Optimization and Engineering (2021) 22:913–974
https://doi.org/10.1007/s11081-020-09552-5

RESEARCH ARTICLE

Stable honeycomb structures and temperature based trajectory optimization for wire-arc additive manufacturing

Martin Bähr[1] · Johannes Buhl[1] · Georg Radow[1] · Johannes Schmidt[1] · Markus Bambach[1] · Michael Breuß[1] · Armin Fügenschuh[1]

Received: 20 December 2019 / Revised: 3 April 2020 / Accepted: 13 August 2020 /
Published online: 14 September 2020

Abstract

We consider two mathematical problems that are connected and occur in the layer-wise production process of a workpiece using wire-arc additive manufacturing. As the first task, we consider the automatic construction of a honeycomb structure, given the boundary of a shape of interest. In doing this, we employ Lloyd's algorithm in two different realizations. For computing the incorporated Voronoi tesselation we consider the use of a Delaunay triangulation or alternatively, the eikonal equation. We compare and modify these approaches with the aim of combining their respective advantages. Then in the second task, to find an optimal tool path guaranteeing minimal production time and high quality of the workpiece, a mixed-integer linear programming problem is derived. The model takes thermal conduction and radiation during the process into account and aims to minimize temperature gradients inside the material. Its solvability for standard mixed-integer solvers is demonstrated on several test-instances. The results are compared with manufactured workpieces.

Keywords Mixed-integer linear programming · Heat transmission · Additive manufacturing · Centroidal Voronoi tesselation · Geometric optimization · Eikonal equation

1 Introduction

Additive manufacturing (AM) processes evolved in the past decades into a notable alternative to classical material-removing production techniques. Especially in the aircraft industry their potential of building components on demand, besides the possibility of reducing weight and material loss, is appreciated (Allen 2006).

✉ Johannes Schmidt
johannes.schmidt@b-tu.de

Extended author information available on the last page of the article

We consider the AM process called wire-arc additive manufacturing (WAAM). It uses the conventional welding technology to print parts with direct energy deposition. A welding torch fed by wire moves over the workpiece. The wire is molten by an electrical arc using high temperatures, and then the material is deposited in droplets in the proposed area. In this way, the workpiece can be built layer-wise or even subpart-by-subpart, if the geometry is more complex (Nguyen et al. 2018). Although it is desired to weld in a continuous path, most structures can not be handled in this way. So eventually movements of the welding torch without welding, called transition moves, are necessary. Since this may lead to more abrasion and reduced quality, it should be avoided if possible.

Given the shape of a workpiece, the problems of (i) finding a good inner structure in terms of functionality and stability and (ii) the best path for printing the desired layer arise. These two consecutive subproblems of structure and path optimization are the ones that we tackle.

Nowadays, the manufacturing process of a workpiece is closely linked to its corresponding weight and material cost. The minimal weight of the manufactured product is preferable for more flexibility and functionality in further work processes, without neglecting its strength or stability properties. Moreover, the production cost of the produced workpiece is directly related to the volume of material employed. To this end, typically an economic minimization is performed for achieving minimal material cost under the observance of certain structural stabilities. The latter minimization concept is strongly connected to honeycomb structures which are known to be of minimal material cost and weight (Hales 2001; Lyon and Colyvan 2008) while providing high strength. Additionally, often workpieces have to absorb impacts and tackle additional external force constraints, i.e., a method for structural optimization should allow for the consideration of the related, additional constraints.

For this reason, in a first step, we consider the automatic construction of a honeycomb structure through centroidal Voronoi tesselations (CVTs), given the boundary shape of a structure. At this point, a CVT is a special type of a Voronoi tesselation (VT) that converges towards a hexagonal honeycomb-like structure when increasing the number of Voronoi cells within the technique, cf. Bronstein et al. (2009). In doing this, modeling of external force constraints for fulfilling certain structural stabilities can be done by incorporating a user-defined density function.

In this context, we employ Lloyd's algorithm for constructing a CVT in two different realizations. For computing the incorporated VT as well as for the precise computation of the center of mass, which are both crucial points when implementing this method, we investigate either the use of an exact or an approximative method. In this work, we consider a geometric optimization method based on a Delaunay triangulation (DT) as well as the eikonal equation which is a hyperbolic partial differential equation (PDE). While finding a DT as the dual graph of the Voronoi diagram is based on geometric arguments, the eikonal based approach makes use of a discretization of the corresponding PDE. Thereby, the geometric approach constructs an exact VT, whereby the PDE-based approach provides an approximated version.

Beyond that, let us mention that within the construction process of a CVT there exist some design parameters which are of practical relevance. For instance, the number of generators used for constructing the VT is an important aspect in practice, since it yields a way to take into account for instance total weight of a planned structure in the design process. Another design parameter is the mentioned user-defined density function. This function can be chosen such that the stability of the final structure is increased in those regions, where more (mechanical) stress is anticipated for the workpiece.

Let us turn to the second step of our construction. The deposition of the hot metal drops on the substrate causes a rise in the temperature of the substrate plate. The thermal expansion of the substrate plate in the heat-affected region causes changes in residual stresses within itself and solidified beads (Leuders et al. 2013) and yields distortions in the part and substrate (Hollander et al. 2006). In the past, different authors, e.g. Zhang et al. (2018), Rodrigues et al. (2019), and Israr et al. (2018), described the development of the residual stresses and distortions depending on the sequence, initial temperature and the method of clamping.

The resulting strain can reduce the quality of a workpiece significantly, making time-consuming post-processing steps necessary or the whole product may become unusable. Therefore, the chosen path of the welding source is crucial for process efficiency.

To support the decision of finding the best welding path, we consider the computation of an optimal welding path in terms of the temperature distribution within the workpiece. For providing a feasible path we specify a binary linear model, that describes a connected sequence of welding and transition moves. The number of transition moves without welding is thereby limited to its minimum to achieve a rapid process.

Furthermore, we derive discrete approximations for heat conduction using finite differences, and for radiation. The latter is described by two different approaches, the one using a piecewise linear approximation of the power function, the other one dealing with a constant factor and an additive constant. We also set up a rough approximation of the substrate plate the workpiece is built on, to include its ability for heat transmission into our calculation.

By combining the path generation and one of the approaches for temperature calculation into a single model we obtain mixed-integer linear problems (MILPs), that are investigated on several test instances. Besides, we compare their results in a qualitative way to the temperature distribution of real processed workpieces, built using the path computed by the models.

In the literature, the main aspect about path optimization in WAAM is the contour accuracy between the manufactured workpiece and its pattern, since the contour of the processed layers should be near the final shape of the workpiece to avoid post-processing work. But in most cases the temperature aspects are neglected.

Ding et al. (2014, 2015) presented different algorithms for tool path optimization. One procedure divides the considered layer up into convex polygons, whose tool path is calculated and then connected to a continuous path of the welding source. Another one is based on Medial-Axis-Transformation to compute a tool path suitable for geometries with thin walls and areas.

Michel et al. (2019) took a similar approach to Ding et al. (2014) by segmentation of the considered layer into parts with easy geometry, that allows contour accurate processing. Then only the welding sequence of these parts has to be specified.

Venturini et al. (2016) did an extensive study about the optimal tool path for T-shaped crossings over several layers, including experiments with several different welding strategies.

Montevecchi et al. (2018) gave a short review about the influence of temperature for the quality of the workpiece and discussed the computation of the optimal idle time between two layers using a finite element approach. But here the initial temperature distribution is given and no travelled tool path is considered.

This paper summarizes and extends our work presented in the conference papers (Bähr et al. 2019; Fügenschuh et al. 2019). Compared to the notes on structure optimization in Bähr et al. (2019), we give here a much more detailed discussion of the involved concepts and techniques. Additionally, we propose a novel approach for computing CVTs, aiming to combine advantages of the approaches discussed previously. This is accompanied by a detailed comparison of the presented approaches, on the basis of theoretical observations as well as numerical experiments. By contrast with Fügenschuh et al. (2019) we drop the assumption of isolated nodes and extend the presented mixed-integer linear model by heat transmission within the workpiece. Furthermore, we rework the existing constraints for path generation to reduce their number along with simplifying the computation and compare the obtained models with real processed workpieces based on the optimum computed before.

The three main parts of this paper are (i) given the outer shape, find an optimal internal structure, (ii) given the structure of a workpiece, compute an optimal printing process, and (iii) combine and test the devised methodology at hand of synthetic and real workpieces. Thus it is structured as follows. In Sect. 2 we discuss the first task in our application, which is to find optimal structures to be printed, given the boundary shape of the planned workpiece. We present and compare multiple ways to obtain such optimal structures. The section is concluded by some numerical experiments, focusing on the discussed first main task. Then, in Sect. 3, we discuss the second main task consisting of finding an optimal trajectory of the welding torch during the printing process in WAAM. We derive a family of MILPs, using different approximation methods, to describe the welding of a single layer including heat transmission. The derived models are tested on several instances to confirm the applicability to a state-of-the-art MILP solver. In Sect. 4 we demonstrate the full pipeline by means of an example. Additionally, we discuss a strategy to make the output of the structure optimization more suitable as an input for path optimization and compare our numerical results to real processed workpieces. Before concluding this paper, we also discuss a number of directions for future work.

2 Voronoi tesselations

In this section, we discuss the first subproblem in our targeted application of WAAM, which is to find an—in some sense—optimal inner structure given the boundary shape of a planned workpiece. For this task, we introduce the basic mathematical concept of VTs, especially CVTs. In this context, we present possible implementations based on geometric and PDE-based approaches. Due to the fact that the geometric version delivers an exact VT and the PDE-based approach only an approximation, we examine more closely the corresponding effects on constructing the CVT. In doing this, we will discuss the mentioned approaches in more detail on the basis of numerical investigations including a special choice of density functions.

2.1 Problem formulation

The region of interest is a simply connected compact set $\Theta \subset \mathbb{R}^2$, which represents one layer of the planned workpiece in the printing process. Especially, for cost reduction, we aim to find a subset of Θ that is to be printed. This subset should include the boundary $\partial\Theta$ so that the exterior shape of the printed workpiece stays the same. A trivial solution might be to only print the boundary $\partial\Theta$, but the resulting workpiece may not be usable for the intended purpose due to mechanical stability issues.

Instead of the trivial solution, we aim to find a planar graph where the edges represent the printed segments. This can be realized through a VT consisting of a set of generators $X = \{\mathbf{x}_1, \dots, \mathbf{x}_{N^g}\}$ with $\mathbf{x}_i \in \Theta$ and associated Voronoi cells $\{A_i\}_{i=1}^{N^g}$ defined as

$$A_i = \left\{\mathbf{x} \in \Theta : d(\mathbf{x}, \mathbf{x}_i) < d(\mathbf{x}, \mathbf{x}_j),\ j \in \{1, \dots, N^g\} \backslash \{i\}\right\} \subset \Theta, \tag{1}$$

where $d(\cdot, \cdot)$ denotes some distance function. Let us remark that the concept of VTs is also valid in higher dimensions.

One can see that $A_i \cap A_j = \emptyset$ for all $i \neq j$ and $\bigcup_{i=1}^{N^g} \overline{A}_i = \Theta$ with $\overline{A}_i$ denoting the closure of A_i. The printed segments then consist of $\bigcup_{i=1}^{N^g} \partial A_i$, i.e. the borders between any two Voronoi cells, and the boundary $\partial\Theta$. Since the Voronoi cells depend on the generators X, we also write $\mathscr{A}(X)$ for the set of all Voronoi cells and with $A(\mathbf{x}_i) \in \mathscr{A}(X)$ we may also refer to the cell generated by $\mathbf{x}_i$.

If in each cell the generator coincides with the center of mass, a VT is called a CVT. To compute the center of mass, a density function or stress map $\rho : \Theta \to [0, \infty)$ is introduced. This stress map may be chosen to enhance mechanical stability in certain regions (Lu et al. 2014). Basically, in regions with higher density more Voronoi cells will be accumulated in a CVT. However, since we consider a fixed amount of generators N^g, global scaling of the density through a constant multiplier will have no impact on the structure of a CVT.

Formally the generators X of a CVT can be characterized through the minimization of an energy functional (Du et al. 1999):

$$\mathscr{E}(X) = \sum_{i=1}^{N^g} \int\limits_{A(\mathbf{x}_i)} \rho(\mathbf{x}) \|\mathbf{x} - \mathbf{x}_i\|^2 \mathrm{d}\mathbf{x}, \tag{2}$$

where $\|.\|$ denotes the Euclidean norm. One of the basic methods for finding generators X of a CVT is Lloyd's algorithm (Lloyd 1982), which is a fixed point iteration consisting of alternatingly computing the Voronoi cells $\mathscr{A}(X)$ and replacing the generator $\mathbf{x}_i$ with the center of mass in A_i, i.e.

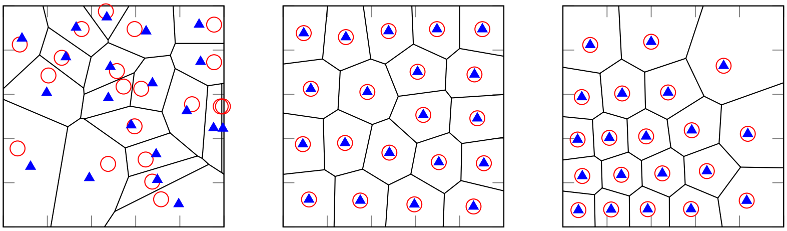


Fig. 1 Computed VTs for $\Theta = [0,1]^2$, $N^g = 20$ generators (red circles) and centroids (blue triangles). *(left)* Randomly distributed generators, *(middle)* CVT with constant density $\rho \equiv 1$, *(right)* CVT where the density is a Gaussian centered at the origin. (Color figure online)

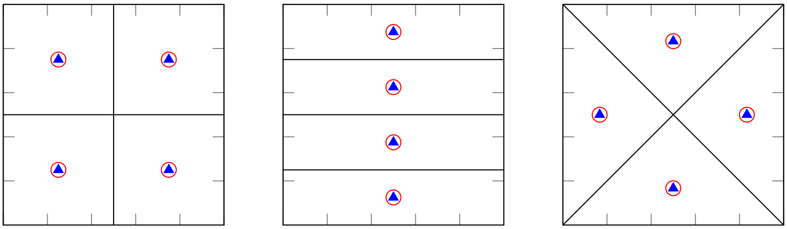


Fig. 2 Multiple CVTs for $\Theta = [0,1]^2$, $\rho \equiv 1$ and $N^g = 4$. All tesselations are fixed points of Lloyd's algorithm, the energies according to (2) are from left to right: 0.0417, 0.0885, 0.0556. (Color figure online)

$$\mathbf{x}_i \leftarrow \frac{\int_{A(\mathbf{x}_i)} \mathbf{x}\rho(\mathbf{x})\mathrm{d}\mathbf{x}}{\int_{A(\mathbf{x}_i)} \rho(\mathbf{x})\mathrm{d}\mathbf{x}}, \qquad \forall i \in \{1, \ldots, N^g\}. \tag{3}$$

Other methods for computing CVTs can be found e.g. in Du and Emelianenko (2006), Liu et al. (2009), Hateley et al. (2015) and the references therein.

Increasing the number of generators leads to the hexagonal honeycomb form of the Voronoi cells, cf. Bronstein et al. (2009), which is a structure of high strength-to-weight ratio. The differences between VT as well as CVT for different ρ are shown in Fig. 1.

Let us remark that, given Θ, ρ and N^g, the problem of finding a CVT may have multiple solutions. Simple examples can be seen in Fig. 2. Especially if there are symmetries within Θ and ρ, the result of Lloyd's algorithm depends significantly on the initialization of the generators.

2.2 Computing Voronoi tesselations

In the following paragraphs, we compare two methods for finding VTs, namely the geometric or graph based approach utilizing a DT, and an approach based on PDEs utilizing the fast marching (FM) method. After a comparison based on theoretical properties, we present a modification of the geometric approach, aiming to integrate some advantages of the PDE-based approach. To our knowledge, this modified approach has not been discussed in the literature up to now.

2.2.1 Geometric approach

Voronoi tesselations can be constructed by computing a DT of the generators. This can be regarded as the typical approach for finding VTs (Tournois et al. 2010; Hateley et al. 2015).

Given points in $\mathbb{R}^2$, a DT is a triangulation, such that no point lies within the circumcircle of any triangle (Preparata and Shamos 1985). Indeed, if one finds a DT for the generators X, then the circumcenters of the triangles coincide with the nodes of the graph that consists of the boundary lines between the Voronoi cells. With this approach, special care must be taken for finding the nodes on the boundary of $\partial\Theta$.

There are multiple resources available for finding VTs. One of these is the Computational Geometry Algorithms Library (CGAL) (Fabri et al. 2000). Also in MATLAB one can compute VTs with the command `voronoin` based on the Qhull algorithm (Barber et al. 1996).

In our numerical experiments, we use MATLAB. To get the boundary points on $\partial\Theta$ with the geometric approach, we mirror those generators on $\partial\Theta$ which are 'close enough' to this boundary. If Θ is a polygon, this means to find those generators $\mathbf{x}_i$ which are the closest generators to some part of the boundary $\partial\Theta$. Then we mirror them on the corresponding line segment since some part of the line segment will be a part of $A(\mathbf{x}_i)$.

For the realization of Lloyd's algorithm for computing CVTs with the discussed geometric approach, the centers of mass are computed in an approximate manner via a geometric decomposition, assuming that Θ is a polygon. In each Voronoi cell $A_i = A(\mathbf{x}_i)$ we consider each triangle Δ_{ij} between $\mathbf{x}_i$ and some boundary edge $e_j \in \partial A_i$. Here we understand the cell boundary ∂A_i as a set of edges between the vertices of the cell. Then, assuming a constant density in Δ_{ij}, we compute the corresponding center of mass $\mathbf{x}_i^{(j)}$ and sample the density $\rho(\mathbf{x}_i^{(j)})$. Now the second step (3) in Lloyd's algorithm is approximated through

$$\mathbf{x}_i \leftarrow \frac{\sum_{j,\ e_j\in\partial A_i} \mathbf{x}_i^{(j)} \rho\left(\mathbf{x}_i^{(j)}\right) F_{ij}}{\sum_{j,\ e_j\in\partial A_i} \rho\left(\mathbf{x}_i^{(j)}\right) F_{ij}}, \qquad \forall i \in \{1, \dots, N^g\}, \tag{4}$$

where F_{ij} denotes the surface area of Δ_{ij}.

2.2.2 PDE-based approach

The computation of a VT is connected to the geodesic distance. In differential geometry, a geodesic is a length-minimizing curve between two connecting points on a surface and relies on the intrinsic geometry, cf. Bronstein et al. (2009). In other words, the geodesic is a generalization of a straight line on a plane surface. It is well-known, that for a convex planar surface Θ the geodesic distance d, defined as the shortest path contained in Θ between two points, is equivalent to the Euclidean distance. For non-convex planar surfaces the mentioned distances are no longer identical. In particular, the geodesic distance is a curve within the surface, whereby the Euclidean distance is a straight line and may not lie completely within Θ. This means that in the non-convex case the Euclidean metric can be replaced by the geodesic metric. Due to this fact, any method computing geodesic distances can also be used to generate a VT.

To compute the discrete geodesic two broad classes of methods exist. The first class computes the discrete geodesic exactly on triangle meshes, two common methods are the Mitchell–Mount–Papadimitriou (MMP) algorithm (Mitchell et al. 1987) and the Chen–Han (CH) algorithm (Chen and Han 1990), however both methods have many variants. The MMP and CH algorithms have in general a worst-case time complexity of $\mathcal{O}(N^2 \log N)$ and $\mathcal{O}(N^2)$, respectively, but are in practice often faster. Some recent and more efficient methods for computing exact geodesics are developed by Chunxu et al. (2015) and Wang et al. (2015) in the context of CVTs. On the other hand, discrete geodesics can be computed by solving PDEs via numerical methods on a mesh. In contrast to the exact methods which may be computationally expensive, the PDE-based methods are very easy to implement and very efficient. However, they provide only an approximation of the geodesic distance. There also exist efficient methods for computing approximations of geodesic distances not based on PDEs, see e.g. Ying et al. (2013). Let us remark that they are often much more complex to implement. In this work, we are more interested in efficient VT computation coupled with a straightforward implementation and neglect the first class of exact discrete geodesic solvers.

Common methods for computing an approximative geodesic distance by using PDEs rely on the eikonal equation (Sethian 1996), the heat method (Crane et al. 2013) or a variational interpretation (Belyaev and Fayolle 2015) built upon the heat method. A recent related work (Zayer et al. 2018) is based on a growth model such that the tesselation arises as the solution of a set of time-dependent PDEs that describe concurrently evolving fronts. In this process, the computational costs depend only on the addition as well as the multiplication of two matrices. Due to this fact, the method is extremely efficient, but it is not connected theoretically to a geodesic metric.

The central idea of the heat method introduced by Crane et al. (2013) is based on the fact that the normalized gradient of a special heat flow coincides with the gradient of the geodesic distance function. In doing this, the heat method algorithm consists of three basic steps. At first the heat flow initiated by a Dirac delta heat distribution is computed by solving a sparse linear system. Subsequently, the normalized gradient field of the solution is evaluated such that, finally, finding the closest scalar potential by energy minimization is equivalent to solving a Poisson equation. By using a Cholesky factorization of the Laplacian matrix, both the heat and the Poisson equation can easily be computed with sub-quadratic time complexity. However, let us mention that the accuracy of geodesic distances obtained with the heat method relies to some degree on the temporal step size when solving the diffusion equation. Moreover, the method has to be adapted for bounded domains by using the average heat field calculated from two different boundary conditions, more precisely averaging the solution for Neumann and Dirichlet boundary conditions. Inspired by the heat method, in the work of Herholz et al. (2017b) the VT is constructed more efficiently by using the heat diffusion directly as a "pseudo" distance. To speed up the approach, Herholz et al. (2017a, b) also propose a localized version via an appropriate reordering of the Cholesky factorization. However, for bounded domains under consideration of Neumann and Dirichlet boundary conditions, changes in the factorization are required, increasing the implementation effort and the computational costs.

Alternatively to the heat method, the geodesic distance can also be approximated by solving the eikonal equation, which is a time-independent hyperbolic PDE and describes an expanding wave propagation. Eikonal-based VTs are applied in several works, see Sethian (1999) and Peyré and Cohen (2003). For PDEs of eikonal-type, a solution can be computed efficiently by the widely used FM (Sethian 1996) or fast sweeping (FS) (Zhao 2005) methods, for a comprehensive overview see Gómez et al. (2019). In the past, both methods were generalized (Kimmel and Sethian 1998; Xu et al. 2010), e.g. to triangle meshes as well as to manifolds, whereby FM requires a non-obtuse triangulation to avoid violation of causality of the method. The benefits of FM and FS are their relatively low complexity of $\mathcal{O}(N \log N)$ and $\mathcal{O}(N)$, respectively. However, in the case of geodesic distance computations (with a constant speed function) the FS method is much faster, cf. Xu et al. (2010) and Gómez et al. (2019). Nevertheless, the implementation of FS is much more cumbersome when considering non-rectangular domains or higher-order approximations of the derivatives within the numerical scheme. In this context, FM is very simple and flexible. Let us note that its computational efficiency is linked to the implementation of a heap-based priority queue. Based on these arguments, we will make use of the FM method in this work.

Let us also note, that recently Peter et al. (2014) have derived a relationship between the eikonal equation and the time-independent Schrödinger equation. On this basis, an inhomogeneous, screened Poisson equation arises, which may serve as an alternative efficient approach to VT computation.

