEARNING

Similar to the routing via ILP and EA, the computational capability of modern workstations provided the means for M/DL techniques to expand in many new applications. That is also true for analog and RF IC design, where M/DL techniques are already paving their way in many steps of the design flow, such as modeling, synthesis, layout, and even testing [9], [163], [164]. These tools opened new perspectives for developing push-button solutions that simultaneously incorporate legacy data or expert design insights in a manner that was not possible in the previous generations of EDA tools.

While some previous TbRGs or SPfGs have incorporated different forms of computational intelligence (mostly evolutionary algorithms) either to minimize wire length during template migration [50], [51] or reroute nets while minimizing a performance impact factor [140], [141] and more recently, complete A@O routing design via multiobjective optimization [153], [154], which may fall into the boundaries of the machine learning field, the focus of this Section is different. It intends to highlight the recent efforts made in the application of M/DL to exploit existent legacy data and use them to enhance/assist established routing techniques.

However, only a few works were reported so far. GeniousRoute [165] was the first work to extract knowledge from a library of legacy layouts and apply it to guide an SPfG routing algorithm. In the pre-processing of the training data, placement and routing are represented as 2-D images, where routing-relevant information is extracted. For each data point, the pins of the entire design and pins for the given net are mapped into two separate 64 $\times$ 64 channels. These channels were then used on a two-step model training, where, first, an artificial neural network used as variational autoencoder (VAE) was initialized in an unsupervised fashion, and only after, supervised decoder training. GeniousRoute underlying routing mechanism was a classical A* path-finding algorithm assisted by the model's inference, which generates the routing probability map to guide the search. In the A* cost function a routing guidance component composed of two parts is used: a penalty for violating the guidance provided by the model, and a penalty for routing in the probability region of other nets. Traditional rip-up and reroute techniques are still used; however, the legacy design patterns will be present in the automatically generated routing solutions.

In REINFORCE [166], an attention-based reinforcement learning model was applied to solve the track-assignment problem for detailed routing. The algorithm is designed to encode the design rules into the track-assignment steps, and the model determines the best sequence of the set of device pairs to be routed, such that the overall solution quality is maximized. In [167], the training of the reinforced learning model was supervised with past solutions, leveraging the information previously produced. Both methods provide nearly 100$\times$ runtime speed-ups when comparing with a plain schedular based on an EA, while generating solutions that are comparable in quality. As a final note and in a slightly different direction, DPRoute [176] applies multi-agent deep reinforced learning to learn the routing schedules between nets and solve the net ordering problem. Each net is handled as an agent, which needs to consider the actions of other agents while making pathing decisions to avoid routing conflict. Unlike previous tools that attempt to solve block or system-level routing, DPRoute is intended for severe package routing problems and has proved to reduce design violations and packaging wirelength when compared with non-ML packaging routing algorithms [177]. These preliminary routing methodologies assisted by M/DL are resumed in Table 5.

TABLE 5. Summary of the routing generators assisted by machine/deep learning (M/DL).

Tool	Year	Key Routing Spec.	Tech.	Base Code
GeniousRoute [165]	2019	Probability maps generated by VAE guide a SPfG	40nm	Python & C++
REINFORCE [166]	2020	Reinforcement learning solves the track-assignment	16nm	Python
DPRoute [176]	2023	Reinforcement learning solves the net ordering	n/s	Python

VIII. DISCUSSION AND FUTURE RESEARCH LINES

Automatic routers are not widely deployed in the frameworks of A/RF IC design provided by major computer-aided vendors. In those that incorporate them, it is possible to confirm that the early approaches are the ones still present, combined in a manual/assisted design fashion, e.g., the use of sequential routers with rip-up and reroute on demand and with partial manual design for critical nets.

However, as shown, there have been considerable efforts in the research community in proposing routers for A/RF ICs. Nowadays, IC designers and design automation developers have at their reach a panoply of analog routing techniques and ideas. Fig. 10 illustrates the spread of those contributions among the identified types of routing techniques from 1980 to 2022.

Starting with PRGs and later by TbRGs, although the interconnects are generated quickly, layout changes are not easily accommodated, i.e., if the relative positioning between the cells' terminals to be routed change, the procedural scripts may have to be re-coded or the templates adapted. Still, these approaches keep resurging as they are the ones that provide greater control over the generated routing. Another limitation pointed to both is that since designer knowledge is followed, no explicit parasitic impact due to the routing is accounted for during generation, which can lead to unforeseen design iterations. This aspect can be bypassed by its successful application within PiS/LaS flows [29], [30], [31], [32], [33], [34], [35], [36], [37], [78], [79], [80], [81], [82], but at the cost of execution time. SPfGs are closer to digital IC routing, and are immensely popular with two bursts of innovation in the 1990's and 2010's. Differently from PRGs and TbRGs, SPfGs encode several routing good practices and parasitic estimations as constraints. Still, due to their sequential nature and since an analog circuit/system contains a considerable number of nets to be routed, the routing of a single wire/net may inadvertently hinder the subsequent routing. Therefore, non-systematic heuristics for net ordering, rip-up, and rerouting are mandatory to attain valid solutions.

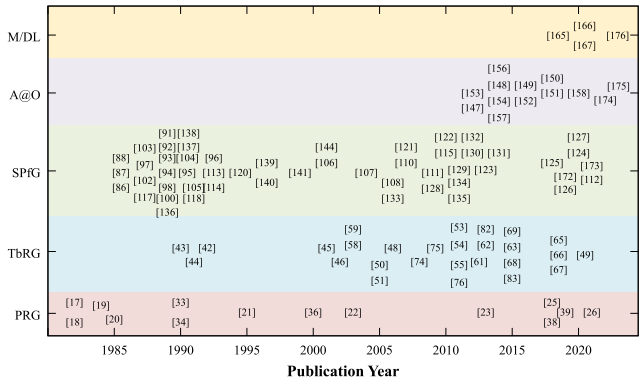

FIGURE 10. Timeline of routers for analog IC over the last 40 years.

Since analog ICs still have a manageable number of nets, an alternative that is getting some attention as the available processing power increases is to handle all nets of the circuit simultaneously towards global optimum solutions with ILP or EA. In ILP A@O even though certain types of constraints are prioritized to reduce the number of variables for problem-solving, it has a severe scalability problem as the number of grids increases with the routable area, often requiring prohibitive memory allocations. In EA-based A@O, memory allocation is not a concern, but the time required to fully evaluate the design rules and short circuits of a single solution is. Moreover, it has a quite low sampling efficiency, generating a high number of solutions without valuable information for the optimization process. Both SPfGs and EAs provided higher flexibility to LaS flows, where the floorplans can be varied without compromising the success of the following automatic routing step.

Regarding functional constraints, in the SPfGs since each net is routed one after another, the only option to handle parasitic coupling (generically computed as a function of the distance between two parallel wire segments) is minimizing it case-by-case as a wire is being drawn. This process assumes that minimizing coupling for each wire as it is being routed, i.e., considering only the ones that were already routed, will result in a satisfactory layout solution at the end. In A@O, minimizing the sum of all coupling components is possible, forcing the algorithm to move towards a global minimum. Another option for all the overviewed approaches is to bound the routing generation by performance sensitivities. However, since the parasitic bounds are generated before optimization, they follow a worst-case approach. These parasitic-related geometry limits may over-constrain the problem, and no routing solution may be found. But in many cases, parts of routing will be subject to less armful parasitic effects, which would allow for other parts to endure more aggressive parasitic without arming the circuit performance easing the job of the router. Nonetheless, due to the nonlinear nature of analog performance space and the nonlinear correlation between parasitics and post-layout performance degradation, even if the routing is bounded, the final performance is ultimately unpredictable. Still, there seems to be a trend to include as many details on all nets as possible. This is justified by the sensitive nature of analog signals and the wide range of harmful interactions that can arise due to a badly chosen routing.

