Automatic Layout Design for Power Electronics PCBs

Yidong Tian
Department of Electrical and Electronic Engineering
The University of Manchester
Manchester, U.K.
yidong.tian@manchester.ac.uk

Andrew J. Forsyth
Department of Electrical and Electronic Engineering
The University of Manchester
Manchester, U.K.
andrew.forsyth@manchester.ac.uk

Zhuoru Li
Department of Electrical and Electronic Engineering
The University of Manchester
Manchester, U.K.
zhuoru.li@manchester.ac.uk

Cheng Zhang
Department of Electrical and Electronic Engineering
The University of Manchester
Manchester, U.K.
cheng.zhang@manchester.ac.uk

***Abstract*—Automatic PCB layout design is one of the emerging research interests in the theme of design automation in power electronics (DAPE), as the industry has an increasing demand in shortened design and development cycles. Several layout design strategies are available in EDA software for very-large-scale integration (VLSI) circuits and power modules, but currently not suitable for power electronics PCBs – as the components are allowed to have arbitrary position, angle, layer, and trace are in any shapes, the layout design has a solution space of infinite dimensions which is hard to be optimized with conventional methods. In this paper, an iterative layout design workflow from circuit schematic to CAM files for power electronics PCBs based on genetic algorithm (GA) is proposed. The conceptual and technical details are explained and discussed. Two practical power converters are tested to verify the efficacy and performance of the proposed methodologies and implementation.**

Keywords—design automation, power electronics PCB, layout, genetic algorithm

I. Introduction

With the increasing demand for new power electronic products and the trend of shorter design cycles and product life-cycles in industry, design automation in power electronics (DAPE) has been raised recently to delegate the time-consuming iterative design processes to computer algorithms and minimize the massive costs in the highly specialized product design and development cycle [1-2]. Recent studies are mainly focused on relatively small-scale power modules with limited numbers of devices. Ning et al. [3] has proposed an automatic layout design method for power modules based on genetic algorithm (GA), limited to a single-layer structure (2-D). Al Razi et al. [4] proposed an integrated software tool, PowerSynth, for 2.5-D multichip power modules design. In assurance of tremendous computational cost saving, it also optimizes for the electric and thermal performance of the device. However, at the level of power electronics PCBs, despite that there are methodologies and algorithms for individual steps [5-7], the viability of a complete toolchain from schematics to CAM files is yet to be studied.

Along the workflow of power electronics PCB design process, several performance metrics are of concern at the same time. These include the power density, signal integrity, EMI, and thermal distributions, some of which may be conflicting with each other. Currently almost all power electronics PCB layouts are still plotted manually by experienced engineers, which restricts scaling up the number of designs and iterations in the development cycles.

Two difficulties are recognized that hinder the realization of automatic layout design for power electronics PCBs. First, compared with the highly automation-incorporated VLSI design process where empirical equations are used to evaluate the signal integrity related parameters, power electronics PCBs have further considerations on power efficiency. The ideal switching operations intrinsically carry infinite frequency components in the spectrum. The minimum width requirements for power traces increase the difficulty of automatic routing, especially when the PCB area needs to be optimized for higher power density. The design rule check (DRC) function becomes extremely critical to the optimizations. Second, there are versatile packages of components used in circuits. The irregular footprint shapes and four degree-of-freedom (x, y, angle and layer) placement parameters for components on the PCBs, and signal traces can also be routed in multiple layers. This infinite design solution space limits the applicability of the abovementioned power module design methods where a lot more position restrictions have been applied.

Some of the design strategies can still be borrowed both from the VLSI and power module design methodologies to help generate a better automatic layout design for power electronics PCBs. Table I displays the different characteristics of VLSI, power module, and power electronics PCB layout designs [3]. During the design process, the layout for power electronics PCBs needs to consider the large component numbers and multiple PCB layers as in VLSI design, and also the thermal stability as in power module design, but could have different design and optimization objectives. Based on these challenges, this paper proposes an iterative automatic layout design workflow for power electronics PCBs. The design overview and algorithms are elaborated in detail with demonstrations and verification on examples.

