Thermal Science and Engineering Progress 61 (2025) 103530

Contents lists available at ScienceDirect

Thermal Science and Engineering Progress

journal homepage: www.elsevier.com/locate/tsep

TIMtrace: Coverage Path Planning for Thermal Interface Materials[☆]

S. Baeuerle [a,c], A. Steimer [b,1], R. Mikut [a,*,1]

[a] Karlsruhe Institute of Technology (KIT) - Institute for Automation and Applied Informatics (IAI), Hermann-von-Helmholtz-Platz 1, Eggenstein-Leopoldshafen, 76344, Germany
[b] Robert Bosch GmbH - Bosch Center for Artificial Intelligence (BCAI), Robert-Bosch-Campus 1, Renningen, 71272, Germany
[c] Robert Bosch GmbH - Mobility Electronics, Markwiesenstrasse 46, Reutlingen, 72770, Germany

ARTICLE INFO

Keywords:
Thermal management
Coverage Path Planning
Optimization
Automated design

ABSTRACT

Thermal Interface Materials are used to transfer heat from a semiconductor to a heatsink. They are applied along a dispense path onto the semiconductor and spread over its entire surface once the heatsink is joined. To plan this application path, design engineers typically perform an iterative trial-and-error procedure of elaborate simulations and manual experiments. State-of-the-art path planning approaches are mainly focused on applications with a constant path width. They cannot be transferred to Thermal Interface Materials since the resulting area coverage depends on their complex flow behavior. We propose the fully automated optimization approach *TIMtrace*, which clearly outperforms the current manual path planning and respects all relevant manufacturing constraints. Generally, the design of an optimization problem highly influences its convergence properties. Our chosen path parameterization keeps the dimensionality of the independent variables reasonably low. We analyze different loss function components and respective weighting strategies with regard to both optimization convergence and result quality. *TIMtrace* enables an automated planning of dispense paths that do not contain voids, do not violate taboo zones and can be transferred easily to automated dispensing machines. An optimum dispense path helps to increase the reliability of the thermal interface and to make manufacturing more sustainable by reducing material waste. We show results on multiple real products from automotive series production, including experimental validation on actual series manufacturing equipment.

1. Introduction

Thermal Interface Material (TIM) is used to transfer heat from a semiconductor to a heatsink. During electronic packaging, TIM is applied onto the semiconductor and spreads over the cooling surface once the heatsink is joined. This is valid for both for small Electronic Control Units (ECUs) and for larger power electronics modules such as inverters or chargers. These products play a central role for electric and autonomous vehicles. As power ratings increase and assembly space is limited, the thermal performance of the electronic package is a critical factor in the competitive market of the automotive industry.

The demands regarding thermal performance for the entire electronic package translate to specific requirements regarding the TIM layer as follows. To ensure an efficient heat flow, a high area coverage percentage of the cooling surface with TIM is desired. The area coverage is shown on the right of Fig. 1. A special case of a low area coverage is the formation of air entrapments or voids during the joining process of the heatsink. For example, if TIM is applied in a circular shape onto the heatsink, air will be trapped inside and form a void area. The effect of voids on the thermal performance can be *devastating* [1]. In automotive industry, the application of TIM onto to the semiconductor is typically carried out with selective dispensing technologies. Prevalent technologies are time-pressure dispensing, rotary screw dispensing and positive displacement piston dispensing. A high degree of automation is achieved by combining the dispensing system with a Computerized Numerical Control (CNC) machine. The dispensing toolhead is programmed to travel along a predefined path. While traveling along this path, the dispensing pump or piston can be (de-)activated as needed. This results in a desired TIM dispensing pattern, i.e. a distribution of TIM material on the cooling surface. This is shown on the left in Fig. 1. This manufacturing process introduces additional requirements, which need to be considered during the design of the dispensing path: (A) A low cycle-time is desired to maximize the number of manufactured parts per timeframe. (B) The amount

[☆] The work of R. Mikut was supported by the Helmholtz Association's Initiative and Networking Fund through Helmholtz AI and the program Energy System Design.

[*] Corresponding author.
E-mail address: ralf.mikut@kit.edu (R. Mikut).

[1] A. Steimer and R. Mikut contributed equally to this work.

https://doi.org/10.1016/j.tsep.2025.103530
Received 16 November 2024; Received in revised form 17 March 2025; Accepted 18 March 2025
Available online 29 March 2025

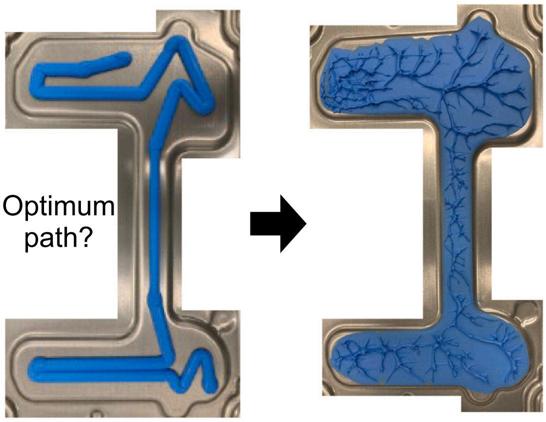


Fig. 1. Left: Thermal Interface Material dispensed along path. Right: Compressed state of Thermal Interface Material after joining the heatsink (heatsink removed for visual assessment). Aim of this work: Plan an optimum path (left) to achieve – among other criteria – a high area coverage of the cooling surface (right). The shown path neither leads to a formation of air entrapments nor to a violation of taboo zones.

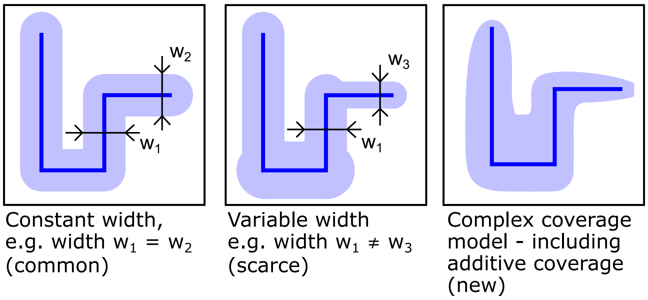
of material dispensed per timeframe, i.e. the dispense rate, may not exceed a minimum and a maximum rate. This is dependent e.g. on the material viscosity or the dispensing nozzle size. (C) The individual beads of the dispensing pattern should be separated from each other. This enables an easier quality inspection and mitigates disturbances during the dispensing process. (D) Excess material flowing beyond the cooling surface should be minimized. Excess material introduces unnecessary material cost. (E) Excess material can further violate sensitive areas – so-called taboo zones – in the vicinity of the cooling surface. Those are e.g. screw holes or other electronic components. This case is a violation of functional requirements and needs to be prevented.

The dispensing pattern is typically designed manually by an engineer. The engineer designs the dispense path based on experience. The design process is carried out in an iterative trial-and-error cycle. It is supported by elaborate Computational Fluid Dynamics (CFD) simulations or by new heuristics [2,3]. Furthermore, manual experiments are carried out. Restrictions regarding development time together with the complexity of the planning problem typically lead to resulting dispense patterns, which are sub-optimal with regard to above requirements.

2. Related work

Traditional path planning aims to find a path, which connects a start point with a goal point while avoiding obstacles. Well-known algorithms in this field are e.g. A* [4] and Dijkstra [5]. Coverage Path Planning (CPP) aims to find a path *that passes over all points of an area or volume of interest while avoiding obstacles* [6]. It relates to the Traveling Salesman Problem (TSP), which aims to find the shortest route for visiting multiple cities. TSP is a NP-hard problem to solve and can be efficiently approached with approximation algorithms. A well-known example is the algorithm of Christofides and Serdyukov. The problem of dispense pattern design for TIM closely fits the problem formulation of CPP, but introduces an additional aspect: the area coverage depends on the complex material flow during the joining process. This is in contrast to many related CPP applications, where the area within a fixed distance to the traveled path is being covered.

The CPP problem has been studied for many years. Choset [7], Galceran and Carreras [6] and Tan et al. [8] provide an overview over approaches to the CPP problem. Choset and Pignon [9] proposed the Boustrophedon decomposition. This decomposition splits the target area into cells, which can be covered by a back-and-forth motion. Zelinsky et al. [10] use both the distance to the goal and the distance to the next obstacle to compute a numeric value for each grid cell. This transformation is used as a basis to guide a robot through the target area. Gabriely and Rimon [11,12] introduced an algorithm called Spiral-STC. They compute a minimum spanning tree for a discretized target area. The coverage path is created by traveling along the spanning tree. Luo and Yang [13–15] propose a CPP approach that relies on the activation of a neural network. The information about obstacles and unclean areas is propagated through an activity landscape and either repulses or attracts the robot. Soltero et al. [16,17] use gradient-based optimization to adjust the waypoints of one or more quadcopters. Coverage is maximized by pushing the waypoints towards the centroids of Voronoi cells, while keeping neighboring waypoints close to each other. Kiemel et al. [18] present a framework to enable CPP for industrial spray painting tasks using Reinforcement Learning. Picciarelli and Foresti [19] focus on CPP for Unmanned Aerial Vehicles (UAVs). They use a double deep Q-network and use a relevance map with spatially defined coverage requirements as an input. Theile et al. [20] add a movement budget to take into account not only the coverage goal, but also power constraints. They use a 2D input map, which is processed by convolutional layers. Ellefsen et al. [21] use Genetic Algorithms to perform a multi-objective coverage path optimization on a discretized space for Autonomous Underwater Vehicles. They treat the coverage, energy usage and collision occurrence as an objective rather than a constraint. This yields paths that may not achieve complete coverage, but represent a good trade-off between all targets. Batista and Zampirolli [22] discretize the target area and use a Genetic Algorithm to plan a coverage path for a tool cleaner. They allow repeated visiting of both path edges and path vertices. They further allow both open and closed paths. Yakoubi and Laskri [23] split the coverage path of a vacuum cleaner robot into short sub-paths. They optimize the combination of the sub-paths using Genetic Algorithms. Popovic et al. [24] optimize the area coverage path of an UAV surveillance task while considering constraints such as a time budget. They assume a pyramid-like field of view of a camera mounted on a UAV. The planning procedure is split into two parts. The first part is the selection of viewpoints in 3D space. In a second step, a continuous trajectory is defined based on the chosen viewpoints using the optimization algorithm Covariance Matrix Adaption Evolution Strategies (CMA-ES) [25]. Strubel [26] presents a Coverage Path Planning approach, which is suitable for an area surveillance task using a UAV. He divides the problem into two separate tasks: waypoint finding and connecting the waypoints to form a path. For the waypoint finding, the assumption of a rectangular field of view of a camera mounted to an UAV is used. Thus, the area coverage is composed of a multitude of individual rectangles. The rectangles are allowed to overlap each other. Coverage Path Planning has also been studied in the context of Additive Manufacturing (AM) [27,28]. Some approaches mainly rely on back-and-forth patterns (e.g. [29]), optimize velocity profiles rather than the global path pattern (e.g. [30]) or adjust the path pattern only locally (e.g. [31]). The workspace can be processed with a medial axis transform [32], which can be used as a basis for path planning with a variable path width [33]. Using a variable path width in AM has been further studied, e.g. by Ding et al. [34], Wang et al. [35], Xiong et al. [36] and Hornus et al. [37]. Xiong et al. further perform an optimization of process parameters such as material feed rate to achieve a desired height and width of the deposited material. In order to perform this optimization, they use Gaussian Processes to model the relationship between in- and outputs. This data-driven model is efficient in terms of computing time and accuracy. Similarly, Ding et al. [38] use an Artificial Neural Network (ANN) instead of Gaussian Processes, since ANNs are well-suited to model complex relationships. Advantages of printing with a variable path width generally include prevention of void areas, increased production speed and a higher geometrical accuracy [36]. Additional criteria to be studied in this setting are e.g. the influence on residual stress [36] and possible voids between individual printing planes [37]. Flaig et al. [39] propose to invert a recently introduced flow behavior model for adhesives [3]. This inversion results in dispense patterns, which cover the target area. This



Constant width, e.g. width $w_1 = w_2$ (common)

Variable width e.g. width $w_1 \neq w_3$ (scarce)

Complex coverage model - including additive coverage (new)

Fig. 2. Area coverage types used in related literature during Coverage Path Planning problem settings. The dispense path is shown in dark blue. The resulting area coverage is shown in light blue. The coverage path width w of exemplary path segments is indicated. Most approaches in literature consider only a constant path width as shown on the left.

has been tested with manual experiments [40]. The resulting patterns exhibit a branched shape rather than a continuous path. This is similar to the result of a medial axis transform, which has been used for path planning in additive manufacturing before [33]. The branched shape of the resulting patterns would require to an automated dispensing system to travel multiple times over the same edge. Assuming a constant dispense rate, this will make it difficult to match the proposed material distribution in practice. Furthermore, the beads are not separated from each other. This yields to difficulties during quality inspection. Flaig et al. themselves state that the resulting patterns *may not yet be well suited for practical application, e.g. by automatized dosing systems* [39]. A more recent publication of the same research group exhibits similar issues with the resulting patterns [41]. Due to the failure to comply with technical constraints, these approaches cannot be used for automated dispensing systems which are used for automotive series manufacturing.