Finally, M/DL approaches are revolutionizing many fields in engineering and EDA [163], [164], and the first steps of applying them to the analog routing problem have already been made. Future research directions are highlighted and summarized below to facilitate more research activities within this field.

A. RF AND MILLIMETER WAVELENGTH

IC design at RF and millimeter wavelength bands has peaked in the last few years. However, it is heavily characterized by time-consuming redesign iterations due to the catastrophic impact of cross-coupling noise from nanometer interconnects. These circuits/systems demand performance-driven

routing mechanisms (including parasitic inductive phenomena), further highly-specific routing constraints, and 45° routing [148], which has not been broadly supported in existing routing tools and requires a profound investigation.

B. MULTI-PATTERNING

Analog design automation in advanced integration nodes, including FinFET, opened an optimal place for PRGs, such as BAG [23] and BAG2 [25], but also mixed-PRG/TbRG LAYGO [26]. However, more studies are foreseen in a near future exploiting the complex and unintuitive multi-patterning design rules to reduce the design space of routing mechanisms with higher automation levels, whose work conducted so far is limited.

C. ML/DL PREDICTION

The way to enhance existent A/RF routing methods with ML/DL techniques is still in an embryonic state, and many ideas are likely to be drawn from the path already paved in digital routing assisted by ML/DL. Several different works on digital routing violation prediction have been proposed, as rip-up and reroute techniques are also costly processes in digital techniques. In [178] different ML techniques were explored to predict design rule violations after digital global routing, using information related to density, connectivity, and pin proximity. References [179], [180], and [181] discard the global routing information and predict routing short violations right after placement, either using artificial neural networks or convolutional neural networks (CNNs). Slightly different from violation prediction is design rule hotspot prediction, where critical areas with multiple violations are identified. Traditional hotspot prediction algorithms forecast hotspots after global routing, however, with the help of CNNs, [182] achieves hotspot prediction after placement with high accuracy. These routing violation predictions or hotspot detection techniques have been used to enhance the efficiency and performance of routing processes, such as in [183], where the routing problem is converted into a reinforcement learning-based collaborative system, where one agent detects violations and the other agent is trained to perform routing.

Some of these ideas have been preliminarily explored for analog routing by the previously presented GeniousRoute [165], with the use of routing probability maps. However, many studies with ML/D techniques cooperating with traditional methods are likely to appear in the literature within the next few years [9]. In these, probability maps or violation/congestion regions predicted by M/DL will be used to: (1) assist SPfGs by reducing/adapting the underlying grids or defining net ordering; (2) assist SPfGs by providing predictive information in the cost function; (3) improve the sampling efficiency of EA-based methods by exploring regions of the search space with higher potential, and thus pursuing drastic run-time reductions; or even (4) reduce problem complexity of an ILP formulation by constraining the search space, pursuing the applicability of these methods for large design problems.

D. GENERATIVE ML/DL

While previous methods extract limited features from legacy data, recent surveys suggest that M/DL may play a supporting role in routing problems only in the short term [188]. The latest studies in the digital domain [184], [187] reveal that M/DL techniques may automatically exploit massive datasets and be trained to effectively generate the paths. These methods are attempting to learn the design styles from the dataset via CNNs or deep reinforcement learning, however, at this stage, most of the successful generative capabilities were attained on the global routing only. At the time of writing, detailed interconnect prediction via ML/DL is still on the horizon and remains to be proved, nonetheless, current research efforts point in that direction. Similarly, for A/RF, methodologies capable of taking advantage of legacy layout libraries, either drawn by circuit designers or generated by automatic tools, using methods such as convolutional variational autoencoders or adversarial networks to identify patterns and use its generative capabilities, are to be studied. While, at first, these techniques are expected to produce detailed paths of a significantly lower quality than established routing techniques, further breakthroughs may be attained in long term.

E. LAYOUT-AWARE SYNTHESIS

Regarding parasitic coupling, three different types of routing algorithms were overviewed: no parasitic information is considered; different classifications for the nets are assumed, e.g., noisy/sensitive, and then the algorithm attempts to departure them; and, by quantifying them in the exact amount of parasitic (R and C) or coupling (CC). Still, even in the last case correlation between minimized parasitic structures and the circuit's functional behavior is far from simple, and the circuit's functional constraints can only be verified through post-layout circuit/system simulations and the assessment of performance degradation made at this stage.

Overviewed LaS approaches are a major step towards closing the gap between electric and physical design steps, however, its convergence is strongly dependent on the quality of the automatic layout generator, including the router. Clarifying, in most of the LaS proposed none of the routings generated in-the-loop balanced parasitic coupling and noise [34], [35], [36], [37], [38], [39], [40], [41], [78], [79], [80], [81], [82], [142], [159], [160], [161], [162]. Thus, the LaS adapts the sizing according to the layout generator's peculiarities or deficiencies. The lack of accurate and thorough performance-aware routing generators in the loop is expected to be investigated since it is currently compromising the optimality of the automatic analog flow. Beyond this aspect, near-optimal results will still be difficult to achieve within acceptable time frames due to the computational effort required to simulate a high number of extracted netlists. Here, helpers based on M/DL, that keep simulator-grade accuracy

in-the-loop, have been recently proposed [189] and must be extensively studied within LaS concept.

F. HIERARCHICAL ROUTING

Due to its increased complexity, the automatic routing of a complete system has to be tackled differently from the circuit level. These concepts have been discussed in several works [126], [135], where the routing constraints have to be successfully propagated through the system hierarchy, and, the different pins/ports made available to the levels below/above. Similarly, in ALIGN [168], [169], [170], [171], electrical performance metrics for the system are percolated down to individual sub-blocks and translated into layout rules, with the specific routing tasks being integrated into each hierarchical level.

G. ROUTING BENCHMARK

The lack of open benchmarks, mostly due to technological and intellectual property barriers, prevents the proper comparison between approaches developed for A/RF IC automatic routing. This is also reflected in the lack of training data, which has been a major challenge in the development of M/DL-based EDA algorithms. Recent developments in open-source PDKs may open doors to the elaboration of a unified test circuit benchmark suite that allows to efficiently compare different routing algorithms, while simultaneously generating a high volume of training data at low cost.

IX. CONCLUSION

This paper extensively reviewed A/RF IC routing techniques from the digitally-inspired earliest approaches to state-of-the-art developments. A/RF IC routing tools try to follow digital IC trends. However, the need for detailed accounting for unwanted behaviors due to routing keeps pushing for techniques that leverage the smaller scale (compared to digital ICs) to the more detailed and holistic routing approaches. This review provides a complete and comprehensive guide for circuit designers and design automation developers while defining research lines to facilitate more activities within this field.

REFERENCES

[1] BSIM. *(Berkeley Short-Channel IGFET Model) Group*. Accessed: Mar. 18, 2023. [Online]. Available: https://bsim.berkeley.edu/

[2] A. K. W. Chee, "Quantitative dopant profiling by energy filtering in the scanning electron microscope," *IEEE Trans. Device Mater. Rel.*, vol. 16, no. 2, pp. 138–148, Jun. 2016.

[3] A. K. W. Chee, "Fermi level pinning characterisation on ammonium fluoride-treated surfaces of silicon by energy-filtered doping contrast in the scanning electron microscope," *Sci. Rep.*, vol. 6, no. 1, p. 32003, Aug. 2016.

[4] P. Schweizer, P. Denninger, C. Dolle, and E. Spiecker, "Low energy nano diffraction (LEND)—A versatile diffraction technique in SEM," *Ultramicroscopy*, vol. 213, Jun. 2020, Art. no. 112956.