TABLE I. Characteristics of VLSI, Power Module and Power Electronics PCB Layout Design [2]

Design characteristics	VLSI	Power module	Power electronics PCB
Component number	Large	Small	Based on topology
Routing layer	Multiple	Single	Multiple
Component clustering	Yes	No	Yes
Component placement	Yes	Yes	Yes
Routing	Yes	Yes	Yes
Copper pouring	No	Yes	Yes
Design process	Iterations	Iterations	Iterations

II. The Interative Layout Design Workflow

In mainstream PCB design software, the PCB layout process is divided into two stages. The circuit schematic is

2022 IEEE Energy Conversion Congress and Exposition (ECCE) | 978-1-7281-9387-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/ECCE50734.2022.9947957

978-1-7281-9387-8/22/$31.00 ©2022 IEEE

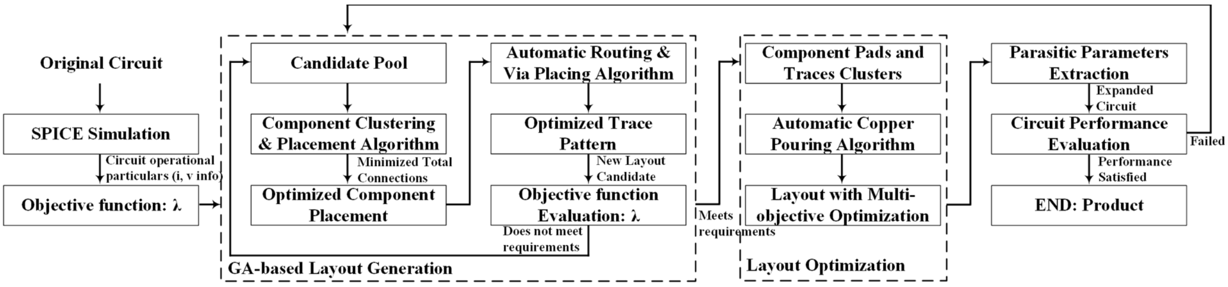

Original Circuit
SPICE Simulation
Circuit operational particulars (i, v info)
Objective function: λ
Candidate Pool
Component Clustering & Placement Algorithm
Minimized Total Connections
Optimized Component Placement
Automatic Routing & Via Placing Algorithm
Optimized Trace Pattern
New Layout Candidate
Objective function Evaluation: λ
Does not meet requirements
GA-based Layout Generation
Meets requirements
Component Pads and Traces Clusters
Automatic Copper Pouring Algorithm
Layout with Multi-objective Optimization
Layout Optimization
Parasitic Parameters Extraction
Expanded Circuit
Circuit Performance Evaluation
Failed
Performance Satisfied
END: Product

Fig. 1. The overall design cycle for power electronics PCBs.

drawn in the first stage, and the actual PCB product is designed in the second stage. The layout workflow proposed in this paper takes the power circuit netlist exported from the first stage as input, therefore all component packaging and connection information are accessible. This is followed by an iterative layout process, aiming to output an optimized circuit board design that can be directly used to generate CAM files.

A genetic algorithm (GA) is used to generate and down-select the layout options. As a global optimization algorithm based on stochastic search methods, the GA solves the problem of evaluating a large number of layout candidates resulting from the unrestricted design space of power electronics PCBs. In the proposed layout design workflow, the potential search space for the layout candidates includes the position coordinate, the rotation angle, and the placed layer, which are also considered elementary elements for the objective function evaluation. Theoretically, in each cycle of the GA-based iterative design process, a PCB layout with the best objective function value is searched and used for final product performance evaluation.

However, the selection procedure is highly reliant on the definition of the objective function in the GA and it may take an unacceptable time to generate a satisfying result. Therefore, circuit operational parameters including the maximum values of rms current, rate-of-change of current, rms voltage, and rate-of-change of voltage are applied to accelerate the overall design process and guide the generation of practical solutions. For example, component pad connections with higher rms current and rate-of-change of current should be preferentially processed to ensure they are placed as close as possible. This reflects an attempt to reduce circuit power loss and parasitic parameters, because the connection length is always positively related to the parasitic resistance and inductance values of copper traces. Meanwhile, the absolute extreme values of rms voltage and rate-of-change of voltage highly affect the creepage of the copper layers and the stray capacitance between the relative copper area on the different PCB layers.

Fig.1 illustrates the proposed overall design cycle for power electronics PCBs. Assuming all design-related SPICE models are available, the circuit operational particulars are extracted from a transient analysis in the SPICE simulation. There are two design steps in the layout design process. The first step is a GA-based layout generation loop. For layout candidates in the candidate pool, their layouts are designed in the order of component clustering, since this better integrates the power circuit on the PCB. This process follows a step-by-step placement and routing sequence. After a number of iterations and selections, the layout candidate that best meets the user-defined objectives will go through the second layout optimization step of copper pouring to guarantee a reasonable layout design with thermal stability. After that, the parasitic parameters of the optimized PCB copper layers can be extracted by FEM computations, then inserted back into the original power circuit simulation for performance evaluation. The final design result usually needs several design iterations in the overall design process, but will be the layout that most satisfies all the design considerations.