Another relevant field of research is the optimization of trajectories in the field of robotics. It typically involves not only planning the path in terms of a path through space, but also the speed or even the acceleration along this path. Gradient-based optimizers can be utilized for such problems [42]. However, gradient-based optimizers strongly depend on the initial trajectory guess and may get stuck in a local minimum. This corresponds to a non-optimal solution.

The above approaches can be clustered according to the underlying area coverage behavior for the three cases, which are shown in Fig. 2. Almost all of the classic Coverage Path Planning approaches consider a constant path width as shown on the left of Fig. 2. A constant width reflects the nature of many of the relevant physical systems accurately. Examples are lawn mowers, milling machines, cleaning robots, 3D printers or harvesting machines.

We indicate the constant path width by depicting width measurements at two positions: width w_1 is equal to width w_2. Very few approaches consider a varying path width as shown in the center of Fig. 2. During additive manufacturing, the bead size can be slightly modified. However, this variation of the path width is bound to narrow limits, since increasing the bead width also increases the bead height and the bead height is generally desired to be constant within a printed layer. Furthermore, the feasible range of bead widths for the printed materials is rather limited. Applying the approaches from additive manufacturing to dispensing of TIM is not feasible, since air entrapments would form for the resulting patterns. We indicate the varying path width in the center image of Fig. 2 by depicting width measurements at two positions. In this case, w_1 is not equal to width w_3.

Furthermore, a few approaches with varying path width came up for UAVs. They consider a pyramid-like field of view leading to a rectangular coverage area on the target surface for an individual viewpoint. These approaches [24,26] are limited to a rectangular coverage for a single viewpoint. For many practical use-cases, the horizontal dimensions are proportionally much larger than the vertical dimension. In this case, the approach will become similar to a constant path width.

The nature of TIM dispensing systems is fundamentally different. For example, air entrapments are naturally not considered for UAVs at all. The Coverage Path Planning problem for TIM has not only a varying path width, but a complex underlying flow behavior model as shown on the right of Fig. 2. To compute the area coverage (light blue), simply considering the area within a width value w left and right of the traveled path (dark blue) is not sufficient anymore in this case. A core difference to many related CPP approaches is that the TIM material will add up – and increase the covered area – if the same spatial area is covered twice.

This kind of complex coverage model is missing in almost all related work. An automated dispense pattern optimization based on such a flow behavior model has first been shortly mentioned in a preprint [43] of Baeuerle et al. [2]. In parallel to our research activity, the research group of Flaig and Kaufmann et al. [39,41] has also tried to optimize the area coverage based on a complex coverage model. The limitations of their approach have been outlined above.

To our knowledge, no further approaches considering a complex coverage behavior for the CPP problem have been proposed. Up to now, a feasible solution for this kind of problem is still missing. We propose the automated dispense path design approach *TIMtrace*, which yields a dispense pattern for a given cooling surface geometry. It considers the requirements, which are relevant for the dispensing equipment used in series manufacturing. We elaborate the detailed setup of our approach and the influence of particular parameters on the quality of the resulting dispense patterns. We show results for real automotive products. We further benchmark our automated method against manually designed dispense patterns, which have been proposed in past by design engineers for regular products of a major automotive company.

Section 2 outlines our new CPP approach for TIM. Section 3 describes the evaluations, which we carry out. This includes a study of the optimization setup and the benchmarking of specific target product geometries. Section 4 contains the respective results. Section 5 makes both advantages and disadvantages transparent and highlights the key features, which we found to be relevant for a successful path planning. Section 6 concludes our work.

3. Methodology

In this section, we give an overview of our optimization setup. All modules including their in- and outputs are depicted in the functional overview in Fig. 3. In this figure, a reference is made for each module to one of the following subsections, where its respective functionality is explained in detail. On a higher level, our approach has the following building blocks:

- An optimization algorithm — module *gradient-free optimizer*,
- two models for the physical manufacturing processes — module *2D discretization model* for the dispensing process and module *flow behavior model* for the joining process,
- multiple modules, which compute the individual components of the objective function — remaining modules.

3.1. Gradient-free optimizer

We aim to find the optimum dispense path. The dispense path is the output of the gradient-free optimizer and depicted in the upper left of Fig. 3. It is parameterized with a series of point coordinates (e.g. P1, P2, P3), which are connected by lines. For each line, one parameter (e.g. V12, V23) corresponding to the respective TIM material feedrate is set. The lines form a polygonal chain. The dispense path is fed into the models for the physical manufacturing processes, which are described in Section 3.2. The outputs of those models are fed into further

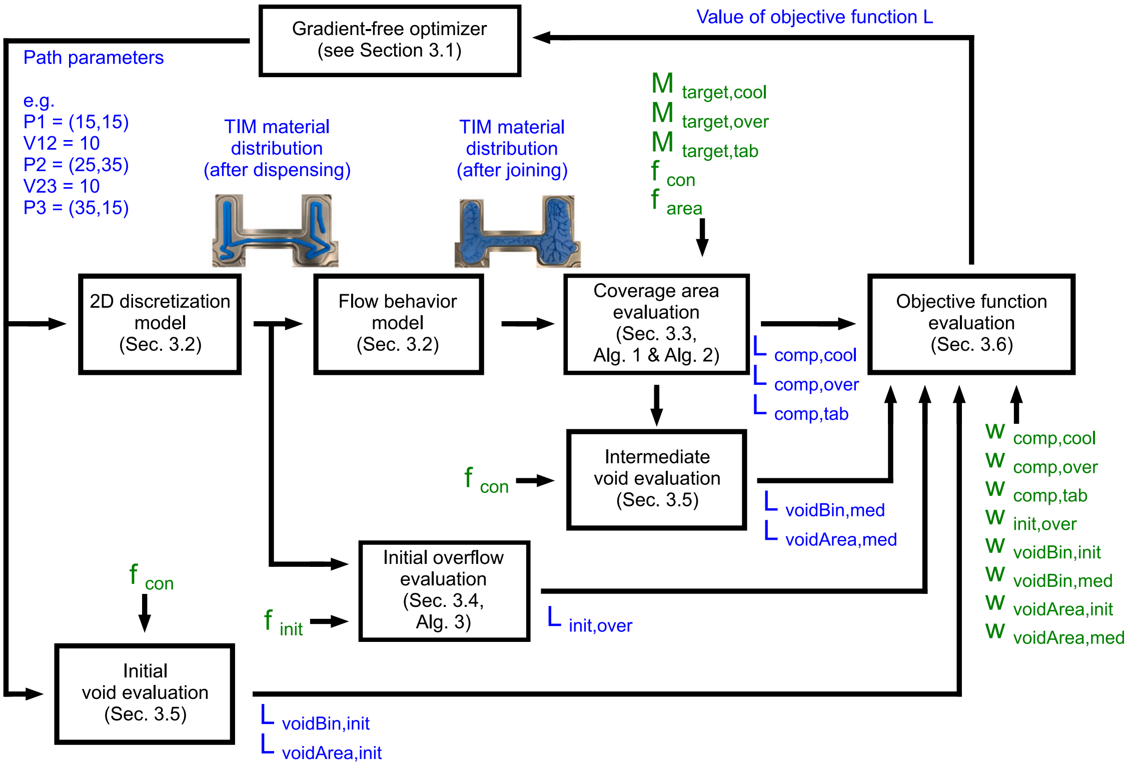

Gradient-free optimizer (see Section 3.1)
Value of objective function L
Path parameters
e.g.
P1 = (15,15)
V12 = 10
P2 = (25,35)
V23 = 10
P3 = (35,15)
TIM material distribution (after dispensing)
TIM material distribution (after joining)
M target,cool
M target,over
M target,tab
f con
f area
2D discretization model (Sec. 3.2)
Flow behavior model (Sec. 3.2)
Coverage area evaluation (Sec. 3.3, Alg. 1 & Alg. 2)
Objective function evaluation (Sec. 3.6)
L comp,cool
L comp,over
L comp,tab
f con
Intermediate void evaluation (Sec. 3.5)
L voidBin,med
L voidArea,med
W comp,cool
W comp,over
W comp,tab
W init,over
W voidBin,init
W voidBin,med
W voidArea,init
W voidArea,med
f con
Initial overflow evaluation (Sec. 3.4, Alg. 3)
f init
L init,over
Initial void evaluation (Sec. 3.5)
L voidBin,init
L voidArea,init

Fig. 3. Our approach sketched with all functional modules and intermediate in- and outputs. Each module is described in the referenced subsection.

modules, which compute the individual components of the objective function and which are explained in detail the later subsections. These components can be summarized to the following groups: Coverage area evaluation, initial overflow evaluation and void evaluation. During the computation of the objective function components, we apply several weighting functions using the following identifiers:

$$\begin{aligned} \text{none} &: f(x) = 0 \\ \text{con} &: f(x) = x \\ \text{lin} &: f(x) = ax \\ \text{squ} &: f(x) = x^2 \\ \text{log} &: f(x) = -1 \cdot \log(1-x). \end{aligned} \tag{1}$$

After computing all individual components, they are aggregated to a single value of the objective function. This value is fed back to the gradient-free optimizer, which then proceeds to update the path parameters of the dispense path iteratively.

Thus, the objective function serves as a dependent variable for the optimizer. The independent variables to be adjusted by the optimizer are the point coordinates of the dispense path. While the feedrate of the dispense path could be set individually for each path segment, we keep it constant along the entire dispense path. The feedrate is computed to match the total amount of TIM, which is needed to cover a given surface geometry. As a gradient-free optimizer, we utilize the optimizer CMA-ES as it is implemented in the *Optuna* library [44]. CMA-ES can efficiently walk through a large search space. It considers the correlations of the parameters. CMA-ES has shown a very good performance on black box optimization problems [45].

3.2. 2D discretization model and flow behavior model

The module *2D discretization model* for the dispensing process and the module *flow behavior model* for the joining process have been introduced previously by Baeuerle et al. [2]. In this subsection we shortly recapitulate the findings of this previous work and state how we utilize those two modules in the scope of this work.

