Brandenburg
University of Technology
Cottbus - Senftenberg

Mixed-Integer Programming for Additive Manufacturing

Jesse Beisegel
Johannes Buhl
Rameez Israr
Johannes Schmidt
Markus Bambach
Armin Fügenschuh

Cottbus Mathematical Preprints
COMP # 23(2021)

Mixed-Integer Programming for Additive Manufacturing

Jesse Beisegel* Johannes Buhl† Rameez Israr‡ Johannes Schmidt§
Markus Bambach¶ Armin Fügenschuh‖

December 21, 2021

Abstract

Since the beginning of its development in the 1950s, mixed integer programming (MIP) has been used for a variety of practical application problems, such as sequence optimization. Exact solution techniques for MIPs, most prominently branch-and-cut techniques, have the advantage (compared to heuristics such as genetic algorithms) that they can generate solutions with optimality certificates. The novel process of additive manufacturing opens up a further perspective for their use. With the two common techniques, Wire Arc Additive Manufacturing (WAAM) and Laser Powder Bed Fusion (LPBD), the sequence in which a given component geometry must be manufactured can be planned. In particular, the heat transfer within the component must be taken into account here, since excessive temperature gradients can lead to internal stresses and warpage after cooling. In order to integrate the temperature, heat transfer models (heat conduction, heat radiation) are integrated into a sequencing model. This leads to the problem class of MIPDECO: MIPs with partial differential equations (PDEs) as further constraints. We present these model approaches for both manufacturing techniques and carry out test calculations for sample geometries in order to demonstrate the feasibility of the approach.

Keywords: Wire Arc Additive Manufacturing, Laser Powder Bed Fusion, Mixed-Integer Programming, Partial Differential Equations, Finite Element Method, Finite Difference Method, Optimization.

1 Introduction

In this section we give a brief introduction to additive manufacturing, simulation of physical phenomena using the finite element method, and mixed-integer programming as our method of choice to model the scheduling of the production process.

*Brandenburg University of Technology, Platz der Deutschen Einheit 1, 03046 Cottbus, Germany, jesse.beisegel@b-tu.de

†Brandenburg University of Technology, Platz der Deutschen Einheit 1, 03046 Cottbus, Germany, johannes.buhl@b-tu.de

‡Brandenburg University of Technology, Platz der Deutschen Einheit 1, 03046 Cottbus, Germany, rameez.israr@b-tu.de

§Brandenburg University of Technology, Platz der Deutschen Einheit 1, 03046 Cottbus, Germany, johannes.schmidt@b-tu.de

¶ETH Zurich, Advanced Manufacturing Lab, Technoparkstrasse 1, 8005 Zurich, Switzerland, mbambach@ethz.ch

‖Brandenburg University of Technology, Platz der Deutschen Einheit 1, 03046 Cottbus, Germany, fuegenschuh@b-tu.de

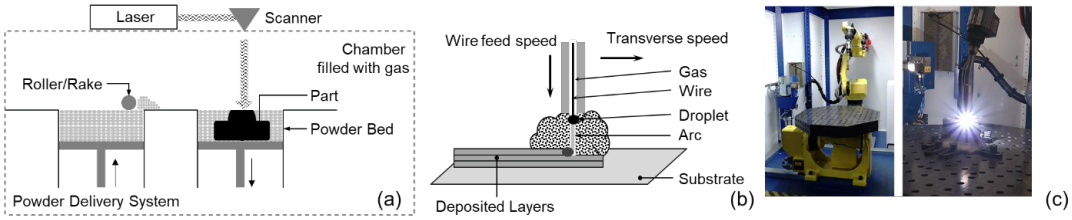

Figure 1: Principle sketch of the material feed before the heating process (a), material feed during the heating process (b), and welding robot for deposition on a turn/tilt table (www.b-tu.de) (c).

1.1 3D Printing Systems and Their Classification

1.1.1 Material Feeding Before the Heating Process

The largest group of processes with material feed before heating to the melting point is performed are the powder bed processes. As shown in Figure 1a, powder is spread over the powder bed via a powder bed feed system. The energy source, for example a laser beam [39, 53, 58], melts or sinters the powder locally at the points that will later form the component. The excess powder in the build space is not heated and is removed and recycled after the printing process. The layer-by-layer coating and melting, or sintering, process is repeated until the complete geometry is built up. The powder bed system can be used to build up very fine geometries with cavities inside and other high-resolution features. Laser Powder Bed Fusion (LPBF) is selected in this work as an example of trajectory optimization.

1.1.2 Material Feeding During the Heating Process

In contrast to the LPBF process, in powder feed systems the applied material is delivered in the form of powder into the interaction zone between the heat source and the substrate material. Alternatively, wire can be fed into the heat source or a wire can be used directly as a consumable electrode (see Figure 1b). When material is fed during the heating process, it is melted with electron beams, laser beams [2], an electric arc, or a plasma [64]. The deposition process is usually performed on a stationary workpiece, with the deposition head moving, or the deposition head stationary and the workpiece moving. For some years now, a variety of motion systems have been explored, whereby systems such as industrial robots are also used to guide the deposition head and the workpiece is mounted, e.g., on a positioner (see Figure 1c) [48]. This allows maximum degrees of freedom for positioning the component and the deposition head. Very high deposition rates are possible, especially with arc welding, so that even large-volume components can be built up. A particular challenge of these processes is the trajectory planning of the deposition head, because holes or material accumulation occur if the trajectories and deposition speeds are not optimally set. Material accumulations in nodes very quickly cause process instabilities and generate considerable post-processing costs [48]. With such application systems, even worn or damaged components can be reworked.

1.2 Physical Phenomena During AM Processes

Due to the strong temperature gradients in almost all 3D printing processes, residual stresses build up, which can only be removed with the help of cost-intensive thermal post-treatments. A major disadvantage of AM processes is that, depending on the component geometry and the printing strategy, the intensive heat input also leads to undesirable

macroscopic effects in the component. One effect is that a high geometric inaccuracy occurs due to thermal deformations during the printing process and during cooling. In addition, it is possible that components crack during production due to the high thermal stresses [59]. Notch-like defects or plate-stack-like stacking defects can occur on the surface [40, 52] and defects such as gas porosity can occur inside the component [36], which can be controlled by adjusting the process parameters [36, 42]. Macroscopic effects in particular are difficult to avoid and often require additional post-processing such as chemical etching [57] or conventional machining. These undesirable macroscopic effects mainly depend on the tool path, the process settings and the cooling strategy. The locally acting heat source causes inhomogeneous thermal expansion, which leads to contraction over the process time and causes mechanical stresses and strains [18]. In almost all AM processes, the temperature profile is strongly dependent on the heat conduction in the component, the heat transfer coefficient, and the radiation into the surrounding medium (powder in LPBD and air in WAAM).

1.3 Application of the FEM Method for AM Processes

With numerical simulations it is possible to calculate the heat transfer with conduction, convection and radiation. Thus, in principle, the macroscopic effects such as the distortion, residual stresses and plastic deformation can be investigated numerically, which has also been done for the WAAM process [8, 32, 17]. Especially when thermal, mechanical, and even the microstructural solvers are used in a coupled way, the simulation time for the LPBD and the WAAM increases tremendously. Due to the complexity of the models, convergence problems occur when solving large deformations, thermal stresses, and residual stresses, which can lead to the premature termination of the computation [9, 41, 7, 19]. Even for larger geometries, the temperature, stress, and thermal distortion could be simulated during the WAAM process and during cooling, with extremely high computation times [49]. With the LPBD method, a simulation of a few hatches is possible [3, 45, 51], but the representation of a layer or even a simple component appears impossible due to the very long laser paths. However, in order to obtain a numerical estimation for real components as well, reference volumes are defined across many layers and calculated with different meso-models [63]. Depending on the resolution of the reference volumes, these approaches enable a very fast global estimation of the temperature and the temperature gradient. Due to the long calculation time, or because the reference volumes are too large, both types of FEM models cannot be used to optimize the trajectory of the heat source with regard to the local temperature gradient in a layer.

1.4 Direct Solution of PDEs for AM Processes

An alternative to highly simplified FEM models is the direct solution of PDEs to calculate the temperature distribution in a layer. For this purpose, a method was first set up to model the nodes of thin-walled hollow structures produced with the WAAM method and to calculate their cooling by thermal radiation. The trajectory of the arc is realized with the sequential activation of the nodes. An optimal node sequence is calculated under the constraints of (i) single node visit, (ii) minimum number of start and end points, and (iii) minimum temperature gradient using MILP [24]. Building on this work, heat conduction was integrated by implementing beam elements (1D) between the nodes [4]. To apply the optimization strategy to the LPBD method, the temperature calculation in the pressure plane (2D) could be adapted, initially considering only the heat conduction in the layer [6]. All works have in common that a steady state process is assumed and the temperature of the lower layer is stable.