III. Design Algorithms in Each Design Cycle

In the layout design workflow, every individual component is represented as a 2-D boundary polygon with a set of internal pad polygons having the associated offset, angle, and layer parameters. Algorithm 1 illustrates the initial component generation algorithm. The greatest advantage of using polygon sets is that after the best placement with no design rule issues has been determined based on the external component boundaries, the component pad polygons can be directly used for further routing and copper pouring processes.

After the design initialization, an objective function is required for the GA process and the user is tasked to merge all the design objectives into a single monotonic function for easy evaluation [3]. Considering the numerical property of the objective function, all the descriptive optimization objectives are simplified into approximate mathematical expressions with geometry considerations on the PCB for easy and fast evaluation in layout design. For example, the consideration of parasitics can be transferred to the optimization of length and width of traces between component pads [6].

In the GA-based layout generation loop, Algorithm 2 is implemented as the solution to generate the PCB layout candidate. The design of each layout candidate starts with a component clustering algorithm to provide order for the layout

```
Algorithm 1: Component generation
  input  : The circuit schematic file
  output: Program processable components
circuit_netlist, package_library = parse-circuit(circuit_schematic_file);
foreach package in the package_library do
    processable_package = create-package(package.name);
    foreach pad in the package.parse_pads do
        x, y, angle, layer, shape = pad.parse_element;
        offset = create-point(x, y) // Pad offset relative to the
          package center;
        pad_polygon = create-polygon(shape) // Create pad boundary
          polygons for specific pad shapes such as rectangular,
          circle, and octagon;
        processable_package.add_pad(offset, angle, layer, pad_polygon);
    processable_package.boundary_polygon =
      create-boundary-polygon(processable_package.pads) // Create
      the minimum rectangular polygon containing all the pads;
    processable_packages.add(processable_package);
foreach component in the circuit_netlist do
    component_package = find-package(processable_packages,
      component.package_name);
    x, y, angle = component.parse_element;
    offset = create-point(x, y) // Component position coordinate;
    processable_component = create-component(component.name,
      offset, angle, component_package);
    processable_components.add(processable_component);
```

generation process. This is because the circuit complexity grows rapidly when the number of components increases, and component clustering can streamline the pad-to-pad connection network extracted from the circuit netlist information. The component clustering algorithm outputs a component processing order based on a connection weighting table describing the relative compactness between component pairs, which can be built up by the simulated maximum rms current and rate-of-change of current values.

After that a placement and routing process is applied to each component. In this process, a common goal for placement algorithms, namely minimizing the total connection distances with weightings for relative compactness considerations between components is adopted since it satisfies most of real-world design targets, and provides a clear criterion for objective function evaluation in GA [5]. Therefore, during each component process, other relevant components as well as the between connections are extracted in advance for layout evaluation. The automatic placement procedure used here has two phases [6]. The first phase implements a placement detection method with component moving operations such as translation, rotation and flip to ensure all the components are placed at the optimal position within an acceptable time frame. But the 'optimal' result from this kind of method often has many defects, such as components becoming stuck while rotating, which will finally lead to an unavailable result. Therefore, fine-tuning of all component positions and orientations are made in the second stage to generate a usable layout with as little impact on the total weighted distance.

The auto-routing takes place based on the optimized placement of each component. To keep the shortest total weighted distance, the principle of the proposed routing algorithm is very similar to path planning [8]. For each pad-to-pad connection, the boundaries of component pads, fixed traces and vias belonging to other signal nets are extracted as obstacles in the path planning algorithm. From the start point on one pad, path planning is continuously applied on different layers to find the next optimal route choices without any obstacles in between until the end point on the other pad is reached. A cost function based on the total trace length and number of vias is used to help find the best trace path. Fig.2 shows a series of path plan choices for pads on the same and different layers with different priorities. For example, in this case, only when there are obstacles that cannot be avoided through vias, are traces diverted with turning points (regarded as virtual vias in the algorithm).

The layout optimization stage mainly applies a copper pouring algorithm to ensure that the layout has sufficient copper area for current handling and thermal constraints.