The dispense path, which is parameterized as polygonal chain as introduced in the previous subsection, is transferred to a two-dimensional representation with a 2D discretization model. This represents the dispensed state of material. The dispensing process is followed by the joining process. Two flow behavior models have been introduced by Baeuerle et al. [2], which model the flow of material during the joining process, i.e. during pressing the heatsink onto the dispensed TIM. The output is the compressed state of TIM. The first flow behavior model is a heuristic based on the material volume conservation and does not include any further effects such as viscosity or friction. The second flow behavior model is an ANN trained on the data of the heuristic. Both models were validated on experimental data. The previous work was mainly focused on the flow behavior models and their computation speed. Both the material distribution after dispensing and the material distribution after joining are discretized to 50 × 50 grid cells. A number in each cell represents the respective TIM amount per grid cell [2].

Both the ANN and the heuristic are a flow behavior model as depicted in the center of Fig. 3. While the ANN is quicker in terms of computation speed, it introduces a small error with respect to the predicted compressed material distribution. In order to analyze the performance of the optimization as accurately as possible within this work, we forego the advantages of the ANN with regard to computation speed and instead work with the original heuristic to retain the best accuracy. To summarize: we use the 2D discretization model and the heuristic flow behavior model to model the behavior of the physical manufacturing processes.

Both are essential modules in our approach as they model the physical process output for a given dispense path. This enables a subsequent evaluation of the process output.

3.3. Coverage area evaluation

The coverage area evaluation is the most obvious component of our objective function. We compare the actual coverage with the desired

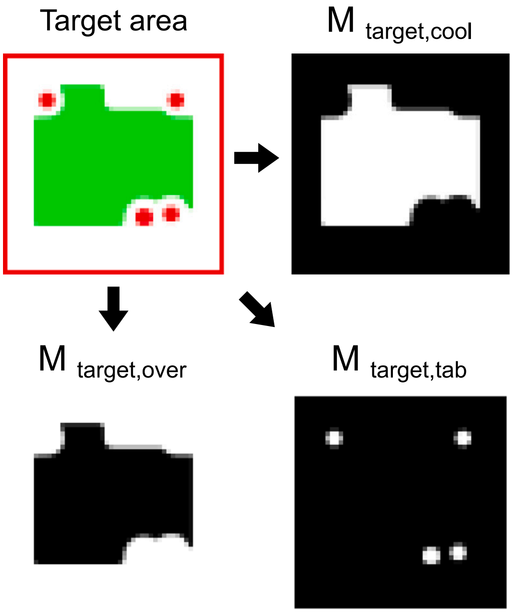


Fig. 4. Split of overall target area in three individual target area types. $M_{target,cool}$ (green color) defines the cooling area, $M_{target,over}$ (white and red color) the non-cooling area and $M_{target,tab}$ (red color) the taboo zones. In each grayscale image, the white color hue indicates the respective spatial location belonging to the specified target area type.

coverage. We distinguish three area types in our desired coverage as shown in Fig. 4. The green color represents the cooling surface, which should be covered. Its area definition is stored in the matrix $M_{target,cool}$, which can be interpreted as a grayscale image with values in the range from zero to one. The white color represents the surface of surrounding parts, where the TIM may overflow into, but which would increase material waste. This surface is stored in the matrix $M_{target,over}$. The red color represents so-called taboo zones, which must not be touched by TIM. These are stored in $M_{target,tab}$.

A reduced color saturation represents the grid cell having several area assignments in proportion to the saturation. I.e. a cell with a light-green color hue should ideally be covered only proportionally to its color saturation. A cell with a light-red color hue is weighted less than a fully red cell during evaluation. Since we work with a 50 × 50 grid resolution, the resolution is limited to the discretized cells. When working without the reduced color saturation, the accuracy would be bound to the discretization, i.e. a cell can be either fully green, white or red. Allowing a reduced color saturation along with several area assignments/colors enables us to represent an area boundary, which ends between two grid cells. Effectively, this enables a higher resolution for the area definitions.

We test different evaluation strategies for the material coverage. The first strategy *S-con* simply sums up the material amounts in each area type and applies either no weighting (*con*), square weighting (*squ*) or logarithmic weighting (*log*). The second strategy *S-area* applies the weighting on the area itself already, i.e. before summing up the individual values. For this kind of weighting we utilize linear (*lin*), square (*squ*) or logarithmic (*log*) weighting.

The strategy *S-con* is calculated according to the pseudocode in Algorithm 1. The function has the following inputs: M_{comp}, M_{target}, $M_{target,cool}$, f_{con}. M_{comp} is a matrix with float values within the interval from zero to one. It represents the actual compressed state. M_{target} is a matrix with integer values of either zero or one. It can be one of the predefined target states $M_{target,cool}$, $M_{target,over}$ or $M_{target,tab}$, which have been introduced above. f_{con} is a string and specifies the weighting function to be used as defined in Eq. (1). Algorithm 1 returns the float value L_{comp}. Depending on the given M_{target}, this is the corresponding objective function term for each of the area types $L_{comp,cool}$, $L_{comp,over}$ or $L_{comp,tab}$. Algorithm 1 needs to be executed first for $M_{target,cool}$. This will store a normalization value in a global variable, which is used for the remaining target area types. For each area type, the respective grayscale image ($M_{target,cool}$, $M_{target,over}$ or $M_{target,tab}$) is multiplied cell-wise with the material distribution after the joining process M_{comp} and summed up. This yields one actual coverage sum per area type. This sum is then normalized with the sum of a fully covered cooling area and clipped to a maximum equal to one. Dependent on the desired weighting function, the normalized area sum is then squared, processed with a logarithmic function or left as-is (no weighting). The results for each target area type are weighted relatively to each other by multiplying them with another weighting factor w_{comp}:

$$\begin{aligned} L_{comp,total} &= w_{comp,cool} \cdot L_{comp,cool} \\ &+ w_{comp,over} \cdot L_{comp,over} \\ &+ w_{comp,tab} \cdot L_{comp,tab}, \end{aligned} \tag{2}$$

with $w_{comp,cool}$ being the weighting factor for the cooling surface, $w_{comp,over}$ being the weighting factor for the overflow area and $w_{comp,tab}$ being the weighting factor for the taboo zones.

Algorithm 1 S-con

```
S-con (M_comp, M_target, M_target,cool, f_con)
targetCover = multiply(M_comp, M_target)
normCover = targetCover / sum(M_target,cool)
clipCover =min(normCover, 1)
If M_target = “M_target,cool”:
    L_comp = f_con(1-clipCover)
Else:
    L_comp = f_con(clipCover)
Return L_comp
```

Algorithm 2 S-area

```
S-area (M_comp, M_target, M_target,cool, f_area)
distTrafo = cv2.distanceTrafo(M_target)
If M_target = M_target,cool:
    Global maxDistCool = max(distTrafo)
clipDistTrafo = clip(distTrafo, [0, maxDistCool])
normDistTrafo = clipDistTrafo / maxDistCool
fctDistTrafo = f_area(normDistTrafo)
binaryCover = where(M_comp > 0, 1, 0)
If M_target = M_target,cool:
    Global maxMatSumCool = sum(fctDistTrafo)
wghtCover = multiply(binaryCover, fctDistTrafo)
normCover = sum(wghtCover) / maxMatSumCool
L_comp = clip(normCover, [0, 1])
Return L_comp
```

The strategy *S-area* is calculated according to the pseudocode in Algorithm 2. It is an alternative to the strategy *S-con*. Its input values, identifiers and output values are the same, i.e. Eq. (2) is valid here as well. In contrast to *S-con*, the weighting is applied at the individual grid cells based on the spatial location thereof. The grayscale image for each area type is processed with the function *distanceTransform()* from the *OpenCV* library [46] using an Euclidean distance metric. An example of a transformed area with a reduced resolution of 10 × 10 grid cells is shown on the right of Fig. 5. Its grayscale values correspond to the weighting factors for each grid cell. Grid cells close to the target area in the center exhibit low values (dark gray). The weights increase proportionally to the distance to the target area, i.e. outer regions have a maximum weight (white color). In the case of the overflow

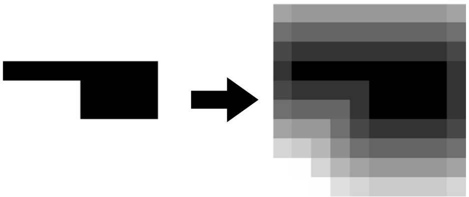


Fig. 5. Spatial weighting of target area. The weighting factors are specified for each grid cell. They increase proportionally to the distance between the respective grid cell and the reference area.

or taboo area, its values are clipped to the same maximum distance-transformed value as calculated for the cooling area. All resulting cell values are normalized to a range from zero to one. For each grid cell, the values are weighted using the linear, square or logarithmic function or left as-is in the case of no cell-wise weighting. The result is cell-wise multiplied with the actual coverage of TIM after joining. The sum is then normalized again using the sum of the cooling area and clipped to a range from zero to one. The results for each target area type are weighted relatively to each other in analogy to Eq. (2).

Two hyperparameters are set to control the weighting of the area loss. The parameter f_{area} is set to be either *con*, *lin*, *squ* or *log*. The parameter f_{con} is set to be either *con* or *log*. Setting f_{area} = *con* will set L_{comp} to the result of *S-con()* as defined in Algorithm 1 (*S-area()* will not be called). Setting f_{area} to either *lin*, *squ* or *log* will compute L_{comp} for the respective target area type using the result of *S-area()* as defined in Algorithm 2 (*S-con()* will not be called). Thus, only one of the weightings can be active at the same time, i.e. L_{comp} as defined in Eq. (2) will take the value of only one of the outputs of either *S-con()* or *S-area()*. *S-con* and *S-area* utilize the same relative weighting factors w_{comp} and return the resulting loss term with the same identifier L_{comp}.

Applying weighting functions to such a loss component affects the shape of the overall objective function and can have an influence on the convergence behavior during the optimization. For example, the objective function might exhibit a plateau, i.e. a region in its input parameter space for which its output values do not change significantly. Here, using such a weighting function would turn this flat plateau into an inclining region and enable an optimization algorithm to identify a better search direction. The strategy *S-area* applies this line of thought to the formulation of the area coverage loss. A violation of the taboo zone, which affects only a small portion of the taboo zone margin but reaches deep into it will be penalized heavily. A violation of the taboo zone, which affects a wider portion of the taboo zone margin but does not reach deep into it will be penalized less. This shapes the objective function in a way that supports the optimizer to find paths which avoid the taboo zone. The strategy *S-con* without any weighting can be considered as benchmark for any weighting strategies. Both the squared and the logarithmic weighting will penalize large violations of taboo zones heavier as compared to not applying a weighting — this is true both for *S-con* and *S-area*. Since the feedrates can be adjusted, it is important to avoid large violations under consideration of our target application. Due to the stochastic nature of CMA-ES such a formulation can yield a diversity of reasonable result paths.

3.4. Initial overflow evaluation

Our optimization problem is set up unconstrained, i.e. we do not pose explicit constraints on the independent variables. Thus, the optimizer may set path points beyond the cooling surface. Those paths will lead to bad coverage results also, but they can easily be accounted for separately. We formulate an additional loss term, which is added to the objective function. It directly penalizes initial paths outside of the cooling surface area and can be interpreted as a soft constraint in the space of the independent variables. We expect this term to benefit the convergence properties of the optimization problem.