1.5 Mixed-Integer Linear Programming

Mixed-Integer Programming refers to the mathematical field of modeling and solving problems from a certain class of optimization problems of the form

$$\begin{array}{rlll} z^* = & \min & cx & \\ & \text{s.t.} & Ax & \leqslant b \\ & & x & \in \mathbb{Z}^p \times \mathbb{Q}^{n-p}, \end{array} \tag{MILP}$$

where c is an n-dimensional row vector, b is an m-dimensional column vector, and A is an m by n matrix, all containing rational numbers. For a fixed integer $p \in \{1, \ldots, n-1\}$ we speak of a *mixed-integer (linear) program*. Here cx is the *objective* and $Ax \leqslant b$ are the *constraints*. Two special cases are worth mentioning: when $p = 0$ we deal with a *linear program (LP)* and $p = n$ is a pure *integer (linear) program (ILP)*. Any column vector $x \in \mathbb{Z}^p \times \mathbb{Q}^{n-p}$ with $Ax \leqslant b$ is a *feasible solution* for (MILP). A feasible solution x^* is an *optimal solution* for (MILP) if its objective function value cx^* is equal to z^*. The study of LPs began in the mid 20th century with the work of Kantorovich [35], Koopmans [38], and most notably, Dantzig's invention of the Simplex algorithm to solve general LPs [16] - although earlier attempts can be dated back to Fourier [22] and Motzkin [46]. A few years later, LPs with integrality conditions came into focus. Dantzig, Fulkerson, and Johnson suggested integer variables to model binary yes-no decisions, with the Traveling Salesman Problem as prime application example. For the actual solution of such model they introduced the LP relaxation and a cutting plane approach [15]. Since these method are still in use today, we give a brief survey on the solution process of (MILP) (for $p > 1$, i.e., MILP or ILP). We first note that these problems are difficult to solve. From a theoretical perspective, they fall into the call of NP-hard problems [25], so that a theoretically efficient algorithm for the solution of general (M)ILP would imply P=NP (which is an open Millennium problem [33]). From a practical perspective, the solution process of MILPs attacks them from two sides, called the primal and the dual side. On the *primal side* one is concerned with finding good feasible solutions fast. This is the realm of heuristics, such as Taboo Search [26], Simulated Annealing [11, 1], or Evolutionary Algorithms [56], to name just a few. When the objective function emphasizes a minimization, as it does in (MILP), then the objective value cx of every primal solution x gives an upper bound on z^*. On the *dual side* one tries to give lower bounds on z^*. Most prominently, the integrality conditions are dropped (relaxed), so that (MILP) is turned into an LP problem, which can be solved much easier. Searching over a larger space now, an LP feasible solution $\hat{x}$ that satisfies $A\hat{x} \leqslant b$ gives a lower bound $c\hat{x}$ on z^*. In case $\hat{x} \in \mathbb{Z}^p \times \mathbb{Q}^{n-p}$, this relaxation would give already a feasible solution. However, this rarely happens in practice, where it can be expected that some of the variables with integrality constraints have fractional values in the solution. Then the integrality conditions are gradually re-introduced by either a cutting plane approach, such as Gomory's [28] (see also [12]), or branch-and-bound [43, 14]. In fact, both approaches can be combined to a branch-and-cut method, which was pioneered by Padberg and Rinaldi [50] and Balas et al. [5]. These methods are today readily available in several software packages, such as IBM ILOG CPLEX [30], Gurobi [29], or Fico Xpress [21], which can solve instances of (MILP) in the order of 100,000 variables and constraints. For further details on mixed-integer programming we refer to the textbooks of Nemhauser and Wolsey [47] and Wolsey [62].

In the additive manufacturing application we introduce below, mixed-integer programming is used to formulate the problem of scheduling the production process within one layer. The sequencing of the printing process is modeled by integer (in fact, binary) decision variables, and the constraints represent restrictions on the possible sequences and consequences of routing decisions, such as the material temperature and temperature gradients, which in turn lead to internal stresses and warpage after cooling. Typical for

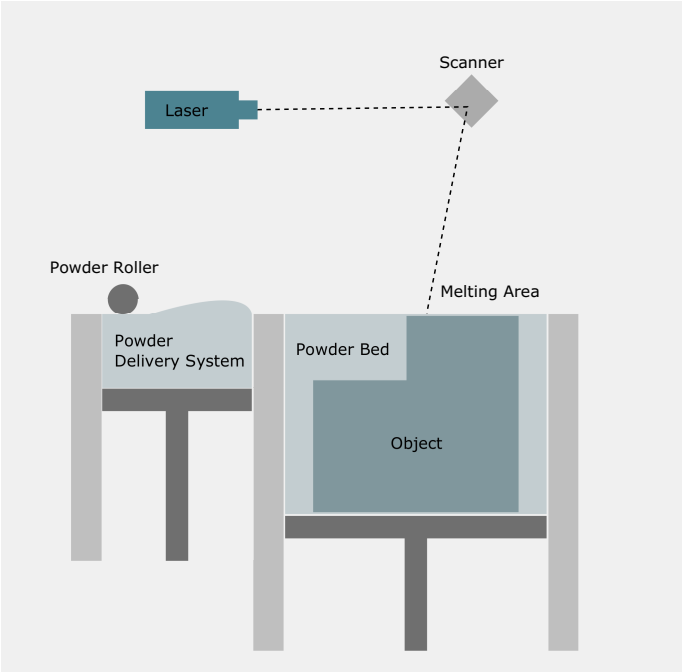

Figure 2: An illustration of the LPBF process.

applied problems with a physical or technical background, they oftentimes come with relations that are not given by explicit functions but by ordinary (ODE) or partial differential equations (PDE) instead. For example, the conduction of heat is described by the heat equation, a well-known parabolic PDE. It is a modeling as well as solving challenge to embed these conditions as constraints into the framework of mixed-integer programming. Approaches for the integration of PDEs can be found in Frank et al. [23], where a PDE was integrated into an MILP using a semi-discretization approach. The term MIPDECO for this problem class was suggested by Leyffer [44]. A usual approach to tackle a PDE is to use a finite difference approach for its discretization [55]. If the discretized equation is linear, it can directly be used as a constraint in (MILP). This approach was used, for instance, by Gnegel et al. [27]. However, the downside of this approach is the enormous size of the constraint system (in terms of number of variables and constraints), which is usually beyond the scope even of modern numerical MILP solvers, and the structure of discretized PDE systems (such as band matrices), which are exploited by special-purpose PDE solvers, but vastly ignored by current MILP solvers.

2 Laser Powder Bed Fusion

In laser powder bed fusion (LPBF), a laser beam melts a thin layer of metallic powder, which solidifies during cooling and bonds to the already processed solid material. The laser beam is controlled by a scanner which uses two rotating mirrors to direct the laser to the desired part of the surface. This way the laser can process any point on the surface, and the delay times requires to jump between two different areas is much shorter than the actual process time. After finishing one layer, a powder recoater covers the surface with a new thin powder layer, and the melting process iterates, until an entire three-dimensional object emerges. An illustration of this process can be found in Figure 2.

As the heat source provided by the beam is concentrated on a tiny point of the surface, the temperature of the whole object is usually very uneven. This uneven distribution leads to significant thermal stress and after cooling to warpage. Thermal stress is mainly affected

Figure 3: Example of an object subdivided into pixels.

by the material properties, the substrate height and the scanning strategy used by the laser beam [37]. As the material type is usually fixed for a given part and the substrate height is fixed by the machine, the best way to reduce thermal stress and thus warpage is to optimize the scanning strategy. In fact, in [13] the authors show that a simple heuristic on the scanning strategy can lead to a reduction of thermal stress of up to one third compared to a standard strategy. More examples of scanning strategies influencing warpage can be found in [54].

A common strategy for LPBF is called the island strategy. Here, the surface of the printing is divided into smaller surfaces called islands which are then processed consecutively to merge into the desired pattern. In [54] the authors use a heuristic approach to decide in which order these islands are processed. Common to all of these approaches is the use of heuristics to find a good solution, which is in some way better than a standard or random order.

In this section we present an approach which aims to compute mathematically optimal solutions for the printing order problem using a mixed-integer formulation with an integrated temperature distribution model. This model is an extension of the model presented in [6]. We discuss different methods to compute the temperature distribution and present several different objective functions for optimization. The computed results are used in a refined FEM simulation, which gives a good picture of the temperature distribution during printing and an evaluation of the thermal stress and warpage.

2.1 Printing Order

As we are printing with an island strategy, the printing order will describe the sequence in which these islands are processed. Therefore, we will section the surface of the object into equal squares or pixels, each of which describes one island. As the scanner can target any point of the surface at any given time without a significant delay, we can assume that any permutation of the islands is an admissible printing order. The path used by the scanner inside the island is usually preset by the given machine and the only information we are given is the printing time of a single island.

For the model we assume that the surface is divided into pixels $(i, j) \in \mathcal{P}$ (see Figure 3) and the time steps are given in a set $\mathcal{T} = \{1, \dots, |\mathcal{P}|\}$. The binary decision variable $x_{i,j,t}$ represents the printing decision for pixel (i, j) at time step t, with $x_{i,j,t} = 1$ if (i, j) is printed in time step t and $x_{i,j,t} = 0$ otherwise. We assume that each pixel of the surface must be printed at some time step of the process, which is described by the following equation.

$$\sum_{t \in \mathcal{T}} x_{i,j,t} = 1 \qquad \forall (i, j) \in \mathcal{P}. \tag{1}$$

Due to the production process, we can also assume that at every time step exactly one pixel is printed, which is represented as follows.

$$\sum_{(i,j)\in\mathcal{P}} x_{i,j,t} = 1 \qquad \forall t \in \mathcal{T}. \tag{2}$$

With these restrictions it is easy to see that the number of permutations of the pixels $|\mathcal{P}|!$ is equal to the number of admissible printing orders.