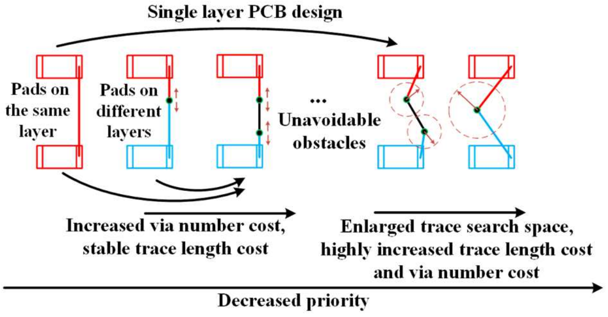

Single layer PCB design
Pads on the same layer
Pads on different layers
...
Unavoidable obstacles
Increased via number cost, stable trace length cost
Enlarged trace search space, highly increased trace length cost and via number cost
Decreased priority

Fig. 2. The path plan choices with different priorities.

Algorithm 2: Layout generation

```
input  : Program processable components
output: Layout candidate
component_processing_order =
  component-clustering-algorithm(program_processable_components);
foreach component in the component_processing_order do
    relevant_components, relevant_connections =
      get-connection-components(circuit_netlist, component) // Find
      all the components connect to the component;
    process_set = [component, relevant_components,
      relevant_connection];
    placement-detection(process_set) // Placement phase I;
    placement-fine-tuning(process_set) // Placement phase II;
    foreach connection in the relevant_connection do
        start, end, obstacles = get-routing-elements(component,
          connection) // Elements for path planning;
        current = start // Current turning point on the trace;
        while current != end do
            trace_points.add(current);
            process_set = [current, end, obstacles];
            foreach layer in the layers do
                next = signal-layer-routing(process_set, layer)
                  // Potential next turning point;
                next_points.add(next);
            optimal_next_point = cost-evaluation(next_points);
            current = optimal_next_point // Multi-layer routing;
        trace_points.add(end);
        trace = create-trace(trace_points);
        traces.add(trace);
layout = create-layout(program_processable_components, traces);
```

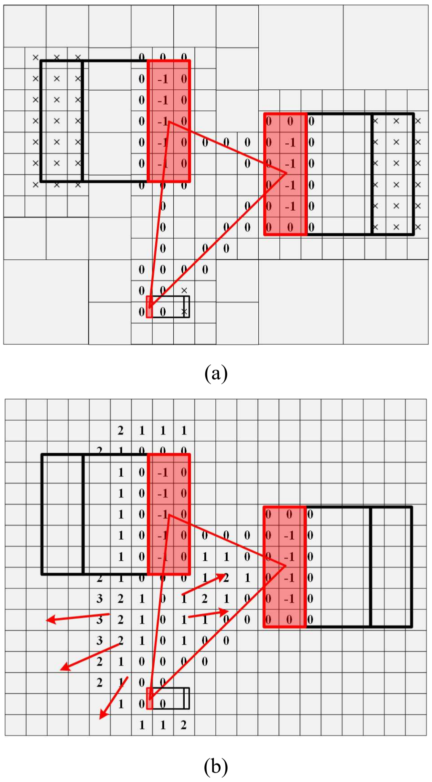


Fig. 3. (a) Initial pattern determination. (b) The generation process.

Starting from the component pads and trace clusters from the different signal nets, the concept of a signed distance field (SDF) is introduced to ensure a uniformly generated copper layer pattern. The SDF is a grid plane, but with positive and negative distance numbers for outside and inside the plane polygons, respectively. During copper pouring, the whole design plane is iteratively segmented into grids from large sizes to small to determine the initial pattern for each copper layer, as shown in Fig.3 (a). After that, Fig.3 (b) shows the generation process. In the process, the boundary grids of each processed copper layer are expanded to the adjacent blank areas with different expansion speeds calculated by the circuit current and voltage parameters until the creepage between signal nets is reached.

IV. Examples of Automatic Layout Design

Two design examples are used to demonstrate the proposed automatic layout design workflow. To generate more intuitive results, all components in each circuit are limited to the top layer with a 0.3 mm clearance. Also the objective function in the GA is defined to minimize the total weighted pad-to-pad distances and layout boundary area.