[5] *Design Rule Complexity Rising*. Accessed: Mar. 18, 2023. [Online]. Available: https://semiengineering.com/design-rule-complexity-rising/

[6] H. Graeb, Ed., *Analog Layout Synthesis: A Survey of Topological Approaches*. New York, NY, USA: Springer, 2011.

[7] M. P.-H. Lin, Y.-W. Chang, and C.-M. Hung, "Recent research development and new challenges in analog layout synthesis," in *Proc. 21st Asia South Pacific Design Autom. Conf. (ASP-DAC)*, Jan. 2016, pp. 617–622.

[8] R. Martins, N. Lourenço, and N. Horta, *Analog Integrated Circuit Design Automation: Placement, Routing and Parasitic Extraction Techniques*. Cham, Switzerland: Springer, 2017.

[9] E. Afacan, N. Lourenço, R. Martins, and G. Dündar, "Review: Machine learning techniques in analog/RF integrated circuit design, synthesis, layout, and test," *Integration*, vol. 77, pp. 113–130, Mar. 2021.

[10] S. N. Ong et al., "A 22 nm FDSOI technology optimized for RF/mmWave applications," in *Proc. IEEE Radio Freq. Integr. Circuits Symp. (RFIC)*, Philadelphia, PA, USA, Jun. 2018, pp. 72–75, doi: 10.1109/RFIC.2018.8429035.

[11] E.-H.-V. Yeh, A.-H. Lo, W.-S. Chen, T.-J. Yeh, and M. Chen, "A 16 nm FinFET 0.4 V inductor-less cellular receiver front-end with 10 mW ultra-low power and 0.31 mm^2 ultra-small area for 5G system in sub-6 GHz band," in *Proc. Symp. VLSI Circuits*, Kyoto, Japan, Jun. 2017, doi: 10.23919/VLSIC.2017.8008553.

[12] H. Graeb, F. Balasa, R. Castro-Lopez, Y.-W. Chang, F. V. Fernandez, P.-H. Lin, and M. Strasser, "Analog layout synthesis—Recent advances in topological approaches," in *Proc. Design, Autom. Test Eur. Conf. Exhib.*, Apr. 2009, pp. 274–279.

[13] H. Zhou, C.-W. Sham, and H. Yao, "Revisiting routability-driven placement for analog and mixed-signal circuits," *ACM Trans. Design Autom. Electron. Syst.*, vol. 23, no. 2, pp. 1–17, Mar. 2018.

[14] H.-Y. Chi, H.-C. Chang, C.-H. Yang, C.-N. Liu, and J.-Y. Jou, "Performance-driven routing methodology with incremental placement refinement for analog layout design," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Feb. 2021, pp. 1218–1223.

[15] G. Van der Plas, G. Gielen, and W. Sansen, *A Computer-Aided Design and Synthesis Environment*. Norwell, MA, USA: Kluwer, 2002.

[16] A. K. Sharma, M. Madhusudan, S. M. Burns, S. Yaldiz, P. Mukherjee, R. Harjani, and S. S. Sapatnekar, "Performance-aware common-centroid placement and routing of transistor arrays in analog circuits," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Design (ICCAD)*, Nov. 2021, pp. 1–9.

[17] R. J. Lipton, S. C. North, J. Valdes, G. Vijayan, and R. Sedgewick, "ALI: A procedural language to describe VLSI layouts," in *Proc. ACM/IEEE Design Autom. Conf.*, Jun. 1982, pp. 467–474.

[18] T. Davis and J. Clark, "SILT: A VLSI design language," Digit. Syst. Lab., Stanford Univ., Stanford, CA, USA, Tech. Rep. 226, Oct. 1982.

[19] W. H. Evans, J. C. Ballegeer, and N. H. Duyet, "ADL: An algorithmic design language for integrated circuit synthesis," in *Proc. DAC*, Jun. 1984, pp. 66–72.

[20] W. E. Cory, "Layla: A VLSI layout language," in *Proc. ACM/IEEE DAC*, Jun. 1985, pp. 245–251.

[21] B. R. Owen, R. Duncan, S. Jantzi, C. Ouslis, S. Rezania, and K. Martin, "BALLISTIC: An analog layout language," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, Aug. 1995, pp. 41–44.

[22] H. Sampath and R. Vemuri, "MSL: A High-level language for parameterized analog and mixed signal layout generators," in *Proc. IFIP Int. Conf. VLSI*, Jan. 2003, pp. 416–421.

[23] J. Crossley, A. Puggelli, H.-P. Le, B. Yang, R. Nancollas, K. Jung, L. Kong, N. Narevsky, Y. Lu, N. Sutardja, E. J. An, A. L. Sangiovanni-Vincentelli, and E. Alon, "BAG: A designer-oriented integrated framework for the development of AMS circuit generators," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2013, pp. 74–81.

[24] S. Alassi and B. Winter, "PyCells for an open semiconductor industry," 2016, *arXiv:1607.00859*.

[25] E. Chang, J. Han, W. Bae, Z. Wang, N. Narevsky, B. NikoliC, and E. Alon, "BAG2: A process-portable framework for generator-based AMS circuit design," in *Proc. IEEE CICC*, Apr. 2018, pp. 1–8.

[26] J. Han, W. Bae, E. Chang, Z. Wang, B. Nikolić, and E. Alon, "LAYGO: A template-and-grid-based layout generation engine for advanced CMOS technologies," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 3, pp. 1012–1022, Mar. 2021.

[27] *Git, a Free and Open Source Distributed Version Control System*. Accessed: Mar. 18, 2023. [Online]. Available: https://git-scm.com

[28] *Conda, a Package, Dependency and Environment Management for Any Language*. Accessed: Mar. 18, 2023. [Online]. Available: https://docs.conda.io/en/latest

[29] A. Agarwal, H. Sampath, V. Yelamanchili, and R. Vemuri, "Fast and accurate parasitic capacitance models for layout-aware," in *Proc. DAC*, Jun. 2004, pp. 145–150.

[30] M. Ranjan, W. Verhaegen, A. Agarwal, H. Sampath, R. Vemuri, and G. Gielen, "Fast, layout-inclusive analog circuit synthesis using pre-compiled parasitic-aware symbolic performance models," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Feb. 2004, pp. 604–609.

[31] A. Pradhan and R. Vemuri, "Efficient synthesis of a uniformly spread layout aware Pareto surface for analog circuits," in *Proc. 22nd Int. Conf. VLSI Design*, Jan. 2009, pp. 131–136.

[32] M. Ranjan, A. Bhaduri, W. Verhaegen, B. Mukherjee, R. Vemuri, G. Gielen, and A. Pacelli, "Use of symbolic performance models in layout-inclusive RF low noise amplifier synthesis," in *Proc. IEEE Int. Behav. Modeling Simulation Conf.*, Oct. 2004, pp. 130–134.

[33] H. Onodera, H. Kanbara, and K. Tamaru, "Operational amplifier compilation with performance optimization," in *Proc. CICC*, May 1989, pp. 17.4/1–17.4/6.

[34] H. Onodera, H. Kanbara, and K. Tamaru, "Operational-amplifier compilation with performance optimization," *IEEE J. Solid-State Circuits*, vol. 25, no. 2, pp. 466–473, Apr. 1990.

[35] M. Dessouky and M.-M. Louerat, "A layout approach for electrical and physical design integration of high-performance analog circuits," in *Proc. IEEE 1st Int. Symp. Quality Electron. Design (ISQED)*, Mar. 2000, pp. 291–298.

[36] M. Dessouky, M.-M. Louërat, and J. Porte, "Layout-oriented synthesis of high performance analog circuits," in *Proc. DATE*, Jan. 2000, pp. 53–57.