Let us now focus on how to compute VTs with the eikonal equation. As mentioned above, a common approach for approximating geodesic distances is by solving the non-linear eikonal equation

$$\|\nabla \widetilde{d}(\mathbf{x})\| = 1, \qquad \mathbf{x} \in \Theta \backslash \Theta_0 \tag{5}$$

with the boundary condition

$$\widetilde{d}(\mathbf{x}) = 0, \qquad \mathbf{x} \in \Theta_0, \tag{6}$$

where Θ_0 is a subset of Θ, see also Bronstein et al. (2009). We use the notation $\widetilde{d}(\mathbf{x}) = d(\mathbf{x}, \Theta_0)$ for the minimal geodesic distance from $\mathbf{x}$ to any point in Θ_0. The underlying PDE represents the shortest arrival time of a wavefront from the initial point $\mathbf{s}$ to every point $\mathbf{x}$ in the computational domain, whereby the wavefront moves in its normal direction with constant unit speed. A solution of (5)–(6) can be computed efficiently by the FM method proposed by Sethian (1996).

We will now elaborate on the usual discretization for the eikonal equation. For simplicity we consider a rectangular domain with equidistant mesh size

$h = \Delta x = \Delta y$ in x- and y-direction, where $\widetilde{d}_{i,j}$ denotes an approximation of the unknown function $\widetilde{d}$ at grid point (x_i, y_j). Equation (5) can be transformed into

$$\widetilde{d}_x^2 + \widetilde{d}_y^2 = 1 \tag{7}$$

with $\nabla \widetilde{d} = (\frac{\partial \widetilde{d}}{\partial x}, \frac{\partial \widetilde{d}}{\partial y})^\top = (\widetilde{d}_x, \widetilde{d}_y)^\top$. Approximating the partial derivatives $\widetilde{d}_x$ and $\widetilde{d}_y$ in (7) with first-order forward differences

$$\widetilde{d}_x(x_i, y_j) \approx \frac{\widetilde{d}_{i+1,j} - \widetilde{d}_{i,j}}{h}, \qquad \widetilde{d}_y(x_i, y_j) \approx \frac{\widetilde{d}_{i,j+1} - \widetilde{d}_{i,j}}{h} \tag{8}$$

and backward differences

$$\widetilde{d}_x(x_i, y_j) \approx \frac{\widetilde{d}_{i,j} - \widetilde{d}_{i-1,j}}{h}, \qquad \widetilde{d}_y(x_i, y_j) \approx \frac{\widetilde{d}_{i,j} - \widetilde{d}_{i,j-1}}{h} \tag{9}$$

combined with the upwind-scheme proposed by Godunov and Bohachevsky (1959) leads to

$$\max\left\{\frac{\widetilde{d}_{i,j} - \widetilde{d}_{i-1,j}}{h}, \frac{\widetilde{d}_{i,j} - \widetilde{d}_{i+1,j}}{h}, 0\right\}^2 + \max\left\{\frac{\widetilde{d}_{i,j} - \widetilde{d}_{i,j-1}}{h}, \frac{\widetilde{d}_{i,j} - \widetilde{d}_{i,j+1}}{h}, 0\right\}^2 = 1. \tag{10}$$

The latter equation can be rewritten as

$$\max\left\{\frac{\widetilde{d}_{i,j} - \min\left\{\widetilde{d}_{i-1,j}, \widetilde{d}_{i+1,j}\right\}}{h}, 0\right\}^2 + \max\left\{\frac{\widetilde{d}_{i,j} - \min\left\{\widetilde{d}_{i,j-1}, \widetilde{d}_{i,j+1}\right\}}{h}, 0\right\}^2 = 1 \tag{11}$$

and by setting

$$\widetilde{d} = \widetilde{d}_{i,j}, \qquad \widetilde{d}_1 = \min\left\{\widetilde{d}_{i-1,j}, \widetilde{d}_{i+1,j}\right\}, \qquad \widetilde{d}_2 = \min\left\{\widetilde{d}_{i,j-1}, \widetilde{d}_{i,j+1}\right\} \tag{12}$$

we obtain

$$\max\left\{\frac{\widetilde{d} - \widetilde{d}_1}{h}, 0\right\}^2 + \max\left\{\frac{\widetilde{d} - \widetilde{d}_2}{h}, 0\right\}^2 = 1, \tag{13}$$

which is the upwind discretized version of the eikonal equation. Concretely the upwind nature of the discretization means that the derivative detects the direction along wave information flows and selects the derivative with respect to the smallest neighbor values.

During FM, the quadratic equation (13) is solved on every grid point in $\Theta \backslash \Theta_0$, therefore the following two cases arise:

(i) For $\widetilde{d} > \max\left\{\widetilde{d}_1, \widetilde{d}_2\right\}$ we get

$$\widetilde{d} = \frac{\widetilde{d}_1 + \widetilde{d}_2 + \sqrt{2h^2 - (\widetilde{d}_1 - \widetilde{d}_2)^2}}{2}. \tag{14}$$

(ii) For $\widetilde{d}_2 \geq \widetilde{d} > \widetilde{d}_1$ the derivative in y-direction is zero and we have

$$\widetilde{d} = \widetilde{d}_1 + h \tag{15}$$

and the case $\widetilde{d}_1 \geq \widetilde{d} > \widetilde{d}_2$ is handled analogously.

This basic numerical approach can be modified into more sophisticated semi-Lagrangian (Cristiani and Falcone 2007) or multistencil discretization (Hassouna and Farag 2007). It can also be extended to non-uniform grids and triangulated meshes (Kimmel and Sethian 1998) or to higher-order upwind discretization (Sethian 1999). In the following, we describe the FM algorithm which solves the discretized eikonal equation pointwise by using a specific causality relationship, where the information of arrival times is propagated downwind.

The principle behind FM is that information advances monotonically from smaller values of $\widetilde{d}$ to larger values of $\widetilde{d}$, starting from the known minimum with $\widetilde{d} = 0$ to the rest of the domain. To this end, one may employ three disjoint sets of nodes as discussed in detail in Sethian (1999): the accepted nodes S_1, the trial nodes S_2 and the far nodes S_3. The values $\widetilde{d}_{i,j}$ of set S_1 are considered as known and will not be changed. The set S_2 consists of all nodes that have a neighbor in S_1. This is the set where the computation actually takes place and the values $\widetilde{d}_{i,j}$ can still change. In set S_3 are all other nodes, where an approximate solution $\widetilde{d}_{i,j}$ has not yet been computed as these are not in a neighborhood of a member of S_1. The FM algorithm can then be described by the following procedure:

(a) Find a grid point $\mathbf{x}$ in S_2 with the smallest value and change it to S_1.
(b) Place all neighbors of $\mathbf{x}$ into S_2 if they are not there already and update the arrival time according to (14) or (15) for all of them, if they are not already in S_1.
(c) If the set S_2 is not empty, return to (a).

In fact, any node can not be accepted more than one time. Additionally, in the case of a rectangular domain, each node has four neighbors at most and can therefore be updated up to four times. An efficient implementation amounts to storing the nodes in S_2, e.g. in a heap data structure, so that the smallest element $\mathbf{x}$ in step (a) can be chosen as fast as possible. Let us finally note that for initialization, one takes the nodes X which bear the boundary condition (6) of the PDE and puts them into the set S_1.

Let us now describe how to use the FM method based on the eikonal equation to construct a CVT. The computation of a VT can be done in the following manner: set $\Theta_0 = \{\mathbf{x}_i\}$ and solve (5)–(6) separately for each generator $\mathbf{x}_i$ with $i = 1, \dots N^g$. Through this one obtains N^g different geodesic distance maps $\widetilde{d}_i(\mathbf{x})$, where $\widetilde{d}_i$ is related to $\Theta_0 = \{\mathbf{x}_i\}$, such that the Voronoi cells can be constructed as follows

$$A(\mathbf{x}_i) = \left\{ \mathbf{x} \in \Theta \,:\, \widetilde{d}_i(\mathbf{x}) \leq \widetilde{d}_j(\mathbf{x}),\ j \in \{1, \ldots, N^g\} \backslash \{i\} \right\}. \tag{16}$$

Let us stress that there exists another strategy for constructing a VT, where only one distance map is computed. In this process, in particular, several wavefronts are started simultaneously which then converge together, cf. Peyré and Cohen (2003). At the points where two wavefronts collide the border between two Voronoi cells is obtained. This strategy can be realized by solving (5)–(6) for $\Theta_0 = X = \{\mathbf{x}_1, \ldots, \mathbf{x}_{N^g}\}$ and is much more efficient. However, the technique is also more cumbersome in terms of correct border detection of the Voronoi cells within the FM method.

Afterwards, the CVT based on the computed VT can be constructed. Let $\mathbf{x}^{(j)}$ denote the location on the grid and $\rho(j)$ be the corresponding density value. Then a discretization of (4) is

$$\mathbf{x}_i \leftarrow \frac{\sum_{j \in \mathscr{J}_i} \rho^{(j)} \mathbf{x}^{(j)}}{\sum_{j \in \mathscr{J}_i} \rho^{(j)}}, \qquad \mathscr{J}_i = \left\{ j \,:\, \mathbf{x}^{(j)} \in A(\mathbf{x}_i) \right\}, \qquad \forall i \in \{1, \ldots, N^g\}. \tag{17}$$

More precisely, the new center of mass $\mathbf{x}_i = (\bar{x}_i, \bar{y}_i)$ of a Voronoi cell A_i can be calculated in discrete form by simple summation for each dimension separately

$$\bar{x}_i = \frac{\sum_{j \in \mathscr{J}_i} \rho^{(j)} x^{(j)}}{\sum_{j \in \mathscr{J}_i} \rho^{(j)}}, \qquad \bar{y}_i = \frac{\sum_{j \in \mathscr{J}_i} \rho^{(j)} y^{(j)}}{\sum_{j \in \mathscr{J}_i} \rho^{(j)}}. \tag{18}$$

Let us mention that due to the discrete handling of the PDE-based approach, it can occur that one grid point $\mathbf{x}^{(j)}$ belongs to several Voronoi cells A_i with $i \in l \subset \{1, \ldots, N^g\}$. In this situation, the grid point $\mathbf{x}^{(j)}$ will be used for all corresponding Voronoi cells within the computation (18) with the factor $\frac{1}{\#l}$, where $\#l$ is the number of elements in l.

Lastly, as the starting point for trajectory optimization in the WAAM process, the planar graph has to be extracted from the PDE-based Voronoi cells. The technical realization of this processing step is described in the following.

The planar graph being sought is characterized via nodes and edges. Due to the fact that the PDE-based approach relies on an underlying grid, the borders between Voronois cells are in some sense not sharp edges, but a discrete representation of a continuous line. Therefore, in a first step, the nodes of the graph are identified. To this end, the corresponding grid points will be located that have neighbors in at least three different Voronoi cells. Let us remark that grid points who mark the nodes of the graph at the boundary of the given shape require special consideration. Moreover, such special grid points are often not single points but rather they accumulate. Therefore, the final nodes are determined by averaging over all coordinates of the detected grid points. Subsequently, the edges between two nodes can be set in a straightforward manner.

2.2.3 Comparison of geometric and PDE-based approach

The two presented methods for computing a CVT have some important properties. In the following we discuss these methods on a theoretical level, based on the fundamental differences of the two approaches. A more practically oriented discussion by means of numerical investigations will be given in Sect. 2.3.

The geometric approach generates the exact VT, whereas the PDE-based method can only deliver an approximation. In consequence, the discrete realization of the PDE-based approach affects the accuracy in both steps of Lloyd's algorithm. When computing a VT in the first step, grid locations near the boundary between multiple Voronoi cells are usually assigned to only one Voronoi cell, even if the area represented by the grid location is part of multiple cells. On the other hand, when computing the centroids in the second step, the new generators are always shifted towards the nearest grid point, introducing another approximation. Of course, the implications of these approximations greatly depend on the target application.

In both methods, the centers of mass $\mathbf{x}_i$ for constructing a CVT are computed by numerical approximation. For this reason, a significant factor in terms of the accuracy of $\mathbf{x}_i$ is the sampling rate of the underlying density. The sampling points of the geometric approach depend on the shape of the Voronoi cells, which is changing during the method. In contrast, for the PDE-based approach, the sampling points consist of the grid points for all iterations. Let us emphasize that in both approaches an approximation of (2) is minimized. However, with the geometric approach the sampling locations of the density change between the iterates of Lloyd's algorithm, and therefore the objective for the minimization is also changing.

With the geometric approach, the threshold for the stopping criteria used for Lloyd's algorithm can be set to arbitrarily small values. When using the PDE-based approach the generators can only move on a discrete grid, therefore a natural stopping criterion is reached if the generators of two consecutive iterations are identical.

In AM often non-convex shapes are of interest. With the geometric approach, this case requires more advanced implementations, see e.g. Tournois et al. (2010). With the PDE-based approach, non-convex shapes are significantly easier to handle.

Curved boundaries pose another hurdle for the geometric approach wherein the boundary is represented by linear segments. Approximating a curved boundary in this way may require a high number of such segments. In general this will result in a high number of mirrored generators near the boundary when computing a VT, leading to a higher computational workload. In contrast, the complexity of the PDE-based approach mostly relies on the number of grid points that Θ is approximated with. Therefore, much more complex shapes can be handled without introducing additional computational cost.

2.2.4 Combination of geometric and PDE-based approach

We now present a third approach to computing CVTs, where we combine some advantages of both the geometric and the PDE-based approach. In the following, this will be denoted as hybrid approach. To our knowledge, this method has not been presented in the literature before.

In this third strategy we start any iteration of Lloyd's algorithm by computing a VT analogously to the geometric approach, i.e. by mirroring the generators close to the boundary and utilizing a DT. In this way we obtain the exact vertex coordinates of the Voronoi cells, up to machine accuracy. Then we project these cells onto a regular grid and compute the centroids as in the PDE-based approach, cf. (17). In this procedure the density is sampled more regularly and usually more accurate than in the pure geometric method. Still, since the centroids are computed as an average of discrete grid points, the proposed hybrid approach delivers an approximation of a CVT.

Let us now give some details on the projection of Voronoi cells onto a grid. Any location $\mathbf{x}^{(j)}$ on the discrete grid that is in the closure $\overline{A_i}$ is labeled as part of the i-th cell. Therefore, it is not necessary that the rectangular area surrounding the grid location j lies completely within A_i. If the grid location j lies on the boundary between two cells, then it is labeled as part of both cells and the corresponding densities $\rho^{(j)}$ in (17) are weighted with $\frac{1}{2}$. The case where j is on the boundary between more than two cells is handled analogously.

2.3 Experimental evaluation of CVTs

We discuss the presented realizations of Lloyd's algorithm by means of numerical experiments. At first, we briefly explore properties of the PDE-based approach by comparing results of FM, when using first and second order upwind discretizations. Then we proceed by evaluating the geometric, PDE-based and hybrid approaches on the basis of qualitative as well as quantitative experiments.

All experiments in this section are conducted within the area $\Theta = [0, 1]^2$, approximated through uniform rectangular grids. The density is chosen either as constant, Gaussian or Rosenbrock function, i.e.

$$\rho^c(\mathbf{x}) = 1, \qquad \rho^g(\mathbf{x}) = e^{-4(x^2+y^2)}, \qquad \rho^r(\mathbf{x}) = (1-x)^2 + 100(y-x^2)^2, \quad (19)$$

with $\mathbf{x} = (x, y)^\top \in \Theta$. The Gaussian and Rosenbrock functions are also displayed in Fig. 3.

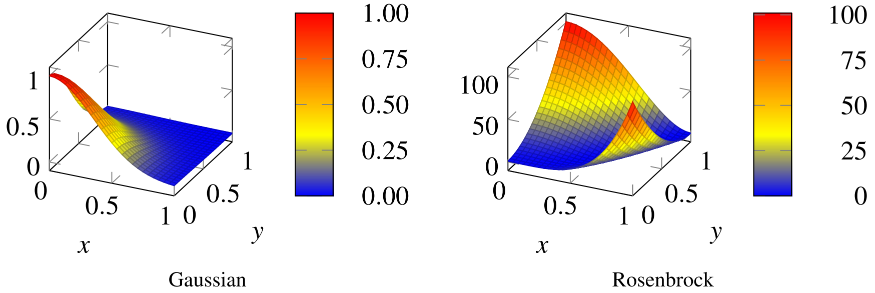


Fig. 3 Gaussian ρ^g and Rosenbrock function ρ^r in $\Theta = [0, 1]^2$. (Color figure online)

The geometric realization of Lloyd's algorithm is usually carried out for 1000 iterations, after which the progress of generators is zero up to machine accuracy in all tested cases. The PDE-based and the hybrid approach are stopped if there is no more movement of the generators. For the PDE-based approach, this means that the new generators shifted towards the underlying grid are identical to the old generators. With the hybrid approach usually, more iterations are carried out, since the generator locations are not rounded.

For quantitative comparisons, we evaluate approximations of the energy (2). To compare the energies of different methods after the respective last iteration we consider the following notation for the relative difference of reached energies, considering the density ρ^i with $i \in \{c, g, r\}$:

$$\Delta_i(A, n_A, B, n_B) = \frac{\mathscr{E}_i(B, n_B) - \mathscr{E}_i(A, n_A)}{\mathscr{E}_i(A, n_A)} \cdot 100\%, \tag{20}$$

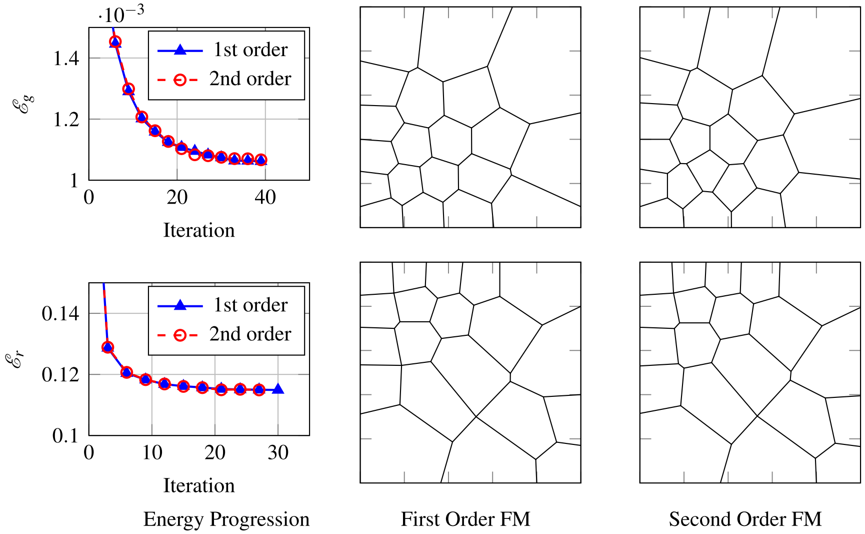
where $\mathscr{E}_i(A, n_A)$ is the energy of method A after n_A iterations and $\mathscr{E}_i(B, n_B)$ is defined analogously. Therefore, positive values indicate that method A reached a lower (better) energy in comparison to method B. This relative difference is based on the energy evaluated in the PDE-based approach, where the integral in (2) is approximated by a weighted sum over all grid points. For the energy minimized during the geometric approach we use the notation $\tilde{\mathscr{E}}_i$ and $\tilde{\Delta}_i$ denotes the respective relative difference. We remember that in this approximation of (2) the integral is split into a sum of integrals over triangles, which admit a closed form solution assuming a constant density in each triangle.

Let us note that the problem of finding a CVT may have multiple solutions, cf. Fig. 2. Therefore, the structure of a fixed point of Lloyd's algorithm may depend heavily on the initial set of generators. In our experiments, an initial set of generators is computed randomly for each considered number of generators N^g. Consequently, different methods always start with the same set of generators, up to differences induced by shifting towards grids of different grid lengths.

2.3.1 Fast marching of first and second order

As mentioned in Sect. 2.2.2, FM can be realized using upwind schemes of different order (Sethian 1999). In general high order schemes lead to more accurate approximations of the geodesic distance. For the experiment displayed in Fig. 4 first and second order FM is tested on a 200 × 200 grid. Both methods lead to almost identical energy values, i.e. $\Delta_g(\text{1st}, 39, \text{2nd}, 41) = 0.42\%$ and $\Delta_r(\text{1st}, 31, \text{2nd}, 28) = 0.0028\%$. However the resulting CVTs are different for the Gaussian density, while for the Rosenbrock function they are almost identical. When repeating this experiment on a 400 × 400 grid, the CVTs are visually almost indistinguishable and we observe the differences $\Delta_g(\text{1st}, 70, \text{2nd}, 74) = -0.082\%$ and $\Delta_r(\text{1st}, 41, \text{2nd}, 41) = 0.0016\%$.

We conclude that the influence of the upwind discretization order is relatively small in our application of computing CVTs and proceed by using first order FM for the PDE-based approach.



Energy Progression First Order FM Second Order FM

Fig. 4 Energy progression and CVTs for $\Theta = [0,1]^2$, $N^g = 20$ on a 200×200 grid computed with the PDE-based approach realized with FM of first and second order for *(top)* the Gaussian ρ^g and *(bottom)* the Rosenbrock function ρ^r. The energy progression and CVT for ρ^r are almost identical. For ρ^g there are some variations in the computed CVTs. (Color figure online)

2.3.2 Approximating exact CVTs

The discussed geometric approach can be used to compute exact CVTs, but it relies on the assumption of constant densities within triangular areas due to change during Lloyd's algorithm. Both the PDE-based and the hybrid approach rely on an underlying grid and are used to compute an approximation of a CVT.

To investigate the quality of these approximations, some visual results are displayed in Fig. 5. For $N^g = 10$ and 20 generators we compute an exact CVT with constant density ρ^c. Then we compare the approximations of the PDE-based and hybrid approaches computed on a 100×100, 200×200 and 400×400.

Let us note some visual observations based on Fig. 5. Obviously, the approximations from each method become more similar to the exact CVT when increasing the number of grid points. Although this is not a surprising property, based on it we conjecture that the PDE-based and hybrid approaches converge in some sense to the exact solution.

Additionally, we observe that the results of the hybrid approach are in general closer to the exact solution than those of the PDE-based approach on the same grid. One reason for this is that the hybrid approach shares the routine for computing VTs with the geometric approach, which is exact for a constant density. Furthermore, the hybrid approach enables more iterations and therefore finer tuning of the generators, since they are not shifted towards a grid in each iteration.