2.2 Temperature Calculation

The most common way to mathematically calculate the temperature distribution in a physical object is using Fourier's heat equation:

$$\frac{\partial\theta}{\partial t} - \alpha\nabla^2\theta = C, \tag{3}$$

where θ is the temperature function, α the thermal diffusivity of the material, and C is a heat source term. Furthermore, the operator ∇^2 describes the Laplacian operator, i.e., $(\frac{\partial^2}{\partial x_1^2} + \ldots + \frac{\partial^2}{\partial x_n^2})$. Given an initial temperature distribution at time 0 and heat sources depending on time and space, the heat distribution of the object is given by θ. In LPBF manufacturing we are dealing with a three-dimensional object in space over the duration of the printing process. Therefore, the function θ is of the form $\theta(x, y, z, t) : \mathbb{R}^4 \to \mathbb{R}$. While in some printing processes it makes sense to deal with the temperature distribution in two dimensions, i.e., neglecting the height (as we will see later for wire-arc additive manufacturing), in this case the build platform acts as a strong heat sink which makes the third dimension essential to modeling. However, a simpler two-dimensional model for LPBF was presented in [6].

Fourier's heat equation is an important example of a *parabolic partial differential equation* and has been widely studied since its introduction the 19th century. While there are many approaches to solve these equations analytically, we are interested in a numerical solution which can be embedded into a mixed-integer program. To this end, it is necessary to describe everything with linear (in-) equalities and the procedure should be able to deal with the heat source given by the laser beam. Furthermore, it is not essential that the computation be very precise, as we are mainly interested in the relative distribution of temperature in the object. However, as the overall optimization task will be computationally expensive, the subtask of updating the temperature distribution should be as efficient as possible.

All of these properties are fulfilled by finite differences schemes: They are made up of linear equalities, can deal with changing momentary heat sources, and can be computed quite efficiently. Moreover, this procedure pairs very well with the pixel structure imposed on the printing pattern by the island strategy described in Section 2.1.

Most finite differences schemes are based on the forward time central space (FTCS) and the backward time central space (BTCS) schemes. Being an explicit method, FTCS is computationally very efficient. However, the drawback is that it is only conditionally stable for Fourier's heat equation, i.e., if we use the same step size h for all three dimensions, it is only stable for:

$$\Delta t \leqslant \frac{h^4}{6\alpha}. \tag{4}$$

For the parameters used in our experiments (see Table 1) this condition is far from being fulfilled. The BTCS scheme is an implicit method and unconditionally stable, however, in contrast to the explicit method it is necessary to solve a system of linear equations. While there are several adaptations of BTCS which either improve computation time by generating more structured matrices (ADI) or which have more accurate solutions (Crank-Nicholson), our model computes most efficiently with the standard BTCS scheme.

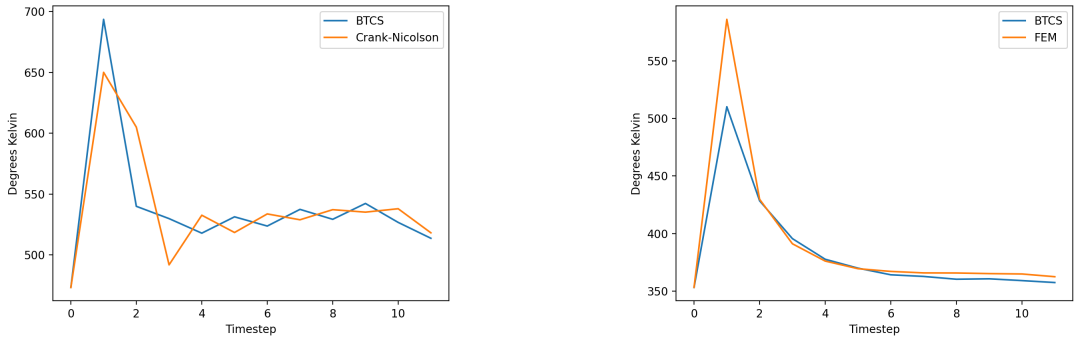

Figure 4: Here we see a comparison of the temperature development of one block for the nine time steps of the simulation of a 3×3 square, as computed by BTCS, CN and a simple FEM simulation. The temperature peak signifies the point at which the laser welds this particular pixel.

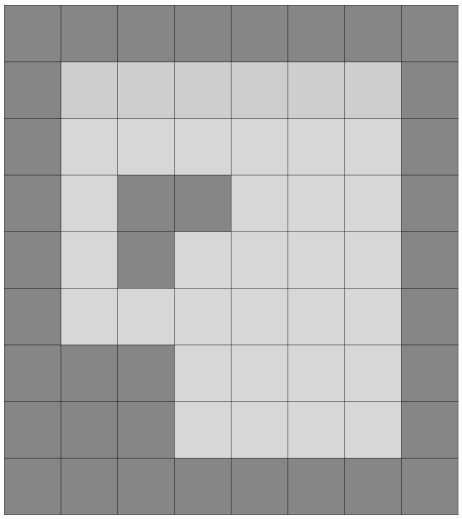

Figure 5: The model as seen from above. The pale pixels are part of the object $\mathcal{P}$ and the dark pixels form the surrounding powder.

Comparison in a simple 3×3 square have also shown that the results obtained by BTCS are very similar in behavior to those computed using CN and even by a simple FEM simulation (see Figure 4).

Following the method of Section 2.1 we partition the printing surface into pixels $(i, j) \in \mathcal{P}$ of size $\Delta x \times \Delta x$. The object is then separated into blocks $(i, j, k) \in \mathcal{P}_{obj}$ of size $\Delta x \times \Delta x \times \Delta z$ matching with the pixels of the surface. As the object is surrounded by powder, we embed $\mathcal{P}_{obj}$ in a larger cuboid $\mathcal{C}$ which is discretized in the same manner, i.e., $\mathcal{C} = \{(i, j, k) \ : \ i \in \{1, \dots, N\}, j \in \{0, \dots, M\}, k \in \{0, \dots, H\}\}$, where N denotes the number of steps in x-direction, M the number of steps in y- direction and H denotes the number of steps in z-direction. All blocks $\mathcal{E} = \mathcal{C} \backslash \mathcal{P}$ consist of the metal powder and are modeled with different material qualities. Only the blocks in the bottommost level, i.e., the set $\{(i, j, k) \in \mathcal{C} \ : \ k = 0\}$ are of the same material as the object itself and model the baseplate of the printing machine.

Now we introduce temperature variables $\theta_{i,j,k,t}$ for each block $(i, j, k) \in \mathcal{C}$ and each time step $t \in \mathcal{T}$. As the environment of the process is temperature controlled, we are given a fixed initial temperature Θ_0 in time step 0:

$$\theta_{i,j,k,0} = \Theta_0 \qquad \forall (i,j,k) \in \mathcal{C}. \tag{5}$$

For the other time steps we use a BTCS discretization scheme applied to the heat equation. In the top layer, i.e., the *printing layer*, the heat transfer is marginal into the surrounding gas. Thus, we assume for simplicity that transfer upwards in z-direction is 0. Furthermore, the temperature equations of the printing layer also include the heat source S which is activated or deactivated by the laser control-variables $x_{i,j,t}$:

$$\begin{aligned} \theta_{i,j,k,t-1} + x_{i,j,t} \cdot S = \\ \left(1 + 4\alpha\frac{\Delta t}{(\Delta x)^2}\right)\theta_{i,j,k,t} + \left(1 + \alpha\frac{\Delta t}{(\Delta z)^2}\right)\theta_{i,j,k,t} \\ - \alpha\frac{\Delta t}{(\Delta x)^2}\sum_{(i',j')\in N_{i,j}} \theta_{i',j',k,t} - \alpha\frac{\Delta t}{(\Delta z)^2}\theta_{i,j,k-1,t}, \end{aligned} \tag{6}$$

where $N_{i,j}$ is the set of blocks adjacent to (i,j) in the printing layer. For all other layers of the model the heat transfer is computed in all directions:

$$\begin{aligned} \theta_{i,j,k,t-1} = \\ \left(1 + 4\alpha\frac{\Delta t}{(\Delta x)^2}\right)\theta_{i,j,k,t} + \left(1 + 2\alpha\frac{\Delta t}{(\Delta z)^2}\right)\theta_{i,j,k,t} \\ - \alpha\frac{\Delta t}{(\Delta x)^2}\sum_{(i',j')\in N_{i,j}} \theta_{i',j',k,t} - \alpha\frac{\Delta t}{(\Delta z)^2}\left(\theta_{i,j,k+1,t} + \theta_{i,j,k-1,t}\right). \end{aligned} \tag{7}$$

Note that the heat transfer coefficient α is given here in a simplified form. It is reliant on the thermal diffusivity of the analyzed material. In this model we are usually given two types of material: the welded solid material and the powder. These two materials have very different thermal diffusivity, which is accounted for in the computations in a later section.