[37] A. Bhaduri, V. Vijay, A. Agarwal, and R. Vemuri, "Parasitic-aware synthesis of RF LNA circuits considering quasi-static extraction of inductors and interconnects," in *Proc. 47th Midwest Symp. Circuits Syst.*, Nov. 2004, p. 477.

[38] S. J. Patel and R. A. Thakker, "Parasitic aware automatic analog CMOS circuit design environment using ABC algorithm," in *Proc. 31st Int. Conf. VLSI Design 17th Int. Conf. Embedded Syst. (VLSID)*, Jan. 2018, pp. 445–446.

[39] S. J. Patel and R. A. Thakker, "Parasitic-aware automatic analog CMOS circuit design environment," in *Proc. 32nd Int. Conf. VLSI Design 18th Int. Conf. Embedded Syst. (VLSID)*, Jan. 2019.

[40] *Magic VLSI Layout Tool*. Accessed: Mar. 18, 2023. [Online]. Available: http://opencircuitdesign.com/magic/

[41] K. Hakhamaneshi, N. Werblun, P. Abbeel, and V. Stojanovic, "BagNet: Berkeley analog generator with layout optimizer boosted with deep neural networks," in *Proc. ICCAD*, Nov. 2019, pp. 1–8.

[42] J. D. Conway and G. G. Schrooten, "An automatic layout generator for analog circuits," in *Proc. Eur. Conf. Design Autom.*, Mar. 1992, pp. 513–514.

[43] M. F. Chowdhury and R. E. Massara, "An expert system for general purpose analogue layout synthesis," in *Proc. 33rd Midwest Symp. Circuits Syst.*, 1990, pp. 1171–1174.

[44] M. F. Chowdhury, R. E. Massara, and H. Tang, "Analogue layout synthesis based on a planning scheme using artificial intelligence," in *Proc. IEEE Int. Symp. Circuits Syst.*, Jun. 1991, pp. 3094–3097.

[45] R. Castro-López, F. Fernández, M. Delgado-Restituto, F. Medeiro, and A. Rodriguez-Vazquez, "Creating flexible analogue IP blocks," in *Proc. Eur. Solid-State Circuits Conf.*, Sep. 2001, pp. 437–440.

[46] R. Castro-López, F. Fernández, F. Medeiro, and A. Rodríguez-Vázquez, "Generation of technology-independent retargetable analog blocks," *Analog Integr. Circuits Signal Process.*, vol. 33, no. 2, pp. 157–170, Nov. 2002.

[47] *Cadence Virtuoso Analog Design Environment*. Accessed: Mar. 18, 2023. [Online]. Available: https://www.cadence.com/virtuoso-analog-design-environment

[48] S. Hammouda, H. Said, M. Dessouky, M. Tawfik, Q. Nguyen, W. Badawy, H. Abbas, and H. Shahein, "Chameleon ART: A non-optimization based analog design migration framework," in *Proc. ACM/IEEE DAC*, Jul. 2006, pp. 885–888.

[49] F. A. Naguib, S. Ahmed, S. Hamed, and M. Dessouky, "Expert guided analog layout placement and routing automation for deep nanotechnologies," in *Proc. 37th Nat. Radio Sci. Conf. (NRSC)*, Sep. 2020, pp. 240–247.

[50] N. Lourenco and N. Horta, "LAYGEN—An evolutionary approach to automatic analog IC layout generation," in *Proc. 12th IEEE Int. Conf. Electron., Circuits Syst.*, Dec. 2005, pp. 1–4.

[51] N. Lourenço, M. Vianello, J. Guilherme, and N. Horta, "LAYGEN—Automatic layout generation of analog ICs from hierarchical template descriptions," in *Proc. Ph.D. Res. Microelectron. Electron.*, Jun. 2006, pp. 213–216.

[52] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, MA, USA: MIT Press, 1996.

[53] A. Unutulmaz, G. Dundar, and F. V. Fernandez, "A template router," in *Proc. 20th Eur. Conf. Circuit Theory Design (ECCTD)*, Aug. 2011, pp. 334–337.

[54] A. Unutulmaz, G. Dündar, and F. V. Fernández, "Template coding with LDS and applications of LDS in EDA," *Anal. Integr. Circuits Signal Process.*, vol. 78, no. 1, pp. 137–151, Jan. 2014.

[55] A. Unutulmaz, G. Dundar, and F. V. Fernandez, "LDS—A description script for layout templates," in *Proc. ECCTD*, Aug. 2011, pp. 857–860.

[56] N. Nilsson, *Problem-Solving Methods in Artificial Intelligence*. New York, NY, USA: McGraw-Hill, 1971.

[57] M. Hanan, "On Steiner's problem with rectilinear distance," *SIAM J. Appl. Math.*, vol. 14, no. 2, pp. 255–265, Mar. 1966.

[58] N. Jangkrajarng, S. Bhattacharya, R. Hartono, and C.-J.-R. Shi, "IPRAIL—Intellectual property reuse-based analog IC layout automation," *Integration*, vol. 36, no. 4, pp. 237–262, Nov. 2003.

[59] S. Bhattacharya, N. Jangkrajarng, R. Hartono, and C.-J.-R. Shi, "Correct-by-construction layout-centric retargeting of large analog designs," in *Proc. ACM/IEEE DAC*, Jun. 2004, pp. 139–144.

[60] N. Jangkrajarng, S. Bhattacharya, R. Hartono, and C.-J.-R. Shi, "Multiple specifications radio-frequency integrated circuit design with automatic template-driven layout retargeting," in *Proc. ASP-DAC*, Jan. 2004, pp. 394–399.

[61] A. Unutulmaz, G. Dundar, and F. V. Fernandez, "LDS based tools to ease template construction," in *Proc. Int. Conf. SMACD*, Sep. 2012, pp. 61–64.

[62] C.-Y. Chin, P.-C. Pan, H.-M. Chen, T.-C. Chen, and J.-C. Lin, "Efficient analog layout prototyping by layout reuse with routing preservation," in *Proc. IEEE/ACM ICCAD*, Nov. 2013, pp. 40–47.

[63] P.-C. Pan, C.-Y. Chin, H.-M. Chen, T.-C. Chen, C.-C. Lee, and J.-C. Lin, "A fast prototyping framework for analog layout migration with planar preservation," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 9, pp. 1373–1386, Sep. 2015.

[64] J. R. Shewchuk, "Delaunay refinement algorithms for triangular mesh generation," *Comput. Geometry*, vol. 22, nos. 1–3, pp. 21–74, May 2002.

[65] H.-Y. Chi, Z.-J. Lin, C.-H. Hung, C.-N.-J. Liu, and H.-M. Chen, "Achieving routing integrity in analog layout migration via Cartesian detection lines," in *Proc. IEEE/ACM ICCAD*, Nov. 2019, pp. 1–6.

[66] H.-Y. Chi, Z.-J. Lin, C.-H. Hung, C.-N.-J. Liu, and H.-M. Chen, "A style-based analog layout migration technique with complete routing behavior preservation," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 40, no. 12, pp. 2556–2567, Dec. 2021.

[67] M. P.-H. Lin, H.-Y. Chi, A. Patyal, Z.-Y. Liu, J.-J. Zhao, C.-N.-J. Liu, and H.-M. Chen, "Achieving analog layout integrity through learning and migration," in *Proc. IEEE/ACM ICCAD*, Nov. 2020, pp. 1–8.

[68] P.-H. Wu, M. P.-H. Lin, and T.-Y. Ho, "Analog layout synthesis with knowledge mining," in *Proc. ECCTD*, Aug. 2015, pp. 1–4.

[69] P.-H. Wu, P.-H. Lin, T.-C. Chen, C.-F. Yeh, X. Li, and T.-Y. Ho, "A novel analog physical synthesis methodology integrating existent design expertise," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 2, pp. 199–212, Feb. 2015.