Finally, let us note that for a higher number of generators also more grid points are necessary to adequately approximate the exact solution. More generators

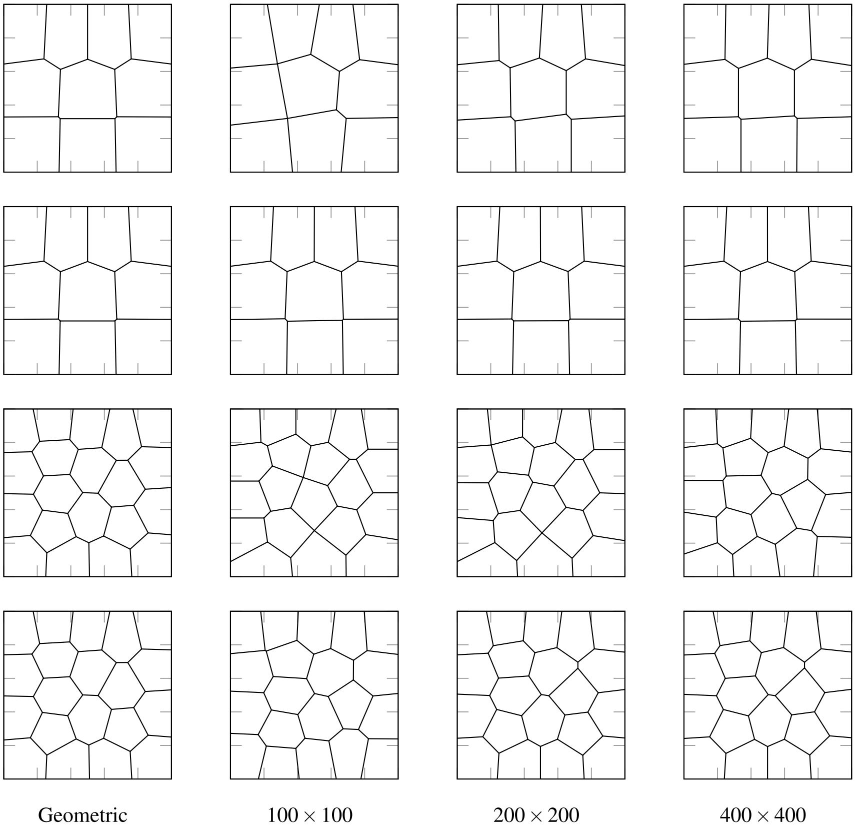


Fig. 5 Computed CVTs for $\Theta = [0, 1]^2$, constant density ρ^c and *(upper half)* $N^g = 10$ and *(lower half)* $N^g = 20$ generators. The first column holds the results with the geometric approach, the other columns display results of the PDE-based approach in the first and third row and the hybrid approach in the second and fourth row with varying grid length

automatically lead to smaller Voronoi cells requiring a higher resolution to adequately approximate their cell boundaries.

2.3.3 Minimization of the energy approximatively

In the following paragraphs, we discuss the three presented approaches for CVT computation in a more quantitative manner based on the energy progression. In Fig. 6 the continuous as well as the discrete energy approximations are displayed for $N^g = 20$.

Let us at first discuss the discrete energies in the upper half of Fig. 6, which we consider as a more accurate approximation of (2) on the utilized 400 × 400 grid since the density is sampled far more accurate.

For the constant density ρ^c the energies are almost indistinguishable for all three methods. The assumption, that the density is constant in certain regions,

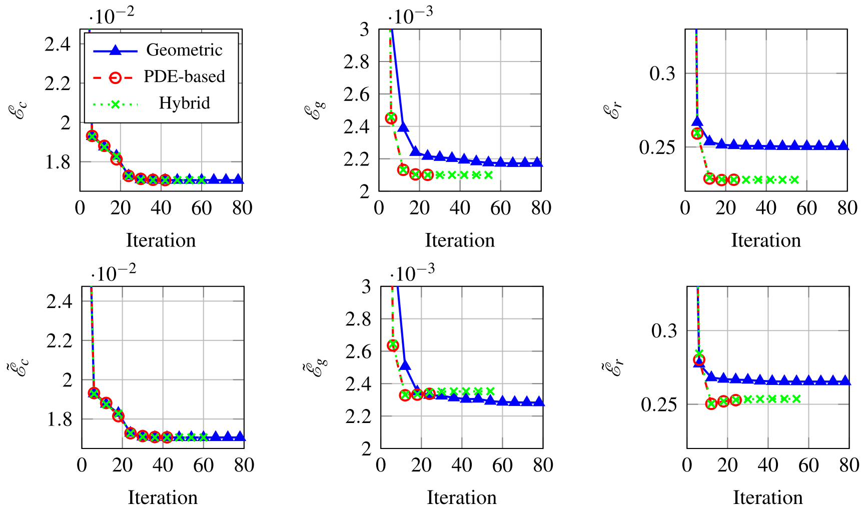


Fig. 6 Energy progression for $\Theta = [0,1]^2$ and $N^g = 20$ on a 400×400 grid with densities *(left-to-right)* ρ^c, ρ^g and ρ^r. *(top)* Discrete energy computed as a weighted sum on a uniform rectangular grid. *(bottom)* Continuous energy based on a closed form for triangles. (Color figure online)

actually holds true in this example and deviations due to approximations of cell boundaries and generator locations are very minor. With the Gaussian ρ^g the geometric approach leads to higher energies due to the less accurate density sampling. For the Rosenbrock function ρ^r these differences may be more pronounced due to the steeper slope, cf. Fig. 3.

In this context, we also consider the relative differences of the discrete energies after the respective last iteration of Lloyd's algorithm. They are given by

$$\Delta_c(\text{geom}, 1000, \text{PDE}, 44) = -0.119\%, \quad \Delta_c(\text{geom}, 1000, \text{hybr}, 65) = -0.001\%, \tag{21}$$

$$\Delta_g(\text{geom}, 1000, \text{PDE}, 26) = -3.217\%, \quad \Delta_g(\text{geom}, 1000, \text{hybr}, 55) = -3.243\%, \tag{22}$$

$$\Delta_r(\text{geom}, 1000, \text{PDE}, 25) = -9.030\%, \quad \Delta_r(\text{geom}, 1000, \text{hybr}, 55) = -9.031\%. \tag{23}$$

The PDE-based and hybrid approaches deliver very similar results in terms of energy, but with the hybrid approach, more iterations are executed before the generators do not change anymore.

The evaluation of the continuous energy in the lower half of Fig. 6 should favor the geometric approach since this method actually minimizes the continuous energy. However, the results are a bit more intricate. For a constant density, the energy progression for all methods is visually indistinguishable, while the Gaussian ρ^g eventually favors the geometric approach. For the Rosenbrock function ρ^r, the geometric approach actually delivers higher energies than the other approaches. The increasing continuous energies are due to the fact that with the PDE-based and hybrid approaches the discrete energy is minimized.

For the continuous energies, we obtain the following relative differences:

$$\tilde{\Delta}_c(\mathrm{geom}, 1000, \mathrm{PDE}, 44) = 0.035\%, \quad \tilde{\Delta}_c(\mathrm{geom}, 1000, \mathrm{hybr}, 65) = 0.001\%, \tag{24}$$

$$\tilde{\Delta}_g(\mathrm{geom}, 1000, \mathrm{PDE}, 26) = 2.371\%, \quad \tilde{\Delta}_g(\mathrm{geom}, 1000, \mathrm{hybr}, 55) = 3.082\%, \tag{25}$$

$$\tilde{\Delta}_r(\mathrm{geom}, 1000, \mathrm{PDE}, 25) = -4.730\%, \quad \tilde{\Delta}_r(\mathrm{geom}, 1000, \mathrm{hybr}, 55) = -4.407\%. \tag{26}$$

Although the continuous energy is the approximation that is minimized during the geometric approach, the other methods get to a solution with lower energy for ρ^r. This may again be due to the fact, that the slope of the Rosenbrock function may become very steep, especially towards the edges $(1, 0)^\top$ and $(0, 1)^\top$.

2.3.4 Comment on results

Let us conclude this numerical study by conjecturing that the hybrid approach outperforms or is equal to the geometric approach in most cases. The PDE-based and hybrid approaches deliver similar results under the assumption that the chosen grid is sufficiently fine for the task at hand. The choice of the preferred algorithm may finally depend on the specific application, taking into account especially the shape of the boundary $\partial\Theta$.

3 Trajectory optimization

In our application, every produced workpiece, like the prototypical one in Fig. 7 (left), is built layer by layer onto an underlying massive block of material. This so-called substrate plate is the basis for the first layer. Due to its layer-wise structure,

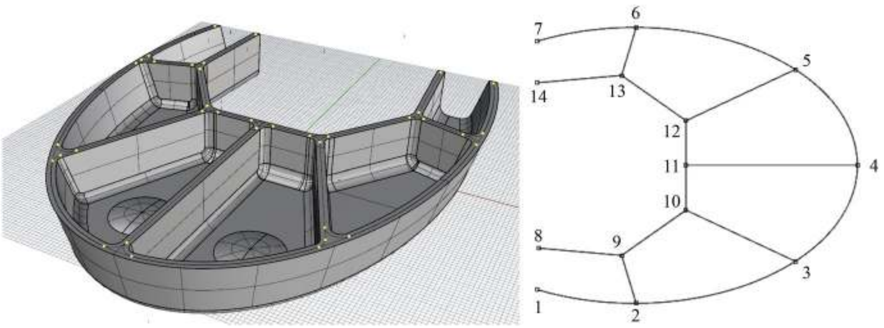


Fig. 7 A prototypical workpiece to be produced with WAAM in 3D *(left)*, and a single layer of it as an undirected graph *(right)*

the problem can be reduced to finding an optimal path through all segments for one layer given some initial temperature. A single layer can be considered as an undirected graph with an edge for every segment to print and nodes at all intersection points of two segments, as it is shown in Fig. 7 (right).

Because the total time for building a layer should be minimized, an optimal path will visit every segment only once and thus is Eulerian, if such a path exists. Hierholzer and Wiener (1873) proved in 1873, that a graph can have at most two nodes of odd node degree to contain an Eulerian path. But this can not be guaranteed for arbitrary workpieces since their structure is limited by their expected functionality. So additional transitions from one node to another without welding, called transition moves, are required. Hence the problem of finding an optimal welding sequence can be related to the Chinese Postman Problem (Edmonds and Johnson 1973).

3.1 Mathematical and physical observations

The undirected graph representing a single layer can be described by a set of n nodes $\mathscr{V} = \{1, \dots, n\}$ and a set of segments $\mathscr{W} \subset \mathscr{V} \times \mathscr{V}$. Since every segment can be printed in either direction, the set $\overline{\mathscr{W}} = \{(i,j) | (i,j) \in \mathscr{W} \vee (j,i) \in \mathscr{W}\}$ is defined. Furthermore, the sets $\mathscr{V}_{odd} \subseteq \mathscr{V}$ and $\mathscr{V}_{even} = \mathscr{V} \backslash \mathscr{V}_{odd}$ of all nodes with odd and even node degree are used as abbreviations.

The welding source has a maximal welding velocity v^w and a maximal transition speed v^m, which is much higher than the welding velocity. Thus it is assumed that a transition move requires no measurable time. In particular, it is much faster than a single time-step. Using this, the overall time T to build a single layer is a priori known and the interval $[0, T]$ can be discretized with grid points $\tilde{k} \cdot \Delta t$, $\tilde{k} \in \mathscr{T}_0 = \{0, \dots, T^{max}\}$ with step size Δt and total number of time steps T^{max}. Later we will use only $t \in \mathscr{T}_0$ to describe the grid point $t \cdot \Delta t$. Furthermore, $\mathscr{T} = \mathscr{T}_0 \backslash \{0\}$ is used as abbreviation.

Since the length $l_{i,j}$ of the segment $(i,j) \in \mathscr{W}$ can be measured, the number of necessary time steps $\tau_{i,j}$ for welding segment $(i,j) \in \mathscr{W}$ can be calculated and it holds

$$T^{max} = \sum_{(i,j) \in \mathscr{W}} \tau_{i,j} = \sum_{(i,j) \in \mathscr{W}} \left\lceil \frac{l_{i,j}}{v^w \Delta t} \right\rceil. \tag{27}$$

In the Chinese Postman Problem, additional edges are inserted between nodes of odd degree, to achieve an Eulerian graph. In the same way, the necessary transition moves in our model are restricted to nodes with odd degree, since a transition move to a node of even degree would cause this node to have odd degree afterwards. This would make another transition necessary to get an even degree again, which is contrary to our goal of minimizing the production time and the number of transition moves. Thus moves to nodes with even degree without welding are prohibited. Similarly, the welding torch must neither start nor end at nodes of even degree, if there are any nodes of odd degree since this would also cause additional transition moves.

In the following paragraphs, we will introduce several constraints considered for modeling our WAAM application.

3.1.1 Conduction

By Fourier's law (Fourier 1878), the heat flux density q inside a material between two points in a rod can be expressed by

$$q = -\lambda \frac{\Delta\theta}{\Delta x}, \tag{28}$$

with thermal conductivity λ, the temperature difference of the points $\Delta\theta$, and the distance between the points Δx. Besides the heat flux, the total transferred thermal energy Q^{con} depends on the cross-sectional area D of the rod and the time difference Δt, and it holds

$$Q^{con} = Dq\Delta t \overset{(28)}{=} -\lambda D \frac{\Delta\theta}{\Delta x}\Delta t. \tag{29}$$

Hence the thermal energy transfer by conduction from node $i \in \mathscr{V}$ to node $j \in \mathscr{V}$ can be expressed by

$$Q^{con}(i,j,t) = \lambda D \frac{\theta_{i,t} - \theta_{j,t}}{l_{i,j}} \Delta t, \tag{30}$$

where $l_{i,j}$ is again the length of segment $(i,j) \in \overline{\mathscr{W}}$. Using the equations

$$Q = cm\Delta\theta \tag{31}$$

$$\lambda = \alpha c \rho \tag{32}$$

$$m = \rho V \tag{33}$$

with heat capacity c, mass m, thermal diffusivity α, density ρ, and volume V, the change of temperature $\Delta\theta$ at node $i \in \mathscr{V}$ at time step t due to conduction can be computed by

$$\begin{aligned}\Delta\theta^{con}_{i,t} &= \frac{\sum_{j\in\mathscr{N}(i)} Q^{con}(i,j,t)}{cm} \overset{(30)}{=} \frac{\lambda D}{cm} \sum_{j\in\mathscr{N}(i)} \frac{\theta_{i,t} - \theta_{j,t}}{l_{i,j}} \Delta t \\ &\overset{(32),(33)}{=} \frac{\alpha D}{V} \sum_{j\in\mathscr{N}(i)} \frac{\theta_{i,t} - \theta_{j,t}}{l_{i,j}} \Delta t,\end{aligned} \tag{34}$$

where $\mathscr{N}(i)$ denotes the set of adjacent nodes to node $i \in \mathscr{V}$. A positive value of $\Delta\theta^{con}_{i,t}$ refers to a decrease of the temperature at node $i \in \mathscr{V}$.

The substrate plate that the workpiece is built on describes the possibility of heat transmission through conduction between two arbitrary nodes without restriction due to adjacency. The distance between them depends on their Euclidean distance $d^e_{i,j}$ on the substrate plate and the height of both nodes relative to the plate. Using (34), it can be modeled by

$$\Delta\theta^p_{i,t} = \frac{\alpha D}{V}\Delta t \sum_{j\in\mathscr{V}} \frac{\theta^m_{i,t} - \theta^m_{j,t}}{d^e_{i,j} + 2h^w(n_l - 1)}, \tag{35}$$

where h^w is the height of a single layer and both nodes are located in the n_l-th layer.

Next to this approach, the heat conduction within one edge $(i,j) \in \mathscr{W}$ can be modeled as a one-dimensional heat equation (Jost 2007) of the form

$$\frac{\partial\theta}{\partial t}(x,t) = \alpha\frac{\partial^2\theta}{(\partial x)^2}(x,t) \quad \forall x \in \left[0, l_{i,j}\right],\ t \in (0,T), \tag{36}$$

$$\theta(0,t) = \theta_{i,t} \quad \forall t \in [0,T], \tag{37}$$

$$\theta(l_{i,j},t) = \theta_{j,t} \quad \forall t \in [0,T], \tag{38}$$

$$\theta(x,0) = \theta^0(x) \quad \forall x \in \left(0, l_{i,j}\right), \tag{39}$$

with the initial temperature distribution $\theta^0(x)$. Now finite differences can be applied to discretize (36) for any edge on the interval $\left[0, l_{i,j}\right]$ in the time horizon $[0,T]$. Let $N^{int}_{i,j}$ be the number of inner discretization points and $\mathscr{L}_{i,j} = \{1,\dots,N^{int}_{i,j}\}$ the set of their indices for the edge $(i,j) \in \mathscr{W}$. They are positioned equidistantly along the edge at $\frac{k\cdot l_{i,j}}{N^{int}_{i,j}+1}$, $k \in \mathscr{L}_{ij}$, resulting in the stepsize $\Delta x_{i,j} = \frac{l_{i,j}}{N^{int}_{i,j}+1}$. The boundary points are represented by the indices 0 on the left and $N^{int}_{i,j} + 1$ on the right boundary respectively. In the following, we will refer to this discretization points by using their index $k \in \mathscr{L}_{i,j} \cup \{0, N^{int}_{i,j} + 1\}$.

Furthermore, we choose in our model formulation $\tau_{i,j} = N^{int}_{i,j} + 1$, $(i,j) \in \mathscr{W}$, resulting in one discretization point per time step. Since the processing time of any edge is transformed into a number of time steps (27) and thus is rounded up to be integer, we get one additional time step per edge. Hence the velocity of the welding torch is not the assumed value v^w, but a little lower.

The alternative would be a discretization to keep the velocity v^w. Then every segment would have the length $v^w\Delta t$ and if the length of the whole edge is no multiple of this value, the segment between the last inner discretization point and the end point of the edge has to be smaller than the other edge segments. This results in a discretization depending on the welding direction of the edge. Thus the position of the inner discretization points would change subject to the assignment of the starting point and the end point of the edge, leading to a more complex and less robust model.

For discretization the finite difference schemes derived in Recktenwald (2004) can be used, e.g. backward time centered space (BTCS) leads to

$$\theta_{k,t-1} = -\tilde{\alpha}_{i,j}\theta_{k-1,t} + (1 + 2\tilde{\alpha}_{i,j})\theta_{k,t} - \tilde{\alpha}_{i,j}\theta_{k+1,t} \qquad \forall k \in \mathscr{L}_{i,j},\ t \in \mathscr{T}, \tag{40}$$

with $\tilde{\alpha}_{i,j} = \frac{\alpha\Delta t}{(\Delta x_{i,j})^2}$.

By adding a heat source $f(x, t)$ to the original heat equation and doing the same derivation, we get for (40)

$$\theta_{k,t-1} = -\tilde{\alpha}_{i,j}\theta_{k-1,t} + (1 + 2\tilde{\alpha}_{i,j})\theta_{k,t} - \tilde{\alpha}_{i,j}\theta_{k+1,t} - \Delta t f_{k,t} \qquad \forall k \in \mathscr{L}_{i,j},\ t \in \mathscr{T}, \tag{41}$$

with discretized heat source $f_{k,t}$, the function value of $f(x, t)$ at the position of interior point $k \in \mathscr{L}_{i,j}$ at time step $t \in \mathscr{T}$. The boundary conditions

$$\theta_{0,t} = \theta_{i,t} \qquad \forall t \in \mathscr{T}_0, \tag{42}$$

$$\theta_{N_{i,j}^{int}+1,t} = \theta_{j,t} \qquad \forall t \in \mathscr{T}_0 \tag{43}$$

according to (37) and (38) have to hold, besides the discretized initial condition, i.e. (39)

$$\theta_{k,0} = \theta_k^0 \qquad \forall k \in \mathscr{L}_{i,j}, \tag{44}$$

where θ_k^0 is the value of $\theta^0(x)$ at the position of the k-th interior point.

3.1.2 Radiation

The net rate of radiation heat transfer (Çengel 2002) between two surfaces r and s can be described by

$$\dot{Q}_{r\to s} = \sigma\epsilon A_r^O F_{r\to s}\left(\theta_r^4 - \theta_s^4\right), \tag{45}$$

with Stefan–Boltzmann constant σ, the material dependent emissivity factor ϵ, surface area A_r^O, view factor $F_{r\to s}$, and surface temperatures θ_r and θ_s. Their computation for our purpose is explained in Sect. 3.1.3, while an overview about the calculation of view factors for simple configurations and some properties can be found in Çengel (2002).

In the following, the summation rule

$$\sum_{s\in\mathscr{R}} F_{r\to s} = 1 \qquad \forall r \in \mathscr{R} \tag{46}$$

for a set of surfaces $\mathscr{R}$, building an enclosed half space, is used.

Since we consider both half spaces, denoted by $\mathscr{R}_1$ and $\mathscr{R}_2$, all view factors add up to two. As abbreviation we use $\mathscr{S} = \mathscr{R}_1 \cup \mathscr{R}_2$. Furthermore, we add another surface for the ambient with $F_{r\to amb} = 2 - \sum_{s\in\mathscr{S}} F_{r\to s}$ and ambient temperature θ_{amb}, thus the net rate of radiation heat transfer of surface r can be written as

$$
\begin{aligned}
\dot{Q}_r &= \sum_{s\in\mathscr{S}} \dot{Q}_{r\to s} + \dot{Q}_{r\to amb} \\
&= \sigma\epsilon A_r^O \left(\sum_{s\in\mathscr{S}} F_{r\to s}\left(\theta_r^4 - \theta_s^4\right) + F_{r\to amb}\left(\theta_r^4 - \theta_{amb}^4\right) \right) \\
&= \sigma\epsilon A_r^O \left(2\left(\theta_r^4 - \theta_{amb}^4\right) - \sum_{s\in\mathscr{S}} F_{r\to s}\left(\theta_s^4 - \theta_{amb}^4\right) \right).
\end{aligned}
\tag{47}
$$

Since we are interested in the temperature distribution of the workpiece, we assign edge segments to every node and all interior points of the discretized heat equation. The segments for interior points $k \in \mathscr{L}_{ij}$ of edge $(i,j) \in \mathscr{W}$ have length $\frac{l_{ij}}{N_{ij}^{int}+1}$ and the point is located in its center. Every node $i \in \mathscr{V}$ has one segment per incident edge $(i,j) \in \overline{\mathscr{W}}$ starting in i and with length $\frac{l_{ij}}{2(N_{ij}^{int}+1)}$. As abbreviation for the length of all segments related to i

$$
l_i = \sum_{(i,j)\in\overline{\mathscr{W}}} \frac{l_{ij}}{2(N_{ij}^{int}+1)}
\tag{48}
$$

is used. Further, we denote with m_i and A_i^O the mass and the surface area of the segment related to the node or interior point i.

Taking the formula for the density $\rho = \frac{m}{V} = \frac{m}{A^O a^w}$ with mass m and width of the edge segments a^w, the transferred energy due to the radiation of node $i \in \mathscr{V}$ can be computed by

$$
Q_i = \dot{Q}_i \Delta t = \frac{\sigma\epsilon m_i}{\rho a^w} \left(2\left(\theta_i^4 - \theta_{amb}^4\right) - \sum_{j\in\mathscr{V}} F_{i\to j}\left(\theta_j^4 - \theta_{amb}^4\right) \right) \Delta t.
\tag{49}
$$

Hence, the change of temperature $\Delta\theta$ of node $i \in \mathscr{V}$ due to the radiation at time step t can be obtained by combination of (31) and (49) as

$$
\Delta\theta_{i,t}^{rad} = \frac{Q_i}{cm_i} = \frac{\sigma\epsilon}{c\rho a^w} \left(2\left(\theta_i^4 - \theta_{amb}^4\right) - \sum_{j\in\mathscr{V}} F_{i\to j}\left(\theta_j^4 - \theta_{amb}^4\right) \right) \Delta t.
\tag{50}
$$

Here again a positive value of $\Delta\theta_{i,t}^{rad}$ refers to a decrease in temperature.