Algorithm 3 S-init

```
S-init (M_initial, M_target,over, f_init)
distTrafo = cv2.distanceTrafo(M_target,over)
maxDist = 25 # Half of the overall grid height/width
clipDistTrafo = clip(distTrafo, [0, maxDist])
normDistTrafo = clipDistTrafo / maxDist
fctDistTrafo = f_init(normDistTrafo)
binaryCover = where(M_ initial >0, 1, 0)
wghtCover = multiply(binaryCover, fctDistTrafo)
maxMatSum = sum(fctDistTrafo)
L_ init,over = sum(wghtCover) / maxMatSum
Return L_ init,over
```

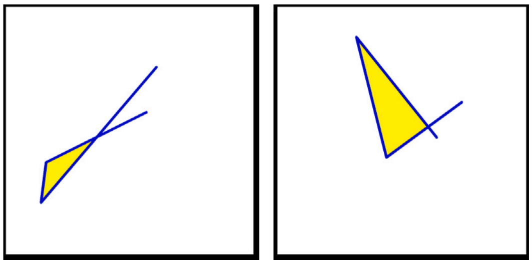


Fig. 6. Two exemplary TIM dispense patterns with the material itself depicted in blue color. The highlighted areas (yellow) indicate initial voids as they may appear directly after the dispensing process.

We utilize part of the computation approach presented during the coverage area evaluation as described in pseudocode in Algorithm 3. The function has the following inputs: $M_{initial}$, $M_{target,over}$, f_{init}. It works in analogy to Algorithm 2 and is calculated for the initially dispensed material (instead of the compressed material). Here, the spatial distance values for the individual grid cells are clipped to a maximum distance of half of the entire grid size (maxDist = 25). Similar to f_{area}, f_{init} is set to be either *lin* or *log* as defined in Eq. (1). We refer to the output of this computation with $L_{init,over}$.

3.5. Initial void evaluation and intermediate void evaluation

Voids can be prevalent directly after the dispensing. If the path segments cross each other, this will yield a direct formation of a void. This case is easy to detect, since it is obvious from the dispense path. We call this kind of voids *initial voids*. In order to detect them with an algorithm, we scale up the resolution from 50 × 50 to 1000 × 1000 and re-apply the discretization. We then apply the *OpenCV* [46] function *cv2.connectedComponentsWithStats()* to detect the number of connected areas and to measure the size of the areas. If the number of connected areas is larger than two, a void is present. The detection of an initial void is shown in Fig. 6. We assess two types of integrating this kind of void into our objective function. First, we define the binary term $L_{void\,Bin,init}$ to be either 0 (in case of no void) or 1 (in case of at least one void). Second, we describe the computation of the term $L_{void\,Area,init}$. The area *voidArea* is equal to the area of the void as computed by *cv2.connectedComponentsWithStats()*. In case of multiple voids, the areas are summed up. We divide *voidArea* by the sum of $M_{target,cool}$ and assign the result to $L_{void\,Area,init,norm}$. We clip the value range to a maximum of one and apply the weighting function defined by *f_con*. The result is identified by $L_{void\,Area,init}$.

Voids can also form during the joining process. This is the case, when the TIM fluid fronts collide in a way that traps air within. Especially for complex dispense paths, this may not be obvious at first sight and we refer to this kind of void as an *intermediate void*. In order to detect this kind of voids, we leverage the iterative nature

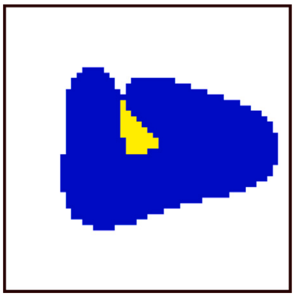


Fig. 7. An exemplary TIM dispense pattern. The shown state is captured halfway through the joining process. The material (blue) is spreading, but has not reached its final state yet. The highlighted area (yellow) indicates an intermediate void, which can occur during the joining process.

of the heuristic flow behavior model of Baeuerle et al. [2]. Since the compression of the material is computed at discretized gap heights, which decrease iteratively, we can apply the function *cv2.connectedComponentsWithStats()* at each iteration. This enables us to detect most of the intermediate voids. The presence of these voids is captured by the binary term $L_{void\,Bin,med}$, which is defined in analogy to $L_{void\,Bin,init}$. We apply the same procedure of measuring and normalizing the void area as with the initial voids and obtain $L_{void\,Area,med}$. The void area is calculated at the iteration of the flow behavior model, where the intermediate void is detected first. If the void is present during subsequent iterations of the flow behavior model, the areas are not aggregated. A visual depiction of an intermediate void is shown in Fig. 7.

A binary term to penalize voids can be considered as a simple benchmark: voids are to be avoided at all and their occurrence is penalized. Voids occur with closed path contours. When the path coordinates are adjusted during the optimization procedure, a path might suddenly change from an open contour to a closed contour. This introduces a jump in the objective function value. As long as the contour stays closed, the loss term for the void stays at a constant high value. This manifests as a plateau in the objective function. As the void is resolved, another jump occurs in the objective function. Formulating the void loss term using an area-based formulation will resolve the plateau and instead enable the optimizer to identify a good search direction. Furthermore, it resolves some (but not all) jumps. While the binary formulation is more straightforward we expect better convergence properties by using an area-based loss term.

3.6. Objective function evaluation

We obtain the overall objective function by summing up the above terms. Each term is weighted with a relative weighting factor w:

$$\begin{aligned} L = {} & w_{comp,cool} \cdot L_{comp,cool} \\ & + w_{comp,over} \cdot L_{comp,over} \\ & + w_{comp,tab} \cdot L_{comp,tab} \\ & + w_{init,over} \cdot L_{init,over} \\ & + w_{void\,Bin} \cdot L_{void\,Bin,init} \\ & + w_{void\,Bin} \cdot L_{void\,Bin,med} \\ & + w_{void\,Area} \cdot L_{void\,Area,init} \\ & + w_{void\,Area} \cdot L_{void\,Area,med}. \end{aligned} \tag{3}$$

Note that the individual components L_{comp} and $L_{void\,Area}$ depend on the value of f_con. L_{comp} further depends on f_{area}. $L_{init,over}$ depends on f_{init}.

4. Experimental setup

This section describes the setup of our analysis in order to evaluate our proposed optimization approach.

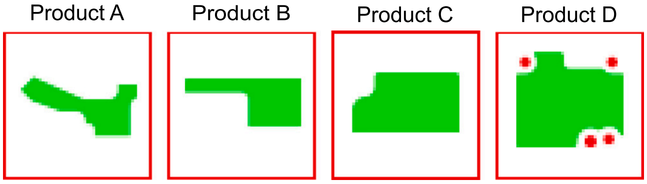


Fig. 8. Target area definition of four real-world automotive products. These are used during the benchmarking of our proposed optimization approach against expert paths.

4.1. Evaluation on products A-D (existing products)

We evaluate our new optimization approach on four existing automotive products from the fields of electronic control units and power electronics. The definition of the desired target coverage is shown in Fig. 8. The color coding follows the area definitions above: Green is the cooling surface, white is the non-critical overflow area and red is the taboo zone area. We run the stochastic optimization multiple times for several days. This yields many result paths for each product. The results are sorted by the area coverage of the cooling surface and a suitable path is chosen manually from the top of the list. Since those products are being manufactured already, we can acquire the corresponding expert path, i.e. the dispense path which has been designed by an engineer. At that time, the optimization was not available to support the manual planning process. We transfer the expert path into a representation which is compatible to our parameterization of the path. This enables a direct comparison of the optimization result with the expert path using the flow behavior model proposed by Baeuerle et al. [2]. The transfer of the expert path into a compatible parameterization is carried out as follows. The coordinates can be transferred directly from the coordinates of the technical drawing. Some products exhibit a rather large mechanical tolerance with respect to the gap height after joining (which is the same as the bondline thickness of TIM layer). This makes it difficult to set the dispense amounts correctly. To enable a fair comparison, we adjust the dispense amounts of the expert path in a way that yields an almost complete coverage of the cooling surface. We notice that at some point further increasing the dispense amounts leads to a disproportionally high material overflow. We stop adjusting the dispense amounts before exceeding this point. Thus, not all expert paths yield a 100 % coverage of the cooling surface. This coverage is reported as *coverage of cooling surface* in Table 1. We set the dispense amounts of the optimized path in way that yield the very same coverage percentage of the cooling surface. At this exact coverage percentage, we compare the optimized path against the expert path. For both the expert path and the optimized path, we measure the amount of TIM that was overflowing beyond the cooling surface and divide it by the amount of material within the cooling surface. This is reported as overflow ratio in Table 1.

4.2. Evaluation on products E & F (new products)

In parallel to our research activity, two new products were being introduced. We offered to assist the manual dispense path design with our proposed optimization approach *TIMtrace*. Due to the large cooling surface area, we split the area into smaller sub-areas and run the optimization on each sub-area separately. To ensure a continuous dispense path for the aggregated area, we set the path points on the connecting edges to a fixed coordinate. The product area and its split into the sub-areas are shown in Fig. 9 for Product E and in Fig. 10 for Product F. The proposals from our approach are slightly adjusted manually and used during the series ramp-up of the respective product. For Product E, TIM is applied onto the mounted electronic components. The underlying flow behavior model assumes plane surfaces. Thus, the real-world surface geometry formally violates the constraints of the flow behavior model. Even in the case of such a violation, the flow behavior model can be utilized to enable an optimization of the dispense path.

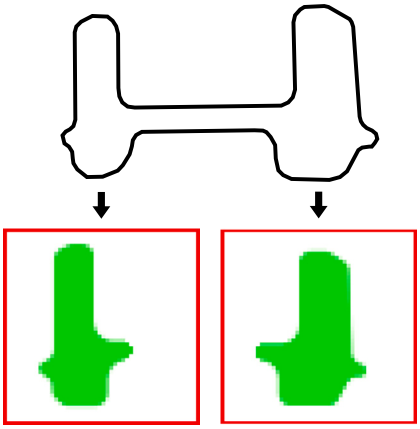


Fig. 9. Test on Product E, which is a new product. The cooling surface of Product E is sketched at the top. It is split into two sub-areas as shown on the bottom, which are to be optimized individually. The connecting region in the center is not included in the optimization.

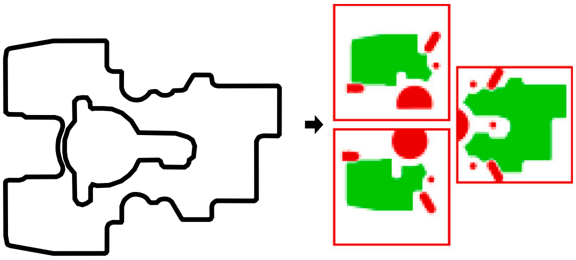


Fig. 10. Test on Product F, which is also a new product. We split the target area surface – similar as with Product E – into sub-areas, which are to be optimized individually.

4.3. Configuration of objective function

We have proposed multiple components of the objective function, which can be weighted and then combined in many different ways. This can be interpreted as a hyperparameter optimization problem in itself. Due to the stochastic nature of the optimizer, results vary between multiple optimization runs. Different cooling surface geometries might have different sets of optimal hyperparameters. We manually define a set of hyperparameters to be assessed for a single product in the scope of this work. We are aware, that this might not constitute an overall optimum and hope to gain insights for future studies. We formulate the optimization as a minimization problem. In the best case, the cooling surface is covered by 100 %. We set the feedrate to be constant along the entire path and the total initial TIM amount equal to the theoretically needed TIM amount to cover exactly 100 % of the cooling surface. In this setting, $w_{comp,cool}$ can be set to zero: this overall fixed TIM amount combined with a penalty for the overflowing material will yield an optimum value for a complete coverage of the cooling surface. An alternative to this setup would be to treat the overall TIM amount as an independent variable (to be optimized) along with setting $w_{comp,cool}$ to a negative value.