[70] U. Choudhury and A. Sangiovanni-Vincentelli, "Constraint generation for routing analog circuits," in *Proc. DAC*, 1990, pp. 561–566.

[71] G. Gad-El-Karim and R. S. Gyurcsik, "Generation of performance sensitivities for analog cell layout," in *Proc. DAC*, 1991, pp. 500–505.

[72] S. Bhattacharya, N. Jangkrajarng, and C.-J.-R. Shi, "Template-driven parasitic-aware optimization of analog integrated circuit layouts," in *Proc. DAC*, 2005, pp. 644–647.

[73] N. Jangkrajarng, L. Zhang, S. Bhattacharya, N. Kohagen, and C.-J. Shi, "Template-based parasitic-aware optimization and retargeting of analog and RF integrated circuit layouts," in *Proc. IEEE/ACM ICCAD*, Nov. 2006, pp. 342–348.

[74] L. Zhang, N. Jangkrajarng, S. Bhattacharya, and C.-J.-R. Shi, "Parasitic-aware optimization and retargeting of analog layouts: A symbolic-template approach," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 5, pp. 791–802, May 2008.

[75] Z. Liu and L. Zhang, "A performance-constrained template-based layout retargeting algorithm for analog integrated circuits," in *Proc. ASP-DAC*, Jan. 2010, pp. 293–298.

[76] Z. Liu and L. Zhang, "Performance-constrained template-driven retargeting for analog and RF layouts," in *Proc. 20th Symp. Great lakes Symp. VLSI*, May 2010, pp. 429–434.

[77] L. Zhang and Z. Liu, "Directly performance-constrained template-based layout retargeting and optimization for analog integrated circuits," *Integration*, vol. 44, no. 1, pp. 1–11, Jan. 2011.

[78] R. Castro-López, O. Guerra, E. Roca, and F. V. Fernández, "An integrated layout-synthesis approach for analog ICs," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 7, pp. 1179–1189, Jul. 2008.

[79] A. Toro-Frías, R. Castro-López, E. Roca, and F. V. Fernández, "An automated layout-aware design flow," in *Proc. Int. Conf. SMACD*, Sep. 2012, pp. 73–76.

[80] A. Toro-Frías, R. Castro-López, E. Roca, and F. V. Fernández, "Layout-aware Pareto fronts of electronic circuits," in *Proc. ECCTD*, Aug. 2011, pp. 345–348.

[81] G. Berkol, A. Unutulmaz, E. Afacan, G. Dundar, F. V. Fernandez, A. E. Pusane, and F. Baskaya, "A two-step layout-in-the-loop design automation tool," in *Proc. IEEE 13th Int. New Circuits Syst. Conf. (NEWCAS)*, Jun. 2015, pp. 1–4.

[82] Y.-C. Liao, Y.-L. Chen, X.-T. Cai, C.-N. Liu, and T.-C. Chen, "LASER: Layout-aware analog synthesis environment on laker," in *Proc. 23rd ACM Int. Conf. Great Lakes Symp. VLSI*, May 2013, pp. 107–112.

[83] H.-J. Chang, Y.-L. Chen, C. Yeh, and C.-N.-J. Liu, "Layout-aware analog synthesis environment with yield consideration," in *Proc. ISQED*, Mar. 2015, pp. 589–593.

[84] *Laker*. Accessed: Mar. 18, 2023. [Online]. Available: https://www.synopsys.implementation-and-signoff/custom-design-platform/laker-custom-design.html

[85] P. M. Maurer, "Automatic routing of integrated circuit connections: A tutorial," in *Proc. IEEE Int. Conf. Commun., Including Supercomm Tech. Sessions*, Apr. 1990, pp. 801–805.

[86] H. H. Chen and E. S. Kuh, "Glitter: A gridless variable-width channel router," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. CAD-5, no. 4, pp. 459–465, Oct. 1986.

[87] H. Shin and A. Sangiovanni-Vincentelli, "Mighty: A 'rip-up and reroute' detailed router," in *Proc. ICCAD*, 1986, pp. 2–5.

[88] H. Shin and A. Sangiovanni-Vincentelli, "A detailed router based on incremental routing modifications: Mighty," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. CAD-6, no. 6, pp. 942–955, Nov. 1987.

[89] K. Han Young, C. H. Sequin, and P. R. Gray, "Automatic layout generation for CMOS operational amplifiers," in *Proc. ICCAD*, 1988, pp. 548–549.

[90] H. Koh, C. Sequin, and P. Gray, "OPASYN: A compiler for CMOS operational amplifiers," *IEEE TCAD*, vol. 9, no. 2, pp. 113–125, Feb. 1990.

[91] D. J. Chen and B. J. Sheu, "Generalised approach to automatic custom layout of analogue ICs," *IEE Proc. G, Circuits, Devices Syst.*, vol. 139, no. 4, p. 481, 1992.

[92] D. J. Chen, J.-C. Lee, and B. J. Sheu, "SLAM: A smart analog module layout generator for mixed analog-digital VLSI design," in *Proc. ICCD*, 1989, pp. 24–25.

[93] J.-C. Lee, S. M. Gowda, and B. J. Sheu, "Fully automated layout generators for high-performance analog VLSI modules," in *Proc. 4th IEEE Region 10 Int. Conf.*, Nov. 1989, pp. 893–896.

[94] R. Gyurcsik and J.-C. Jeen, "A generalized approach to routing mixing analog and digital signal nets in a channel," *IEEE J. Solid-State Circuits*, vol. 24, no. 2, pp. 436–442, Apr. 1989.

[95] U. Choudhury and A. Sangiovanni-Vincentelli, "Constraint-based channel routing for analog and mixed analog/digital circuits," in *Proc. ICCAD*, 1990, pp. 198–199.

[96] U. Choudhury and A. Sangiovanni-Vincentelli, "Constraint-based channel routing for analog and mixed analog/digital circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Systems*, vol. 12, no. 4, pp. 497–510, Apr. 1993.

[97] J. Rijmenants, T. Schwarz, J. Litsios, and R. Zinszner, "ILAC: An automated layout tool for analog CMOS circuits," in *Proc. CICC*, May 1988, pp. 7.6/1–7.6/4.

[98] J. Rijmenants, J. B. Litsios, T. R. Schwarz, and M. G. R. Degrauwe, "ILAC: An automated layout tool for analog CMOS circuits," *IEEE J. Solid-State Circuits*, vol. 24, no. 2, pp. 417–425, Apr. 1989.

[99] M. G. R. Degrauwe, B. L. A. G. Goffart, C. Meixenberger, M. L. A. Pierre, J. B. Litsios, J. Rijmenants, O. J. A. P. Nys, E. Dijkstra, B. Joss, M. K. C. M. Meyvaert, T. R. Schwarz, and M. D. Pardoen, "Towards an analog system design environment," *IEEE J. Solid-State Circuits*, vol. 24, no. 3, pp. 659–671, Jun. 1989.

[100] M. Mogaki, N. Kato, Y. Chikami, N. Yamada, and Y. Kobayashi, "LADIES: An automatic layout system for analog LSI's," in *Proc. ICCAD*, 1989, pp. 450–451.

[101] C.-P. Hsu, "General river routing algorithm," in *Proc. 20th Design Autom. Conf.*, Jun. 1983, pp. 578–583.

[102] M. Kayal, S. Piguet, M. Declercq, and B. Hochet, "An interactive layout generation tool for CMOS analog ICs," in *Proc. IEEE ISCAS*, Jun. 1988, pp. 2431–2434.

[103] M. Kayal, S. Piguet, M. Declercq, and B. Hochet, "SALIM: A layout generation tool for analog ICs," in *Proc. IEEE CICC*, May 1988, pp. 7.5/1–7.5/4.