Because of the non-linearity of the power function $p(x) = x^4$, Eq. (50) has to be linearized, so that a MILP solver can deal with it. For that a piecewise linear function is used. Due to the numerical comparison of different piecewise linearization techniques in Fügenschuh et al. (2014), it is modeled by the incremental method. This approach was first described by Markowitz and Manne (1957), but here it is implemented according to Fügenschuh et al. (2014) with K^{pwl} intervals $\left[\Phi_{k,0}, \Phi_{k,1}\right]$, $k \in \{1, \dots, K^{pwl}\}$, satisfying $\Phi_{k,0} = \Phi_{k-1,1}$, $k \in \{2, \dots, K^{pwl}\}$. Therefore, variables $\delta_{i,t,k} \in [0,1]$ and $b_{i,t,k} \in \{0,1\}$ are defined. The former describes the partition of interval k, that is less or equal to the argument of the original function for node $i \in \mathscr{V}$ at time step t, while the latter determines the active interval k of the piecewise linear function for node $i \in \mathscr{V}$

at time step t. Using the incremental method the power function is approximated in a piecewise linear way with

$$(\theta^m_{i,t})^4 \approx \sum_{k=1}^{K^{pwl}} \delta_{i,t,k}(\Phi^4_{k,1} - \Phi^4_{k,0}) \qquad \forall i \in \mathcal{V},\ t \in \mathcal{T}_0, \tag{51}$$

where the auxiliary variables are modeled subject to

$$\theta^m_{i,t} = \sum_{k=1}^{K^{pwl}} \delta_{i,t,k}(\Phi_{k,1} - \Phi_{k,0}) \quad \forall i \in \mathcal{V},\ t \in \mathcal{T}, \tag{52}$$

$$b_{i,t,k} \leq \delta_{i,t,k} \quad \forall i \in \mathcal{V},\ t \in \mathcal{T}_0,\ k \in \{1, \dots, K^{pwl}\}, \tag{53}$$

$$b_{i,t,k} \geq \delta_{i,t,k+1} \quad \forall i \in \mathcal{V},\ t \in \mathcal{T}_0,\ k \in \{1, \dots, K^{pwl} - 1\}. \tag{54}$$

A second approach to linearize the radiation heat flux is based on the Rosseland diffusion equation (Howell et al. 2015) for optically thick media

$$q = -\frac{16\sigma\theta^3}{3\beta_R}\frac{\partial\theta}{\partial n}, \tag{55}$$

where β_R is the Rosseland mean attenuation coefficient and $\frac{\partial\theta}{\partial n}$ the derivative of θ in normal direction n. Isolating the partial derivative on one side, the slope can be written as

$$\frac{\partial\theta}{\partial n} = -\frac{3\beta_R q}{16\sigma\theta^3}. \tag{56}$$

Hence, by discretizing the time, the temperature at the next time step is calculated by

$$\theta_{t+1} = \left(1 - \frac{3\beta_R q_t \Delta t}{16\sigma\theta_t^4}\right)\theta_t. \tag{57}$$

Using the Stefan–Boltzmann equation $q = \sigma\theta^4$ according to Howell et al. (2015), the fraction in (57) can be simplified into

$$\theta_{t+1} = \left(1 - \frac{3\beta_R \Delta t}{16}\right)\theta_t. \tag{58}$$

In the following, we will use $\kappa^e = 1 - \frac{3\beta_R\Delta t}{16}$ as abbreviation. Since a negative value of κ^e would cause alternating signs between two time steps and it should describe a cooling process, we set $\kappa^e \in (0, 1)$.

3.1.3 View factors

According to Çengel (2002), the view factor $F_{r\to s}$ between two surfaces r and s is defined by the surface integral

$$F_{r\to s} = \frac{1}{A_r^O \pi} \int\limits_{A_r} \int\limits_{A_s} \frac{\cos(\beta_r)\cos(\beta_s)}{(d_{rs}^e)^2} \mathrm{d}A_r \mathrm{d}A_s, \tag{59}$$

where β_r and β_s are the angles of the surfaces r and s, set up by the respective normal vector and the connection line between them. Furthermore, A_r and A_s are the set of all points in the respective surface. All these quantities are shown in Fig. 8 for an arbitrary configuration. The view factor describes the portion of the emitted energy of r, which strikes s directly.

The angles $\cos(\beta_r)$ and $\cos(\beta_s)$ in (59) can be substituted by the cosine rule for vector spaces with scalar products

$$\cos(\beta_r) = \frac{\langle \mathbf{n}_r, \mathbf{d}\rangle}{\|\mathbf{n}_r\|\|\mathbf{d}\|}, \qquad \cos(\beta_s) = \frac{\langle \mathbf{n}_s, -\mathbf{d}\rangle}{\|\mathbf{n}_s\|\|\mathbf{d}\|}, \tag{60}$$

where $\mathbf{d}$ is the vector pointing from r to s with $\|\mathbf{d}\| = d_{rs}^e$. Thus the surface integral (59) can be rewritten as

$$F_{r\to s} = \frac{1}{A_r^O \pi} \int\limits_{A_r} \int\limits_{A_s} \frac{\langle \mathbf{n}_r, \mathbf{d}\rangle\langle \mathbf{n}_s, -\mathbf{d}\rangle}{\|\mathbf{n}_r\|\|\mathbf{n}_s\|\|\mathbf{d}\|^4} \mathrm{d}A_r \mathrm{d}A_s. \tag{61}$$

Every edge segment related to a node $i \in \mathcal{V}$ can be described by the parameter form $\phi_i(u,v) = \mathbf{p}_i + u\mathbf{d}_i + v\mathbf{h}^w$, $u, v \in [0,1]$ with $\mathbf{p}_i$ one end point of the segment projected to the xy-plane, $\mathbf{d}_i$ the vector pointing from $\mathbf{p}_i$ to the other end point projected to the xy-plane, and $\mathbf{h}^w = \left(0,\ 0,\ h^w\right)^T$ is the height of a single layer. The resulting rectangular surface of node $i \in \mathcal{V}$ is used as a parameterization to transform (61) into

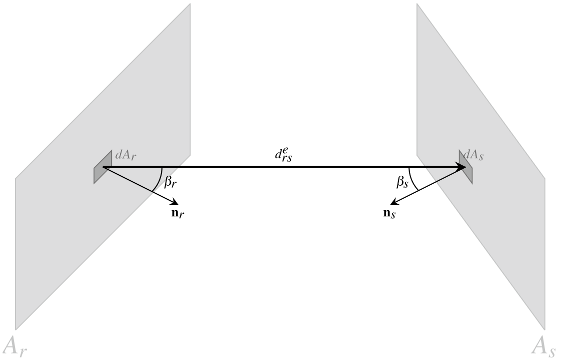


Fig. 8 Arbitrary configuration of two rectangles with normals $\mathbf{n}_r, \mathbf{n}_s$

$$F_{i\to j} = \frac{1}{A_i^O \pi} \int\limits_0^1 \int\limits_0^1 \int\limits_0^1 \int\limits_0^1 f(u_1, u_2, v_1, v_2) \|\mathbf{d}_i \times \mathbf{h}^W\| \|\mathbf{d}_j \times \mathbf{h}^W\| \mathrm{d}u_1 \mathrm{d}v_1 \mathrm{d}u_2 \mathrm{d}v_2, \tag{62}$$

with

$$f(u_1, u_2, v_1, v_2) = \frac{\langle \mathbf{n}_i, \phi_j(u_2, v_2) - \phi_i(u_1, v_1)\rangle \langle \mathbf{n}_j, \phi_i(u_1, v_1) - \phi_j(u_2, v_2)\rangle}{\|\mathbf{n}_i\| \|\mathbf{n}_j\| \|\phi_j(u_2, v_2) - \phi_i(u_1, v_1)\|^4}. \tag{63}$$

The resulting four-dimensional integral (62) is solved numerically using a Quasi-Monte-Carlo-method, motivated by Kuo and Sloan (2005). Therefore, every interval $[0, 1]$ is divided up into N^{QMC} subintervals of similar length and then a random point is chosen for any combination of subintervals to compute the value of the integrand. The average of these values is taken as an approximation for the integral. This procedure is rerun a fixed number L^{QMC} of times and the average of all runs is used for the calculation of the view factor. Since every node is related to a number of segments and the view factors are computed for them separately, they have to be merged to get the view factor for the whole geometry. Therefore, the reciprocity rule of view factors

$$A_r F_{r\to s} = A_s F_{s\to r} \tag{64}$$

and the superposition rule

$$F_{r\to(s,t)} = F_{r\to s} + F_{r\to t} \tag{65}$$

for surfaces $r, s, t \in \mathscr{S}$ are used, where (s, t) is the surface built by combining of s and t.

Let $s_{i,1}, \dots, s_{i,n}$ be the segments related to node $i \in \mathscr{V}$ and $\mathscr{S}$ the set of all segments in the workpiece. Then it holds

$$F_{r\to(s_{i,1},\dots,s_{i,n})} = \sum_{j=1}^{n} F_{r\to s_{i,j}} \qquad \forall r \in \mathscr{S}. \tag{66}$$

Applying the reciprocity rule to (66) results in

$$\frac{A^O_{(s_{i,1},\dots,s_{i,n})}}{A^O_r} F_{(s_{i,1},\dots,s_{i,n})\to r} = \sum_{j=1}^{n} \frac{A^O_{i,j}}{A^O_r} F_{s_{i,j}\to r}. \tag{67}$$

Since the surface related to node $i \in \mathscr{V}$ is the combined surface of $s_{i,1}, \dots, s_{i,n}$, the view factor $F_{i\to r}$ from i to r can be obtained from (67)

$$F_{i\to r} = \sum_{j=1}^{n} \frac{A^O_{i,j}}{A^O_i} F_{s_{i,j}\to r}. \tag{68}$$

3.2 Model formulation

In the following paragraphs, we want to derive a mixed-integer linear model, consisting of two main parts. On one side there are constraints motivated by Fügenschuh et al. (2019) to describe a feasible path of the welding torch, passing all edges in minimal time with the necessary number of transition moves. The second part calculates the temperature progression of all nodes and interior discretization points of the workpiece using the results of Sects. 3.1.1 and 3.1.2. Both parts are linked by constraints for the heating process if the welding torch arrives at one node.

To track the path of the welding head it is necessary to know, which edge is welded at a specific time step and when the transition moves are done. For path tracking, a set $\mathscr{W}^* \subset \mathscr{V} \times \mathscr{T}_0 \times \mathscr{V} \times \mathscr{T}$ of pairs (i, t_i, j, t_j), such that $(i, j) \in \overline{\mathscr{W}}$ and $t_j = t_i + \tau_{i,j}$ is defined. For transiting, the possible moves are defined according to Sect. 3.1 by

$$\mathscr{U}^* = \{(i, j, t) \in \mathscr{V}_{odd} \times (\mathscr{V}_{odd} \backslash \{i\}) \times \{1, \dots, T^{max} - 1\}\}$$

and the number of necessary moves is

$$\omega = \max\left\{ \frac{|\mathscr{V}_{odd}|}{2} - 1, 0 \right\}. \tag{69}$$

In $t = 0$ and $t = T^{max}$ no transition move is needed, since the welding head can be placed arbitrarily above any node with odd degree at the beginning and there is no determined node with odd degree to end the welding process.

For generality we assume for the model $\mathscr{V}_{odd} \neq \emptyset$. If there are no nodes of odd degree, then the constraints respective to $\mathscr{V}_{even}$ can be dropped and in the remaining constraints $\mathscr{V}_{odd}$ is replaced by $\mathscr{V} = \mathscr{V}_{even}$.

Next to their introduction in the following sections, a list of all used sets, parameters, and variables can be found in Tables 1 and 2, respectively.

3.2.1 Path generation

The working sequence of the segments is represented by a family of binary variables $w_{i,t_i,j,t_j} \in \{0, 1\}$ for every pair $(i, t_i, j, t_j) \in \mathscr{W}^*$, equal to 1 if the welding head starts at node $i \in \mathscr{V}$ at time step t_i and arrives at node $j \in \mathscr{V}$ in time step t_j, and 0 otherwise. A transition move is described using binary variables $u_{i,j,t} \in \{0, 1\}$ for every $(i, j, t) \in \mathscr{U}^*$, equal to 1 if the welding head moves from node $i \in \mathscr{V}$ to node $j \in \mathscr{V}$ at time step t, and 0 otherwise.

The welding process has to start in an arbitrary node $i \in \mathscr{V}_{odd}$ at time step 0, while every node with even degree cannot be chosen as a starting point:

$$\sum_{\substack{i,j,t_j \,:\, (i,0,j,t_j) \in \mathscr{W}^* \\ i \in \mathscr{V}_{odd}}} w_{i,0,j,t_j} = 1, \tag{70}$$

Table 1 Overview of the sets and parameters used in the mixed-integer linear model

Symbol	Definition
$\mathscr{L}_{i,j} = \{1, \dots, N_{i,j}^{int}\}$	Set of all interior points of segment (i, j)
$\mathscr{N}(i)$	Set of all nodes adjacent to node i
$\mathscr{P}_k$	k-th ring of the discretized weld pool
$\mathscr{T} = \{1, \dots, T^{max}\}$	Set of discrete time steps
$\mathscr{T}_0 = \{0, \dots, T^{max}\}$	Set of discrete time steps including the initial situation
$\mathscr{U}^*$	Set of possible transition moves during the welding process
$\mathscr{V} = \{1, \dots, n\}$	Set of all nodes
$\mathscr{V}_{even} \subseteq \mathscr{V}$	Set of all nodes with even node degree
$\mathscr{V}_{odd} \subseteq \mathscr{V}$	Set of all nodes with odd node degree
$\mathscr{W}$	Set of all segments
$\overline{\mathscr{W}}$	Set of all segments in either direction
$\mathscr{W}^*$	Set of all segments in either direction with welding process starting in a specified time step
a^w	Width of the segments of the workpiece
c	Heat capacity
$d_{i,j}^e$	Euclidian distance between nodes i and j
$F_{i \to j}$	View factor between the surfaces associated to the nodes i and j
K^{pwl}	Number of intervals for the piecewise linear approximation
K^w	Number of rings of discretized weld pool
M	Sufficiently large constant
n	Number of nodes
n_l	Number of layer l
$N_{i,j}^{int}$	Number of interior points in segment (i, j)
r^w	Radius of the circular weld pool
$r_k \in [0, r^w]$	Outer radius of the k-th ring of the discretized weld pool
T^{max}	Number of necessary time steps to weld whole layer
$\tilde{\alpha}_{i,j}$ (cf. (40))	Scaled thermal diffusity of the segment (i, j)
Δt	Length of one discrete time step
$\boldsymbol{\Phi}_{k,0}$	Left limit of the k-th interval of the piecewise linear approximation
$\boldsymbol{\Phi}_{k,1}$	Right limit of the k-th interval of the piecewise linear approximation
ϕ_t^{amb}	Ambient temperature at time step t
φ_i^{init}	Initial temperature of node i
$\varphi_{i,j,k}^{init}$	Initial temperature of the k-th interior point of segment (i, j)
φ^{max}	Welding temperature
$\varphi_{n_l}^{rad}$	Temperature increase due to incoming radiation in layer n_l
$\kappa^e \in (0, 1)$	Cooling factor
$\kappa_k^w \in [0, 1]$	Portion of the welding temperature transferred to a node in the k-th ring of the discretized weld pool
σ	Stefan–Boltzmann constant
ρ	Material density
ϵ	Material dependent emissivity factor
ω (cf. (69))	Number of necessary transition steps

Table 2 Overview of the variables used in the mixed-integer linear model

Symbol	Definition
$a^m_{i,t} \in \{0,1\}$	Binary variable indicating whether the node i has been visited by the welding torch not later than time step t
$a^{fd}_{i,j,k,t} \in \{0,1\}$	Binary variable indicating whether the k-th interior point of segment (i,j) has been visited by the welding torch not later than time step t
$b_{i,t,k} \in \{0,1\}$	Binary variable indicating whether the k-th interval of the piecewise linear approximation at node i at timestep t is active
$b_{i,j,h,t,k} \in \{0,1\}$	Binary variable indicating whether the k-th interval of the piecewise linear approximation at the h-th interior point of segment (i,j) at timestep t is active
$u_{i,j,t} \in \{0,1\}$	Binary variable indicating whether the welding torch makes a transition move from node i to node j at time step t
$w_{i,t_i,j,t_j} \in \{0,1\}$	Binary variable indicating whether the segment (i,j) is welded from time step t_i to time step t_j
$\delta_{i,t,k} \in [0,1]$	Variable indicating the partition of the k-th interval of the piecewise linear approximation that is less or equal to the argument of the original function at node i at time step t
$\delta_{i,j,h,t,k} \in [0,1]$	Variable indicating the partition of the k-th interval of the piecewise linear approximation that is less or equal to the argument of the original function at the h-th interior point of segment (i,j) at time step t
$\Delta\theta^{p}_{i,t}$ (cf. (35))	Temperature change at node i at time step t due to heat transfer by the substrate plate
$\Delta\theta^{rad}_{i,t}$ (cf. (93))	Temperature change at node i at time step t due to heat transfer by radition
$\Delta\theta^{rad}_{i,j,k,t}$ (cf. (94))	Temperature change at the k-th interior point of segment (i,j) at time step t due to heat transfer by radition
$\theta^m_{i,t} \in \mathbb{R}_+$	Temperature of node i at time step t
$\theta^{fd}_{i,j,k,t} \in \mathbb{R}_+$	Temperature of the k-th interior point of segment (i,j) at time step t
$\vartheta^+_{i,j,t} \in \mathbb{R}_+$	Positive portion of the temperature gradient of segment (i,j) at time step t
$\vartheta^-_{i,j,t} \in \mathbb{R}_+$	Negative portion of the temperature gradient of segment (i,j) at time step t

$$\sum_{i,j,t_j:(i,0,j,t_j)\in\mathscr{W}^*} w_{i,0,j,t_j} = 0 \qquad \forall i \in \mathscr{V}_{even}. \tag{71}$$

Similarly, the welding torch must arrive in some node $j \in \mathscr{V}_{odd}$ at time step T^{max}, while all edges to nodes of even degree have to be finished before:

$$\sum_{\substack{i,t_i,j\,:\,(i,t_i,j,T^{max})\in\mathscr{W}^* \\ j\in\mathscr{V}_{odd}}} w_{i,t_i,j,T^{max}} = 1, \tag{72}$$

$$\sum_{i,t_i,j:(i,t_i,j,T^{max})\in\mathscr{W}^*} w_{i,t_i,j,T^{max}} = 0 \qquad \forall j \in \mathscr{V}_{even}. \tag{73}$$

Each segment of the considered layer has to be welded in any direction:

$$\sum_{t_i,t_j:(i,t_i,j,t_j)\in\mathscr{W}^*} w_{i,t_i,j,t_j} + \sum_{t_j,t_i:(j,t_j,i,t_i)\in\mathscr{W}^*} w_{j,t_j,i,t_i} = 1 \qquad \forall\,(i,j)\in\mathscr{W}. \tag{74}$$

Since the number of nodes with odd degree is known, the number of necessary transition moves ω is given and should not be exceeded to avoid abrasion:

$$\sum_{(i,j,t)\in\mathscr{U}^*} u_{i,j,t} = \omega. \tag{75}$$

The resulting path has to be continuous, but for nodes $i\in\mathscr{V}_{odd}$ transition moves have to be taken into account:

$$\begin{aligned}&\sum_{h,t_h:(h,t_h,i,t)\in\mathscr{W}^*} w_{h,t_h,i,t} + \sum_{h:(h,i,t)\in\mathscr{U}^*} u_{h,i,t}\\ &\quad= \sum_{j,t_j:(i,t,j,t_j)\in\mathscr{W}^*} w_{i,t,j,t_j} + \sum_{j:(i,j,t)\in\mathscr{U}^*} u_{i,j,t}\\ &\quad\forall\,i\in\mathscr{V}_{odd},\ t\in\mathscr{T}\backslash\{T^{max}\},\end{aligned} \tag{76}$$

whereas for even nodes no transitions are possible:

$$\sum_{h,t_h:(h,t_h,i,t)\in\mathscr{W}^*} w_{h,t_h,i,t} = \sum_{j,t_j:(i,t,j,t_j)\in\mathscr{W}^*} w_{i,t,j,t_j} \qquad \forall\,i\in\mathscr{V}_{even},\ t\in\mathscr{T}\backslash\{T^{max}\}. \tag{77}$$

Since the number of transition moves is minimal due to (75), consecutive transitions would cause the path to be infeasible. So the following constraint is not necessary for feasibility, but reduces the computational effort by restricting the number of transitions per time step:

$$\sum_{i,j:(i,j,t)\in\mathscr{U}^*} u_{i,j,t} \le 1 \qquad \forall\,t\in\mathscr{T}. \tag{78}$$

3.2.2 Temperature calculation

To track the node temperatures we define variables $\theta^m_{i,t}\in\mathbb{R}_+$ for every node $i\in\mathscr{V}$ at time step $t\in\mathscr{T}_0$. To model conduction inside the workpiece, we apply the finite difference approach discussed in Sect. 3.1.1 by defining variables $\theta^{fd}_{i,j,k,t}\in\mathbb{R}_+$, describing the temperature of the k-th interior point $\left(k\in\mathscr{L}_{i,j}\right)$ of segment $(i,j)\in\mathscr{W}$ at time step $t\in\mathscr{T}_0$.

It is assumed that the temperature of every node $i\in\mathscr{V}$ and every interior point $k\in\mathscr{L}_{i,j}$ is equivalent to the initial ambient temperature ϕ^{amb}_0, until it is reached by the welding torch the first time. Since all edges have to be welded at the last time step, every node must be visited until then and thus the assumption has to hold for $t<T^{max}$ for all nodes. Similarly all interior points have to be visited until $t=T^{max}-1$, because the welding process ends in a node. To fix the node temperature to the ambient temperature, further binary variables $a^m_{i,t}$ and $a^{fd}_{i,j,k,t}$ are introduced. Thereby, $a^m_{i,t}$ is equal to 1 if node $i\in\mathscr{V}$ is visited not later than time step $t\in\mathscr{T}\backslash\{T^{max}\}$, and 0 otherwise.