2.3 Objective Functions

The main difficulty for the construction of the model is the choice of the objective function. Simulating the resulting warpage of a printing process is very complicated, and even in a refined FEM simulation with fixed printing order the warpage can only be inferred by the resultant displacement. One natural way to circumvent this problem is to infer the behavior of the warpage from a feature that is easier to simulate, such as the temperature distribution. Uneven temperature distribution and successive cooling leads to thermal stress in the material, which is a key cause of warpage. From this observation an obvious choice for the optimization objective is the minimization of the temperature gradients between the blocks of the discretization, i.e., an objective of the form:

$$\min \frac{1}{T^2}\sum_{t\in\mathcal{T}}\sum_{(i,j)\in P}\sum_{(i',j')\in\mathcal{N}_{i,j}}\left|\frac{\theta_{i,j,\text{top},t} - \theta_{i',j',\text{top},t}}{\Delta x}\right|, \tag{8}$$

where we divide by T^2 to normalize by the number of time steps and pixels in the geometry. As the absolute value function is non-linear, we require additional values $\delta^+_{i,j,i',j',t}, \delta^-_{i,j,i',j',t} \in \mathbb{R}_+$ which represent the positive and the negative proportion of the absolute value function, respectively. We can now reformulate the above objective function as:

$$\min \frac{1}{T^2} \sum_{t \in \mathcal{T}} \sum_{(i,j) \in P} \sum_{(i',j') \in \mathcal{N}_{i,j}} \left(\delta^+_{i,j,i',j',t} + \delta^-_{i,j,i',j',t} \right), \tag{9}$$

with the additional constraint

$$\delta^+_{i,j,i',j',t} - \delta^-_{i,j,i',j',t} = \frac{\theta_{i,j,\text{top},t} - \theta_{i',j',\text{top},t}}{\Delta x} \quad \forall (i,j) \in \mathcal{P}, (i',j') \in \mathcal{N}_{i,j}, \; t \in \mathcal{T}. \tag{10}$$

Here we sum up only the gradients of the printing layer for the sake of simplicity. However, it is also possible to sum up over all blocks of the object. In the following we will refer to this objective as *Grad*.

This objective is not only complicated and involves the computation of many absolute values, one can also easily see by assigning the value $\frac{1}{|\mathcal{P}|}$ to each variable $x_{i,j,t}$ that the value of an LP-relaxation is very close to zero. This can be interpreted as spreading the energy of the laser beam evenly across the whole surface of the object. In fact, in practice this is a significant problem and for example, a test using the geometry seen in Figure 5 could not be solved with a gap exceeding 95%, even if run for multiple days. In [6] the authors use the same objective function for a much simpler two-dimensional model and have similar difficulties.

These circumstances show that different objectives are needed in order to solve this problem efficiently. One possibility is to evaluate the deviation in temperature from a given fixed target-temperature Θ^* and sum these up for all blocks in the printing layer and all time steps:

$$\min \frac{1}{T^2} \sum_{t \in T} \sum_{(i,j) \in \mathcal{P}} \left| \theta_{i,j,\text{top},t} - \Theta^* \right|, \tag{11}$$

where we again divide by T^2 to normalize by the number of time steps and pixels in the geometry and the absolute value function is linearized just as in the objective *Grad*.

As can be see in the computational results presented in the next section, this already significantly improves running times and yields good results for the printing order. In the following we will refer to this objective as *Dev*. However, in order to make the computations more efficient we also present the following objective:

$$\min \frac{1}{T^2} \sum_{t \in T} \sum_{(i,j) \in \mathcal{P}} \theta_{i,j,\text{top},t}. \tag{12}$$

In this function we sum up all temperatures in the printing layer for all blocks and all time steps. In the following we will refer to this objective as *Sum*. While at first glance the value computed here seems trivial, i.e., minimizing the amount of energy added to the surface layer by the laser beam, this can also be interpreted as maximizing the energy passed to the boundary in form of the powder bed and the baseplate. As the printing layer will always be the hottest part of the object, it creates a large amount of the thermal stress within the object. Therefore, choosing a printing order which keeps this area as cool as possible can make an impact on the resulting warpage. This is the case even more so if the object is very irregular in the z-direction for example, in the form of an overhang. In this case, the material on the overhang cannot pass energy towards the baseplate as efficiently as material which has a direct connection to the baseplate. We will see in the following section that this objective yields much faster computation times than both of the other presented objectives.

parameter	unit	value	description
Δx	mm	6	edge length of a pixel
Δz	mm	1.3 and 0.13	height of a block
Δt	s	3.6864	time step
λ	$\mathrm{W}/(\mathrm{m}\cdot\mathrm{K})$	15	thermal conductivity
ρ	kg/m^3	8000	density
c	$\mathrm{J}/(\mathrm{kg}\cdot\mathrm{K})$	500	specific heat capacity
α	m^2/s	3.75×10^{-6}	thermal diffusivity
α_{powder}	m^2/s	$0.03\cdot\alpha$	thermal diffusivity unwelded powder
τ	W	250	laser power
Θ_0	K	773.15	initial temperature
Θ^*	K	973.15	target temperature

Table 1: Parameters and values used for computations.

2.4 Computational Results

In order to test our model we used a standard setup for the printing process. The material chosen was stainless steel of type 1.4571 whose material properties can be found in Table 1. The shapes to be printed were designed to be asymmetrical, with some thinner areas and a hole. Both instances can be found in Figure 6 with their respective surface and pixel structure. More irregular shapes are more likely to warrant a customized printing order and edges and holes are likely to cause more thermal stress. We use an edge length for each pixel of 6mm. With regard to the height, we tested two different approaches. The height of the printed object is chosen as 1.3 mm. As an actually printed layer has a height of approximately 30 µm, we would have to model about 45 layers to reach a height of 1.3 mm. Simulating a thinner object is not as interesting, as the temperature needs several rounds of printing to accumulate in the object. Therefore, we first use one height layer (plus the boundary condition representing the baseplate) and a heat source which is adjusted to this height. This approximates the process in which all printing layers use the same order. In a second approach, we use ten height layers of 0.13 mm each, and optimize the strategy for each of these subsequently. This approximates the process where about 4-5 subsequent layers use the same strategy and then a new strategy is computed using the previous temperature distribution. We assume a laser beam of 250 W and as a starting temperature we choose 773.15 K. The heat source is computed using the laser energy and is adjusted for the time step and pixel size.

These setups were then computed on the two different geometries (*Geometry A* and *Geometry B* in Figure 6) for all presented objectives. The strategies denoted by *big* correspond to the setup using one layer of 1.3 mm. The strategies denoted by *layer* correspond to the setup where ten layers of 0.13 mm were used and are numbered for the layers processed. The different objectives are denoted by *grad*, *dev* and *sum*. The computations were made using IBM ILOG CPLEX 20.1.0.0 [30] with a time limit of 3600 s for the *layer-dev* and *layer-sum* strategies and a time limit of 36 000 s for the *big-dev* and *big-sum* strategies. The LP-method was set to *parallel* and the *numerical emphasis* was switched on to improve the performance of LP relaxation. In the case of the *layer-dev* and *layer-sum* strategies, we used MIP-starts in CPLEX to pass the solution of a previous layer to the next computation. All computations were performed on a Linux system with a Intel Xeon Gold 6136 CPU at 3 GHz using up to 32 kernels and 240 GiB RAM.

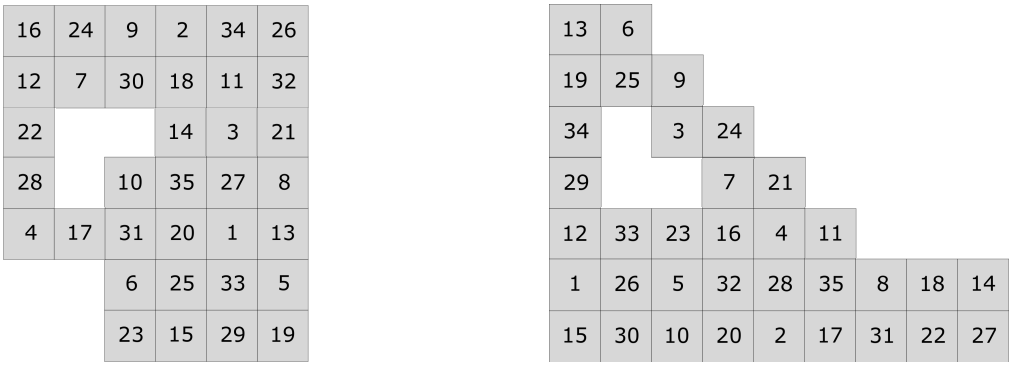

Figure 6: The image on left represents the surface of Geometry A, and on the right we see Geometry B. The strategy *big-dev* is given by the numbering on the pixels.

Strategy	***Geometry A***			***Geometry B***		
	runtime	gap	value	runtime in s	gap	value in K
big-grad	36000	95.50%	121184.19	3600	94.47%	114286.61
big-dev	36000	1.20%	391.58	36000	1.45%	393.46
layer1-dev	2.12	0%	197.29	3.72	0%	197.29
layer2-dev	3.85	0%	194.58	4.99	0%	194.58
layer3-dev	3602	2.66%	197.12	3601	2.66%	197.13
layer4-dev	3600	5.23%	199.61	3601	5.23%	199.62
layer5-dev	3603	7.67%	201.97	3611	7.68%	201.99
layer6-dev	3603	10.00%	204.2	3601	10%	204.22
layer7-dev	3604	12.2%	206.26	3620	12.21%	206.30
layer8-dev	3604	14.29%	208.15	3601.73	14.3%	208.2
layer9-dev	3607	16.26%	209.85	3627.77	16.27%	209.92
layer10-dev	3602	18.13%	211.38	3600	–	–
big-sum	1.43	0%	1039.9	1.49	0%	1038.78
layer1-sum	0.41	0%	775.86	0.55	0%	775.86
layer2-sum	3.31	0%	778.57	5.47	0%	778.57
layer3-sum	7.73	0%	781.27	15.31	0%	781.27
layer4-sum	82.35	0%	783.98	125.82	0%	783.97
layer5-sum	130.04	0%	786.67	81.46	0%	786.67
layer6-sum	52.65	0%	789.37	165.17	0%	789.36
layer7-sum	51.05	0%	792.06	113.87	0%	792.04
layer8-sum	141.87	0%	794.74	347.90	0%	794.72
layer9-sum	99.68	0%	797.42	448.05	0%	797.40
layer10-sum	709.81	0%	800.09	360.69	0%	800.07

Table 2: Computational results displaying runtime, optimality gap, and objective value for all constructed models. Note that *layer10-dev* did not compute a solution in the defined time frame of 3600s.