[104] S. Piguet, F. Rahali, M. Kayal, E. Zysman, and M. Declercq, "A new routing method for full custom analog ICs," in *Proc. IEEE CICC*, May 1990, pp. 27.7/1–27.7/4.

[105] Z.-M. Lin, "Global routing techniques for an automatic mixed analog/digital IC layout compiler," in *Proc. IEEE SOUTHEASTCON*, Apr. 1991, pp. 392–396.

[106] L. Zhang, U. Kleine, and M. Wolf, "Automatic inner wiring for integrated analog modules," in *Proc. Int. Conf. Mixed Design Integr. Circuits Syst.*, Jun. 2001, pp. 109–114.

[107] L. Zhang and U. Kleine, "A novel analog layout synthesis tool," in *Proc. IEEE ISCAS*, May 2004, p. 5.

[108] L. Zhang, U. Kleine, and Y. Jiang, "An automated design tool for analog layouts," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 8, pp. 881–894, Aug. 2006.

[109] L. Zhang, U. Kleine, R. Raut, and Y. Jiang, "Aladin: A layout synthesis tool for analog integrated circuits," *Anal. Integr. Circuits Signal Process.*, vol. 46, no. 3, pp. 215–230, Mar. 2006.

[110] E. Yilmaz and G. Dundar, "New layout generator for analog CMOS circuits," in *Proc. 18th Eur. Conf. Circuit Theory Design*, Aug. 2007, pp. 36–39.

[111] E. Yilmaz and G. Dundar, "Analog layout generator for CMOS circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 28, no. 1, pp. 32–45, Jan. 2009.

[112] P. H. Wei and B. Murmann, "Analog and mixed-signal layout automation using digital place-and-route tools," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 29, no. 11, pp. 1838–1849, Nov. 2021.

[113] E. Malavasi and A. Sangiovanni-Vincentelli, "Area routing for analog layout," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 12, no. 8, pp. 1186–1197, Aug. 1993.

[114] V. Bexten, C. Moraga, R. Klinke, W. Brockherde, and K.-G. Hess, "ALSYN: Flexible rule-based layout synthesis for analog ICs," *IEEE J. Solid-State Circuits*, vol. 28, no. 3, pp. 261–268, Mar. 1993.

[115] L. Xiao, E. F. Y. Young, X. He, and K. P. Pun, "Practical placement and routing techniques for analog circuit designs," in *Proc. ICCAD*, Nov. 2010, pp. 675–679.

[116] W. Heyns, W. Sansen, and H. Beke, "A line-expansion algorithm for the general routing problem with a guaranteed solution," in *Proc. 17th Design Autom. Conf.*, Jun. 1980, pp. 243–249.

[117] D. J. Garrod, R. A. Rutenbar, and L. R. Carley, "Automatic layout of custom analog cells in ANAGRAM," in *Proc. IEEE ICCAD*, Nov. 1988, pp. 544–545.

[118] J. M. Cohn, D. J. Garrod, R. A. Rutenbar, and L. R. Carley, "KOAN/ANAGRAM II: New tools for device-level analog placement and routing," *IEEE J. Solid-State Circuits*, vol. 26, no. 3, pp. 330–342, Mar. 1991.

[119] J. M. Cohn, D. J. Garrod, R. A. Rutenbar, and L. R. Carley, "New algorithms for placement and routing of custom analog cells in ACACIA," in *Proc. IEEE CICC*, May 1990, pp. 27.6/1–27.6/5.

[120] J. Dion and L. Monier, "Contour: A tile-based gridless router," Western Res. Lab., 250 Univ. Avenue Palo Alto, CA, USA, Res. Rep., 1995.

[121] P.-H. Lin, H.-C. Yu, T.-H. Tsai, and S.-C. Lin, "A matching-based placement and routing system for analog design," in *Proc. Int. Symp. VLSI Design, Autom. Test (VLSI-DAT)*, Apr. 2007, pp. 1–4.

[122] Q. Gao, Y. Shen, Y. Cai, and H. Yao, "Analog circuit shielding routing algorithm based on net classification," in *Proc. 16th ACM/IEEE Int. Symp. Low Power Electron. Design*, Aug. 2010, pp. 123–128.

[123] W. Chen, H. Yao, Y. Cai, and Q. Zhou, "Analog routing considering min-area constraint," in *Proc. IEEE 10th Int. Conf. ASIC*, Oct. 2013, pp. 1–4.

[124] H. Chen, K. Zhu, M. Liu, X. Tang, N. Sun, and D. Z. Pan, "Toward silicon-proven detailed routing for analog and mixed-signal circuits," in *Proc. ICCAD*, Nov. 2020, pp. 1–8.

[125] H.-Y. Chi, H.-Y. Tseng, C.-N.-J. Liu, and H.-M. Chen, "Performance-preserved analog routing methodology via wire load reduction," in *Proc. ASP-DAC*, Jan. 2018, pp. 482–487.

[126] Y.-H. Chen, H.-Y. Chi, L.-Y. Song, C.-N.-J. Liu, and H.-M. Chen, "A structure-based methodology for analog layout generation," in *Proc. Int. Conf. SMACD*, Jul. 2019, pp. 33–36.

[127] H. Y. Chi, C. N. J. Liu, and H. M. Chen, "Wire load oriented analog routing with matching constraints," *ACM Trans. Design Autom. Electron. Syst.*, vol. 25, no. 6, pp. 1–26, Aug. 2020.

[128] M. M. Ozdal and R. F. Hentschke, "Exact route matching algorithms for analog and mixed signal integrated circuits," in *Proc. ICCAD*, Nov. 2009, pp. 231–238.

[129] M. Ozdal and R. F. Hentschke, "An algorithmic study of exact route matching for integrated circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 30, no. 12, pp. 1842–1855, Dec. 2011.

[130] M. M. Ozdal and R. F. Hentschke, "Maze routing algorithms with exact matching constraints for analog and mixed signal designs," in *Proc. ICCAD*, Nov. 2012, pp. 130–136.

[131] M. Ozdal and R. F. Hentschke, "Algorithms for maze routing with exact matching constraints," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 33, no. 1, pp. 101–112, Jan. 2014.

[132] H. Yao, Y. Cai, and Q. Gao, "LEMAR: A novel length matching routing algorithm for analog and mixed signal circuits," in *Proc. IEEE/ACM ASP-DAC*, Jan. 2012, pp. 157–162.

[133] C. Du, Y. Cai, and X. Hong, "A novel analog routing algorithm with constraints of variable wire widths," in *Proc. Int. Conf. Commun., Circuits Syst.*, Jun. 2006, pp. 2459–2463.

[134] J.-T. Yan and Z.-W. Chen, "Obstacle-aware multiple-source rectilinear Steiner tree with electromigration and IR-drop avoidance," in *Proc. DATE*, Mar. 2011, pp. 1–6.

[135] P.-C. Pan, H.-M. Chen, Y.-K. Cheng, J. Liu, and W.-Y. Hu, "Configurable analog routing methodology via technology and design constraint unification," in *Proc. IEEE/ACM ICCAD*, Nov. 2012, pp. 620–626.

[136] E. Malavasi, M. Chilanti, and R. Guerrieri, "A general router for analog layout," in *Proc. VLSI Comput. Peripherals*, 1989, pp. 5–49.

[137] U. Choudhury and A. Sangiovanni-Vincentelli, "Use of performance sensitivities in routing analog circuits," in *Proc. IEEE ISCAS*, May 1990, pp. 348–351.

[138] E. Malavasi, U. Choudhury, and A. Sangiovanni-Vincentelli, "A routing methodology for analog integrated circuits," in *Proc. ICCAD*, 1990, pp. 202–205.