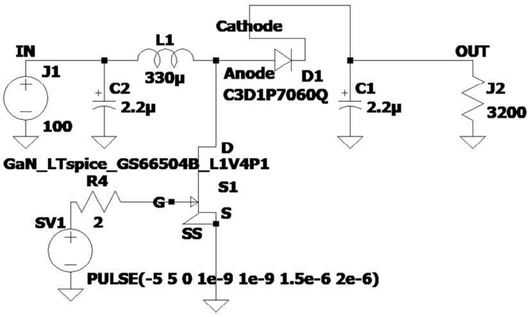
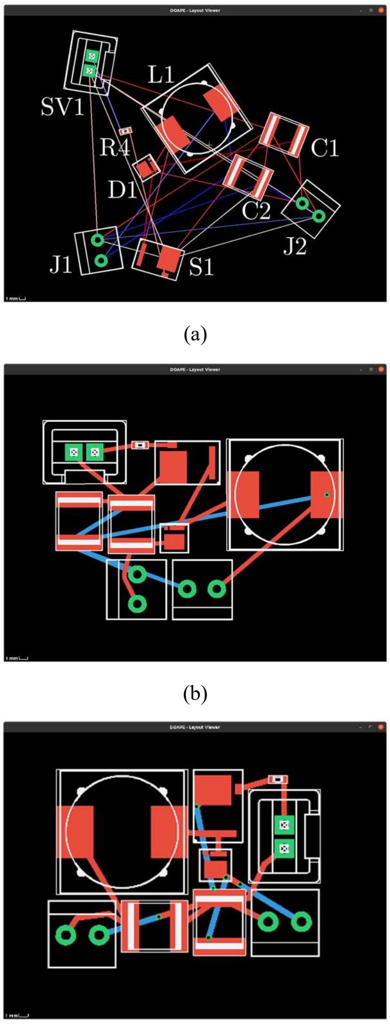


Fig. 4. The LTSPICE model for the boost converter.



(a)

(b)

(c)

Fig. 5. Automatic design process for a boost converter. (a) A random initial state (Red/white/blue indicates the connection weights from high to low). (b) A double-layer layout result. (Red/blue represents the top/bottom layer). (c) Another result.

A. A Boost Converter

The power circuit for a boost converter is selected as the first example, and through this the concept and realization of the workflow can be clearly demonstrated. There are totally 6 signal nets, 26 pad-to-pad connections and 9 components in this circuit. The LTSPICE model is shown in Fig.4 for generating the weightings for component clustering. Starting from a random initial state in Fig.5 (a), Fig.5 (b) and (c) display two different double-layer layout results from the place-and-routing process with 0.5 mm trace width, respectively. The result in Fig.5 (b) was found after 5167 iterations in the GA-based layout generation loop, having 149.69 mm Euclidian distance and 875.25 mm^2 area, while the result in Fig.5 (c) was found after 5335 iterations, having 165.71 mm distance and only 628.78 mm^2 area. Fig.6 gives the final layout optimization result based on the layout in Fig.5 (c), and a fixed rate of copper layer expansion (each step a minimized grid) is applied in this example for easy observation. Typically, it takes about 4 minutes using a laptop (Intel(R) Core(TM) i7-10510U CPU@ 1.80GHz, 16 GB RAM) to generate an optimized layout for this circuit using the design cycle proposed in Fig.1.

To confirm the practical operation and performance of the layout solution in Fig.6, a real PCB is built in Fig.7 (a), and the test platform is shown in Fig.7 (b). Given a 500 kHz, 25%

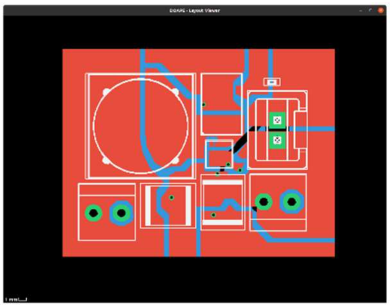
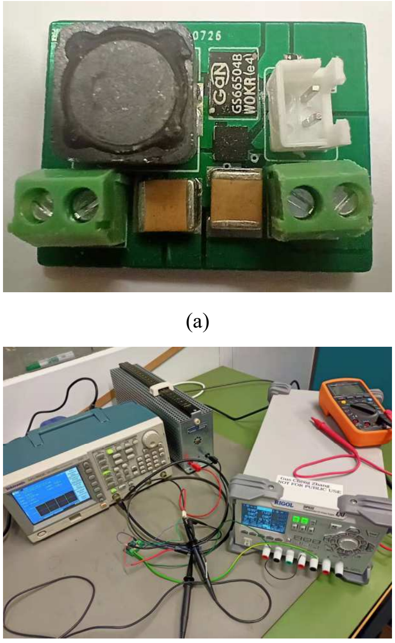


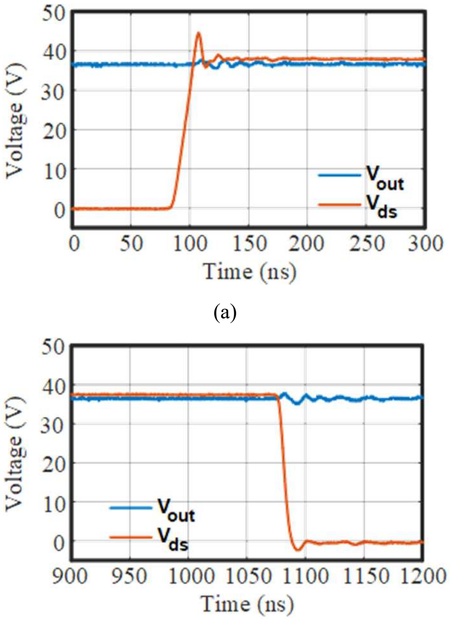
Fig. 6. The optimized layout solution based on the second layout result.