Similiarly, $a^{fd}_{i,j,k,t}$ is equal to 1 if the k-th interior point, $k \in \mathscr{L}_{i,j}$, of segment $(i,j) \in \mathscr{W}$ is visited not later than time step $t \in \mathscr{T} \setminus \{T^{max}-1, T^{max}\}$. Using these variables and a sufficiently large constant M, this constraint can be expressed by

$$\theta^m_{i,t} \leq \phi^{amb}_0 + Ma^m_{i,t} \qquad \forall i \in \mathscr{V},\ t \in \{1, \dots, T^{max}-1\}, \tag{79}$$

$$\theta^m_{i,t} \geq \phi^{amb}_0 - Ma^m_{i,t} \qquad \forall i \in \mathscr{V},\ t \in \{1, \dots, T^{max}-1\} \tag{80}$$

for all nodes $i \in \mathscr{V}$ and

$$\theta^{fd}_{i,j,k,t} \leq \phi^{amb}_0 + Ma^{fd}_{i,j,k,t} \qquad \forall (i,j) \in \mathscr{W},\ k \in \mathscr{L}_{i,j},\ t \in \{1, \dots, T^{max}-2\}, \tag{81}$$

$$\theta^{fd}_{i,j,k,t} \geq \phi^{amb}_0 - Ma^{fd}_{i,j,k,t} \qquad \forall (i,j) \in \mathscr{W},\ k \in \mathscr{L}_{i,j},\ t \in \{1, \dots, T^{max}-2\} \tag{82}$$

for all interior points $k \in \mathscr{L}_{i,j}$ in a discretized edge. The binary variables $a^m_{i,t}$ and $a^{fd}_{i,j,k,t}$ are restricted by

$$a^m_{i,t} \leq \sum_{j,t_j : (i,t_i,j,t_j) \in \mathscr{W}^*} w_{i,t_i,j,t_j} + \sum_{\substack{h,t_h,t_i \,:\, (h,t_h,i,t_i) \in \mathscr{W}^* \\ t_i \leq t}} w_{h,t_h,i,t_i} + \sum_{\substack{h,t_i \,:\, (h,i,t_i) \in \mathscr{U}^* \\ t_i \leq t}} u_{h,i,t_i} \qquad \forall i \in \mathscr{V}_{odd},\ t \in \{1, \dots, T^{max}-1\}, \tag{83}$$

$$a^m_{i,t} \leq \sum_{\substack{h,t_h,t_i \,:\, (h,t_h,i,t_i) \in \mathscr{W}^* \\ t_i \leq t}} w_{h,t_h,i,t_i} \qquad \forall i \in \mathscr{V}_{even},\ t \in \{1, \dots, T^{max}-1\}, \tag{84}$$

$$a^m_{i,t} \geq \frac{1}{\frac{|\mathscr{M}(i)|}{2}+1} \left(\sum_{j,t_j : (i,t_i,j,t_j) \in \mathscr{W}^*} w_{i,t_i,j,t_j} + \sum_{\substack{h,t_h,t_i \,:\, (h,t_h,i,t_i) \in \mathscr{W}^* \\ t_i \leq t}} w_{h,t_h,i,t_i} + \sum_{\substack{h,t_i \,:\, (h,i,t_i) \in \mathscr{U}^* \\ t_i \leq t}} u_{h,i,t_i} \right) \qquad \forall i \in \mathscr{V}_{odd},\ t \in \{1, \dots, T^{max}-1\}, \tag{85}$$

$$a^m_{i,t} \geq \frac{2}{|\mathscr{M}(i)|} \sum_{\substack{h,t_h,t_i \,:\, (h,t_h,i,t_i) \in \mathscr{W}^* \\ t_i \leq t}} w_{h,t_h,i,t_i} \qquad \forall i \in \mathscr{V}_{even},\ t \in \{1, \dots, T^{max}-1\} \tag{86}$$

for all nodes $i \in \mathscr{V}$ and

$$a^{fd}_{i,j,k,t} = \sum_{\substack{i,t_i,t_j \,:\, (i,t_i,j,t_j)\in \mathscr{W}^* \\ t_i + k \le t}} w_{i,t_i,j,t_j} + \sum_{\substack{j,t_j,t_i \,:\, (j,t_j,i,t_i)\in \mathscr{W}^* \\ t_i - k \le t}} w_{j,t_j,i,t_i} \qquad \forall (i,j)\in \mathscr{W},\ k\in \mathscr{L}_{i,j},\ t\in\{1,\dots,T^{max}-2\} \tag{87}$$

for all interior points $k \in \mathscr{L}_{i,j}$.

The welding torch is assumed to heat up nodes and interior points with the constant temperature φ^{max}, limited by its weld pool energy, when it is passing by. Furthermore, it is not only heating up a single spot, but an area around its center assumed circular with radius r^w and all nodes within it. During the heating process the torch is centered over one node $i \in \mathscr{V}$. The distance to any node $j \in \mathscr{V}$ is calculated by the Euclidian distance $d^e_{i,j}$ of their positions. To model this area of effect it is split up into K^w rings with radius r_k, $0 = r_1 < r_2 < \cdots < r_{K^w} = r^w$ represented by intervals $\mathscr{P}_1 = \{0\}$ and $\mathscr{P}_k = (r_{k-1}, r_k]$ for $k \in \{2,\dots,K^w\}$. Numbers $\kappa^w_k \in [0,1]$, $1 \ge \kappa^w_1 \ge \kappa^w_2 \ge \cdots \ge \kappa^w_{K^w} \ge 0$ determine the partial heating of nodes relative to their position to the welding torch and its intensity.

At the beginning one node is used as the starting point of the welding source and thus is heated, all other nodes and interior points have the ambient temperature since they are not yet visited. Hence, only the interval $\mathscr{P}_1 = \{0\}$ has to be taken into account and the initial temperatures of all nodes and interior points are computed according to

$$\theta^m_{i,0} = \phi^{amb}_0 + \kappa^w_1 \varphi^{max} \sum_{j,t_j : (i,0,j,t_j)\in \mathscr{W}^*} w_{i,0,j,t_j} \qquad \forall\, i \in \mathscr{V}, \tag{88}$$

$$\theta^{fd}_{i,j,k,0} = \phi^{amb}_0 \qquad \forall (i,j)\in \mathscr{W},\ k \in \mathscr{L}_{i,j}, \tag{89}$$

where ϕ^{amb}_0 is the right hand side of (44).

From time step $t-1 \in \mathscr{T}_0$ to $t \in \mathscr{T}$ the heating process of a single node $i \in \mathscr{V}$ can be written as

$$\theta^m_{i,t} = \theta^m_{i,t-1} + \sum_{k=1}^{K^w} \kappa^w_k \varphi^{max} \left(\sum_{\substack{h,t_h,j \,:\, (h,t_h,j,t)\in \mathscr{W}^* \\ d^e_{i,j} \in \mathscr{P}_k}} w_{h,t_h,j,t} + \sum_{\substack{h,j \,:\, (h,j,t)\in \mathscr{U}^* \\ d^e_{i,j} \in \mathscr{P}_k}} u_{h,j,t} \right) \qquad \forall i \in \mathscr{V}_{odd},\ t\in\{1,\dots,T^{max}-1\}, \tag{90}$$

$$\theta^m_{i,t} = \theta^m_{i,t-1} + \sum_{k=1}^{K^w} \kappa^w_k \varphi^{max} \left(\sum_{\substack{h,t_h,j \,:\, (h,t_h,j,t) \in \mathscr{W}^* \\ d^e_{i,j} \in \mathscr{P}_k}} w_{h,t_h,j,t} \right) \quad \forall i \in \mathscr{V}_{even},\ t \in \{1,\dots,T^{max}-1\}, \tag{91}$$

$$\theta^m_{i,T^{max}} = \theta^m_{i,T^{max}-1} + \sum_{k=1}^{K^w} \kappa^w_k \varphi^{max} \left(\sum_{\substack{h,t_h,j \,:\, (h,t_h,j,T^{max}) \in \mathscr{W}^* \\ d^e_{i,j} \in \mathscr{P}_k}} w_{h,t_h,j,T^{max}} \right) \quad \forall i \in \mathscr{V}. \tag{92}$$

The radiation has to be computed for every node and interior point. Using the incremental method to approximate the power function, the temperature loss due to radiation (50) can be written as

$$\begin{aligned} \Delta\theta^{rad}_{i,t} =& \frac{2\sigma\epsilon\Delta t}{c\rho a^w} \left(\sum_{k=1}^{K^{pwl}} \delta_{i,t,k} (\boldsymbol{\Phi}^4_{k,1} - \boldsymbol{\Phi}^4_{k,0}) - (\phi^{amb}_t)^4 \right) \\ & - \frac{\sigma\epsilon\Delta t}{c\rho a^w} \sum_{j \in \mathscr{V}} F_{i\to j} \left(\sum_{k=1}^{K^{pwl}} \delta_{j,t,k} (\boldsymbol{\Phi}^4_{k,1} - \boldsymbol{\Phi}^4_{k,0}) - (\phi^{amb}_t)^4 \right) \\ & - \frac{\sigma\epsilon\Delta t}{c\rho a^w} \sum_{(g,j)\in\mathscr{W}} \sum_{h \in \mathscr{L}_{g,j}} F_{i\to gjh} \left(\sum_{k=1}^{K^{pwl}} \delta_{g,j,h,t,k} (\boldsymbol{\Phi}^4_{k,1} - \boldsymbol{\Phi}^4_{k,0}) - (\phi^{amb}_t)^4 \right) \\ & \forall i \in \mathscr{V},\ t \in \mathscr{T} \end{aligned} \tag{93}$$

for nodes $i \in \mathscr{V}$ and

$$\begin{aligned} \Delta\theta^{rad}_{i,j,h,t} =& \frac{2\sigma\epsilon\Delta t}{c\rho a^w} \left(\sum_{k=1}^{K^{pwl}} \delta_{i,j,h,t,k} (\boldsymbol{\Phi}^4_{k,1} - \boldsymbol{\Phi}^4_{k,0}) - (\phi^{amb}_t)^4 \right) \\ & - \frac{\sigma\epsilon\Delta t}{c\rho a^w} \sum_{g \in \mathscr{V}} F_{ijh\to g} \left(\sum_{k=1}^{K^{pwl}} \delta_{g,t,k} (\boldsymbol{\Phi}^4_{k,1} - \boldsymbol{\Phi}^4_{k,0}) - (\phi^{amb}_t)^4 \right) \\ & - \frac{\sigma\epsilon\Delta t}{c\rho a^w} \sum_{(f,g)\in\mathscr{W}} \sum_{m \in \mathscr{L}_{f,g}} F_{ijh\to fgm} \left(\sum_{k=1}^{K^{pwl}} \delta_{i,j,h,t,k} (\boldsymbol{\Phi}^4_{k,1} - \boldsymbol{\Phi}^4_{k,0}) - (\phi^{amb}_t)^4 \right) \\ & \forall (i,j) \in \mathscr{W},\ h \in \mathscr{L}_{i,j},\ t \in \mathscr{T} \end{aligned} \tag{94}$$

for interior points $k \in \mathscr{L}_{i,j}$ with ambient temperature φ_t^{amb} at time step t. Furthermore, $F_{i \rightarrow gjh}$ and $F_{gjh \rightarrow i}$ are describing the view factors from node $i \in \mathscr{V}$ to interior point $h \in \mathscr{L}_{g,j}$ and vice versa, while $F_{gjh_1 \rightarrow ikh_2}$ describes the view factor from one interior point $h_1 \in \mathscr{L}_{g,j}$ to $h_2 \in \mathscr{L}_{i,k}$.

The auxiliary variables $\delta_{i,t,k} \in [0, 1]$ determine the value of the piecewise linear function in the interval $k \in \{1, \dots, K^{pwl}\}$ for node $i \in \mathscr{V}$ at time step $t \in \mathscr{T}$ and $b_{i,t,k} \in \{0, 1\}$ select the active interval $k \in \{1, \dots, K^{pwl} - 1\}$. In the same way $\delta_{i,j,h,t,k} \in [0, 1]$, and $b_{i,j,h,t,k} \in \{0, 1\}$ are defined for the h-th interior point, $h \in \mathscr{L}_{i,j}$, of segment $(i, j) \in \mathscr{W}$.

Since the heating of a node is a process over time, not only the temperature of the last time step has to be taken into account to compute the radiation in (52), but also the additional temperature of a possible heating process. They are computed similarly to (52)–(54) by

$$\begin{aligned} \sum_{k=1}^{K^{pwl}} \delta_{i,t,k}(\Phi_{k,1} - \Phi_{k,0}) = \theta_{i,t-1}^m & \\ + \sum_{k=1}^{K^w} \kappa_k^w \varphi^{max} \left(\sum_{\substack{h,t_h,j:(h,t_h,j,t) \in \mathscr{W}^* \\ d_{i,j}^e \in \mathscr{P}_k}} w_{h,t_h,j,t} + \sum_{\substack{h,j:(h,j,t) \in \mathscr{U}^* \\ d_{i,j}^e \in \mathscr{P}_k}} u_{h,j,t} \right) & \\ \forall i \in \mathscr{V}_{odd},\ t \in \{1, \dots, T^{max} - 1\}, & \end{aligned} \tag{95}$$

$$\begin{aligned} \sum_{k=1}^{K^{pwl}} \delta_{i,t,k}(\Phi_{k,1} - \Phi_{k,0}) = \theta_{i,t-1}^m + \sum_{k=1}^{K^w} \kappa_k^w \varphi^{max} \left(\sum_{\substack{h,t_h,j:(h,t_h,j,t) \in \mathscr{W}^* \\ d_{i,j}^e \in \mathscr{P}_k}} w_{h,t_h,j,t} \right) & \\ \forall i \in \mathscr{V}_{even},\ t \in \{1, \dots, T^{max} - 1\}, & \end{aligned} \tag{96}$$

$$\begin{aligned} \sum_{k=1}^{K^{pwl}} \delta_{i,T^{max},k}(\Phi_{k,1} - \Phi_{k,0}) = \theta_{i,T^{max}-1}^m & \\ + \sum_{k=1}^{K^w} \kappa_k^w \varphi^{max} \left(\sum_{\substack{h,t_h,j:(h,t_h,j,T^{max}) \in \mathscr{W}^* \\ d_{i,j}^e \in \mathscr{P}_k}} w_{h,t_h,j,T^{max}} \right) & \\ \forall i \in \mathscr{V}, & \end{aligned} \tag{97}$$

$$b_{i,t,k} \leq \delta_{i,t,k} \qquad \forall i \in \mathscr{V},\ t \in \mathscr{T},\ k \in \left\{1, \dots, K^{pwl} - 1\right\}, \tag{98}$$

$$b_{i,t,k} \geq \delta_{i,t,k+1} \qquad \forall i \in \mathscr{V},\ t \in \mathscr{T},\ k \in \left\{1, \dots, K^{pwl} - 1\right\} \tag{99}$$

for every node $i \in \mathscr{V}$ and

$$\begin{aligned}&\sum_{k=1}^{K^{pwl}} \delta_{i,j,h,t,k}(\boldsymbol{\Phi}_{k,1} - \boldsymbol{\Phi}_{k,0})\\ &\quad = \theta^{fd}_{i,j,h,t-1} + \kappa^w_1 \varphi^{max}\left(\sum_{\substack{t_i, t_j \,:\, (i,t_i,j,t_j) \in \mathscr{W}^* \\ t_i + h = t}} w_{i,t_i,j,t_j} + \sum_{\substack{t_j, t_i \,:\, (j,t_j,i,t_i) \in \mathscr{W}^* \\ t_i - h = t}} w_{j,t_j,i,t_i}\right)\\ &\quad \forall (i,j) \in \mathscr{W},\ h \in \mathscr{L}_{i,j},\ t \in \mathscr{T},\end{aligned} \tag{100}$$

$$\begin{aligned} b_{i,j,h,t,k} \le \delta_{i,j,h,t,k} \qquad &\forall (i,j) \in \mathscr{W},\ h \in \mathscr{L}_{i,j},\\ k \in \left\{1, \dots, K^{pwl} - 1\right\},\ &t \in \mathscr{T}, \end{aligned} \tag{101}$$

$$\begin{aligned} b_{i,j,h,t,k} \ge \delta_{i,j,h,t,k+1} \qquad &\forall (i,j) \in \mathscr{W},\ h \in \mathscr{L}_{i,j},\\ k \in \left\{1, \dots, K^{pwl} - 1\right\},\ &t \in \mathscr{T} \end{aligned} \tag{102}$$

for the interior points $k \in \mathscr{L}_{i,j}$.

To compute the temperature of every node, the heating process (90)–(92) is combined with the radiation term (93) and the approximation of the substrate plate (35). Then the constraints (79) and (80) are incorporated, considering if the node has been visited in the past. Thus, these constraints are written as

$$\begin{aligned}\theta^m_{i,t} \le &\theta^m_{i,t-1} - \Delta\theta^{rad}_{i,t} - \Delta\theta^p_{i,t} + M\left(1 - a^m_{i,t}\right)\\ &+ \sum_{k=1}^{K^w} \kappa^w_k \varphi^{max}\left(\sum_{\substack{h, t_h, j \,:\, (h,t_h,j,t) \in \mathscr{W}^* \\ d^e_{i,j} \in \mathscr{P}_k}} w_{h,t_h,j,t} + \sum_{\substack{h, j \,:\, (h,j,t) \in \mathscr{U}^* \\ d^e_{i,j} \in \mathscr{P}_k}} u_{h,j,t}\right)\\ &\quad \forall\, i \in \mathscr{V}_{odd},\ t \in \{1, \dots, T^{max} - 1\},\end{aligned} \tag{103}$$

$$\begin{aligned}\theta^m_{i,t} \ge &\theta^m_{i,t-1} - \Delta\theta^{rad}_{i,t} - \Delta\theta^p_{i,t} - M\left(1 - a^m_{i,t}\right)\\ &+ \sum_{k=1}^{K^w} \kappa^w_k \varphi^{max}\left(\sum_{\substack{h, t_h, j \,:\, (h,t_h,j,t) \in \mathscr{W}^* \\ d^e_{i,j} \in \mathscr{P}_k}} w_{h,t_h,j,t} + \sum_{\substack{h, j \,:\, (h,j,t) \in \mathscr{U}^* \\ d^e_{i,j} \in \mathscr{P}_k}} u_{h,j,t}\right)\\ &\quad \forall\, i \in \mathscr{V}_{odd},\ t \in \{1, \dots, T^{max} - 1\},\end{aligned} \tag{104}$$

$$\begin{aligned} \theta^m_{i,t} \leq & \theta^m_{i,t-1} - \Delta\theta^{rad}_{i,t} - \Delta\theta^p_{i,t} + M\left(1 - a^m_{i,t}\right) \\ & + \sum_{k=1}^{K^w} \kappa^w_k \varphi^{max} \left(\sum_{\substack{h, t_h, j \,:\, (h, t_h, j, t) \in \mathscr{W}^* \\ d^e_{i,j} \in \mathscr{P}_k}} w_{h,t_h,j,t} \right) \\ & \forall\, i \in \mathscr{V}_{even},\ t \in \{1, \dots, T^{max} - 1\}, \end{aligned} \tag{105}$$

$$\begin{aligned} \theta^m_{i,t} \geq & \theta^m_{i,t-1} - \Delta\theta^{rad}_{i,t} - \Delta\theta^p_{i,t} - M\left(1 - a^m_{i,t}\right) \\ & + \sum_{k=1}^{K^w} \kappa^w_k \varphi^{max} \left(\sum_{\substack{h, t_h, j \,:\, (h, t_h, j, t) \in \mathscr{W}^* \\ d^e_{i,j} \in \mathscr{P}_k}} w_{h,t_h,j,t} \right) \\ & \forall\, i \in \mathscr{V}_{even},\ t \in \{1, \dots, T^{max} - 1\}, \end{aligned} \tag{106}$$

$$\begin{aligned} \theta^m_{i,T^{max}} = & \theta^m_{i,T^{max}-1} - \Delta\theta^{rad}_{i,T^{max}} - \Delta\theta^p_{i,T^{max}} \\ & + \sum_{k=1}^{K^w} \kappa^w_k \varphi^{max} \left(\sum_{\substack{h, t_h, j \,:\, (h, t_h, j, T^{max}) \in \mathscr{W}^* \\ d^e_{i,j} \in \mathscr{P}_k}} w_{h,t_h,j,T^{max}} \right) \\ & \forall\, i \in \mathscr{V}. \end{aligned} \tag{107}$$

To model the heat transmission due to conduction within the workpiece we combine our customized BTCS scheme (41), with the welding torch taken as heat source $f_{i,t}$, with (94) and again incorporate the constraints (81) and (82), thus leading to

$$\begin{aligned} \theta^{fd}_{i,j,k,t-1} \leq & - \tilde{\alpha}_{i,j}\theta^{fd}_{i,j,k-1,t} + (1 + 2\tilde{\alpha}_{i,j})\theta^{fd}_{i,j,k,t} - \tilde{\alpha}_{i,j}\theta^{fd}_{i,j,k+1,t} \\ & + \Delta\theta^{rad}_{i,j,k,t} + M\left(1 - a^{fd}_{i,j,k,t}\right) \\ & - \Delta t \kappa^w_1 \varphi^{max} \left(\sum_{\substack{t_i, t_j \,:\, (i, t_i, j, t_j) \in \mathscr{W}^* \\ t_i + h = t}} w_{i,t_i,j,t_j} + \sum_{\substack{t_j, t_i \,:\, (j, t_j, i, t_i) \in \mathscr{W}^* \\ t_i - h = t}} w_{j,t_j,i,t_i} \right) \\ & \forall (i,j) \in \mathscr{W},\ k \in \mathscr{L}_{i,j},\ t \in \{1, \dots, T^{max} - 2\}, \end{aligned} \tag{108}$$

$$\begin{aligned}\theta^{fd}_{i,j,k,t-1} \geq &- \tilde{\alpha}_{i,j}\theta^{fd}_{i,j,k-1,t} + (1+2\tilde{\alpha}_{i,j})\theta^{fd}_{i,j,k,t} - \tilde{\alpha}_{i,j}\theta^{fd}_{i,j,k+1,t}\\ &+ \Delta\theta^{rad}_{i,j,k,t} - M\left(1 - a^{fd}_{i,j,k,t}\right)\\ &- \Delta t\kappa^{w}_{1}\varphi^{max}\left(\sum_{\substack{t_i,t_j\,:\,(i,t_i,j,t_j)\in\mathscr{W}^*\\ t_i+h=t}} w_{i,t_i,j,t_j} + \sum_{\substack{t_j,t_i\,:\,(j,t_j,i,t_i)\in\mathscr{W}^*\\ t_i-h=t}} w_{j,t_j,i,t_i}\right)\\ &\forall (i,j)\in\mathscr{W},\ k\in\mathscr{L}_{i,j},\ t\in\{1,\dots,T^{max}-2\},\end{aligned} \tag{109}$$

$$\begin{aligned}\theta^{fd}_{i,j,k,t-1} = &- \tilde{\alpha}_{i,j}\theta^{fd}_{i,j,k-1,t} + (1+2\tilde{\alpha}_{i,j})\theta^{fd}_{i,j,k,t} - \tilde{\alpha}_{i,j}\theta^{fd}_{i,j,k+1,t} + \Delta\theta^{rad}_{i,j,k,t}\\ &- \Delta t\kappa^{w}_{1}\varphi^{max}\left(\sum_{\substack{t_i,t_j\,:\,(i,t_i,j,t_j)\in\mathscr{W}^*\\ t_i+h=t}} w_{i,t_i,j,t_j} + \sum_{\substack{t_j,t_i\,:\,(j,t_j,i,t_i)\in\mathscr{W}^*\\ t_i-h=t}} w_{j,t_j,i,t_i}\right)\\ &\forall (i,j)\in\mathscr{W},\ k\in\mathscr{L}_{i,j},\ t\in\{T^{max}-1,T^{max}\}.\end{aligned} \tag{110}$$

Furthermore, all segments are connected by their shared boundary points, represented by the nodes, i.e.

$$\theta^{fd}_{i,j,0,t} = \theta^{m}_{i,t} \qquad \forall (i,j)\in\mathscr{W},\ t\in\mathscr{T}_0, \tag{111}$$

$$\theta^{fd}_{i,j,N^{int}_{i,j}+1,t} = \theta^{m}_{j,t} \qquad \forall (i,j)\in\mathscr{W},\ t\in\mathscr{T}_0. \tag{112}$$

Taking the constraints (79)–(89) and (95)–(110) into account, the whole temperature distribution within the workpiece can be calculated. However, the proposed model is very complex on the basis that for every node and interior discretization point the piecewise approximation for radiation is done at every time step. Consequently, this approach leads to very high computational costs. In the following, the model using this temperature calculation is referred to as ($WAAM_{pwl}$).