[139] E. Malavasi, E. Charbon, E. Felt, and A. Sangiovanni-Vincentelli, "Automation of IC layout with analog constraints," *IEEE Trans. Comput.-Aided Design Integr.*, vol. 15, no. 8, pp. 923–942, Aug. 1996.

[140] K. Lampaert, G. Gielen, and W. Sansen, "Analog routing for manufacturability," in *Proc. IEEE CICC*, May 1996, pp. 175–178.

[141] K. Lampaert, G. Gielen, and W. Sansen, *Analog Layout Generation for Performance and Manufacturability*. Norwell, MA, USA: Kluwer, 1999.

[142] H. Habal and H. Graeb, "Constraint-based layout-driven sizing of analog circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 30, no. 8, pp. 1089–1102, Aug. 2011.

[143] *IC Shape-Based Technology Chip Assembly User Guide*, Cadence Des. Syst., San Jose, CA, USA, Jun. 2003.

[144] P. Vancorenland, G. Van der Plas, M. Steyaert, G. Gielen, and W. Sansen, "A layout-aware synthesis methodology for RF circuits," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Design*, Nov. 2001, pp. 358–362.

[145] C. De Ranter, B. De Muer, G. Van der Plas, P. Vancorenland, M. Steyaert, G. Gielen, and W. Sansen, "CYCLONE: Automated design and layout of RF LC-oscillators," in *Proc. DAC*, 2000, pp. 11–14.

[146] C. R. C. De Ranter, G. van der Plas, M. S. J. Steyaert, G. G. E. Gielen, and W. M. C. Sansen, "CYCLONE: Automated design and layout of RF LC-oscillators," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 21, no. 10, pp. 1161–1170, Oct. 2002.

[147] H.-C. Ou, H.-C.-C. Chien, and Y.-W. Chang, "Non-uniform multilevel analog routing with matching constraints," in *Proc. DAC*, Jun. 2012, pp. 549–554.

[148] H.-C. Ou, H.-C.-C. Chien, and Y.-W. Chang, "Nonuniform multilevel analog routing with matching constraints," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 33, no. 12, pp. 1942–1954, Dec. 2014.

[149] M. Torabi and L. Zhang, "Efficient ILP-based variant-grid analog router," in *Proc. IEEE ISCAS*, May 2016, pp. 1266–1269.

[150] M. Torabi and L. Zhang, "Electromigration and parasitic-aware ILP router for analog and RF circuits," in *Proc. IEEE ISCAS*, May 2018, pp. 1–5.

[151] M. Torabi and L. Zhang, "Electromigration- and parasitic-aware ILP-based analog router," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 10, pp. 1854–1867, Oct. 2018.

[152] C.-Y. Wu, H. Graeb, and J. Hu, "A pre-search assisted ILP approach to analog integrated circuit routing," in *Proc. ICCD*, Oct. 2015, pp. 244–250.

[153] R. Martins, N. Lourenco, and N. Horta, *Generating Analog IC Layouts With LAYGEN II*. Berlin, Germany: Springer, 2012.

[154] R. Martins, N. Lourenço, and N. Horta, "Routing analog ICs using a multi-objective multi-constraint evolutionary approach," *Anal. Integr. Circuits Signal Process.*, vol. 78, no. 1, pp. 123–135, Jan. 2014.

[155] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[156] R. Martins, N. Lourenço, A. Canelas, and N. Horta, "Electromigration-aware and IR-drop avoidance routing in analog multiport terminal structures," in *Proc. DATE*, 2014, pp. 1–6.

[157] R. Martins, R. Póvoa, N. Lourenço, and N. Horta, "Current-flow and current-density-aware multi-objective optimization of analog IC placement," *Integration*, vol. 55, pp. 295–306, Sep. 2016.

[158] R. Martins, N. Lourenço, A. Canelas, and N. Horta, "Electromigration-aware analog router with multilayer multiport terminal structures," *Integration*, vol. 47, no. 4, pp. 532–547, Sep. 2014.

[159] R. Martins, N. Lourenço, N. Horta, S. Zhong, J. Yin, P. I. Mak, and R. P. Martins, "Design of a 4.2-to-5.1 GHz ultralow-power complementary class-B/C hybrid-mode VCO in 65-nm CMOS fully supported by EDA tools," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 11, pp. 3965–3977, Nov. 2020.

[160] R. Martins, N. Lourenco, F. Passos, R. Povoa, A. Canelas, E. Roca, R. Castro-Lopez, J. Sieiro, F. V. Fernandez, and N. Horta, "Two-step RF IC block synthesis with preoptimized inductors and full layout generation in-the-loop," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 38, no. 6, pp. 989–1002, Jun. 2019.

[161] F. Passos, E. Roca, R. Castro-López, and F. Fernández, *Automated Hierarchical Synthesis of Radio-Frequency Integrated Circuits and Systems: A Systematic and Multilevel Approach*. Cham, Switzerland: Springer, 2020.

[162] R. Povoa, N. Lourenco, R. Martins, A. Canelas, N. Horta, and J. Goes, "A folded voltage-combiners biased amplifier for low voltage and high energy-efficiency applications," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 67, no. 2, pp. 230–234, Feb. 2020.

[163] M. Fayazi, Z. Colter, E. Afshari, and R. Dreslinski, "Applications of artificial intelligence on the modeling and optimization for analog and mixed-signal circuits: A review," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 6, pp. 2418–2431, Jun. 2021.

[164] R. Mina, C. Jabbour, and G. E. Sakr, "A review of machine learning techniques in analog integrated circuit design automation," *Electronics*, vol. 11, no. 3, p. 435, Jan. 2022.

[165] K. Zhu, M. Liu, Y. Lin, B. Xu, S. Li, X. Tang, N. Sun, and D. Z. Pan, "GeniusRoute: A new analog routing paradigm using generative neural network guidance," in *Proc. ICCAD*, Nov. 2019, pp. 1–8.

[166] H. Liao, Q. Dong, X. Dong, W. Zhang, W. Zhang, W. Qi, E. Fallon, and L. B. Kara, "Attention routing: Track-assignment detailed routing using attention-based reinforcement learning," 2020, *arXiv:2004.09473*.

[167] H. Liao, Q. Dong, W. Qi, E. Fallon, and L. B. Kara, "Track-assignment detailed routing using attention-based policy model with supervision," in *Proc. MLCAD*, Nov. 2020, pp. 105–110.

[168] K. Kunal, M. Madhusudan, A. K. Sharma, W. Xu, S. M. Burns, R. Harjani, J. Hu, D. A. Kirkpatrick, and S. S. Sapatnekar, "ALIGN: Open-source analog layout automation from the ground up," in *Proc. 56th ACM/IEEE Annu. Design Autom. Conf. (DAC)*, Jun. 2019, pp. 1–4.

[169] T. Dhar, K. Kunal, Y. Li, Y. Lin, M. Madhusudan, J. Poojary, A. K. Sharma, S. M. Burns, R. Harjani, J. Hu, P. Mukherjee, S. Yaldiz, and S. S. Sapatnekar, "The ALIGN open-source analog layout generator: V1.0 and beyond," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2020, pp. 1–2.

[170] J. Liu, S. Su, M. Madhusudan, M. Hassanpourghadi, S. Saunders, Q. Zhang, R. Rasul, Y. Li, J. Hu, A. K. Sharma, S. S. Sapatnekar, R. Harjani, A. Levi, S. Gupta, and M. S.-W. Chen, "From specification to silicon: Towards analog/mixed-signal design automation using surrogate NN models with transfer learning," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Design (ICCAD)*, Nov. 2021, pp. 1–9.