(a)

(b)

Fig. 7. Test for layout solution in Fig.6. (a) The actual PCB (34.0mm × 25.7mm). (b) The test platform.



(a)

(b)

Fig. 8. (a) The rising edge of V_{ds} (orange) and the corresponding circuit output voltage (blue). (b) The falling edge of V_{ds} (orange) and the corresponding circuit output voltage (blue).

duty ratio gate control signal and 10V input voltage, Fig.8 (a) and (b) show the rising edge and falling edge of the MOSFET drain-to-source voltage V_{ds} and the corresponding circuit output voltage, respectively. It can be seen that, during the circuit operation, there are voltage overshoots on V_{ds}, which could affect the normal operation of the design circuit. These overshoots are caused due to the uneven copper layer expansion (a fixed expansion rate for each signal net in this example), but could be reduced by using a weighting-related copper expansion method.

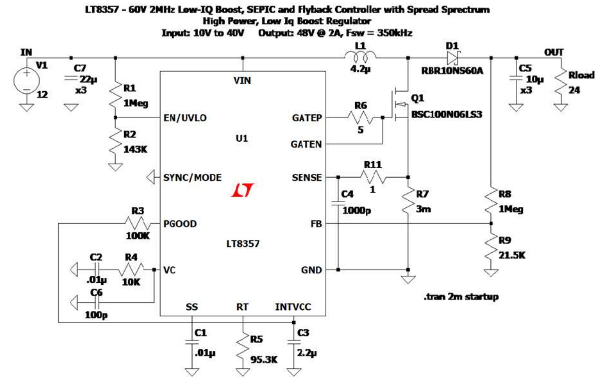
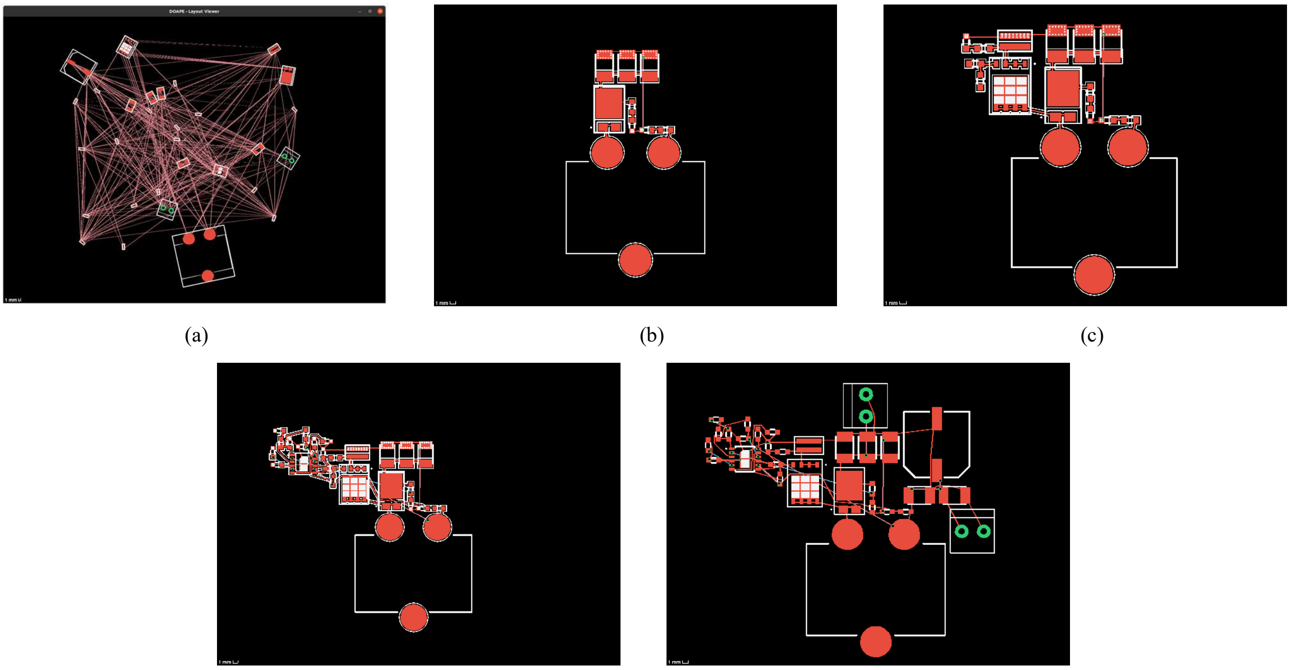


Fig. 9. The LTSPICE model for the evaluation board DC2937A [10].