As the values of the remaining objective function terms are essentially to be optimized in relation to each other and scaling all weights with the same factor will have no effect, we set $w_{comp,over} = 1$. The clearance of a taboo zone constitutes a functional requirement for the resulting product. A violation of a taboo zone leads to the product being sorted out during quality inspection. If we estimate the cost of TIM to be around 0.50 Euro per part and the cost of an entire power electronics product to be around 500 Euros, we assume a weighting of $w_{comp,taboo} = 1000 \cdot w_{comp,over}$ to be a reasonable choice. A similar estimation can be done for the voids that are located in the inner region of the cooling surface area. Since dispensing TIM beyond the cooling surface obviously does not make any sense, we define the weight for $w_{init,over}$ to be in similar orders of magnitude as $w_{comp,taboo}$ and w_{void}. In our trial we test a rather broad range of order of magnitudes of $w \in [10, 100, 1000, 10000]$ for each of $w_{comp,taboo}$, w_{void} and $w_{init,over}$. We further test different weighting functions on grid-cell level as described above by setting f_{con}, f_{area} and f_{init}. A full grid search of this space is not feasible. The exact hyperparameter combinations which we assess is documented in the Appendix. Each hyperparameter configuration is executed 100 times for paths with a length of 5 to 10 segments and with 1000 optimization iterations of CMA-ES.

For each hyperparameter setting, we compute three metrics: the average coverage of the cooling surface, an auxiliary value for the convergence properties and the average of both, which we refer to as *average performance ratio*. The coverage is readily available from the coverage area evaluation procedure. We calculate the average of the coverage percentage over all runs in each individual hyperparameter setting. We obtain a single average coverage value in the range from zero to one. The auxiliary value for the convergence properties is calculated as follows. We define a binary value for each hyperparameter setting with a value of one corresponding to the convergence to a good local minimum. A value of zero corresponds to a not usable dispense path. The conditions are: at least 80 % coverage of the cooling surface area, a maximum violation of a taboo zone area of 1 % relative to the cooling surface area, a maximum void area of 5 % relative to the cooling surface area. These binary values are added up and divided by the total number of optimization runs for each hyperparameter setting. This again yields a single auxiliary value in the range from zero to one.

4.4. Compliance with mechanical tolerances

Real products are subject to mechanical tolerances. With regard to the dispense path planning, especially the tolerances with respect to the final gap height after joining are relevant. The tolerances are in some cases significant and the tolerance width can even be in the same range as the total gap height. This has implications on the dispense path planning: In a maximum gap setting, the coverage of the cooling surface needs to be sufficient. In a minimum gap setting, the overflowing material may not violate any taboo zone. We comply with this kind of tolerance by evaluating the material coverage at the maximum gap height in a first step. In a second step, we compress the TIM material further down to the minimum gap height and evaluate the material coverage again. The objective terms with regard to area coverage are added up for both gap heights. This forces the optimizer to find a dispense path, which both covers as much of the cooling surface as possible in the maximum gap setting and avoids the violation of taboo zones in a minimum gap setting.

4.5. Material compatibility

Similar to the work of Baeuerle et al. [2], we have used Material A for the major part of our study. Material A is a widely used TIM material and is analyzed on the example of Product E. It consists of two components, is addition-cured, room temperature curable, silicon based and has a viscosity of 200 Pa · s. The filler material is composed of an aluminum oxide ceramic material.

Material B is another widely used TIM material and is analyzed on the example of Product F. It consists of two components, is high temperature curable, silicon based and has a viscosity of 350 Pa · s. The filler material is composed of an aluminum oxide ceramic material.

Material C is an adhesive. It is room temperature curable, silicon based and has a viscosity of 130 Pa · s. With this material we extend our study beyond TIM and aim to get an indication on the transferability to other applications. We use the same dispense path that we obtain for Product A and carry out the following test in a laboratory setting. The same path is dispensed onto a plane steel plate using both Material A

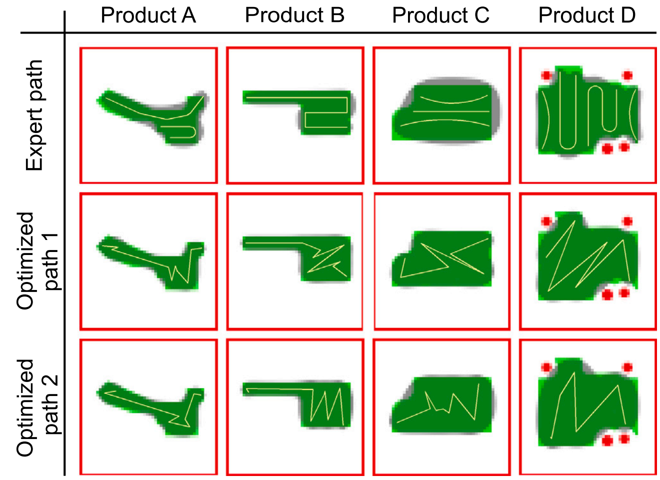


Fig. 11. Comparison of dispense patterns which are either designed by experts or are a result from our proposed Coverage Path Planning. The shown products are real products, which had been in series manufacturing before our approach was available. Thus, the expert paths are the result of the conventional dispense pattern design process under real-world conditions. They serve us as a realistic benchmark.

and Material C. It is compressed by another steel plate pushed down from the top with a professional compression testing machine. The testing machine is normally used for recording force-distance curves and ensures a precise guiding of the steel plate during compression. We compress the paths to the same final gap height and evaluate the compressed coverage area.

5. Results

In this section we present results from our proposed optimization approach *TIMtrace*: we present optimized dispense paths for real-world products and we evaluate the influence of the hyperparameters of the optimization.

5.1. Evaluation on products A-D (existing products)

The optimization results are visualized along with the expert paths in Fig. 11. The expert paths had been designed in past with the conventional manual planning approach, whereas the optimization results are generated using our newly proposed approach. The dispense paths are shown in a light-yellow cover. The resulting TIM coverage is shown as an overlay in a gray color on top of the target area definitions introduced in the previous section.

Table 1 shows the ratio of overflowing TIM material relative to TIM material amount within the cooling surface. The reduction of overflowing TIM material directly translates into a reduction of material waste. This is explicitly listed in Table 1. Reducing the overall dispensed amount translates into a reduction of cycle-time. The cycle-time is further reduced by reducing the number of path segments. For each additional path segment, the dispensing pump needs to be stopped, the toolhead needs to be moved to the starting point of the next segment and the dispensing pump needs to be started again. We report the reduction of cycle-time as listed in Table 1 based solely on the reduced dispensed TIM amounts. The further reduction of cycle-time based on the reduced number of segments – which is also realized by using our proposed approach – thus leads to an additional saving in cycle-time, which we did not quantify yet. The optimized paths all have a fixed material feedrate and they are not broken up into individual path segments at all. Besides a reduction in cycle-time, having a single continuous dispense path is furthermore beneficial with respect to process stability and quality inspection.

Table 1
Comparison of automated path planning results and designs from experts on existing products. The expert path is shown in the top row of Fig. 11. The optimized paths are shown in the respective center and bottom row.

	Product A	Product B	Product C	Product D
Coverage of cooling surface	94.3 %	98.3 %	97.2 %	95.5 %
TIM overflow ratio				
Expert path	20.3 %	13.1 %	19.5 %	8.7 %
Optimized path 1	**7.1 %**	**9.6 %**	**2.5 %**	**4.3 %**
Optimized path 2	**7.1 %**	12.1 %	8 %	5.9 %
Material waste reduction (for optimized path 1)	**65 %**	**27 %**	**87 %**	**51 %**
Number of segments				
Expert path	2	1	3	4
Optimized path 1	**1**	1	**1**	**1**
Optimized path 2	**1**	1	**1**	**1**
Cycle-time reduction (for optimized path 1)	**>13 %**	**4 %**	**>17 %**	**>4 %**
Development effort reduction		**>80 %**		

All of the resulting dispense patterns respect the functional constraints regarding both voids and taboo zones. Manufacturing tolerances (see Section 5.4) are not included in this table, i.e. we optimize for the nominal gap height.

The reported reduction of development effort in Table 1 is based on experience in recent trials involving further actual products. The manual planning by an expert consumes at least two weeks of working time. The two weeks results from the following activities. Laboratory trials during the iterative manual design take up around six days and involve e.g. the alignment of machine capacity, preparation of machine and samples, actual trials and subsequent evaluation. Simulation trials take up three days and involve e.g. alignment with a simulation engineer, setting up the simulation model, carrying out the simulation and evaluation. One day is planned for the work of the design engineer. Another day is needed for any alignment on project level, including e.g. the alignment with the product engineers about the boundary conditions for the individual product. This adds up to ten days (or two working weeks). For multiple design iterations, the efforts both for laboratory and simulation trials increase correspondingly. The automated planning with our proposed approach can be performed with an effort of two days of working time: one day for alignment on project level and one day to convert the technical drawing to the input format of our approach and starting the automated procedure. The reduction from ten days to two days corresponds to a reduction of development effort by 80 percent. Typically, we see multiple iterations during the iterative manual planning approach which leads to a reduction larger than 80 percent.

As reported in Table 1, the advantage of using the optimization in terms of potential material savings is also considerable. This is especially the case for the shown products, since they are produced in high numbers during automotive series manufacturing.

Our proposed optimization approach *TIMtrace* outperforms the conventional manual planning method in all tested products for all criteria.

5.2. Evaluation on products E & F (new products)

The underlying flow behavior model itself was validated on experimental data in a laboratory setting using Material A by Baeuerle et al. [2]. The validation showed a good match with an error of the predicted coverage after joining being around 10 %. Within this work, we validate optimized dispense paths on two actual products (Product E & F). Product E contains Material A and Product F contains Material B. The optimization results were given to the process experts, who made only minor adjustments. The optimization output, the adjusted dispense pattern and the resulting coverage are shown in Fig. 12 for Product

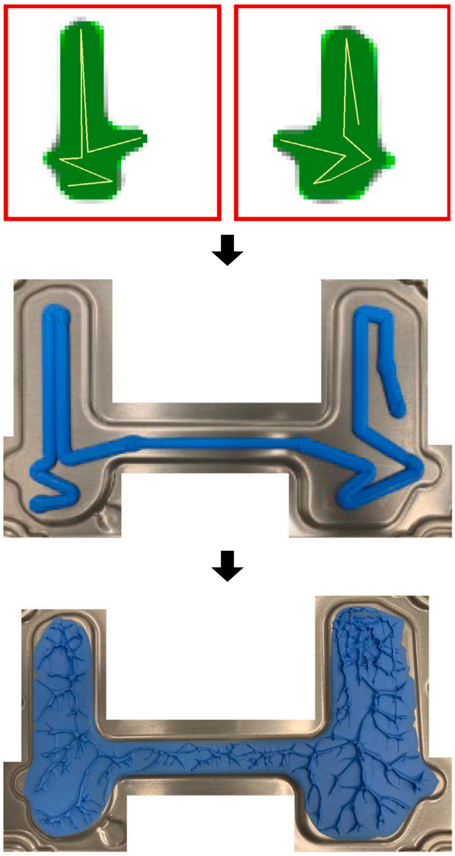


Fig. 12. Optimization results, dispensed path and resulting coverage for the cooling surface for Product E. In this case, our proposed approach has been used during the dispense pattern design (as opposed to Product A-D, where the approach was not available yet). Only small adaptions to the optimized dispense path were made by the process expert. This evaluation was carried out under real-world conditions during a regular series ramp-up. This is a contrast to both Baeuerle et al. [2] and Flaig et al. [40], which provide results in a laboratory setting.