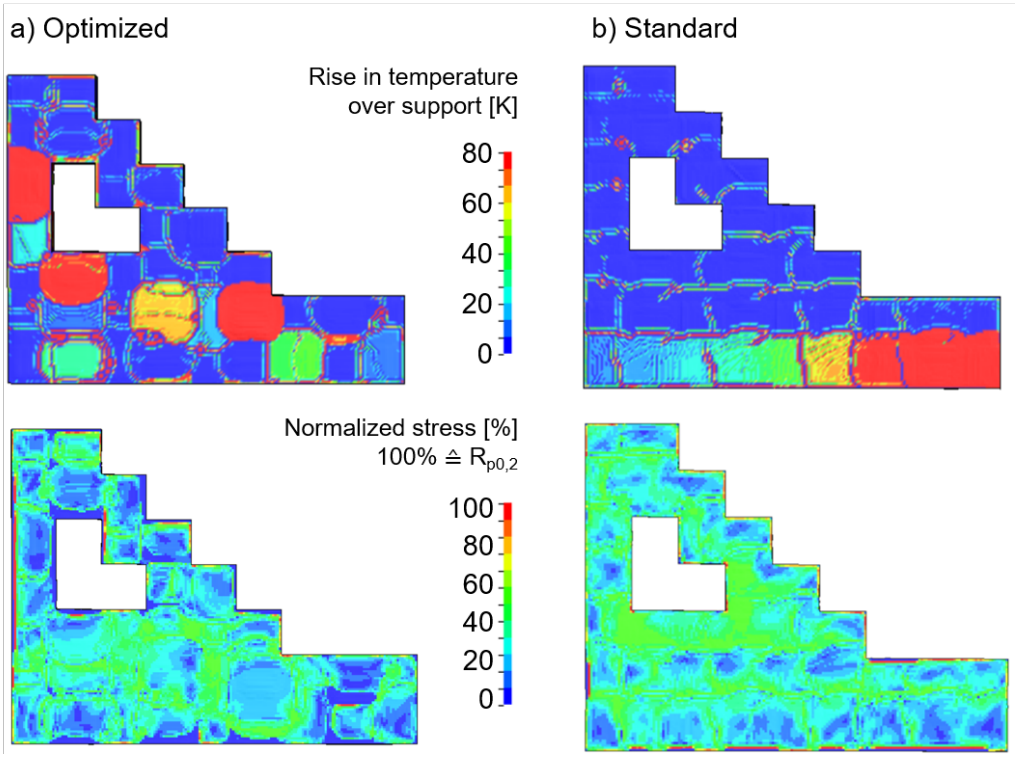

Figure 7: A visualization of the temperature distribution and the normalized stress in the last step of the process using both the optimized and a standard strategy on Geometry B.

The computational results in Table 2 for the *big* model clearly show that the objective *Grad* leads to very inaccurate results when executed in the fixed time span of 36 000 s, while the optimality gap for *Dev* is quite low in the same time frame. Objective *Sum* on the other hand computes to optimality in less than two seconds. In the *layer* model we can see the effect of the inclusion of multiple height layers. Note that the use of solutions when moving from layer_i to layer_{i+1} leads to some jumps in running time, as can be seen for *layer5-sum* and *layer6-sum*. However, apart from this the computation time jumps in an order of magnitude every other layer. This is an issue when computing with a large amount of layers and further motivates, why we did not use the printing height of 30 µm.

To validate the results of the optimization, the computed printing orders were used in a refined FEM simulation which can generate the temperature distribution, as well as material displacement during the printing process. Here, we present the results for the orders generated by the *big-dev* computation for both presented geometries. A visualization of these printing orders can be found in Figure 6. Each of these results is compared to a standard printing order which prints the objects in stripes moving from left to right and from to p to bottom. In the following, the temperature distribution directly after the last exposure step is shown. Since the heat is distributed very quickly, the scale is chosen in such a way that only a temperature increase up to 80 K due to the heat source is resolved.

For Geometry B the temperature distribution and the normalized stress for the optimized and the standard strategy are given in Figure 7. We see that for Geometry B the temperature distribution is more uniform with the optimized strategy, while with the standard strategy the temperatures are much higher in the lower half of the object. Similarly, the stress normalized to the yield stress is slightly reduced for the optimized printing sequence.

The displacement produced by the temperature distribution (see Figure 8) shows clear differences between the two strategies. First, the local z-shift is particularly visible in the

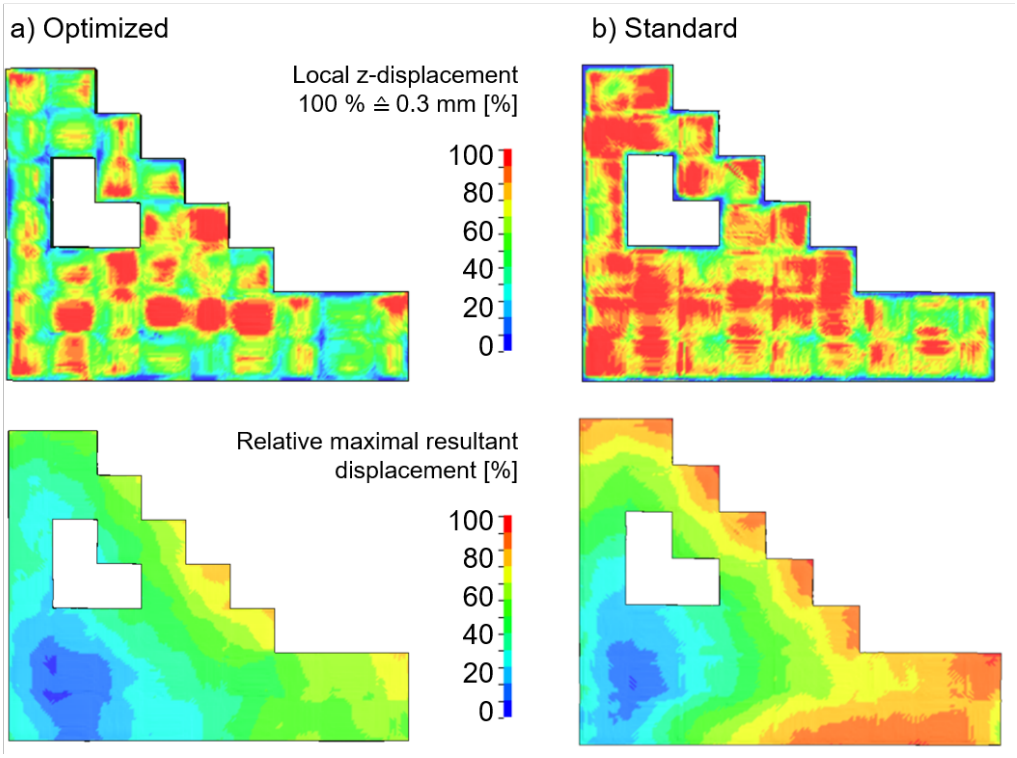

Figure 8: A visualization of the z-displacement and the relative maximal resultant displacement in the last step of the process using both the optimized and a standard strategy on Geometry B.

last processed pixels, which can be seen as red dots. With the standard strategy, the local z-shift can be seen in most areas of the surface. The the highest relative maximal resultant displacement of 100% occurs in the the standard strategy. This resultant displacement is essentially an aggregate of the displacement in all three dimensions and is only about 75% for the optimized strategy in comparison to the standard strategy and is much better distributed in the optimized strategy. For the standard strategy, the maximum value is in the lower right corner, which is the last area to be processed by this strategy.

For Geometry A we see a similar picture with regard to the temperature distribution (see Figure 9). Due to the processing in stripes of the standard strategy, the lower half has much higher temperatures than the top, dividing the object into two temperature zones, while the optimized strategy is again more even. When analyzing the normalized stress, however, the standard strategy seems to have a slight advantage.

In Figure 10 we see that this slight advantage in normalized stress also translates to the displacement. Both z-displacement and relative maximal resultant displacement are slightly higher in the optimized strategy, especially in the corner on the left. As the temperature distribution is in fact better than that of the standard strategy, this seems to imply that displacement (and thus warpage) cannot be inferred from temperature distribution alone, but is also somewhat dependent on the geometry of the printed object.

As the strategy was computed with the *Dev*-objective, the goal of the computation was to heat the object as close to a preassigned temperature as possible. While this seems to have been successful, for some objects this does not necessarily lead to a reduction of the displacement.