As an alternative approach to reduce the model complexity due to the calculation of radiation, we take the approximation based on the Rosseland equation, presented in Sect. 3.1.2. Here we use for the cooling process of a node a factor $\kappa^e \in (0,1)$ for heat loss to the environment, while heat exchange between nodes is not considered. We extend this approach by incorporating an additive constant $\varphi^{rad}_{n_l} \geq 0$ representing the incoming radiation to any node in layer n_l. Thus the variables b and δ, the constraints (95)–(102) and the radiation terms (93) and (94) can be dropped and the temperature calculation (103)–(110) can be substituted with

$$\begin{aligned}\theta^{m}_{i,t} \leq & \kappa^{e}\theta^{m}_{i,t-1} + \varphi^{rad}_{n_l} - \Delta\theta^{p}_{i,t} + M\left(1 - a^{m}_{i,t}\right) \\ & + \sum_{k=1}^{K^w} \kappa^{w}_{k}\varphi^{max}\left(\sum_{\substack{h,t_h,j\,:\,(h,t_h,j,t)\in\mathscr{W}^* \\ d^{e}_{i,j}\in\mathscr{P}_k}} w_{h,t_h,j,t} + \sum_{\substack{h,j\,:\,(h,j,t)\in\mathscr{U}^* \\ d^{e}_{i,j}\in\mathscr{P}_k}} u_{h,j,t} \right) \\ & \forall\, i \in \mathscr{V}_{odd},\; t \in \{1,\dots,T^{max}-1\},\end{aligned} \tag{113}$$

$$\begin{aligned}\theta^{m}_{i,t} \geq & \kappa^{e}\theta^{m}_{i,t-1} + \varphi^{rad}_{n_l} - \Delta\theta^{p}_{i,t} - M\left(1 - a^{m}_{i,t}\right) \\ & + \sum_{k=1}^{K^w} \kappa^{w}_{k}\varphi^{max}\left(\sum_{\substack{h,t_h,j\,:\,(h,t_h,j,t)\in\mathscr{W}^* \\ d^{e}_{i,j}\in\mathscr{P}_k}} w_{h,t_h,j,t} + \sum_{\substack{h,j\,:\,(h,j,t)\in\mathscr{U}^* \\ d^{e}_{i,j}\in\mathscr{P}_k}} u_{h,j,t} \right) \\ & \forall\, i \in \mathscr{V}_{odd},\; t \in \{1,\dots,T^{max}-1\},\end{aligned} \tag{114}$$

$$\begin{aligned}\theta^{m}_{i,t} \leq & \kappa^{e}\theta^{m}_{i,t-1} + \varphi^{rad}_{n_l} - \Delta\theta^{p}_{i,t} + M\left(1 - a^{m}_{i,t}\right) \\ & + \sum_{k=1}^{K^w} \kappa^{w}_{k}\varphi^{max}\left(\sum_{\substack{h,t_h,j\,:\,(h,t_h,j,t)\in\mathscr{W}^* \\ d^{e}_{i,j}\in\mathscr{P}_k}} w_{h,t_h,j,t} \right) \\ & \forall\, i \in \mathscr{V}_{even},\; t \in \{1,\dots,T^{max}-1\},\end{aligned} \tag{115}$$

$$\begin{aligned}\theta^{m}_{i,t} \geq & \kappa^{e}\theta^{m}_{i,t-1} + \varphi^{rad}_{n_l} - \Delta\theta^{p}_{i,t} - M\left(1 - a^{m}_{i,t}\right) \\ & + \sum_{k=1}^{K^w} \kappa^{w}_{k}\varphi^{max}\left(\sum_{\substack{h,t_h,j\,:\,(h,t_h,j,t)\in\mathscr{W}^* \\ d^{e}_{i,j}\in\mathscr{P}_k}} w_{h,t_h,j,t} \right) \\ & \forall\, i \in \mathscr{V}_{even},\; t \in \{1,\dots,T^{max}-1\},\end{aligned} \tag{116}$$

$$\begin{aligned} \theta^{m}_{i,T^{max}} =& \kappa^{e}\theta^{m}_{i,T^{max}-1} + \varphi^{rad}_{n_l} - \Delta\theta^{p}_{i,T^{max}} \\ & + \sum_{k=1}^{K^w} \kappa^{w}_{k}\varphi^{max}\left(\sum_{\substack{h,t_h,j\,:\,(h,t_h,j,T^{max})\in\mathscr{W}^* \\ d^{e}_{i,j}\in\mathscr{P}_k}} w_{h,t_h,j,T^{max}} \right) \\ & \forall\, i\in\mathscr{V} \end{aligned} \tag{117}$$

and

$$\begin{aligned} \kappa^{e}\theta^{fd}_{i,j,k,t-1} + \varphi^{rad}_{n_l} \leq & -\tilde{\alpha}_{i,j}\theta^{fd}_{i,j,k-1,t} + (1+2\tilde{\alpha}_{i,j})\theta^{fd}_{i,j,k,t} \\ & - \tilde{\alpha}_{i,j}\theta^{fd}_{i,j,k+1,t} + M\left(1 - a^{fd}_{i,j,k,t}\right) \\ & - \Delta t\kappa^{w}_{1}\varphi^{max}\left(\sum_{\substack{t_i,t_j\,:\,(i,t_i,j,t_j)\in\mathscr{W}^* \\ t_i+h=t}} w_{i,t_i,j,t_j} + \sum_{\substack{t_j,t_i\,:\,(j,t_j,i,t_i)\in\mathscr{W}^* \\ t_i-h=t}} w_{j,t_j,i,t_i} \right) \\ & \forall (i,j)\in\mathscr{W},\ k\in\mathscr{L}_{i,j},\ t\in\{1,\dots,T^{max}-2\}, \end{aligned} \tag{118}$$

$$\begin{aligned} \kappa^{e}\theta^{fd}_{i,j,k,t-1} + \varphi^{rad}_{n_l} \geq & -\tilde{\alpha}_{i,j}\theta^{fd}_{i,j,k-1,t} + (1+2\tilde{\alpha}_{i,j})\theta^{fd}_{i,j,k,t} \\ & - \tilde{\alpha}_{i,j}\theta^{fd}_{i,j,k+1,t} - M\left(1 - a^{fd}_{i,j,k,t}\right) \\ & - \Delta t\kappa^{w}_{1}\varphi^{max}\left(\sum_{\substack{t_i,t_j\,:\,(i,t_i,j,t_j)\in\mathscr{W}^* \\ t_i+h=t}} w_{i,t_i,j,t_j} + \sum_{\substack{t_j,t_i\,:\,(j,t_j,i,t_i)\in\mathscr{W}^* \\ t_i-h=t}} w_{j,t_j,i,t_i} \right) \\ & (i,j)\in\mathscr{W},\ k\in\mathscr{L}_{i,j},\ t\in\{1,\dots,T^{max}-2\}, \end{aligned} \tag{119}$$

$$\begin{aligned} \kappa^{e}\theta^{fd}_{i,j,k,t-1} + \varphi^{rad}_{n_l} = & -\tilde{\alpha}_{i,j}\theta^{fd}_{i,j,k-1,t} + (1+2\tilde{\alpha}_{i,j})\theta^{fd}_{i,j,k,t} - \tilde{\alpha}_{i,j}\theta^{fd}_{i,j,k+1,t} \\ & - \Delta t\kappa^{w}_{1}\varphi^{max}\left(\sum_{\substack{t_i,t_j\,:\,(i,t_i,j,t_j)\in\mathscr{W}^* \\ t_i+h=t}} w_{i,t_i,j,t_j} + \sum_{\substack{t_j,t_i\,:\,(j,t_j,i,t_i)\in\mathscr{W}^* \\ t_i-h=t}} w_{j,t_j,i,t_i} \right) \\ & (i,j)\in\mathscr{W},\ k\in\mathscr{L}_{i,j},\ t\in\{T^{max}-1,T^{max}\}. \end{aligned} \tag{120}$$

Using the constraints (79)–(89) and (113)–(120), the calculation of the temperature within the workpiece is less computationally intensive compared to ($WAAM_{pwl}$), since there are no additional variables and constraints for the computation of the radiation. Moreover, the constraints are modeled in an easier way. As a result, the quality of the simplified model is low, e.g. the cooling factor is not dependent on the temperature and the increase of temperature due to heating can not be distributed at the same time step, but only at the next. In the following, the model with the above temperature calculation is referred to as ($WAAM_{\kappa}$).

To reduce the complexity of ($WAAM_{pwl}$) and the drawbacks of ($WAAM_{\kappa}$), we combine both models into another alternative approach. Therefore, the temperature calculation for all nodes is done by the piecewise linear approximation of the radiation and for all interior points a factor κ^e combined with an additive constant $\varphi^{rad}_{n_l}$ is used. Then only the radiation term (93) has to be changed to

$$\begin{aligned}\Delta\theta^{rad}_{i,t} =& \frac{2\sigma\epsilon\Delta t}{c\rho a^w}\left(\sum_{k=1}^{K^{pwl}} \delta_{i,t,k}(\Phi^4_{k,1} - \Phi^4_{k,0}) - (\phi^{amb}_t)^4\right)\\ &- \frac{\sigma\epsilon\Delta t}{c\rho a^w}\sum_{j\in\mathcal{V}} F_{i\to j}\left(\sum_{k=1}^{K^{pwl}} \delta_{j,t,k}(\Phi^4_{k,1} - \Phi^4_{k,0}) - (\phi^{amb}_t)^4\right) \quad \forall i \in \mathcal{V},\ t \in \mathcal{T}.\end{aligned} \tag{121}$$

Finally, the node temperatures are calculated as in the first approach, i.e. according to (95)–(99) and (103)–(107) with the changed radiation term (121). For interior points the second approach is used, i.e. (118)–(120). This procedure may be advantageous because the number of binary variables necessary for the approximation of radiation is now independent of the size of the workpiece and the computation of the radiation for nodes is more detailed. In the following, this hybrid model is referred to as ($WAAM_{hyb}$).

3.2.3 Objective

The processing time and the quality of the workpiece are crucial factors for the performance of manufacturing processes. Since the minimal time to build all segments of one layer can be computed by (27) and the number of transition moves is limited by (75), there is no need to take time aspects into account in the objective function. The quality depends greatly on the welding sequence because material stresses are mainly caused by temperature differences within the workpiece. Hence, the trajectory optimization towards a homogeneous temperature distribution is desirable.

Local temperature gradients within the material cause expansion or contraction, so the resulting material deformation can be used to measure stresses. Young's modulus E is a constant for material deformation depending on the considered material. It describes the amount of stress that is required to cause a certain change in length.

According to Carter et al. (1991) it can be calculated by

$$E = \frac{\sigma^{th}}{\epsilon}, \tag{122}$$

with thermal stress σ^{th} and strain $\epsilon = \frac{\Delta l}{l}$, representing the relative expansion of the material of length l. Substituting the strain by $\frac{\Delta l}{l} = \alpha^* \Delta T$ with heat expansion coefficient α^* and temperature gradient ΔT, we compute the thermal stress through

$$\sigma^{th} = E\alpha^* \Delta T. \tag{123}$$

Thus, reduction of possible stress sources can be done by minimizing temperature gradients in the workpiece. This is modeled by

$$\min \sum_{t \in \mathcal{T}_0} \sum_{(i,j) \in \mathcal{W}} |\theta^m_{i,t} - \theta^m_{j,t}|. \tag{124}$$

To obtain a linear objective function, we introduce new auxiliary variables $\vartheta^+_{i,j,t}, \vartheta^-_{i,j,t} \in \mathbb{R}_+$, tracking the absolute value of the positive or negative temperature gradient of segment $(i,j) \in \mathcal{W}$ at time step $t \in \mathcal{T}_0$ and replace (124) with

$$\min \sum_{t \in \mathcal{T}_0} \sum_{(i,j) \in \mathcal{W}} \vartheta^+_{i,j,t} + \vartheta^-_{i,j,t}. \tag{125}$$

Furthermore, a new constraint

$$\theta^m_{i,t} - \theta^m_{j,t} = \vartheta^+_{i,j,t} - \vartheta^-_{i,j,t} \qquad \forall (i,j) \in \mathcal{W},\ t \in \mathcal{T}_0 \tag{126}$$

has to be added to ensure the correct calculation of the objective.

3.2.4 Computation of higher layers

A considered workpiece will in general not consist of only a single layer. So the presented model should be capable of computing the temperature distribution of an arbitrarily chosen layer. Several differences are affecting the model structure between the computation for the first and any following layer.

Considering the first layer of a given workpiece, then the constraints derived in Sects. 3.2.1 and 3.2.2 have to be used with $n_l = 1$.

Regarding the computation for the second or a higher layer, the constraints derived in Sect. 3.2.1 remain unchanged, but several changes in the temperature calculation are necessary. At first, the parameter n_l has to be adjusted to its correct value. Furthermore, since all nodes are already visited at least once, the variables $a^m_{i,t}$ and $a^{fd}_{i,j,k,t}$ are obsolete and have to be neglected. Thus the constraints (79)–(87) can be dropped and in the constraints (103)–(106),(108)–(109), (113)–(116), (118), and (119) the respective term $M\left(1 - a^m_{i,t}\right)$ or $M\left(1 - a^{fd}_{i,j,k,t}\right)$ vanishes and each pair of corresponding inequalities, e.g., (103) and (104), can be combined to one equality constraint.

The initial temperature of all nodes and interior points is given by the final temperature of the corresponding node or interior point in the previously finished layer. This data is stored in new parameters φ_i^{init} and $\varphi_{i,j,k}^{init}$ for all nodes $i \in \mathscr{V}$ and interior points $k \in \mathscr{L}_{i,j}$ of every segment $(i,j) \in \mathscr{W}$. The constraints (88) and (89) are then substituted by

$$\theta_{i,0}^{m} = \varphi_i^{init} \qquad \forall\, i \in \mathscr{V}, \tag{127}$$

$$\theta_{i,j,k,0}^{fd} = \varphi_{i,j,k}^{init} \qquad \forall (i,j) \in \mathscr{W},\ k \in \mathscr{L}_{i,j}. \tag{128}$$

3.3 Input data

In the following, we work with the models ($WAAM_{pwl}$), ($WAAM_{\kappa}$), and ($WAAM_{hyb}$) derived in Sect. 3.2.2. The first one approximates radiation by a piecewise linear function using the incremental method for all nodes and interior points. The second one uses a factor and an additive constant to represent radiation, while the third one combines the piecewise linear approximation of radiation by the incremental method for nodes with the computation of radiation by a factor and an additive constant for all interior points.

All models share the objective (125), the constraints (70)–(78), (79)–(89), (111)–(112), and (126), binary conditions on the variables w, u, a, and non-negativity constraints on the variables θ^m, θ^{fd}, ϑ^+, ϑ^-. The difference within the models is the calculation of the temperature, where ($WAAM_{pwl}$) is subjected to the constraints (95)–(110) and additionally included binary conditions for b. In contrast, the second model ($WAAM_{\kappa}$) uses the constraints (113)–(120), while the binary constraints for b are not necessary. The model ($WAAM_{hyb}$) calculates the temperature by (95)–(99) and (103)–(107) with radiation term (121) and (118)–(120). Here binary conditions for b are required again.

The resulting mixed-integer linear programming problems are considering the first layer of a workpiece and thus it holds $n_l = 1$. Further, we assume the ambient temperature $\varphi_t^{amb} = 20^{\circ}\mathrm{C}\,(t \in \mathscr{T}_0)$ and the length of one time step $\Delta t = 0.5\mathrm{s}$.

3.3.1 Welding source

The heat distribution of the welding torch can be modeled using distribution functions, e.g.

$$q^w(x,y,z) = \frac{2Q^w}{\pi\sqrt{\pi}a^w b^w c^w}\exp^{-\left(\frac{x}{a^w}\right)^2}\exp^{-\left(\frac{y}{b^w}\right)^2}\exp^{-\left(\frac{z}{c^w}\right)^2}, \qquad (x,y,z) \in \mathbb{R}^3, \tag{129}$$

derived by Goldak et al. (1984) based on a normal distribution, which is used here. It describes an ellipsoidal three dimensional heat source, where a^w, b^w, and c^w are the weld pool widths in x, y, and z direction, respectively, and Q^w is the weld pool energy. Splitting the ellipsoid into a front and a rear half and choosing different values for c_f^w and c_r^w, a different heat distribution for every half of the ellipsoidal heat

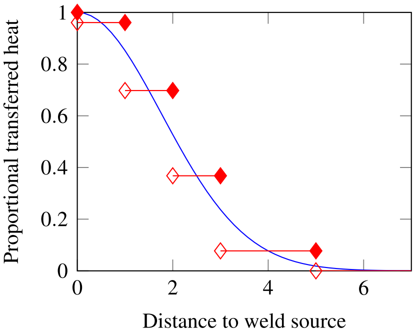


Fig. 9 Normalized heat distribution function *(blue)* for the welding source and its piecewise constant approximation *(red)*. (Color figure online)

Table 3 Weld source parameters for a Fronius TPS/i 500 welding device

Weld pool width	a^w	2.25 mm
Weld pool depth	b^w	1.60 mm
Forward weld pool	c_f^w	2.50 mm
Rearward weld pool	c_r^w	5.00 mm
Weld pool energy	Q^w	1840 W
Heat source velocity	v^w	7.5 mm/s

source can be computed. Because a single layer is considered and the weld pool is assumed to be circular, it holds $y = 0$ and the weld pool widths a^w, c_f^w, and c_r^w can be substituted by the radius $r^w = \max\{a^w, c_f^w, c_r^w\}$. By substitution of the position $\tilde{x}^2 = x^2 + y^2 + z^2 = x^2 + z^2$ by its distance to the origin, the distribution function can be rewritten as

$$q^w(\tilde{x}) = \frac{2Q^w}{\pi\sqrt{\pi}b^w(r^w)^2} \exp^{-\left(\frac{\tilde{x}}{r^w}\right)^2}, \qquad \tilde{x} \in \mathbb{R}. \tag{130}$$

The factors κ_k^w ,$k = 2, \dots, K^w$, for the intensity of the welding source are chosen as $\kappa_k^w = \frac{1}{q^w(0)} q^w\left(\frac{r_{k-1}+r_k}{2}\right)$ in any interval $\mathscr{P}_k = \left(r_{k-1}, r_k\right]$, while $\kappa_1^w = 1$. The normalized heat distribution function for the welding source $\frac{q^w(\tilde{x})}{q^w(0)}$ and its piecewise constant approximation are shown in Fig. 9.

The parameters related to the welding source are chosen according to the welding device Fronius TPS 500i to allow the comparison of our results to a real-world welding machine. These parameters are displayed in Table 3.

3.3.2 Piecewise linear function

The accuracy of the piecewise linear approximation of the power function in the radiation term is crucial for the heat transfer due to radiation. Therefore, the number of subintervals and their length must be chosen appropriately. The incremental method requires $\Phi_{1,0} = 0$ and $\Phi_{k-1,1} = \Phi_{k,0}$ ($k = 2, \dots, K^{pwl}$), and $\Phi_{K^{pwl},1}$ must be

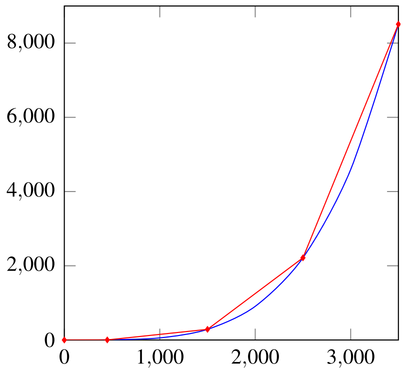


Fig. 10 Power function $f(x) = \sigma x^4$ and its piecewise linear approximation. (Color figure online)

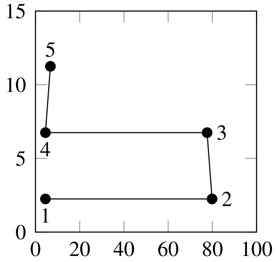


Fig. 11 Test instance for parameter estimation (in mm)

large enough to make sure that the problem is feasible. Otherwise, the temperature that should be approximated can not be handled by $\delta \in [0, 1]$.

In terms of radiation there is some point where the temperature loss is greater than the former temperature and its increase by the heating process. Taking the value $\Phi_{K^{pwl},1} = 3500\,°\mathrm{C}$ with this property, it has proven to be sufficiently large in all computations. Thus, we opt for the interval from 0 to 3500 °C and divide it into $K^{pwl} = 4$ subintervals with the breakpoints 450 °C, 1500 °C and 2500 °C as displayed in Fig. 10.

3.3.3 Material data

Since many of the parameters related to the used material are dependent on the temperature, like the specific heat capacity or the thermal diffusivity, their use would cause non-linearities within our models. To avoid this, we perform a parameter estimation to compute the best average values of these parameters for every model.

Therefore, we consider a small example displayed in Fig. 11 consisting of $|\mathscr{V}| = 5$ nodes and $|\mathscr{W}| = 4$ edges. For this workpiece, we fix the tool-path and compute the temperature progression using simulation software. Then this data is taken as reference and the model parameters are estimated to approximate this simulated temperature distribution as good as possible.

In ($WAAM_{pwl}$) the parameters φ^{max}, $\tilde{\alpha}$, and ρ have to be estimated. Since the temperature gain due to heating in ($WAAM_{\kappa}$) and ($WAAM_{hyb}$) cannot change at the same time step (no node or interior point is heated), the value of the parameter φ^{max} is directly given from the provided data. Hence, in ($WAAM_{\kappa}$) the parameters κ_e, $\tilde{\alpha}$, and $\varphi^{rad}_{n_l}$ have to be estimated, while in ($WAAM_{hyb}$) we require values for κ_e, $\tilde{\alpha}$, $\varphi^{rad}_{n_l}$, and ρ.

The given data for the temperature $\theta^{sim}_{i,t}$ of all nodes $i \in \mathscr{V} = \{1, \dots, 5\}$ at time step $t \in \mathscr{T}_0$ with $T^{max} = 43$ is computed using the simulation tool LS-DYNA R11.0.0 MSP. Basic information about the AM-modeling technique with death–birth elements in the simulation environment of LS-DYNA can be found in Israr et al. (2018). Details like activation of the inactive thermal and mechanical material properties, as well as convergence analysis for coupled and uncoupled, pure implicit/explicit and hybrid solvers, estimations of computation time, time step and mesh size can be seen in Buhl et al. (2019). In this paper the optimal parameters are used.

To ensure the quality of the estimated parameters, the computed temperature via LS-DYNA should have a minimal difference to the simulated temperature obtained through the presented models. Thus, the objective (125) of all models for minimizing temperature gradients is substituted by the new objective function

$$\min \sum_{t \in \mathscr{T}_0} \sum_{i \in \mathscr{V}} (\theta^{sim}_{i,t} - \theta^{m}_{i,t})^2, \tag{131}$$

to minimize the quadratic error between the simulated temperature $\theta^{sim}_{i,t}$ and the computed temperature $\theta^{m}_{i,t}$ for all nodes $i \in \mathscr{V}$ at time step $t \in \mathscr{T}_0$. According to this, constraint (126) can be dropped.

By adding the above-mentioned parameters for each model to its variables, we obtain a non-linear model for ($WAAM_{\kappa}$) and a non-linear mixed-integer model for ($WAAM_{pwl}$) as well as for ($WAAM_{hyb}$).

All models are solved using BARON 19.7.13 (Kılınç and Sahinidis 2018) (time limit 80,000 s) on a MacBookPro with an Intel Core i7 running 8 threads parallel at 3.1 GHz clock speed and 16 GB RAM.

Given this time limit, the solver was able to find a feasible solution for ($WAAM_{\kappa}$) with a gap of 69.68%. For ($WAAM_{pwl}$) and ($WAAM_{hyb}$) no feasible solution was found within the time limit. To achieve comparable parameter values for these models, we choose a possible range for every of the required parameters, based on the material data given in Holman (2009) for ρ and $\tilde{\alpha}$ and former experiments on the others. Dividing these ranges into subintervals result in 3,325 instances for ($WAAM_{pwl}$) and 44,352 instances for ($WAAM_{hyb}$). Solving all instances with the associated model, the objective values in Fig. 12 were obtained and the parameter values of the best solutions (marked by a red star) were chosen. Compared to the simulation data, the maximum and average relative approximation error is 83.3%, respectively 12.72% for ($WAAM_{pwl}$) and 83.3%, respectively 12.67% for ($WAAM_{hyb}$).