E. For the optimization output at the top, the dispense path is shown in a light-yellow color and the resulting coverage in a gray color. It exhibits minor areas of the cooling surface not being covered. This is due to large mechanical tolerances and partially no components being mounted in the areas below the heatsink. Applying an evaluation in analogy to Products A-D, we obtain a coverage of 97.2 % of the cooling area surface with a waste of 6.9 %. The coverage has been assessed by the responsible engineers and found to meet the requirements well. The results from the optimizer served as a very good starting point for the dispense path planning and greatly simplified the design process.

Fig. 13 shows similar results for Product F and also contains the optimization output, the adjusted dispense pattern and the resulting coverage. We received very good feedback from the process experts regarding the output of *TIMtrace*: the surface was covered well despite the complex coverage area shape.

5.3. Configuration of objective function

A full report of the tested hyperparameter configurations is shown in the Appendix. The void area being part of the loss function showed a significant advantage over modeling the void presence in a binary fashion: utilizing $w_{void\,Area}$ instead of $w_{void\,Bin}$ consistently improved the convergence behavior by a large margin. More than 60 % of configurations using $w_{void\,Area}$ converge to a feasible solution, in contrast to only 30 % of configurations using $w_{void\,Bin}$. In this case, the parameter settings have a high influence on the stability of convergence.

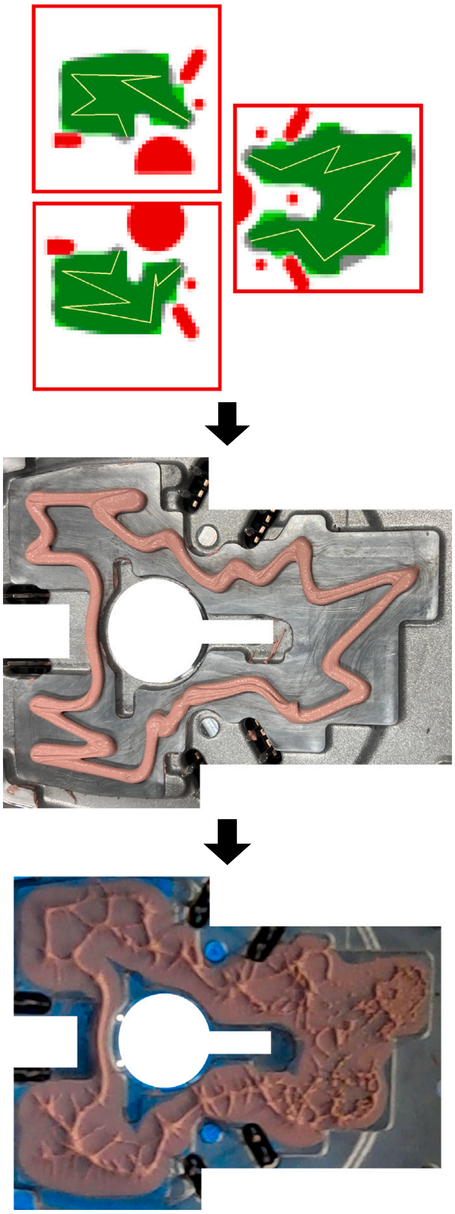


Fig. 13. Optimization results, dispensed path and resulting coverage for the cooling surface for Product F. This validation is carried out in analogy to Product E with *TIMtrace* being used to support the real-world development work. Since Product F uses a different TIM material to Product E, this validation proofs the transferability to a different TIM material in an actual series manufacturing setup.

Furthermore, we note the following effects of the parameter settings. Assigning a proportionally higher weight to $w_{init,over}$ as compared to $w_{comp,tab}$, $w_{void\,Bin}$ and $w_{void\,Area}$ improves the convergence behavior. Assigning a proportionally lower weight to $w_{init,over}$ not only worsens the convergence behavior, but also slightly decreases the achieved cooling surface coverage. We did not see a clear advantage for any of the weighting strategies for the coverage area evaluation. Overall, the combination of setting $w_{void\,Area}$ along with $w_{init,over}$ yielded best results with respect to both convergence properties and cooling surface coverage.

5.4. Compliance with mechanical tolerances

Fig. 14 shows the optimization result when including the mechanical tolerances during the formulation of the objective function. The

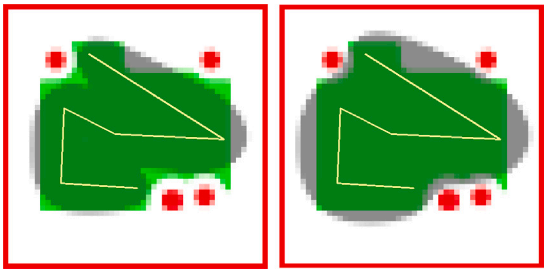


Fig. 14. The proposed optimization approach yields a solution for Product D, which complies with mechanical tolerances as follows: the resulting dispense path simultaneously matches the desired coverage conditions in a trade-off between both the maximum gap height and the minimum gap height. The left image shows the maximum gap setting, where a high coverage of the cooling surface (green) is achieved. The right image shows the minimum gap setting, where taboo zones (red) are being avoided. The shown dispense path – as indicated by the light-yellow line – fulfills both criteria at the same time.

optimizer manages to find a dispense path, which fulfills the requirements well both for a maximum and minimum gap height after joining: It manages to cover a large part of the cooling surface with the maximum gap height setting and to avoid taboo zones with the minimum gap height setting.

The extension of the objective function with mechanical tolerances is an optional extension of *TIMtrace*. We did not include it for our study of Products A-F. However, Product E and F both underwent limit sample tests during series manufacturing ramp-up. Limit sample tests involve the maximum expected mechanical tolerances. The proposed paths as shown in Figs. 12 and 13 passed all limit sample tests. Furthermore, many products with diverse tolerances have been manufactured by now and no error regarding the TIM layer has occurred. This means even without explicitly including the tolerances in our objective, the resulting paths perform well within the full range of real-world mechanical tolerances.

5.5. Material compatibility

The results both on Product E and Product F show the compatibility of *TIMtrace* with both Material A and Material B. Furthermore, we show results for the adhesive Material C in Fig. 15. We compare the coverage of Material C with the coverage of Material A as follows. We calculate the absolute error between both coverage areas and divide by the total coverage area of Material A. This gives us a relative error of 10.7 %. The resulting coverage areas are visually almost identical.

6. Discussion

The key to a successful optimization was to optimize the parameterized path as opposed to optimizing the initial material distribution (after dispensing/before joining). This restricts the parameter space to feasible states with regard to the manufacturing constraints: it significantly reduces the dimensionality of the parameter space, since instead of the full 2D distribution only a few path parameters are taken into account: Even with a rather coarse discretization of 50 × 50 grid cells this would lead to 2 500 degrees of freedom. In contrast, a polygonal chain with 10 segments has only 20 degrees of freedom — assuming constant material feedrate. Furthermore, additional constraints can be formulated straightforward. This includes further constraints which have not been taken into account during this work yet, e.g. maximum or minimum segment length, minimum distance between individual segments or a rounding at the corners to reduce mechanical accelerations of the dispense head.

Our chosen parameterization with a polygonal chain as path definition and a constant feedrate enables a quick and easy transfer to actual manufacturing equipment. It is fully compatible with the constraints of commonly used CNC machines. The effort to write the CNC code for a given polygonal chain is very low compared to alternative path parameterizations such as e.g. Bézier curves or splines. To transfer the paths onto manufacturing machines, the following actions have to be taken: insert a small radius at each corner (if desired), rewrite the path in CNC code, set the feedrate in CNC code. Those actions could largely be automated by a software code, which would accept our path parameterization format as input and deliver ready-to-use CNC machine code as output.

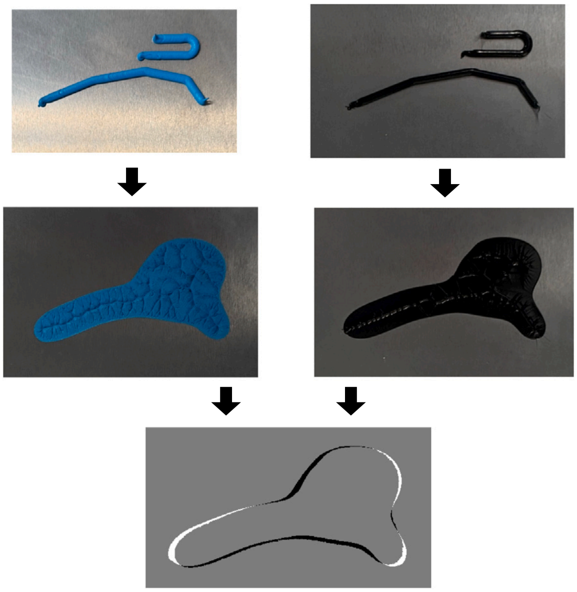


Fig. 15. Comparison of the coverage area for the TIM Material A and the adhesive Material C when using the same dispense path. Top row: dispensed path, center row: compressed path with respective area coverage, bottom row: pixel-wise difference of both area coverages. This experimental validation was carried out in a laboratory setting. Even though material properties such as viscosity are different, the resulting coverage is visually almost identical.

We see the choice of the parameter space to be optimized to be an issue with the studies of Kaufmann and Flaig et al. [39,41]. They closely orient their optimization approach along the lines of classic topology optimization. They optimize the material distribution after dispensing with the help of the flow behavior model of Kaufmann et al. [3] This effectively leads to the material being drawn back to the geometric center lines of the target surface and closely resembles the results of applying a medial axis transform.

We studied our optimization approach *TIMtrace* on TIM dispensing, but it is applicable to adhesive application: we show a very similar coverage area both for TIM and an adhesive in Fig. 15. This would benefit an even wider application range. Our results with three different materials with viscosities from 130 Pa·s to 350 Pa·s indicate that the underlying flow behavior model could potentially be used in an extended application scope. For materials with significantly different rheological properties such as very high or very low viscosities the underlying flow behavior model would need to be exchanged for a suitable flow behavior model. Regarding the transfer to applications beyond TIM the respective dispensing requirements need to be considered. Our path parameterization can be used with selective dispensing technologies such as time-pressure dispensing, rotary screw dispensing and positive displacement piston dispensing. For jetting dispensing or stencil printing we would recommend using a different path parameterization.

A constant feedrate as we implemented it during this work has advantages with regard to quality inspection. One type of quality inspection is carried out by taking a picture of the dispensed path before joining the heatsink. In this picture, the dispense path width is visible very well, but no height information is available. Given a constant feedrate any errors during the dispensing can be detected easily as deviations in path widths, whereas a varying feedrate will lead to changing path widths. This is a feedback we received directly from a manufacturing plant where lean and robust solutions are generally preferred. An uninterrupted path with a high constant feedrate is further preferable in terms of cycle-time. Creating a model for the cycle-time and integrating it into the objective function would be an extension of our approach, which would cover both objectives even better. Furthermore, a model for the joining forces could be included to reduce material stress.

While we worked with a constant feedrate along the entire dispense path, a piece-wise constant or even varying feedrate can be realized easily. A varying feedrate in this case means a dynamically changing feedrate along the length of the dispense path. This is slightly more elaborate to implement on manufacturing machines, which typically rely on Computerized Numerical Control (CNC) to define and execute the given paths. However, it creates additional degrees of freedom for the optimizer while still being within a feasible parameter range of the manufacturing equipment. This leads to an even better area coverage as tested by us in initial virtual trials. Currently, we stick to a constant feedrate. This is due to the already mentioned advantage in terms of quality inspection and it further simplifies the transfer of the proposed paths to the process engineers and finally to the CNC machinery.