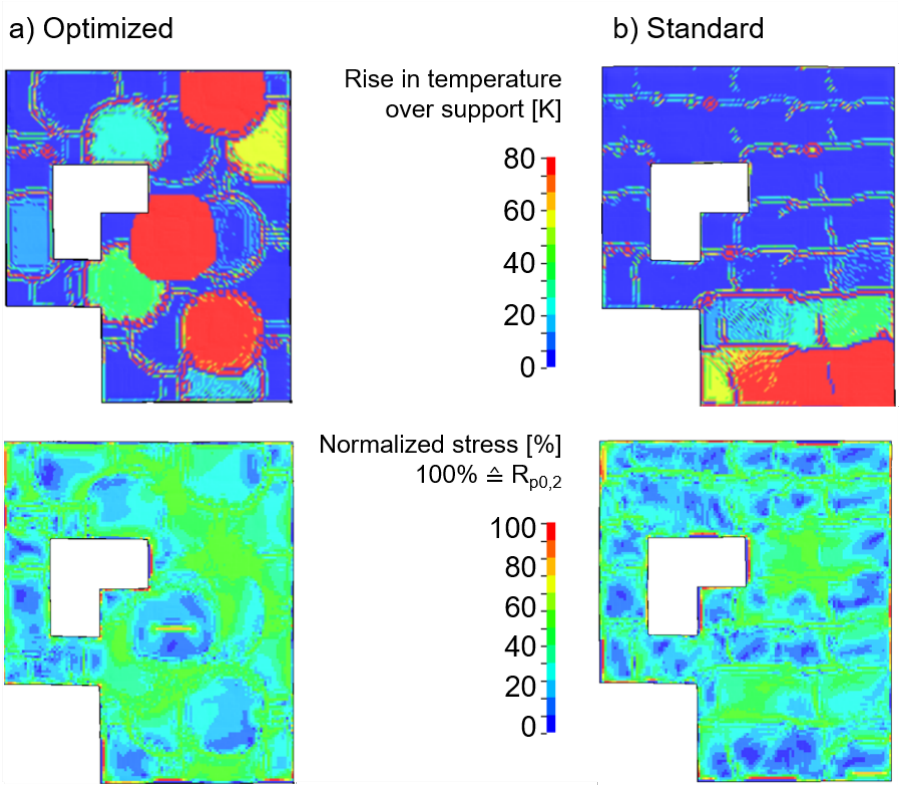

Figure 9: A visualization of the temperature distribution and the normalized stress in the last step of the process using both the optimized and a standard strategy on Geometry A.

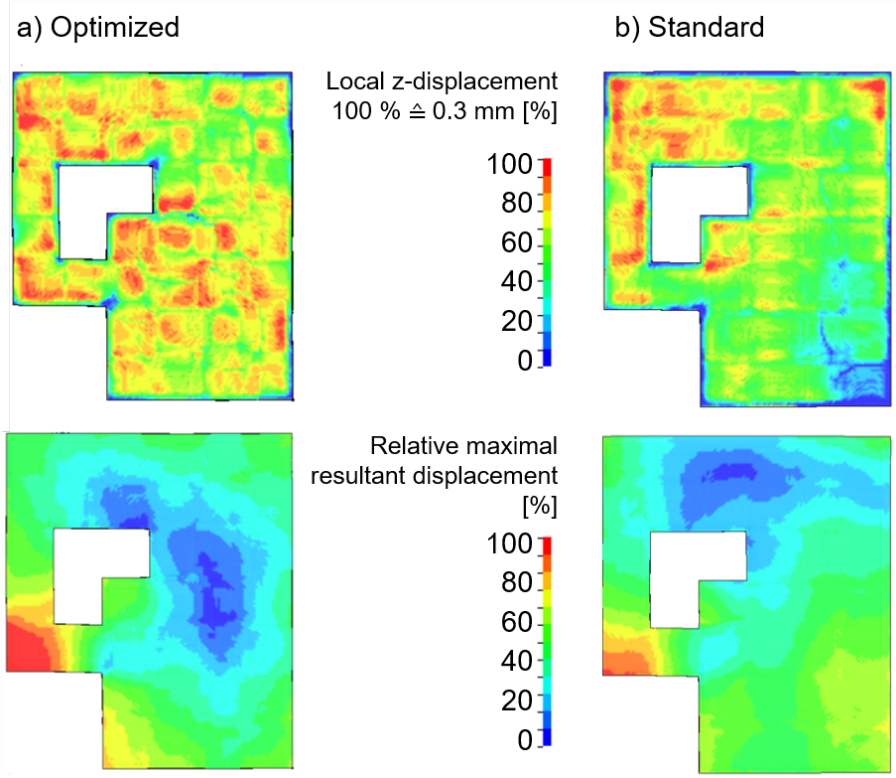

Figure 10: A visualization of the z-displacement and the relative maximal resultant displacement in the last step of the process using both the optimized and a standard strategy on Geometry A

3 Wire-Arc Additive Manufacturing

In the process of Wire-Arc Additive Manufacturing, the desired workpiece is built by progressive deposition of weld beads on an underlying substrate using a weld source moving around the working area freely. The wire is molten with high energy from an electrical arc and deposited in droplets to produce weld beads. Although the process does not necessarily require layer-by-layer deposition, we will consider only cases that can be accomplished by slicing the part geometry and depositing material layer-by-layer. For a single layer, the welding trajectory should be continuous since every time the weld source is moved without welding, there is a chance to introduce bondin defects and the quality of the resulting workpiece reduces. On the contrary, welding parts of the layer more than once leads to material accumulation which affects the shape of the workpiece and increases post-processing efforts. Furthermore, the high temperature of the weld source can cause large temperature gradients with its surrounding, resulting in strain distribution in the welded material, which can lead even to cracks. Thus it is desirable to achieve a homogeneous temperature distribution within the workpiece by adjusting the welding trajectory. Taking these aspects into account, careful planning of the welding trajectory is crucial for process efficiency. In this work, we consider only workpieces with wall thicknesses larger than the width of the weld bead. Since for areas filled with material there are many possible ways to manufacture them [34], we assume that the path strategy is given and only the sequence of the single welding moves can be optimized. A study about thin-walled structures with wall thickness as broad as the width of the weld bead can be found in [4].

3.1 Path Generation

The maximum velocities of the weld source while welding $v^w \in \mathbb{R}_+$ and for transiting $v^m \in \mathbb{R}_+$ are given parameters of the welding process. Let Δt denote the length of one discrete time step. We consider a given geometry of a two-dimensional layer as a graph with nodes $i \in \mathcal{V}$ at the coordinates $\vec{r}_i \in \mathbb{R}^2$ of every intersection point between two welding segments and edges $(i,j) \in \mathcal{W}$ describing the part of the respective welding segment between nodes $i \in \mathcal{V}$ and $j \in \mathcal{V}$ with length $l_{i,j} \in \mathbb{R}_+$. Let in the following $\mathcal{V}^{odd} \subseteq \mathcal{V}$ and $\mathcal{V}^{even} = \mathcal{V} \backslash \mathcal{V}^{odd}$ denote the set of nodes with odd and even node degree, respectively. In this setting we assume that every welding segment $(i,j) \in \mathcal{W}$ is printed at once and transition moves can only be performed between the nodes. To reduce defects and increase the quality of the workpiece, their number should be minimized. Since every segment must be welded to process the whole layer, the problem of finding a feasible welding trajectory can be seen as a Chinese postman problem [20] in the graph $G = (\mathcal{V}, \mathcal{W})$. For this problem it is known that, if necessary, additional edges must be inserted between nodes of odd node degree to keep their number minimal. If G is not connected and contains more than one component, this holds for the trajectory within every component and the transition between two components containing nodes of odd node degree. For components without nodes of odd node degree, every node is a possible start- or endpoint for them. Let $\nu \in \mathbb{N}$ denote the number of components of G, $\mathcal{V}_i^{odd}$ and $\mathcal{V}_i^{even}$ the sets of nodes with odd or even node degree in component $i \in \{1, \dots, \nu\}$, $\mathcal{I} \subseteq \{1, \dots, \nu\}$ the set of all components with $\mathcal{V}_i^{odd} \neq \varnothing$, and $\mathcal{V}^{tran} = \left(\bigcup_{i \in \mathcal{I}} \mathcal{V}_i^{odd}\right) \cup \left(\bigcup_{i \notin \mathcal{I}} \mathcal{V}_i^{even}\right)$. Thus, all transition moves are also restricted to the set $\mathcal{U} = \mathcal{V}^{tran} \times \mathcal{V}^{tran}$ and the trajectory must start in a node $i \in \mathcal{V}^{tran}$, otherwise another transition move is required. To incorporate transition moves into the model, let $\omega \in \mathbb{N}$ be their minimal number required to process the complete layer and $d_{i,j}^e = \|\vec{r}_i - \vec{r}_j\|_2$ the euclidean distance between nodes $(i,j) \in \mathcal{U}$. Note that ω also contains transition moves between components of the graph, if it is not connected.