B. The Evaluation Board DC2937A

The second example is a re-design process for the circuit of an evaluation board DC2937A for a controller IC LT8357 [9]. This circuit is much more complicated having 18 signal nets, 289 pad-to-pad connections and 30 components. The circuit LTSPICE model is online provided and is shown in Fig.9 [10]. Considering the signal paths also need to be placed on the same PCB, weighting-related trace widths from 0.15 mm to 0.2 mm are applied to expand the routable area on the PCB. Also as a starting point, a random initial state is shown in Fig.10 (a), and Fig.10 (b), (c), (d) show three middle stages that have processed 9, 15 and 25 components, respectively. After the place-and-routing process, the 4-layer PCB layout in Fig. 10 (e) has 245.100 mm total pad-to-pad connection distance and 2054.991 mm^2 area.

The final practical PCB for the redesigned DC2937A circuit is shown in Fig.11 (a), which is generated after the copper pouring process with a weighting-related copper expansion rate (maximum speed of 4 grids per copper expansion step). And an additional via farm filling is applied to the final layout solution for better thermal dissipation. The



(a)

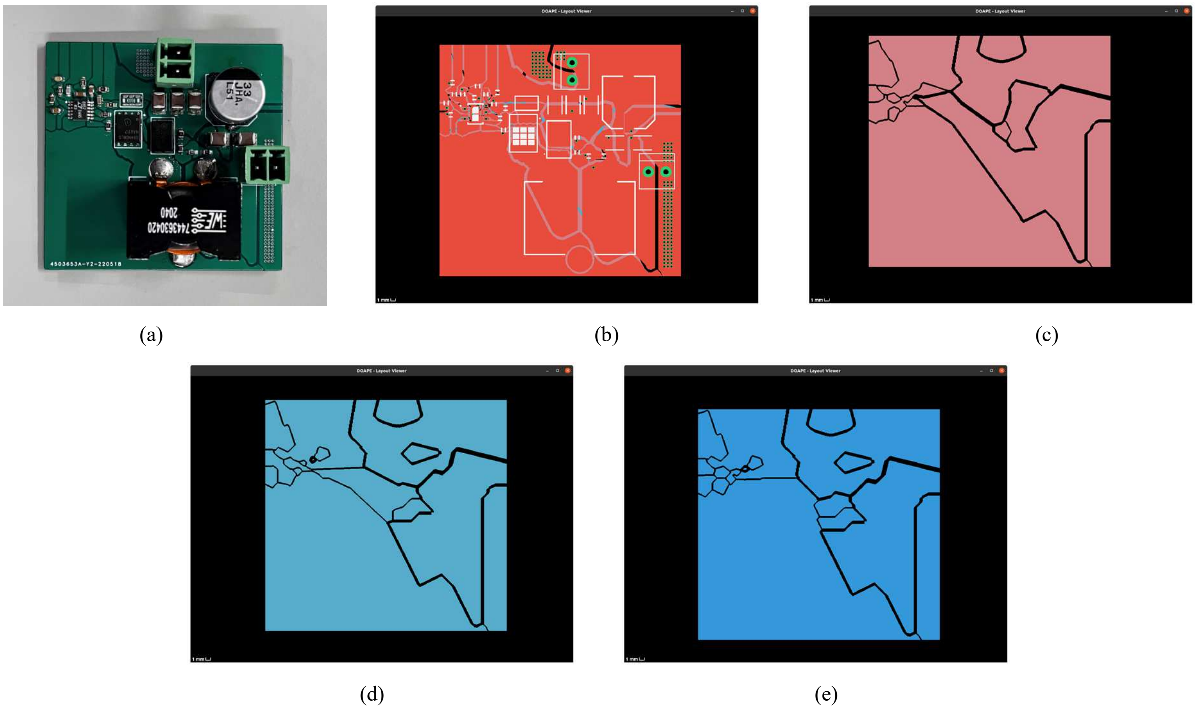
(b)

(c)

(d)

(e)

Fig. 10. Automatic design process for the circuit of an evaluation board DC2937A [9]. (a) A random initial state. (b) The middle stage of 4-layer layout after processing the 9th component. (c) The middle stage of 4-layer layout after processing the 15th component. (d) The middle stage of 4-layer layout after processing the 25th component. (e) The 4-layer generated layout result after the place-and-routing process.