We saw the formulation of boundary conditions as an objective rather than a hard constraint to be advantageous in terms of convergence properties: reducing the void area rather than strictly preventing voids in a binary fashion showed clearly superior results. This validates the findings of Ellefsen et al. [21], who proposed to formulate an objective with continuous values rather than an objective with hard constraints which strictly prohibit any violation.

Overall, the formulation of the objective function significantly affected the quality of the results and the convergence properties: we show that different parameter settings for the various loss terms are an important factor for the optimization setup. While we analyzed a sensibly chosen set of hyperparameters, we do not claim to have found the best hyperparameter setting yet. Introducing an automated hyperparameter optimization could further benefit the setup. However, since the hyperparameters are set for the optimization itself, the computation time for such an optimization would be quite high.

We discuss our approach mainly in the context of automotive industry. The analyzed products are from the field of either power electronics or Electronic Control Units (ECUs). We do not see any limitations regarding the transfer to products from other industries using TIM and having a similar dimensions and functionality. Prominent examples are inverters and chargers for solar power systems and other renewable energy sources. The dimensions of our tested products are in the range of roughly three to thirty centimeters when measuring the shortest and longest edge of the minimum bounding box around each cooling surface area.

A limitation of our approach is the computation speed. Adjusting the hyperparameters has improved the speed significantly. However, we are running the optimizer many times for a single given product surface and it takes about one week to find a very well-tuned dispense path. We run the optimization trials in parallel on each core of an INTEL Xeon E5-2680 processor. Here, introducing the ANN as flow behavior model as proposed by Baeuerle et al. [2] could help. This would not only decrease the computation time of the gradient-free optimization, but would also enable the use of gradient-based optimizers.

A further limitation of our approach is the limitation of the underlying flow behavior model. We have successfully validated our approach with material viscosities in the range of 130 Pa·s to 350 Pa·s, but it might not work for materials with significantly higher or lower viscosity and might not work under special conditions such as very high or low temperatures. We emphasize that this is not an issue of the optimization approach that we present in this work, but rather an issue of the flow behavior model as proposed by Baeuerle et al. [2]. Our optimization approach can work as-is with another flow behavior model. This can be implemented by simply exchanging it in the respective module as shown in Fig. 3. The remaining modules, loss function components and weight will remain largely the same. We have carried out a quick test and appended the respective results in Appendix. Special conditions such as very high or low temperatures are not relevant for our application, since the ambient conditions in manufacturing are well controlled.

An alternative flow behavior model which can model all of the previous aspects accurately would still be required to be executed within a comparably low time. We are not aware of a model that can fulfill both requirements. Typically, more accurate models like CFD models suffer from (very) high computation times. As the accuracy for TIM has been analyzed and quantified for the model we use (see also [2]), this aspect is mainly relevant for the transfer to other applications beyond TIM. We have not done an elaborate quantification yet involving materials other than different TIM materials, thus we currently cannot give a definitive statement regarding the application boundaries in other applications beyond TIM. In our application the utilized flow behavior model performs very well under all conditions that we have tested. This includes testing under real-world conditions at multiple international locations using two commonly used materials and actual products.

Beyond the tests that we carry out ourselves regarding the area coverage and that we show in Figs. 12 and 13, further tests are carried out for new products. This includes electrical testing of the final product and reliability assessment under varying temperature conditions. We did not see any negative effects from using the dispense paths as obtained from our proposed approach. The products underwent regular inspection and release procedures.

7. Conclusion

We extend the research field of Coverage Path Planning beyond constant coverage path widths towards a complex modeling of the area coverage. We propose the new optimization approach *TIMtrace* specifically for the dispensing of Thermal Interface Material. It is transferable to other dispensing applications such as adhesive dispensing. We lay out the needed configuration in detail. We validate our approach against existing manual dispense path planning and present experimental results not only in a laboratory setting but from actual series manufacturing equipment. *TIMtrace* fully complies with the essential manufacturing constraints and can easily be extended to further constraints. We plan to extend it towards further flow behavior models. Furthermore, we aim to study second-order optimization methods and the formulation of further constraints with respect to the dispense path.

CRediT authorship contribution statement

S. Baeuerle: Writing – original draft, Software, Methodology, Investigation, Funding acquisition, Conceptualization. **A. Steimer:** Writing – review & editing, Supervision, Methodology, Conceptualization. **R. Mikut:** Writing – review & editing, Supervision, Project administration, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

We thank Ralph Nyilas and Barnabas Szilagyi for the valuable exchange with respect to the dispensing process and the properties of the manufacturing equipment. The work of Ralf Mikut was supported by the Helmholtz Association's Initiative and Networking Fund through Helmholtz AI and the program Energy System Design.

Table 2
Parameter study for the components of the objective function. Good result values are indicated with a stronger blue color.

w_comp,tab	w_voidArea	w_voidBin	w_init,over	f_con	f_area	f_init	Coverage ratio of cooling surface [%]	Convergence ratio [%]	Average performance [%]
100	100	0	0	con	con	none	0.89	0.82	0.85
1000	1000	0	0	con	con	none	0.91	0.78	0.85
10000	10000	0	0	con	con	none	0.91	0.71	0.81
100	100	0	0	log	con	none	0.92	0.78	0.85
1000	1000	0	0	log	con	none	0.91	0.74	0.83
10000	10000	0	0	log	con	none	0.91	0.79	0.85
100	100	0	100	log	con	lin	0.93	0.82	0.87
100	100	0	1000	log	con	lin	0.93	0.84	0.88
100	100	0	100	log	con	log	0.93	0.81	0.87
100	100	0	1000	log	con	log	0.93	0.95	0.94
100	0	100	0	log	con	none	0.88	0.26	0.57
1000	0	1000	0	log	con	none	0.90	0.27	0.59
100	100	0	0	log	lin	none	0.92	0.77	0.84
100	100	0	0	log	squ	none	0.89	0.75	0.82
100	100	0	0	log	log	none	0.89	0.74	0.82
100	100	0	0	log	con	none	0.93	0.78	0.85
100	100	0	100	con	con	lin	0.92	0.88	0.90
100	100	0	1000	con	con	lin	0.93	0.94	0.93
100	100	0	100	con	con	log	0.92	0.82	0.87
100	100	0	1000	con	con	log	0.92	0.88	0.90
100	100	0	0	con	con	none	0.93	0.83	0.88
100	100	0	0	con	lin	none	0.90	0.74	0.82
100	100	0	0	con	squ	none	0.89	0.82	0.86
100	100	0	0	con	log	none	0.89	0.75	0.82
1000	1000	0	1000	log	con	lin	0.92	0.86	0.89
1000	1000	0	10000	log	con	lin	0.93	0.86	0.89
1000	1000	0	1000	log	con	log	0.92	0.84	0.88
1000	1000	0	10000	log	con	log	0.93	0.90	0.91
100	0	100	0	con	con	none	0.88	0.22	0.55
1000	0	1000	0	con	con	none	0.89	0.21	0.55
1000	0	1000	0	log	con	none	0.86	0.20	0.53
10000	0	10000	0	log	con	none	0.88	0.25	0.57
1000	1000	0	0	con	con	none	0.90	0.76	0.83
1000	1000	0	0	con	lin	none	0.92	0.79	0.85
1000	1000	0	0	con	squ	none	0.89	0.68	0.78
1000	1000	0	0	con	log	none	0.89	0.71	0.80
1000	1000	0	0	log	con	none	0.90	0.73	0.81
1000	1000	0	0	log	lin	none	0.90	0.78	0.84
1000	1000	0	0	log	squ	none	0.87	0.64	0.76
1000	1000	0	0	log	log	none	0.91	0.78	0.84
10	10	0	10	con	con	lin	0.93	0.78	0.86
10	10	0	10	con	lin	lin	0.92	0.78	0.85
10	10	0	10	con	squ	lin	0.92	0.73	0.82
10	10	0	10	con	log	lin	0.92	0.81	0.86
10	10	0	10	log	con	lin	0.93	0.85	0.89
10	10	0	10	log	lin	lin	0.92	0.85	0.88
10	10	0	10	log	squ	lin	0.92	0.82	0.87
10	10	0	10	log	log	lin	0.92	0.86	0.89
100	100	0	10	con	con	lin	0.92	0.79	0.85
100	100	0	10	con	lin	lin	0.90	0.74	0.82
100	100	0	10	con	squ	lin	0.90	0.74	0.82
100	100	0	10	con	log	lin	0.90	0.82	0.86
100	100	0	10	log	con	lin	0.90	0.78	0.84
100	100	0	10	log	lin	lin	0.90	0.77	0.83
100	100	0	10	log	squ	lin	0.89	0.71	0.80
100	100	0	10	log	log	lin	0.90	0.74	0.82

Appendix

Table 2 shows the results of our hyperparameter study. It includes various weighting strategies. The strategies are evaluated with the (a) the convergence ratio, (b) the coverage percentage of the cooling surface and (c) the average of both as reported in the last columns. The following parameters remain constant: $w_{comp,cool} = 0$, $w_{comp,over} = 1$. Relevant violations w.r.t. voids and taboo zones are considered in the convergence ratio. The column *average performance* is the average of the previous two columns. This analysis has been carried out using Product D.

Fig. 16 shows an exemplary dispense path as output by *TIMtrace* when using the Artificial Neural Network (ANN) of Baeuerle et al. as flow behavior model. In this case, we added a small rounding in each corner natively during the optimization routine. The rounding is introduced just before the process parameters are passed to the 2D discretization model. *TIMtrace* allows an easy integration of optional manufacturing constraints such as rounded corners. This trial with the ANN proofs the transferability to a second flow behavior model. This is an important aspect, since different applications apart from TIM might require an adapted flow behavior model. However, this ANN does not support an intermediate void evaluation as shown in Fig. 3. When using

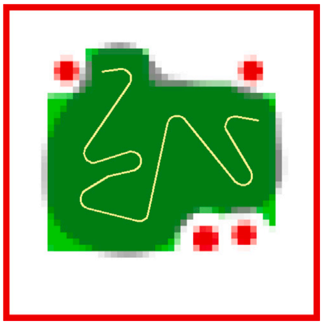


Fig. 16. Resulting area coverage when using the ANN from Baeuerle et al. [2] as underlying flow behavior for *TIMtrace*. This proofs the transferability of this approach to another flow behavior model.

the ANN, a dedicated model to compute the void area would need to be included. Similar to training the ANN on data from the heuristic flow behavior model, a second ANN could be trained to predict the area of intermediate void.

Data availability

Data will be made available on request.

References

[1] A. Gowda, D. Esler, S. Tonapi, K. Nagarkar, K. Srihari, Voids in thermal interface material layers and their effect on thermal performance, in: Proceedings of 6th Electronics Packaging Technology Conference, EPTC 2004, Singapore, 2004, pp. 41–46, http://dx.doi.org/10.1109/EPTC.2004.1396574.

[2] S. Baeuerle, M. Gebhardt, J. Barth, R. Mikut, A. Steimer, Rapid flow behavior modeling of thermal interface materials using deep neural networks, IEEE Access 12 (2024) 17782–17792.

[3] M. Kaufmann, F. Flaig, M. Müller, H. Fricke, T. Vallée, How adhesives flow during joining, Int. J. Adhes. Adhes. 122 (2023) 103315.

[4] P.E. Hart, N.J. Nilsson, B. Raphael, A formal basis for the heuristic determination of minimum cost paths, IEEE Trans. Syst. Sci. Cybern. 4 (2) (1968) 100–107, http://dx.doi.org/10.1109/TSSC.1968.300136.

[5] E.W. Dijkstra, A note on two problems in connexion with graphs, Numer. Math. 1 (1) (1959) 269–271.