For every welding segment $(i,j) \in \mathcal{W}$, the number of time steps to process it is given

by $\tau^w_{i,j} = \left\lceil \frac{l_{i,j}}{v^w \Delta t} \right\rceil$ and the whole layer is processed in $T^{proc} = \sum_{(i,j)\in\mathcal{W}} \tau^w_{i,j}$ time steps. In a similar way, the number of necessary time steps to perform a transition move between nodes $i, j \in \mathcal{V}$ is computed by $\tau^m_{i,j} = \left\lceil \frac{d^e_{i,j}}{v^m \Delta t} \right\rceil$. Due to the varying length of the transition moves, the overall time to perform them is not known apriori, but it can be overestimated by $T^{tran} = \omega \max_{i,j\in\mathcal{V}} \tau^m_{i,j}$. Overall, the discrete time horizon is given by $\mathcal{T} = \{1, \dots, T^{max}\}$ with $T^{max} = T^{proc} + T^{tran}$. As abbreviations we use $\mathcal{T}_0 = \mathcal{T} \cup \{0\}$, $\mathcal{T}^- = \mathcal{T} \backslash \{T^{max}\}$, and $\mathcal{T}^{end} = \{T^{proc} + \omega \min_{i,j\in\mathcal{V}} \tau^m_{i,j}, \dots, T^{max}\}$, the set of discrete time steps where the process could finish, depending on the length of the necessary transition moves.

Since every segment can be processed in both directions, the edge set $\mathcal{W}$ is expanded to $\overline{\mathcal{W}} = \{(i,j) \in \mathcal{V} \times \mathcal{V} \mid (i,j) \in \mathcal{W} \vee (j,i) \in \mathcal{W}\}$ and the number of time steps for processing is adjusted to $\tau^w_{i,j} = \tau^w_{j,i}$ for $(i,j) \in \overline{\mathcal{W}}$. Relating all possible connections with their respective processing time, we obtain the sets $\mathcal{W}^* = \{(i,t_i,j,t_j) \in \mathcal{V} \times \mathcal{T}_0 \times \mathcal{V} \times \mathcal{T} \mid (i,j) \in \overline{\mathcal{W}},\ t_j = t_i + \tau^w_{i,j}\}$ for the welding moves and $\mathcal{U}^* = \{(i,t_i,j,t_j) \in \mathcal{V} \times \mathcal{T} \times \mathcal{V} \times \mathcal{T} \mid (i,j) \in \mathcal{U},\ t_j = t_i + \tau^m_{i,j}\}$ for all possible transition moves. Note that transition moves cannot occur in the first time step since if this happens, there is a feasible welding trajectory starting at the end point of this transition with a smaller ω and less processing time. Binary variables $w_{i,t_i,j,t_j} \in \{0,1\}$, indicating if the weld source moves from node $i \in \mathcal{V}$ to node $j \in \mathcal{V}$ from time step $t_i \in \mathcal{T}_0$ to time step $t_j \in \mathcal{T}$, are used to track the welding trajectory. For the transition moves, there are binary variables $u_{i,t_i,j,t_j} \in \{0,1\}$, equal to one if and only if the weld source moves between node $i, j \in \mathcal{V}$ from time step $t_i \in \mathcal{T}$ to time step $t_j \in \mathcal{T}$ without welding. Due to the varying number of time steps for the transition moves, further binary variables $u_{i,t} \in \{0,1\}$, indicating whether the welding trajectory ends in node $i \in \mathcal{V}^{tran}$ at time step $t \in \mathcal{T}^{end}$, are required. Thus, a feasible welding trajectory is described by the following constraints: The weld source must start its path at some node

$$\sum_{\substack{i,j,t_j:(i,0,j,t_j)\in\mathcal{W}^* \\ i\in\mathcal{V}^{tran}}} w_{i,0,j,t_j} = 1, \tag{13}$$

$$\sum_{j,t_j:(i,0,j,t_j)\in\mathcal{W}^*} w_{i,0,j,t_j} = 0 \qquad \forall i \in \mathcal{V}\backslash\mathcal{V}^{tran}. \tag{14}$$

Furthermore, every segment must be processed

$$\sum_{t_i,t_j:(i,t_i,j,t_j)\in\mathcal{W}^*} w_{i,t_i,j,t_j} + \sum_{t_j,t_i:(j,t_j,i,t_i)\in\mathcal{W}^*} w_{j,t_j,i,t_i} = 1 \qquad \forall (i,j) \in \mathcal{W}. \tag{15}$$

The resulting trajectory must be continuous

$$\begin{aligned} &\sum_{k,t_k:(k,t_k,i,t)\in\mathcal{W}^*} w_{k,t_k,i,t} + \sum_{k,t_k:(k,t_k,i,t)\in\mathcal{U}^*} u_{k,t_k,i,t} \\ &\quad = \sum_{j,t_j:(i,t,j,t_j)\in\mathcal{W}^*} w_{i,t,j,t_j} + \sum_{j,t_j:(i,t,j,t_j)\in\mathcal{U}^*} u_{i,t,j,t_j} \\ &\qquad\qquad \forall i \in \mathcal{V}^{tran},\ t \in \mathcal{T}\backslash\mathcal{T}^{end}, \end{aligned} \tag{16}$$

$$\begin{aligned} &\sum_{k,t_k:(k,t_k,i,t)\in\mathcal{W}^*} w_{k,t_k,i,t} + \sum_{k,t_k:(k,t_k,i,t)\in\mathcal{U}^*} u_{k,t_k,i,t} \\ &\quad = \sum_{j,t_j:(i,t,j,t_j)\in\mathcal{W}^*} w_{i,t,j,t_j} + \sum_{j,t_j:(i,t,j,t_j)\in\mathcal{U}^*} u_{i,t,j,t_j} + u_{i,t} \\ &\qquad\qquad \forall i \in \mathcal{V}^{tran},\ t \in \mathcal{T}^{end}, \end{aligned} \tag{17}$$

$$\begin{aligned} &\sum_{k,t_k:(k,t_k,i,t)\in\mathcal{W}^*} w_{k,t_k,i,t} = \sum_{j,t_j:(i,t,j,t_j)\in\mathcal{W}^*} w_{i,t,j,t_j} \\ &\qquad\qquad \forall i \in \mathcal{V}\backslash\mathcal{V}^{tran},\ t \in \mathcal{T}, \end{aligned} \tag{18}$$

Transition moves cannot be used consecutively since they then can be merged to a single one

$$\sum_{k,t_k:(k,t_k,i,t)\in\mathcal{U}^*} u_{k,t_k,i,t} + \sum_{j,t_j:(i,t,j,t_j)\in\mathcal{U}^*} u_{i,t,j,t_j} \leqslant 1 \qquad \forall i \in \mathcal{V}^{tran},\ t \in \mathcal{T}. \tag{19}$$

Finally, the number of end nodes and transition moves is limited by

$$\sum_{i\in\mathcal{V}^{tran}} \sum_{t\in\mathcal{T}^{end}} u_{i,t} = 1, \tag{20}$$

$$\sum_{(i,t_i,j,t_j)\in\mathcal{U}^*} u_{i,t_i,j,t_j} = \omega. \tag{21}$$

3.2 Temperature Calculation

Considering the temperature distribution within the workpiece, it is affected by the heat input of the weld source, heat conduction, convection, and heat radiation. The progression of the temperature at every point within the layer can be described by the two-dimensional heat equation

$$\frac{\partial\theta}{\partial t}(x,y,t) = \alpha\left(\frac{\partial^2\theta}{(\partial x)^2}(x,y,t) + \frac{\partial^2\theta}{(\partial y)^2}(x,y,t)\right) + q(x,y,t)$$
$$\forall (x,y)\in\Omega,\ t\in(0,T], \tag{22a}$$

$$\frac{\partial\theta}{\partial n}(x,y,t) = \kappa^e\left(\varphi^{add} - \theta(x,y,t)\right) \qquad \forall (x,y)\in\partial\Omega,\ \forall t\in[0,T], \tag{22b}$$

$$\theta(x,y,0) = \theta^{init}(x,y) \qquad \forall (x,y)\in\Omega, \tag{22c}$$

with thermal diffusivity $\alpha \in \mathbb{R}_+$, an artificial cooling factor $\kappa^e \in [0,1]$ and initial temperature distribution $\theta^{init} : \Omega \to \mathbb{R}_+$. We assume the heat source function $q(x,y,t) : \Omega\times[0,T]\to\mathbb{R}_+$ to be piece-wise constant. The Robin boundary (22b) is used to approximate the heat exchange with the environment by heat radiation and convection. It has the slope κ^e and an additive constant $\varphi^{add}\in\mathbb{R}_+$.