(a)

(b)

(c)

(d)

(e)

Fig. 11. The final layout solution. (a) The actual PCB (48.3mm × 46.3mm). (b) The copper pattern on the top layer. (c) The copper pattern on the first inner signal layer. (d) The copper pattern on the second inner signal layer. (e). The copper pattern on the bottom layer.

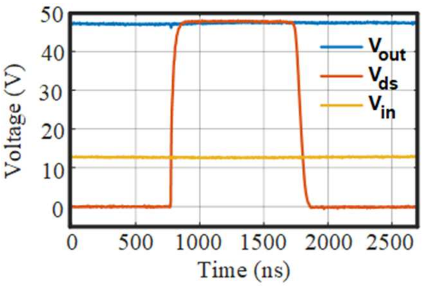


Fig. 12. Test for layout solution in Fig. 11.

total design process costs 30 minutes. Fig.11 (b), (c), (d), (e) illustrate the automatically designed copper layer patterns on the 4 layers, which validate the expansion consistency of the same signal net on the different PCB layers.

A circuit test experiment is also applied here to verify the effect of layout design automation. In practical application, the evaluation board DC2937A is used to boost the input voltage (from 10V to 40V) to a 48V output [9]. The re-design circuit example takes the same control configuration. Fig.12 shows one switching cycle of the circuit operation (picked in the burst-mode). The output voltage is successfully boosted to 48V. Also, the potential overshoots on the V_{ds} are avoided through assigning the corresponding weighting values to different signal nets, which finally ensures a reasonable signal flow on the PCB.

V. Conclusions

This work proposes an iterative automatic layout design workflow for power electronics PCBs based on GA. The design assumptions and overall design cycle are elaborated with detailed design algorithms and data structures. Two different design examples are applied to verify the practicability, reliability, and design speed of the automatic design process. The results meet the expectation. This automatic layout design workflow could become a fundamental tool to enhance the circuit layout design in the DAPE framework, and could also improve the circuit performance of future power electronics circuit design.

References

[1] A. Bindra and A. Mantooth, "Modern Tool Limitations in Design Automation: Advancing Automation in Design Tools is Gathering Momentum," IEEE Power Electronics Magazine, vol. 6, no. 1, pp. 28-33, Mar. 2019.

[2] K. Hermanns, Y. Peng and A. Mantooth, "The Increasing Role of Design Automation in Power Electronics: Gathering What Is Needed," IEEE Power Electronics Magazine, vol. 7, no. 1, pp. 46-50, Mar. 2020.

[3] P. Ning, F. Wang and K. D. T. Ngo, "Automatic layout design for power module," IEEE Transactions on Power Electronics, vol. 28, no. 1, pp. 481-487, Jan. 2013.

[4] I. Al Razi, Q. Le, T. M. Evans, S. Mukherjee, H. A. Mantooth and Y. Peng, "PowerSynth Design Automation Flow for Hierarchical and Heterogeneous 2.5-D Multichip Power Modules," IEEE Transactions on Power Electronics, vol. 36, no. 8, pp. 8919-8933, Aug. 2021.

[5] N. Quinn and M. Breuer, "A forced directed component placement procedure for printed circuit boards," IEEE Transactions on Circuits and Systems, vol. 26, no. 6, pp. 377-388, Jun. 1979.

[6] Y. Tian, A. J. Forsyth, Z. Li, and C. Zhang, "A Component Manipulation Algorithm to Enable Design Automation of Power Electronic PCBs," in 2021 IEEE Design Methodologies Conference (DMC), 2021, pp. 1-6.

[7] Y. Tian, A. J. Forsyth, Z. Li and C. Zhang, "Automated Copper Layer Design and Optimization Tool based on Progressive Point Expansion Algorithm for Switch Mode Power Supplies," in 2021 IEEE 22nd Workshop on Control and Modelling of Power Electronics (COMPEL), 2021, pp. 1-6.

[8] N. Sariff and N. Buniyamin, "An Overview of Autonomous Mobile Robot Path Planning Algorithms," in 2006 4th Student Conference on Research and Development, 2006, pp. 183-188.

[9] Analog Devices, Demo manual DC2937A, https://www.analog.com/media/en/technical-documentation/user-guides/dc2937a.pdf.

[10] Analog Devices, https://www.analog.com/media/en/simulation-models/ltspice-demo-circuits/lt8357_ta03a.asc.