[6] E. Galceran, M. Carreras, A survey on coverage path planning for robotics, Robot. Auton. Syst. 61 (12) (2013) 1258–1276, http://dx.doi.org/10.1016/j.robot.2013.09.004, URL https://linkinghub.elsevier.com/retrieve/pii/S092188901300167X.

[7] H. Choset, Coverage for robotics – a survey of recent results, Ann. Math. Artif. Intell. 31 (1–4) (2001) 113–126, URL https://link.springer.com/article/10.1023/A:1016639210559.

[8] C.S. Tan, R. Mohd-Mokhtar, M.R. Arshad, A comprehensive review of coverage path planning in robotics using classical and heuristic algorithms, IEEE Access 9 (2021) 119310–119342.

[9] H. Choset, P. Pignon, Coverage path planning: The boustrophedon cellular decomposition, in: Field and Service Robotics, Springer London, London, 1998, pp. 203–209.

[10] A. Zelinsky, R.A. Jarvis, J.C. Byrne, S. Yuta, Planning paths of complete coverage of an unstructured environment by a mobile robot, in: Proceedings of International Conference on Advanced Robotics, Vol. 13, Tokyo, Japan, 1993, pp. 533–538, URL http://pinkwink.kr/attachment/cfile3.uf@1354654A4E8945BD13FE77.pdf.

[11] Y. Gabriely, E. Rimon, Spanning-tree based coverage of continuous areas by a mobile robot, Ann. Math. Artif. Intell. 31 (1) (2001) 77–98.

[12] Y. Gabriely, E. Rimon, Spiral-STC: An on-line coverage algorithm of grid environments by a mobile robot, in: Proceedings of the 2002 IEEE International Conference on Robotics and Automation, ICRA, Vol. 1, Washington, DC, USA, 2002, pp. 954–960, http://dx.doi.org/10.1109/ROBOT.2002.1013479, vol. 1.

[13] C. Luo, S.X. Yang, D.A. Stacey, J.C. Jofriet, A solution to vicinity problem of obstacles in complete coverage path planning, in: Proceedings 2002 IEEE International Conference on Robotics and Automation, ICRA, Vol. 1, Washington, DC, USA, 2002, pp. 612–617, URL https://ieeexplore.ieee.org/abstract/document/1013426/.

[14] S. Yang, C. Luo, A neural network approach to complete coverage path planning, IEEE Trans. Syst. Man Cybern. B 34 (1) (2004) 718–724, http://dx.doi.org/10.1109/TSMCB.2003.811769, URL http://ieeexplore.ieee.org/document/1262545.

[15] C. Luo, S.X. Yang, A bioinspired neural network for real-time concurrent map building and complete coverage robot navigation in unknown environments, IEEE Trans. Neural Netw. 19 (7) (2008) 1279–1298, http://dx.doi.org/10.1109/TNN.2008.2000394, URL http://ieeexplore.ieee.org/document/4539807.

[16] D.E. Soltero, M. Schwager, D. Rus, Generating informative paths for persistent sensing in unknown environments, in: Proceedings of the 2012 IEEE International Conference on Intelligent Robots and Systems, Vilamoura, Algarve, Portugal, 2012, pp. 2172–2179, URL https://ieeexplore.ieee.org/abstract/document/6385730/.

[17] D.E. Soltero, M. Schwager, D. Rus, Decentralized path planning for coverage tasks using gradient descent adaptive control, Int. J. Robot. Res. 33 (3) (2014) 401–425, http://dx.doi.org/10.1177/0278364913497241, URL http://journals.sagepub.com/doi/10.1177/0278364913497241.

[18] J.C. Kiemel, P. Yang, P. Meißner, T. Kröger, Paintrl: Coverage path planning for industrial spray painting with reinforcement learning, in: Workshop on Closing the Reality Gap in Sim2Real Transfer for Robotic Manipulation, Freiburg, Germany, 2019.

[19] C. Piciarelli, G.L. Foresti, Drone patrolling with reinforcement learning, in: Proceedings of the 13th International Conference on Distributed Smart Cameras, ICDSC 2019, Association for Computing Machinery, Trento, Italy, 2019, http://dx.doi.org/10.1145/3349801.3349805.

[20] M. Theile, H. Bayerlein, R. Nai, D. Gesbert, M. Caccamo, Coverage Path Planning under varying power constraints using deep Reinforcement Learning, Tech. Rep., 2020, URL https://arxiv.org/abs/2003.02609. arXiv:2003.02609.

[21] K. Ellefsen, H. Lepikson, J. Albiez, Multiobjective coverage path planning: Enabling automated inspection of complex, real-world structures, Appl. Soft Comput. 61 (2017) 264–282, http://dx.doi.org/10.1016/j.asoc.2017.07.051.

[22] V.R. Batista, F.A. Zampirolli, Optimising robotic pool-cleaning with a genetic algorithm, J. Intell. Robot. Syst. 95 (2) (2019) 443–458.

[23] M.A. Yakoubi, M.T. Laskri, The path planning of cleaner robot for coverage region using genetic algorithms, J. Innov. Digit. Ecosyst. 3 (1) (2016) 37–43, http://dx.doi.org/10.1016/j.jides.2016.05.004, URL https://linkinghub.elsevier.com/retrieve/pii/S2352664516300050.

[24] M. Popović, G. Hitz, J. Nieto, I. Sa, R. Siegwart, E. Galceran, Online informative path planning for active classification using UAVs, in: 2017 IEEE International Conference on Robotics and Automation, ICRA, Marina Bay Sands, Singapore, 2017.

[25] N. Hansen, The CMA] Evolution Strategy: A Tutorial, Tech. Rep., 2016, URL http://arxiv.org/abs/1604.00772. arXiv:1604.00772.

[26] D. Strubel, Coverage Path Planning Based on Waypoint Optimization, with Evolutionary Algorithms (Ph.D. thesis), Université Bourgogne Franche-Comté, Petronas, France, 2019.

[27] P. Kulkarni, A. Marsan, D. Dutta, A review of process planning techniques in layered manufacturing, Rapid Prototyp. J. 6 (1) (2000) 18–35, http://dx.doi.org/10.1108/13552540010309859.

[28] W. Oropallo, L.A. Piegl, Ten challenges in 3D printing, Eng. Comput. 32 (1) (2016) 135–148, http://dx.doi.org/10.1007/s00366-015-0407-0, URL http://link.springer.com/10.1007/s00366-015-0407-0.

[29] G. Jin, W. Li, C. Tsai, L. Wang, Adaptive tool-path generation of rapid prototyping for complex product models, J. Manuf. Syst. 30 (3) (2011) 154–164, http://dx.doi.org/10.1016/j.jmsy.2011.05.007, URL https://linkinghub.elsevier.com/retrieve/pii/S0278612511000562.

[30] L. Roveda, B. Maggioni, E. Marescotti, A. Shahid, A.M. Zanchettin, A. Bemporad, D. Piga, Pairwise preferences-based optimization of a path-based velocity planner in robotic sealing tasks, IEEE Robot. Autom. Lett. (2021) http://dx.doi.org/10.1109/LRA.2021.3094479, URL https://ieeexplore.ieee.org/document/9473003/, pp. 1–1.

[31] R. Comminal, M.P. Serdeczny, D.B. Pedersen, J. Spangenberg, Motion planning and numerical simulation of material deposition at corners in extrusion additive manufacturing, Addit. Manuf. 29 (2019) 100753.

[32] H. Blum, A transformation for extracting new descriptors of shape, in: Proceedings of the Symposium on Models for the Perception of Speech and Visual Form, Vol. 43, MIT Press, Cambridge, MA, USA, 1967, pp. 362–380.

[33] J.-H. Kao, F.B. Prinz, Optimal motion planning for deposition in layered manufacturing, in: Proceedings of Design Engineering Technical Conferences 1998, Vol. 80364, American Society of Mechanical Engineers, Atlanta, Georgia, USA, 1998, pp. 13–16.

[34] D. Ding, Z. Pan, D. Cuiuri, H. Li, N. Larkin, Adaptive path planning for wire-feed additive manufacturing using medial axis transformation, J. Clean. Prod. 133 (2016) 942–952, http://dx.doi.org/10.1016/j.jclepro.2016.06.036, URL https://www.sciencedirect.com/science/article/pii/S0959652616307119.

[35] J. Wang, T.-w. Chen, Y.-a. Jin, Y. He, Variable bead width of material extrusion-based additive manufacturing, J. Zhejiang Univ.- Sci. A 20 (1) (2019) 73–82, http://dx.doi.org/10.1631/jzus.A1700236, URL http://link.springer.com/10.1631/jzus.A1700236.

[36] Y. Xiong, S.-I. Park, S. Padmanathan, A.G. Dharmawan, S. Foong, D.W. Rosen, G.S. Soh, Process planning for adaptive contour parallel toolpath in additive manufacturing with variable bead width, Int. J. Adv. Manuf. Technol. 105 (10) (2019) 4159–4170, http://dx.doi.org/10.1007/s00170-019-03954-1.

[37] S. Hornus, T. Kuipers, O. Devillers, M. Teillaud, J. Martínez, M. Glisse, S. Lazard, S. Lefebvre, Variable-width contouring for additive manufacturing, ACM Trans. Graph. 39 (4) (2020) http://dx.doi.org/10.1145/3386569.3392448.

[38] D. Ding, C. Shen, Z. Pan, D. Cuiuri, H. Li, N. Larkin, S. van Duin, Towards an automated robotic arc-welding-based additive manufacturing system from CAD to finished part, Comput.- Aided Des. 73 (2016) 66–75, http://dx.doi.org/10.1016/j.cad.2015.12.003, URL https://www.sciencedirect.com/science/article/pii/S0010448515001748.

[39] F. Flaig, T. Fräger, M. Kaufmann, T. Vallée, H. Fricke, M. Müller, How to find the perfect application pattern for adhesively bonded joints? J. Adv. Join. Process. 8 (2023) 100147, http://dx.doi.org/10.1016/j.jajp.2023.100147, URL https://www.sciencedirect.com/science/article/pii/S2666330923000092.

[40] F. Flaig, T. Fräger, M. Kaufmann, T. Vallée, H. Fricke, M. Müller, A practical strategy to identify appropriate application patterns for adhesively bonded joints, Proc. Appl. Math. Mech. 23 (3) (2023) e202300080, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/pamm.202300080.

[41] M. Kaufmann, F. Flaig, M. Müller, H. Fricke, T. Vallée, Optimized adhesive application, Int. J. Adhes. Adhes. 130 (2024) 103620, http://dx.doi.org/10.1016/j.ijadhadh.2024.103620.

[42] A.E. Bryson, Applied Optimal Control: Optimization, Estimation and Control, Vol. 2, Routledge, 1975.

[43] S. Baeuerle, M. Gebhardt, J. Barth, A. Steimer, R. Mikut, Rapid Flow Behavior Modeling of Thermal Interface Materials using Deep Neural Networks, Tech. rep., 2022, URL https://arxiv.org/abs/2208.04045, arXiv:2208.04045.

[44] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyma, Optuna: A Next-Generation Hyperparameter Optimization Framework, Tech. rep., 2019, URL https://arxiv.org/abs/1907.10902, arXiv:1907.10902.

[45] N. Hansen, A. Auger, R. Ros, S. Finck, P. Pošík, Comparing results of 31 algorithms from the black-box optimization benchmarking BBOB-2009, in: Proceedings of the 12th Annual Conference Companion on Genetic and Evolutionary Computation, GECCO '10, Association for Computing Machinery, New York, NY, USA, 2010, pp. 1689–1696.

[46] G. Bradski, The OpenCV library, Dr. Dobb's Journal of Software Tools, 2000.