The maximum relative approximation error is obtained at timestep 43 at the fifth node where the simulated temperature already increases, but the temperature computed by our models is still bounded to the ambient temperature, since the fifth node is only visited at the last timestep. By repeating the error computation with only those timesteps where the respective node is visited, the maximum relative approximation error reduces to 55.45% for ($WAAM_{pwl}$) and 53.76% for ($WAAM_{hyb}$), while the

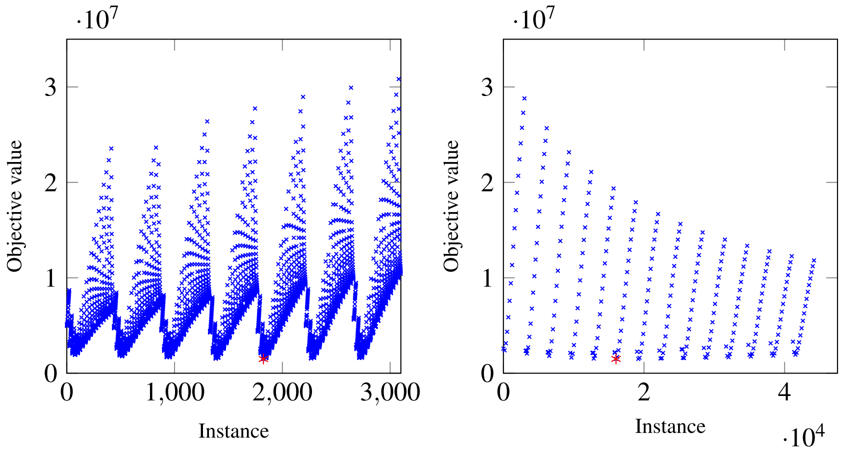


Fig. 12 Objective values of calculated subinterval combinations for ($WAAM_{pwl}$) *(left)* and ($WAAM_{hyb}$) *(right)* applied to the test instance, cf. Fig. 11. (Color figure online)

average relative approximation error increases to 17.01% for ($WAAM_{pwl}$) and 16.89% for ($WAAM_{hyb}$).

Surprisingly, it turns out that in the case of ($WAAM_{hyb}$) the radiation of all interior points does not affect the objective value, resulting in the same objective value for all instances, where only the parameters of the interior points are changing.

In the following, we take the estimated parameter values for κ^e and φ^{rad} of ($WAAM_\kappa$) also for the model ($WAAM_{hyb}$). For all models the estimated parameters as well as the given input data for the material, are reported in Table 4. The temperature approximation of the simulated data for the three models is displayed in Fig. 13.

Table 4 Estimated parameters (upper part) and given data (lower part) related to the material

Parameter	($WAAM_{pwl}$)	($WAAM_\kappa$)	($WAAM_{hyb}$)
$\varphi^{max}(\,^\circ\mathrm{C})$	1500	1500	1500
$\varphi^{rad}(\,^\circ\mathrm{C})$	–	46.0496	46.0496
κ_e	–	0.8315	0.8315
$\alpha\left(\frac{\mathrm{mm}^2}{\mathrm{s}}\right)$	6	3.5315	6
$\rho\left(\frac{\mathrm{g}}{\mathrm{mm}^3}\right)$	0.002	–	0.002
$\varphi^{init}(\,^\circ C)$	1520	1520	1520
$D(\mathrm{mm}^2)$	1500	1500	1500
$c\left(\frac{\mathrm{Ws}}{\mathrm{g}\,^\circ\mathrm{C}}\right)$	0.3	0.3	0.3
ϵ	1	1	1
$V(\mathrm{mm}^3)$	45,000	45,000	45,000
$\sigma\left(\frac{\mathrm{W}}{\mathrm{mm}^2\,^\circ\mathrm{C}^4}\right)$	5.67037×10^{-14}	5.67037×10^{-14}	5.67037×10^{-14}

The value of φ^{max} is estimated only for ($WAAM_{pwl}$) and given for the other models

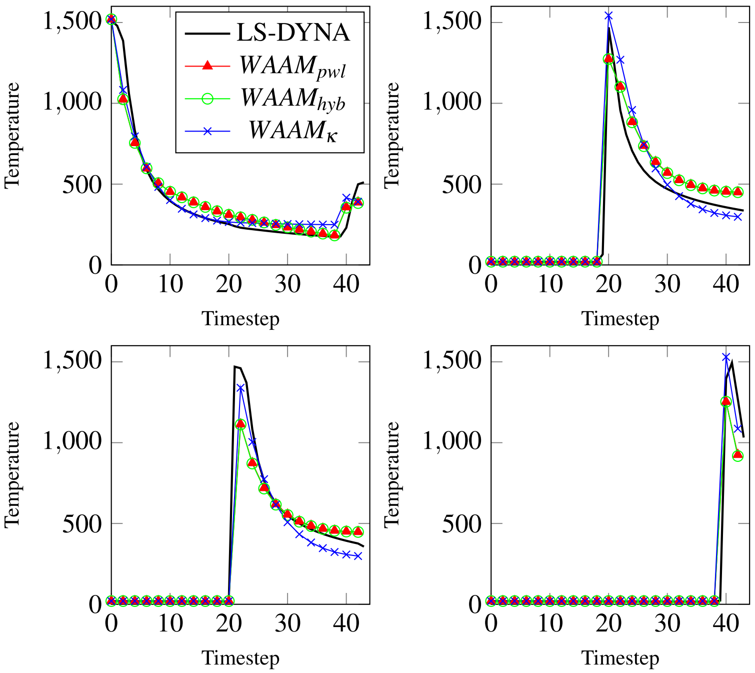


Fig. 13 Graph of the approximated temperature to the data computed by LS-DYNA for the first four nodes and the presented models based on the test instance, cf. Fig. 11. The approximate temperature of the nodes is ordered by columns from top left to right down. (Color figure online)

Only the first four nodes are displayed since the last node is just heated up when it is reached at the last time step.

3.4 Computational results

To review the objective function and to point out differences between the derived models, we conclude this section with the evaluation of the following experiment.

We consider the workpiece displayed left in Fig. 14, consisting of $|\mathscr{V}| = 8$ nodes and $|\mathscr{W}| = 10$ edges and having a side length of 100 mm each. Due to the fact that there are four nodes with odd node degree only a single transition move is required.

At first, we examine the spread of possible solutions for this instance, followed by some comparisons of computation time. All computations were carried out on a MacBookPro with an Intel Core i7 running 8 threads parallel at 3.1 GHz clock speed and 16 GB RAM, using IBM ILOG CPLEX 12.9.0 (default settings).

We consider the ground layer of the given instance, thus $n_l = 1$. To enumerate all paths of the given instance we define a set $\mathscr{M}$ containing all feasible paths $\left(\mathscr{P}_w, \mathscr{P}_u\right)$ that were found so far. In $\mathscr{P}_w$ all tupels of indices $(i, t_i, j, t_j) \in \mathscr{W}^*$ contained, whose related variables hold $w_{i,t_i,j,t_j} = 1$ for this path. The indices of every $u_{i,j,t} = 1$, $(i, j, t) \in \mathscr{U}^*$ of the considered path are stored in $\mathscr{P}_u$.

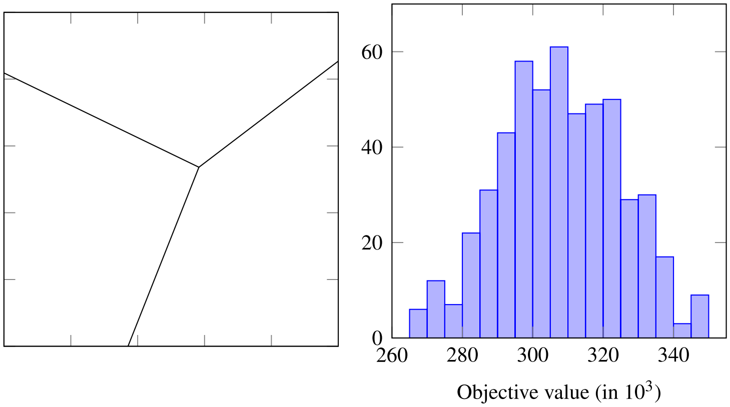


Fig. 14 Test instance with 8 nodes and 10 edges *(left)* and the distribution of its objective values for all feasible solutions *(right)*. The considered layer is a CVT generated using $N^g = 3$ generators

For the computation of all feasible paths a model consisting of constraints (70)–(77) and

$$\sum_{(i,t_i,j,t_j)\in\mathscr{P}_w} w_{i,t_i,j,t_j} + \sum_{(i,j,t)\in\mathscr{P}_u} u_{i,j,t} \leq |\mathscr{W}| + \omega - 1 \qquad \forall\left(\mathscr{P}_w, \mathscr{P}_u\right) \in \mathscr{M} \tag{132}$$

is used, whereby every time a feasible path $\left(\mathscr{P}_w, \mathscr{P}_u\right)$ is found, it is added to $\mathscr{M}$ and the problem is solved again. In contrast to the models presented in Sect. 3.2, constraint (132) is now necessary since for every feasible path $\left(\mathscr{P}_w, \mathscr{P}_u\right) \in \mathscr{M}$ it is not valid. Thus in every run the described model has to find a new feasible path and can not return an already computed one, guaranteeing the desired enumeration.

The enumeration reveals 528 possible paths to weld the test instance of Fig. 14 (left), while the calculation needed 2,274.72 s to enumerate all paths and 2,712.27 s to compute all their heat distributions. For the latter computation the model ($WAAM_\kappa$) was used and all w_{i,t_i,j,t_j}, $(i, t_i j, t_j) \in \mathscr{P}_w$, with $w_{i,t_i j,t_j} = 1$ were set to this value. The allocation of the objective values for all possible paths is displayed in Fig. 14 (right).

It shows, that the solutions are ranging from 266,910.35 to 352,991.05, resulting in a maximum gap of 32.25%.

Furthermore, the histogram in Fig. 14 indicates that the feasible solutions could be normally distributed. This observation is verified by a Shapiro–Wilk test (Shapiro and Wilk 1965) delivering $p = 0.1031$, which is bigger than the chosen alpha level of $\alpha = 0.05$. The computed paths of the four best and the four worst objective values are presented in Fig. 15.

A closer look at the paths of Fig. 15 reveals, that the four best solutions have a similar behavior of the welding process. Each time, the upper surface is firstly processed followed by a jump to the centered node of the considered test instance, before the rest of the unique path is welded. But for the following solutions, no

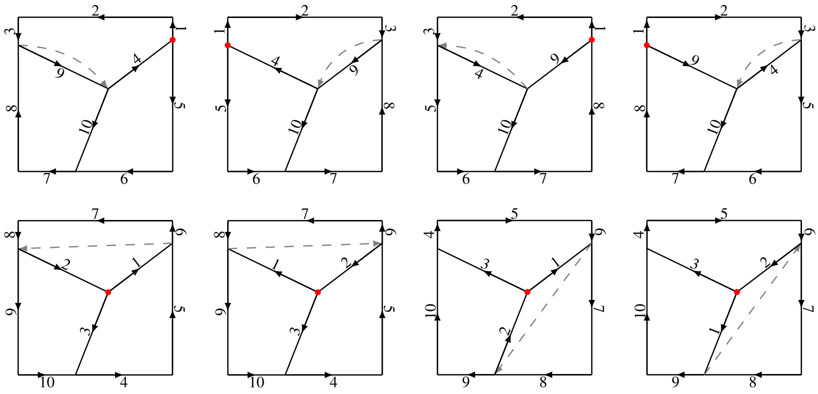


Fig. 15 The four best *(upper row)* and the four worst paths *(lower row)* of the test instance (cf. Figure 14) computed by ($WAAM_\kappa$). The edge labels represent the welding order of the edges, while the dashed line displays the necessary dead-heading move. The starting point of the welding process is marked by a red circle. (Color figure online)

such behavior can be observed. Only the worst solutions have in common, that the inner structure is built first before the outer boundary is welded.

Applying the presented models to this test instance leads to the computation times and solutions displayed in Table 5. The lack of a feasible solution of ($WAAM_{pwl}$) can be explained by the complexity of this model formulation, where for the given time discretization $\Delta t = 0.5s$ around 150 interior points are added and for each the power function $f(x) = x^4$ has to be approximated piecewise linear by the incremental

Table 5 Computational data of all presented models and enumeration applied to the test instance (cf. Fig. 14)

		($WAAM_{pwl}$)	($WAAM_\kappa$)	($WAAM_{hyb}$)	Enumeration	
					($WAAM_\kappa$)	
					Part 1	Part 2
Variables	Before presolve	252,310	60,148	70,102	11,564	60,148
	After presolve	64,786	2528	5413	9869	0
Constraints	Before presolve	320,176	130,124	137,668	1409–1937	130,124
	After presolve	241,948	94,599	98,779	1157–1685	24,616
Solution time (s)		43,200	6279.47	43,200	2274.72	2712.27
Objective value		–	266,910.36	–	–	266,910.36
Gap (%)		100.00	0.00	100.00	–	0.00

In the first part of the enumeration, all feasible ways were computed, followed by the calculation of the temperature distribution for each way in the second part. The models ($WAAM_{pwl}$) and ($WAAM_{hyb}$) found no feasible solution within the time limit of 43,200 s method. The same reason holds for ($WAAM_{hyb}$), although it has already less binary variables. On the other hand, the speed of the enumeration approach depends largely on the complexity of the considered layer and the used model in the temperature calculation.

4 Combining structure and path optimization

We demonstrate the proposed pipeline by means of a simple example. While connecting the methods presented in Sects. 2 and 3, we also propose an optional heuristic to increase suitability of the optimized structure as an input for the path optimization and the target application of WAAM. Finally we also discuss the qualitative impact of different welding paths by means of measurements from workpieces that were actually produced.

4.1 Optimizing structures for printability

We choose the boundary shape to be quadratic, i.e. $\Theta = [0, 10]^2$. As density (or stress map) we choose the Gaussian $\tilde{\rho}^g(\mathbf{x}) = \rho^g(0.1\mathbf{x})$ with ρ^g from (19). A centroidal Voronoi tesselation with $N^g = 6$ generators obtained with the hybrid approach is displayed in Fig. 16a.

As discussed before, during the printing process transition moves should be avoided. The number of necessary and sufficient transition moves is correlated to the number of nodes with odd degree. To this end, we present a heuristic approach to reduce the number of nodes of odd degree. This optional task is to be executed after constructing a CVT and before the resulting graph is passed on to the path optimization routine.

The heuristic consists in simply merging two adjacent nodes of odd degree, thereby creating a node of even degree. If the original nodes are both either on the boundary or inside of Θ, the new node is placed in the middle between them. In the other case the new node is set at the location of the boundary node. The increase of the energy (2) is computed for all adjacent pairs of candidates, i.e. nodes of odd

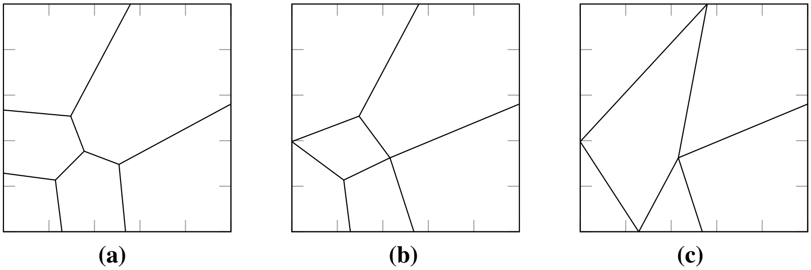


Fig. 16 **a** Centroidal Voronoi tesselation for $\Theta = [0, 10]^2$, $N^g = 6$, scaled density $\tilde{\rho}^g(\mathbf{x}) = \rho^g(0.1\mathbf{x})$ computed with the hybrid approach on a 200×200 grid, **b** graph with merged nodes corresponding to an energy increase of at most 10%, **c** graph with maximum number of merged nodes

Table 6 Computational data of the presented models applied to the CVT (Fig. 16a) including an upper bound for the computational time of 43,200 s

		$(WAAM_{pwl})$	$(WAAM_{\kappa})$	$(WAAM_{hyb})$
Variables	Before presolve	11,788	7066	10,060
	After presolve	5289	3472	4641
Constraints	Before presolve	10,408	6110	8650
	After presolve	7824	4076	6319
Solution time (s)		43,200	43,200	43,200
Objective value		–	184,289.79	–
Gap (%)		100.00	56.12	100.00

Table 7 Computational data of the presented models applied to the merged graph (Fig. 16b) including an upper bound for the computational time of 43,200 s

		$(WAAM_{pwl})$	$(WAAM_{\kappa})$	$(WAAM_{hyb})$
Variables	Before presolve	9215	4877	7343
	After presolve	3290	1657	2607
Constraints	Before presolve	9369	5391	7467
	After presolve	6957	3550	5363
Solution time (s)		43,200	15,742.89	43,200
Objective value		–	155,084.45	115,019.33
Gap (%)		100.00	0.00	96.09

degree. Then the pair with the lowest energy increase is merged. This process can be repeated until the energy would be increased by more than a tolerated amount, or until there are no adjacent nodes of odd degree anymore.

The results of this heuristic are displayed in Fig. 16b and c. On the one hand stability of the resulting structure may be significantly impeded, especially for the last graph, where the maximum number of nodes are merged. On the other hand this graph has only two nodes of odd degree, therefore the structure can be printed without doing transition moves.

4.2 Optimizing the path with or without radiation

The Centroidal Voronoi tesselation and the merged graphs computed in the previous section are taken as input data for the presented models of Sect. 3.2.2.

The results for the three derived models can be found in Tables 6, 7 and 8. The optimal trajectories for the welding head, displayed in Fig. 17, were obtained by $(WAAM_{\kappa})$ since it was the only model that found at least a feasible point in every instance. But for the maximal merged graph, all models found the same optimal path.

Table 8 Computational data of the presented models applied to the maximum merged graph (Fig. 16c) including an upper bound for the computational time of 43,200 s

		$(WAAM_{pwl})$	$(WAAM_{\kappa})$	$(WAAM_{hyb})$
Variables	Before presolve	8716	3998	6124
	After presolve	2220	628	1361
Constraints	Before presolve	9744	5354	7118
	After presolve	6716	3210	4638
Solution time (s)		8738.91	30.11	401.58
Objective value		88,430.04	149,020.04	88,296.81
Gap (%)		0.00	0.00	0.00

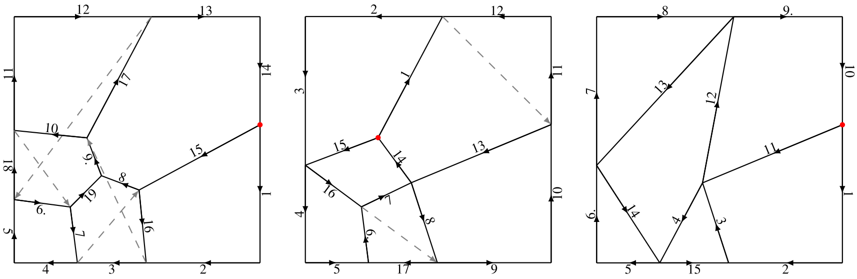


Fig. 17 Best found welding trajectories for the given instances of Fig. 16. For the CVT and the merged graph the solutions of $(WAAM_{\kappa})$ were taken, while for the maximal merged graph all models computed the same optimal path. The edge labels represent the welding order of the edges, while the dashed lines display the necessary dead-heading moves. The starting point of the welding process is marked by a red circle. (Color figure online)

Enumerating all possible ways is no option here. While for the maximal merged graph the enumeration routine, described in Sect. 3.4, needed 208.48 s to find all 608 feasible ways, the computation for the original centroidal Voronoi tesselation and the partly merged graph was aborted after 80,001.80 s with 7,430 feasible ways and 80,000 s with 10,284 ways, respectively. Using a depth-first search, the number of feasible ways for the original centroidal Voronoi tesselation is 196,208,640 and in the partly merged graph there are 248,160 feasible paths.

Although enumeration seems to be an alternative for workpieces with a simple geometry or only a few nodes and edges, for medium complex parts, like the considered ones, it is outperformed by the optimization models.

Furthermore, these computations show the problem of increasing complexity when more transition moves are required in the process. For a small change of the number of edges and the additional transition moves related to this, the number of feasible ways and the computation time grows rapidly. So the number of transitions without welding is a crucial factor for the applicability of our presented models and in terms of computation time it is advantageous to merge nodes to avoid them.

4.3 Application to manufactured workpieces in practice

For an exemplary discussion on the qualitative impact of different welding paths, let us return to the planned workpiece displayed in Fig. 14. To compare the results of the presented models with real processed workpieces, we set the layer height to $h^w = 1.45$ mm, edge width to $b^w = 7.00$ mm and the welding speed to $v^w = 6.67$ mm/s. The instance displayed left in Fig. 14 is again solved using the new parameters and ($WAAM_\kappa$). The resulting optimal path is the same as in Sect. 3.4 with an objective value of 280,987.24 and was computed in 15,691.34 s. According to this result, we chose the optimal and the worst path of the test instance, presented in Fig. 15 on the left in the upper and the lower row, respectively, as templates for the workpieces to be manufactured.

The results of the numerical investigation are manufactured with the fanuc robot system shown in Fig. 18 and the welding source TPS 500i MSG of Fronius. To reduce the required thermal energy, the cold metal transfer process is used, where a moving heat source melts the electrode wire and deposits the metal through an electric arc on the substrate.

The optimized tool paths mentioned above have been translated to a NC-code according the method described in Nguyen et al. (2018). To achieve the best relation of viscosity and cooling of the weld pool for the mild steel ER70S-6, a shielding gas of 80% Ar and 20% CO_2 and a wire diameter of 1.0 mm used. With a welding speed of 6.67 mm/s and a wire-feed rate of 91.67 mm/s, the weld bead features a width of 7 mm and a height of 1.45 mm. The whole structure was built up in 12 layers which results in a height of 17.4 mm.

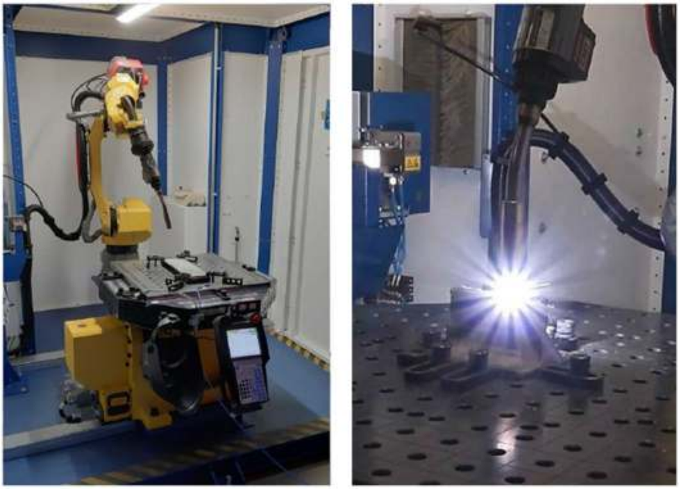


Fig. 18 Fanuc robot system using a TPS 500i MSG of Fronius as weld source, that was used to manufacture the compared workpieces. *(left)* Complete system, *(right)* welding head during printing process

During the AM-process the thermal camera VarioCAM® Infratec with 640 × 480 pixels and an accuracy of 2% measures the temperature distribution. An emissivity of 0.6 is chosen as an average value.