[171] T. Dhar, K. Kunal, Y. Li, M. Madhusudan, J. Poojary, A. K. Sharma, W. Xu, S. M. Burns, R. Harjani, J. Hu, D. A. Kirkpatrick, P. Mukherjee, S. Yaldiz, and S. S. Sapatnekar, "ALIGN: A system for automating analog layout," *IEEE Design Test*, vol. 38, no. 2, pp. 8–18, Apr. 2021.

[172] B. Xu, K. Zhu, M. Liu, Y. Lin, S. Li, X. Tang, N. Sun, and D. Z. Pan, "MAGICAL: Toward fully automated analog IC layout leveraging human and machine intelligence: Invited paper," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design (ICCAD)*, Nov. 2019, pp. 1–8.

[173] H. Chen, M. Liu, B. Xu, K. Zhu, X. Tang, S. Li, Y. Lin, N. Sun, and D. Z. Pan, "MAGICAL: An open-source fully automated analog IC layout system from netlist to GDSII," *IEEE Design Test*, vol. 38, no. 2, pp. 19–26, Apr. 2021.

[174] C.-Y. Chiang, C.-L. Hu, K.-Y. Chang, M. P.-H. Lin, S.-J. Jou, H.-Y. Chen, C.-N.-J. Liu, and H.-M. Chen, "On optimizing capacitor array design for advanced node SAR ADC," in *Proc. 18th Int. Conf. Synth., Modeling, Anal. Simulation Methods Appl. Circuit Design (SMACD)*, Jun. 2022, pp. 1–4.

[175] C.-Y. Chiang, C.-L. Hu, M. P.-H. Lin, Y.-S. Chung, S.-J. Jou, J.-T. Wu, S.-H.-W. Chiang, C.-N.-J. Liu, and H.-M. Chen, "On automating fingercap array synthesis with optimal parasitic matching for custom SAR ADC," in *Proc. 28th Asia South Pacific Design Autom. Conf.*, Jan. 2023, pp. 352–357.

[176] Y.-H. Yeh, S. Y.-H. Chen, H.-M. Chen, D.-Y. Tu, G.-Q. Fang, Y.-C. Kuo, and P.-Y. Chen, "DPRoute: Deep learning framework for package routing," in *Proc. 28th Asia South Pacific Design Autom. Conf.*, Jan. 2023, pp. 277–282.

[177] H.-Y. Chi, S. Y.-H. Chen, H.-M. Chen, C.-N. Liu, Y.-C. Kuo, Y.-H. Chang, and K.-H. Ho, "Practical substrate design considering symmetrical and shielding routes," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2022, pp. 951–956.

[178] W.-T.-J. Chan, P.-H. Ho, A. B. Kahng, and P. Saxena, "Routability optimization for industrial designs at sub-14 nm process nodes using machine learning," in *Proc. ACM Int. Symp. Phys. Design*, Mar. 2017, pp. 15–21.

[179] A. F. Tabrizi, L. Rakai, N. K. Darav, I. Bustany, L. Behjat, S. Xu, and A. Kennings, "A machine learning framework to identify detailed routing short violations from a placed netlist," in *Proc. 55th ACM/ESDA/IEEE Design Autom. Conf. (DAC)*, Jun. 2018, pp. 1–6.

[180] T.-C. Yu, S.-Y. Fang, H.-S. Chiu, K.-S. Hu, P. H.-Y. Tai, C. C.-F. Shen, and H. Sheng, "Pin accessibility prediction and optimization with deep learning-based pin pattern recognition," in *Proc. 56th ACM/IEEE Annu. Design Autom. Conf.*, Jun. 2019, pp. 1–6.

[181] T.-C. Yu, S.-Y. Fang, H.-S. Chiu, K.-S. Hu, P. H.-Y. Tai, C. C.-F. Shen, and H. Sheng, "Lookahead placement optimization with cell library-based pin accessibility prediction via active learning," in *Proc. Int. Symp. Phys. Design*, Mar. 2020, pp. 65–72.

[182] R. Liang, H. Xiang, D. Pandey, L. Reddy, S. Ramji, G.-J. Nam, and J. Hu, "DRC Hotspot prediction at sub-10 nm process nodes using customized convolutional network," in *Proc. Int. Symp. Phys. Design*, Mar. 2020, pp. 135–142.

[183] U. Gandhi, I. Bustany, W. Swartz, and L. Behjat, "A reinforcement learning-based framework for solving physical design routing problem in the absence of large test sets," in *Proc. ACM/IEEE 1st Workshop Mach. Learn. CAD (MLCAD)*, Sep. 2019, pp. 1–6.

[184] J. Chen, J. Kuang, G. Zhao, D. J.-H. Huang, and E. F. Y. Young, "PROS: A plug-in for routability optimization applied in the state-of-the-art commercial EDA tool using deep learning," in *Proc. IEEE/ACM ICCAD*, Nov. 2020, pp. 1–8.

[185] H. Liao, W. Zhang, X. Dong, B. Poczos, K. Shimada, and L. B. Kara, "A deep reinforcement learning approach for global routing," 2019, *arXiv:1906.08809*.

[186] S. Indrapriyadarsini, S. Mahboubi, H. Ninomiya, T. Kamio, and H. Asai, "A Nesterov's accelerated quasi-Newton method for global routing using deep reinforcement learning," 2020, *arXiv:2010.09465*.

[187] D. Utyamishev and I. Partin-Vaisband, "Late breaking results: A neural network that routes ICs," in *Proc. ACM/IEEE DAC*, Jul. 2020, pp. 1–2.

[188] L. Li, Y. Cai, and Q. Zhou, "A survey on machine learning-based routing for VLSI physical design," *Integration*, vol. 86, pp. 51–56, Sep. 2022.

[189] J. L. C. P. Domingues, P. J. C. D. C. Vaz, A. P. L. Gusmão, N. C. G. Horta, N. C. C. Lourenço, and R. M. F. Martins, *Speeding-Up Radio-Frequency Integrated Circuit Sizing With Neural Networks*. Springer, 2023.

RICARDO M. F. MARTINS (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from Instituto Superior Técnico (IST), University of Lisbon (UL), Portugal, in 2015. Since 2011, he has been with Instituto de Telecomunicações developing electronic design automation tools. In 2022, he became an Assistant Professor of the electronics scientific area with the Department of Electrical and Computer Engineering, IST, UL. He has participated in about 100 scientific publications, including several first authorships and supervisions in books and high-impact journals, such as IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, IEEE ACCESS, *Applied Soft Computing*, *Expert Systems With Applications*, and *Engineering Applications of Artificial Intelligence*. His research interests include electronic design automation tools for analog, mixed-signal, radio-frequency, and millimeter-wave integrated circuits, deep nanometer integration technologies, applied soft computing, and machine and deep learning. He is a member of the organizing committee of several other conferences, for example, IEEE ICECS, PRIME, and New Generation of CAS. He was the General Chair of the International Conference on SMACD 2019, held in École Polytechnique Fédérale de Lausanne.

NUNO C. C. LOURENÇO (Senior Member, IEEE) received the Licenciado, M.Sc., and Ph.D. degrees in electrical and computer engineering from Instituto Superior Técnico, University of Lisbon, Portugal, in 2005, 2007, and 2014, respectively.

He has been with Instituto de Telecomunicações, Lisbon, since 2005, where he is currently holding a Postdoctoral Research position. He has coauthored more than 100 international scientific publications, including patents, books, international journals, and conference papers. His current research interests include AMS/RF IC design, artificial intelligence, evolutionary computation, and machine learning applied to electronic design automation. He was involved in the organizing committee of several international conferences, such as IEEE ISCAS 2015, PRIME 2016–2021, and SMACD 2016–2021. He was the General Chair of SMACD 2022, all technically sponsored by IEEE, IEEE CEDA, and IEEE CASS.