In Fig. 19, the processed workpieces and their temperature distribution directly after welding are shown. Both AM-parts show negligibly porosities and feature good welding qualities.

Due to heat transfer into the substrate plate the temperature decreases very fast, which causes thermal shrinkage of the added material and the part. The intensive localized heating results in high residual stresses and distortion during and even after the process. This leads to geometrical inaccuracies and sometimes to thermal cracks (Israr et al. 2018). These unwanted phenomena are an indicator for the quality of the welding strategy. It should be mentioned, that the main distortion occurs during the first two layers, where the welding process is quite instable and more thermal energy is needed to heat the contact surface of the substrate plate. After four or five layers, the temperature balance will be stable, because the thermal energy of the weld source is equal to conduction and radiation of the actual shape. To find a time independent indicator to benchmark the path strategies in experiments, the distortion of the substrate plate is measured after cooling.

The distortion of the substrate plate for the worst welding path according to the thermal calculation is about 2.21 mm. Figure 20 shows, that the distortion depends

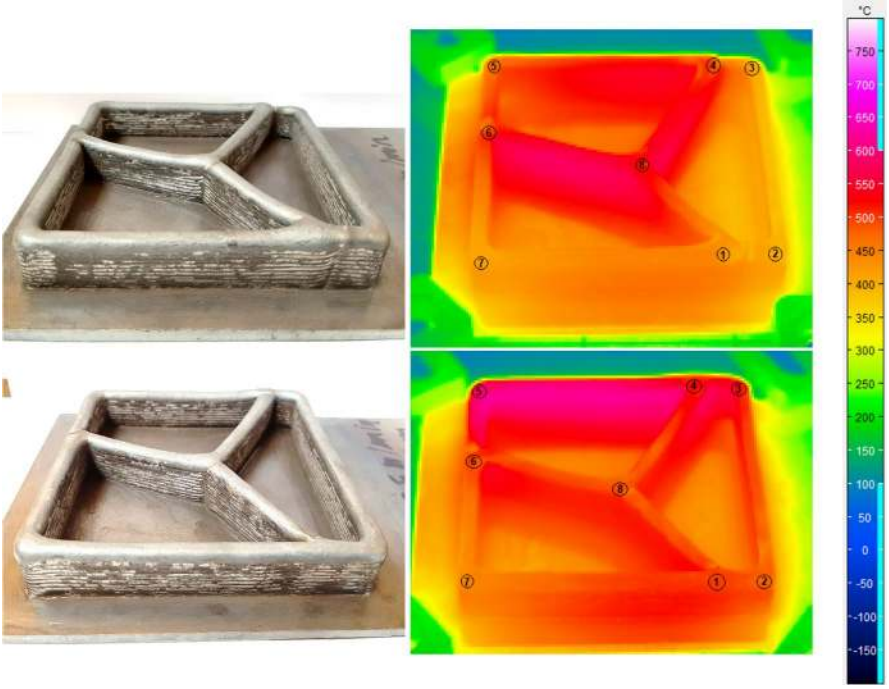


Fig. 19 Real processed workpiece and its final temperature distribution for the optimal *(upper row)* and the worst path *(lower row)* of the test instance, computed in Sect. 3.4. (Color figure online)

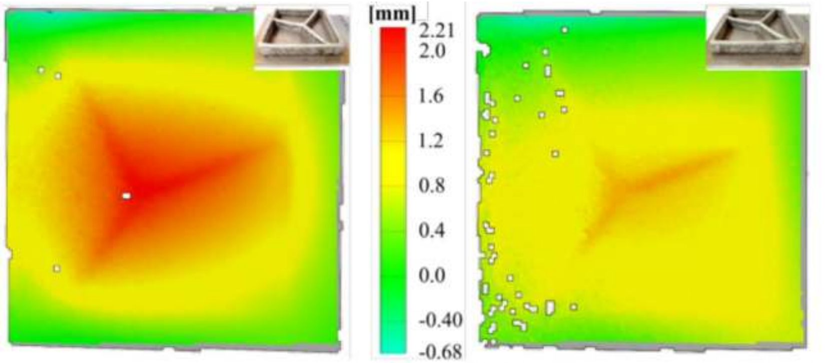


Fig. 20 Distortion distribution of the substrate plate for the worst *(left)* and the best welding path *(right)* according to the thermal calculation. (Color figure online)

highly on the geometry and the position of the processed edges. Both parts show nearly exact the same pattern of distortion. Nevertheless, in case of the best path, the distortion could be reduced about 0.69 mm to 1.52 mm, which is round about 30%.

Now we want to compare the accuracy of the presented models by comparison to the measured data during the manufacturing process for the best case. We consider the second layer, want to compute its temperature distribution and compare the results to the measured data during the manufacturing process.

Thus we apply the necessary changes presented in Sect. 3.2.4 to the models. For the initial temperature φ_i^{init}, the temperature of every node was measured when the first layer was finished and for the interior points we assume

$$\theta_{i,j,k,0}^{fd} = \frac{\varphi_i^{init} + \varphi_j^{init}}{2} \qquad \forall (i,j) \in \mathscr{W},\ k \in \mathscr{L}_{i,j}. \tag{133}$$

To speed up the computation of the presented models, we fix all variables w_{i,t_i,j,t_j}, $(i,t_i,j,t_j) \in \mathscr{W}^*$, and $u_{i,j,t}$, $(i,j,t) \in \mathscr{U}^*$ to their respective values within the optimal solution.

Since the camera has a fixed viewpoint and the welding head is moving along the edges, not all nodes are visible all the time. Furthermore, at the center of the weld pool the metal is evaporating causing a much lower temperature than at its surrounding. This leads to a pair of peaks in the measured data, when a node is heated up. The measured temperature and the computed temperature distribution of all three models for the marked nodes in Fig. 19 are displayed in Fig. 21.

As one can see, the progress of the calculated temperature for a single node is similar to the measured data, only the heating process is a bit too high. Furthermore, the models ($WAAM_{pwl}$) and ($WAAM_{hyb}$) deliver nearly the same results, whereby ($WAAM_{hyb}$) needed less computation time in general. In contrast to the models using the incremental method, the cooling process of the model ($WAAM_{\kappa}$) is too fast compared to the measured data, resulting in a worse temperature distribution. By estimating the related parameters in a different way, this behaviour could be reduced. The increase

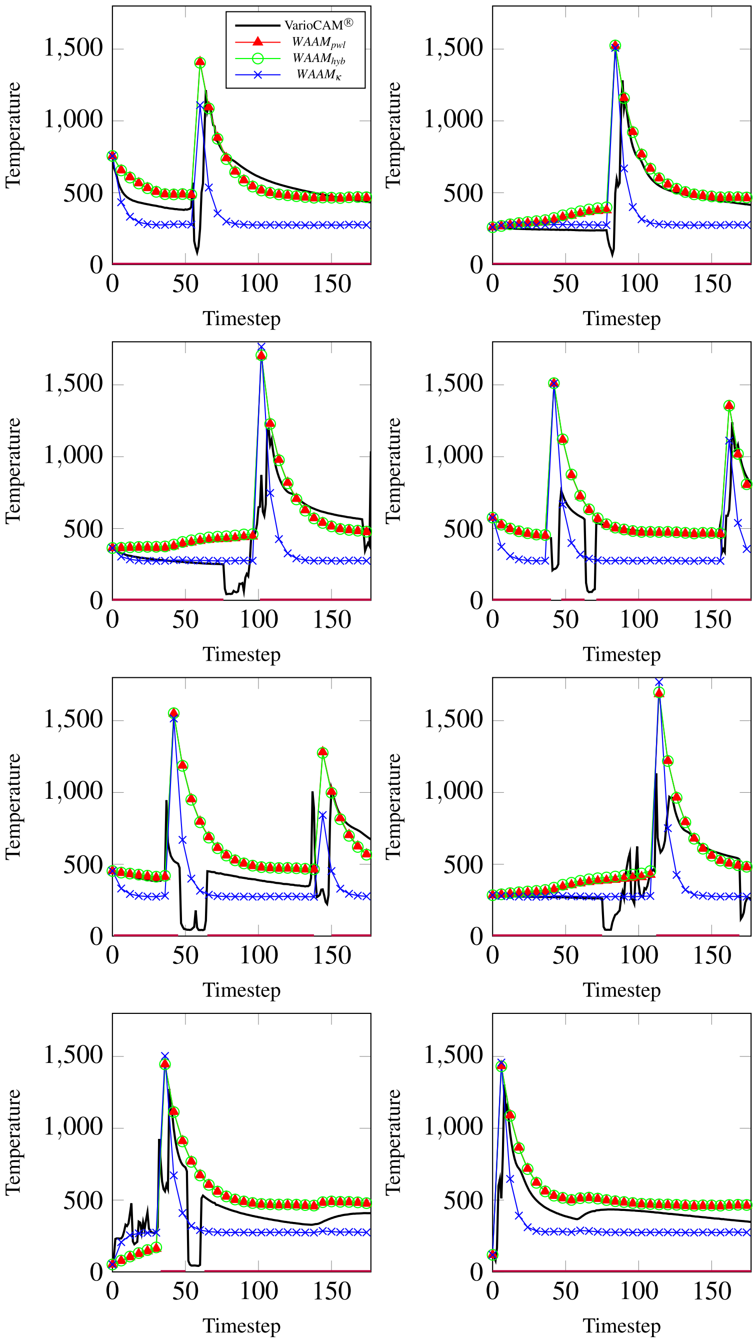


Fig. 21 Measured and calculated temperature of all nodes for the three presented models. The purple line at the *x*-axis indicates if a node is visible for the camera or hidden by the welding head, which leads to irregular behavior for VarioCAM®. (Color figure online)

of the temperature at the beginning of the computation could be caused by the initial values of the interior points, which are approximated to be equal over the whole edge.

5 Future work

The discussed techniques certainly give rise to a number of possible directions for future extensions. First of all, a big challenge is to adapt the presented work in the three dimensional case, which is of course more relevant in the context of AM. The concept of CVTs is also valid in $\mathbb{R}^3$, but there are usually no restrictions on the orientation of cell boundaries. This may pose a problem since the printability of an object in WAAM depends on the angle of the involved surfaces.

For the path optimization in the 3D case, the temperature of the previous layer would have to be taken into account. Furthermore, in practice welding every layer with the same tool path is not desirable due to stability issues, so the solution of the actual layer should be different from the previous one. Here the computational time may be the crucial factor. Currently the optimization of a single layer for relatively simple objects requires several hours.

In terms of accuracy of the temperature calculation a more detailed model of the substrate plate could be added. One linear approach would be to use finite differences on a two-dimensional heat equation. Thereby, the respective discretization points could be connected to the above positioned points of the first layer. If they are assumed to be independent of the path generation variables, then the coefficients of the resulting linear equation system can be generated a priori.

Motivated by the parameter estimation of ($WAAM_{hyb}$) the question arises, what influence can be assigned to the radiation between points within the workpiece for increasing size. A detailed analysis of this relations could be used in a preprocessing step for reducing the model complexity and thus saving computation time.

To increase the sampling rate during the geometric approach for CVT computation without approximating cell boundaries, a possibility would be to divide the arising triangles within the Voronoi cells into four smaller triangles. This process could be repeated until the triangles are sufficiently small. The density would then naturally be sampled in the centers of mass of the smallest triangles. However, there is the need to find out if such an additional effort can be justified by a structural stability gain.

In Sect. 4 we introduced a heuristic to reduce the number of nodes with odd degree. The process of merging nodes may also be combined with the addition of new edges, thus increasing printability at the cost of weight and increased material requirements.

6 Summary and conclusion

In the context of WAAM, we have demonstrated a procedure to optimize the inner structure to be printed as well as the process of actually printing the structure. For the first subproblem of structure optimization we considered the computation of

CVTs in multiple variations. These variations have their advantages and drawbacks. Therefore, the choice on which variant should be used depends on the specific application, taking into account especially the characteristics of the shape of a planned workpiece.

For the subproblem of path optimization we derived multiple mixed-integer linear models and investigated them on several test instances. Among the used models a trade-off between accuracy and computation speed has been revealed. Hence, their usage should depend on the aimed quality of the planned workpiece, but is at the moment limited by its complexity.

Overall, we discussed fundamental principles that have to be considered for optimizing certain tasks in WAAM. With experimental tests, the model could be proved with experimental temperature history and the distortion could be reduced about 0.69 mm to 1.52 mm, which is round about 30%. We believe that the discussed insights will also be helpful in future research on WAAM to achieve a higher geometrical accuracy and increase the process stability. While it remains to optimize the presented pipeline for computational efficiency, the developed methods can be applied to a wide array of use cases.

Acknowledgements The authors are grateful to Fabian Gnegel for his helpful comments on an earlier version of the manuscript and Rameez Israr for his FEM-simulations in LS-DYNA.

Funding Open Access funding enabled and organized by Projekt DEAL.

References

Allen J (2006) An investigation into the comparative costs of additive manufacture vs. machine from solid for aero engine parts. Technical report, Rolls-Royce plc. Derby, UK

Bähr M, Radow G, Breuß M, Fügenschuh A (2019) Computation of stable honeycomb structures for additive manufacturing. In: Fortz B, Labbe M (eds) Operations research proceedings. Springer, Cham, pp 363–369

Barber CB, Dobkin DP, Huhdanpaa H (1996) The quickhull algorithm for convex hulls. ACM Trans Math Softw 22(4):469–483

Belyaev AG, Fayolle PA (2015) On variational and PDE-based distance function approximations. Comput Graph Forum 34(8):104–118

Bronstein AM, Bronstein MM, Kimmel R (2009) Numerical geometry of non-rigid shapes, 1st edn. Springer, Berlin

Buhl J, Israr R, Bambach M (2019) Modeling and convergence analysis of directed energy deposition simulations with hybrid implicit/explicit and implicit solutions. J Mach Eng 19:94–107

Carter G, Giles FC, Donald EP (1991) Materials science and engineering. ASM International, Cleveland

Çengel YA (2002) Heat and mass transfer: a practical approach. McGraw-Hill Higher Education, New York

Chen J, Han Y (1990) Shortest paths on a polyhedron. In: Proceedings of the sixth annual symposium on computational geometry, SCG ’90, pp 360–369

Chunxu X, Wang T, Liu YJ, Liu L, He Y (2015) Fast wavefront propagation (FWP) for computing exact discrete geodesics on meshes. IEEE Trans Vis Comput Graph 21:822–834

Crane K, Weischedel C, Wardetzky M (2013) Geodesics in heat: a new approach to computing distance based on heat flow. ACM Trans Graph 32(5):152:1–152:11

Cristiani E, Falcone M (2007) Fast semi-Lagrangian schemes for the eikonal equation and applications. SIAM J Numer Anal 45(5):1979–2011

Ding D, Pan ZS, Cuiuri D, Li H (2014) A tool-path generation strategy for wire and arc additive manufacturing. Int J Adv Manuf Technol 73(1):173–183

Ding D, Pan ZS, Cuiuri D, Li H (2015) A practical path planning methodology for wire and arc additive manufacturing of thin-walled structures. Robot Comput-Integr Manuf 34:8–19

Du Q, Emelianenko M (2006) Acceleration schemes for computing centroidal Voronoi tessellations. Numer Linear Algebra Appl 13:173–192

Du Q, Faber V, Gunzburger M (1999) Centroidal Voronoi tessellations: applications and algorithms. SIAM Rev 41(4):637–676

Edmonds J, Johnson EL (1973) Matching, Euler tours and the Chinese postman. Math Program 5:88–124

Fabri A, Giezeman GJ, Kettner L, Schirra S, Schönherr S (2000) On the design of CGAL a computational geometry algorithms library. Softw: Pract Exp 30(11):1167–1202

Fourier JBJ (1878) The analytical theory of heat. The University Press, Sacramento

Fügenschuh A, Hayn C, Michaels D (2014) Mixed-integer linear methods for layout-optimization of screening systems in recovered paper production. Optim Eng 15(2):533–573

Fügenschuh A, Bambach M, Buhl J (2019) Trajectory optimization for wire-arc additive manufacturing. In: Fortz B, Labbe M (eds) Operations research proceedings 2018. Springer, Cham, pp 331–337

Godunov SK, Bohachevsky I (1959) Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics. Mat Sb 47(89)(3):271–306

Goldak J, Chakravarti A, Bibby M (1984) A new finite element model for welding heat sources. Metall Trans B 15(2):299–305

Gómez J, Alvarez D, Garrido S, Moreno L (2019) Fast methods for eikonal equations: an experimental survey. IEEE Access 7:39005–39029

Hales TC (2001) The honeycomb conjecture. Discrete Comput Geom 25(1):1–22

Hassouna MS, Farag AA (2007) Multistencils fast marching methods: a highly accurate solution to the eikonal equation on Cartesian domains. IEEE Trans Pattern Anal Mach Intell 29(9):1563–74

Hateley JC, Wei H, Chen L (2015) Fast methods for computing centroidal Voronoi tessellations. J Sci Comput 63:185–212

Herholz P, Davis TA, Alexa M (2017a) Localized solutions of sparse linear systems for geometry processing. ACM Trans Graph 36(6):183:1–183:8

Herholz P, Haase F, Alexa M (2017b) Diffusion diagrams: Voronoi cells and centroids from diffusion. Comput Graph Forum 36:163–175

Hierholzer C, Wiener C (1873) Über die Möglichkeit, einen Linienzug ohne Wiederholung und ohne Unterbrechung zu umfahren. Math Ann 6(1):30–32 [**publication in German**]

Hollander DA, Walter MV, Wirtz T, Sellei R, Schmidt-Rohlfing B, Paar O, Erli HJ (2006) Structural, mechanical and in vitro characterization of individually structured Ti–6Al–4V produced by direct laser forming. Biomaterials 27(7):955–963

Holman JP (2009) Heat transfer, 10th edn. Mc Graw Hill, New York

Howell JR, Menguc MP, Siegel R (2015) Thermal radiation heat transfer. CRC Press, Boca Raton

Israr R, Buhl J, Elze J, Bambach M (2018) Simulation of different path strategies for wire-arc additive manufacturing with Lagrangian finite element methods. In: LS-DYNA forum

Jost J (2007) Partial differential equations, 2nd edn. Springer, Berlin

Kılınç MR, Sahinidis NV (2018) Exploiting integrality in the global optimization of mixed-integer nonlinear programming problems with BARON. Optim Methods Softw 33(3):540–562

Kimmel R, Sethian JA (1998) Computing geodesic paths on manifolds. Proc Natl Acad Sci USA 95(15):8431–8435

Kuo FY, Sloan IH (2005) Lifting the curse of dimensionality. Not AMS 52(11):1320–1328

Leuders S, Thöne M, Riemer A, Niendorf T, Tröster T, Richard H, Maier H (2013) On the mechanical behaviour of titanium alloy TiAl6V4 manufactured by selective laser melting: fatigue resistance and crack growth performance. Int J Fatigue 48:300–307

Liu Y, Wang W, Lévy B, Sun F, Yan DM, Lu L, Yang C (2009) On centroidal Voronoi tessellation—energy smoothness and fast computation. ACM Trans Graph 28(4):1–17

Lloyd S (1982) Least squares quantization in PCM. IEEE Trans Inf Theory 28(2):129–137

Lu L, Sharf A, Zhao H, Wei Y, Fan Q, Chen X, Savoye Y, Tu C, Cohen-Or D, Chen B (2014) Build-to-last: strength to weight 3D printed objects. ACM Trans Graph 33(4):1–10

Lyon A, Colyvan M (2008) The explanatory power of phase spaces. Philos Math 16(2):227–243

Markowitz HM, Manne AS (1957) On the solution of discrete programming problems. Econometrica 25(1):84–110

Michel F, Lockett H, Ding J, Martina F, Marinelli G, Williams S (2019) A modular path planning solution for wire+ arc additive manufacturing. Robot Comput-Integr Manuf 60:1–11

Mitchell JSB, Mount DM, Papadimitriou CH (1987) The discrete geodesic problem. SIAM J Comput 16(4):647–668

Montevecchi F, Venturini G, Grossi N, Scippa A, Campatelli G (2018) Idle time selection for wire-arc additive manufacturing: a finite element-based technique. Addit Manuf 21:479–486

Nguyen L, Buhl J, Bambach M (2018) Decomposition algorithm for tool path planning for wire-arc additive manufacturing. J Mach Eng 18(1):95–106

Peter AM, Gurumoorthy KS, Moyou M, Rangarajan A (2014) A new energy minimization framework and sparse linear system for path planning and shape from shading. In: Proceedings of the 2014 Indian conference on computer vision graphics and image processing, pp 15:1–15:8

Peyré G, Cohen LD (2003) Geodesic remeshing using front propagation. In: VLSM'03, pp 33–40

Preparata FP, Shamos MI (1985) Computational geometry. Springer, New York

Recktenwald GW (2004) Finite-difference approximations to the heat equation. Mech Eng 10:1–27

Rodrigues TA, Duarte V, Miranda R, Santos TG, Oliveira J (2019) Current status and perspectives on wire and arc additive manufacturing (waam). Materials 12(7):1121

Sethian JA (1996) A fast marching level set method for monotonically advancing fronts. Proc Natl Acad Sci USA 93(4):1591–1595

Sethian JA (1999) Level set methods and fast marching methods, 2nd edn. Cambridge Press, Cambridge

Shapiro SS, Wilk MB (1965) An analysis of variance test for normality (complete samples). Biometrika 52(3/4):591–611

Tournois J, Alliez P, Devillers O (2010) 2D centroidal Voronoi tessellations with constraints. Numer Math: Theory Methods Appl 3(2):212–222

Venturini G, Montevecchi F, Scippa A, Campatelli G (2016) Optimization of WAAM deposition patterns for t-crossing features. Procedia CIRP 55:95–100

Wang X, Ying X, Liu YJ, Xin SQ, Wang W, Gu X, Müller-Wittig W, He Y (2015) Intrinsic computation of centroidal Voronoi tessellation (CVT) on meshes. Comput Aided Des 58:51–61

Xu SG, Zhang YX, Yong JH (2010) A fast sweeping method for computing geodesics on triangular manifolds. IEEE Trans Pattern Anal Mach Intell 32:231–241

Ying X, Wang X, He Y (2013) Saddle vertex graph (SVG): a novel solution to the discrete geodesic problem. ACM Trans Graph 32(6):170:1–170:12

Zayer R, Mlakar D, Steinberger M, Seidel HP (2018) Layered fields for natural tessellations on surfaces. ACM Trans Graph 37(6):264:1–264:15

Zhang M, Zhou Y, Huang C, Chu Q, Zhang W, Li J (2018) Simulation of temperature distribution and microstructure evolution in the molten pool of GTAW Ti–6Al–4V alloy. Materials 11(11):2288

Zhao H (2005) Fast sweeping method for eikonal equations. Math Comput 74(250):603–627

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Martin Bähr[1] · **Johannes Buhl**[1] · **Georg Radow**[1] · **Johannes Schmidt**[1] · **Markus Bambach**[1] · **Michael Breuß**[1] · **Armin Fügenschuh**[1]

Martin Bähr
martin.baehr@b-tu.de

Johannes Buhl
buhl@b-tu.de

Georg Radow
radow@b-tu.de

Markus Bambach
bambach@b-tu.de

Michael Breuß
breuss@b-tu.de

Armin Fügenschuh
fuegenschuh@b-tu.de

[1] Brandenburg University of Technology Cottbus-Senftenberg, Platz der Deutschen Einheit 1,
03046 Cottbus, Germany