To transform the partial differential equation system (22) into the discrete framework, we apply the finite element method (FEM) according to [60]. Along every welding segment $(i,j)\in\mathcal{W}$, $\tau^w_{i,j}-1$ equidistantly distributed discretization points are added and stored in the set $\mathcal{V}^{int}$. Let in the following $\overline{\mathcal{V}} = \mathcal{V}\cup\mathcal{V}^{int}$ denote the set of all nodes, $n = |\overline{\mathcal{V}}|$, and $\xi : \mathcal{V}^{int}\to\mathcal{W}\times\{1,\dots,\tau^w_{i,j}\}$ the function assigning every interior node to its position. Then, the FEM is set up with the node set $\overline{\mathcal{V}}$ as discretization points, the shape of the considered geometry as its boundary, and linear triangle elements between the nodes. Using the implicit time approach, it results in the linear equation system

$$(M+\Delta tK)\vec{\theta}_{t+1} = \Delta t(\vec{q}_{t+1}\vec{f^H} + \kappa^e\varphi^{add}\vec{f^R}) + M\vec{\theta}_t, \tag{23}$$

with mass matrix $M = (m_{i,j})\in\mathbb{R}^{n\times n}$, stiffness matrix $K = (k_{i,j})\in\mathbb{R}^{n\times n}$, and load vectors $\vec{f^H}\in\mathbb{R}^n$, $\vec{f^R}\in\mathbb{R}^n$. Note that the stiffness matrix is computed by $K = \alpha K^S + \kappa^e K^R$, where $K^S\in\mathbb{R}^{n\times n}$ and $K^R\in\mathbb{R}^{n\times n}$ are the effects of (22a) and (22b), respectively. The heat input in all nodes at time step $t\in\mathcal{T}_0$ is given by the vector $\vec{q}_t\in\mathbb{R}^n_+$ and the vector $\vec{\theta}_t$ consists of the variables $\theta_{i,t}\in\mathbb{R}_+$, describing the temperature of node $i\in\mathcal{V}$ at time step $t\in\mathcal{T}_0$. Furthermore, we use f^H_i, f^R_i, and $q_{i,t}$ to denote element $i\in\mathcal{V}$ of the above defined vectors.

For the weld source, we use the piece-wise constant approximation of the Goldak heat source model derived in [4]. It assumes the area of effect of the weld source to be circular with a homogeneous energy distribution in every direction and splits it into K^w non-overlapping rings, where a constant proportion $\kappa^w_1 > \kappa^w_2 > \kappa^w_{K^w}$ of the maximum welding

temperature φ^w is added to every node within it. Every ring is identified with the interval $\mathcal{P}_k,\ k = 1, \dots, K^w$, given by the minimum and the maximum distance from the weld source, where the factor κ_k^w applies. Due to the choice of the nodes $\overline{\mathcal{V}}$, the heat source is centered above one node at every time step. Thus, the heat input vector $\vec{q}_t$ is given by

$$q_{i,t} = \varphi^w \left(w_{i,t} + \sum_{k=1}^{K^w} \sum_{\substack{j \in \overline{\mathcal{V}} \\ d_{i,j}^e \in \mathcal{P}_k}} \kappa_k^w w_{j,t} \right) \qquad \forall i \in \overline{\mathcal{V}},\ t \in \mathcal{T}_0, \tag{24}$$

where $w_{i,t}$ is an abbreviation for

$$w_{i,t} = \begin{cases} 0 & , i \in \overline{\mathcal{V}} \backslash \mathcal{V}^{odd}, \\ & \quad t = 0, \\ \sum_{j,t_j:(i,0,j,t_j) \in \mathcal{W}^*} w_{i,0,j,t_j} & , i \in \mathcal{V}^{odd},\ t = 0, \\ \sum_{\substack{(h,t_h,j,t_j) \in \mathcal{W}^* \\ \xi(i)=(h,j,k) \\ t=t_h+k}} w_{h,t_h,j,t_j} + \sum_{\substack{(j,t_j,h,t_h) \in \mathcal{W}^* \\ \xi(i)=(h,j,k) \\ t=t_h-k}} w_{j,t_j,h,t_h} & , i \in \mathcal{V}^{int},\ t \in \mathcal{T}, \\ \sum_{h,t_h:(h,t_h,i,t) \in \mathcal{W}^*} w_{h,t_h,i,t} + \sum_{h:(h,t_h,i,t) \in \mathcal{U}^*} u_{h,t_h,i,t} & , i \in \mathcal{V}, t \in \mathcal{T}. \end{cases} \tag{25}$$

Discretizing the initial temperature distribution $\theta^{init}(x, y)$ to $\theta_i^{init} \in \mathbb{R}_+$ describing the initial temperature of node $i \in \overline{\mathcal{V}}$, the temperature distribution within the layer is calculated by

$$\theta_{i,0} = \theta_i^{init} \qquad \forall i \in \overline{\mathcal{V}}, \tag{26}$$

$$\sum_{j \in \overline{\mathcal{V}}} (m_{i,j} + \Delta t k_{i,j}) \theta_{j,t} = \sum_{j \in \overline{\mathcal{V}}} m_{i,j} \theta_{j,t-1} + \Delta t \left(q_{i,t} f_i^H + \kappa^e \varphi^{add} f_i^R \right) \qquad \forall i \in \overline{\mathcal{V}},\ t \in \mathcal{T}. \tag{27}$$

To obtain a flexible objective function, we minimize the absolute deviation from a given target temperature $\theta_{i,t}^{tar}$ in node $i \in \overline{\mathcal{V}}$ at time step $t \in \mathcal{T}_0$. With this approach, different goals can be achieved by choosing appropriate values. If the target temperature is chosen constant and equal for all nodes and time steps, a welding trajectory with a homogeneous temperature distribution is preferred. Specified material properties can be obtained by setting the target temperature for the desired nodes to the necessary temperature progression. The objective function is given by

$$\min \qquad \sum_{i \in \overline{\mathcal{V}}} \sum_{t \in \mathcal{T}_0} |\theta_{i,t} - \theta_{i,t}^{tar}|. \tag{28}$$

As we have seen in Section 2, additional variables $\vartheta_{i,t}^+, \vartheta_{i,t}^- \in \mathbb{R}_+$ are required to linearize the absolute value function. These represent the positive and the negative proportion of the absolute value function, respectively. Then, the linearized objective function is

$$\min \qquad \sum_{i \in \overline{\mathcal{V}}} \sum_{t \in \mathcal{T}_0} \left(\vartheta_{i,t}^+ + \vartheta_{i,t}^- \right), \tag{29}$$

with the additional constraint

$$\theta_{i,t} - \theta_{i,t}^{tar} = \vartheta_{i,t}^+ + \vartheta_{i,t}^- \qquad \forall i \in \overline{\mathcal{V}},\ t \in \mathcal{T}_0. \tag{30}$$

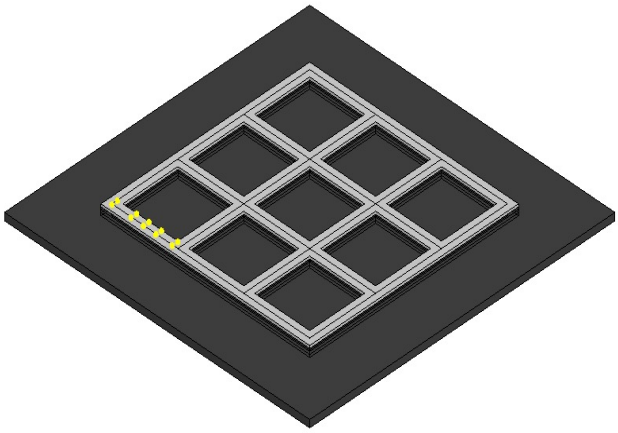

Figure 11: Geometry used to estimate the parameters of the model using the yellow marked points.

3.3 Parameter Estimation

The parameters φ^w, κ^e and φ^{add} are artificial parameters that cannot be identified to physical or material parameters or contain several of them. To achieve good values for them, a parameter estimation is necessary. Therefore, the geometry displayed in Figure 11 was simulated in LS-DYNA with a predefined welding trajectory for every layer, thermal diffusivity $\alpha = 15.6\frac{\text{W}}{\text{mm }^\circ\text{C}}$, and time step length $\Delta t = 0.5\text{s}$. Basic information about the AM-modeling technique with death-birth elements in the simulation environment of LS-DYNA can be found in [31]. The geometry with the chosen points used for the calibration is displayed in Figure 11. To achieve data of the process while in steady state, the temperature of the 10th layer is taken as desired temperature distribution which should be approximated by the mathematical model using a weighted absolute deviation. Let $i \in \mathcal{V}_M$ and $\theta_{i,t}^{sim} \in \mathbb{R}_+$ denote the set of nodes which are identified with the chosen points and their simulated temperature data at time step $t \in \mathbb{T}_0$, respectively. Since the welding trajectory is fixed, only φ^w, κ^e, φ^{add} and the temperature $\theta_{i,t}$ of every node $i \in \mathcal{V}$ at time step $t \in \mathcal{T}_0$ remain as variables. The relevant area of the geometry is modeled using the temperature data of the outer chosen points as boundary set $\mathcal{V}_B$ with a Dirichlet boundary condition. Thus, the optimization model simplifies to

$$\min \sum_{i\in\mathcal{V}_M}\sum_{t\in\mathcal{T}_0} \frac{\theta_{i,t}^{sim}}{M}|\theta_{i,t} - \theta_{i,t}^{sim}|, \tag{31a}$$

$$\theta_{i,0} = \theta_i^{init} \qquad \forall i \in \overline{\mathcal{V}}, \tag{31b}$$

$$\sum_{j\in\overline{\mathcal{V}}}(m_{i,j} + \Delta t k_{i,j})\theta_{j,t} = \sum_{j\in\overline{\mathcal{V}}} m_{i,j}\theta_{j,t-1} + \Delta t\left(q_{i,t}f_i^H + \kappa^e\varphi^{add}f_i^R\right)$$

$$\forall\ i \in \overline{\mathcal{V}}\backslash\mathcal{V}_B,\ t \in \mathcal{T}, \tag{31c}$$

$$\theta_{i,t} = \theta_{i,t}^{sim} \qquad \forall\ i \in \mathcal{V}_B,\ t \in \mathcal{T}, \tag{31d}$$

$$k_{i,j} = \alpha k_{i,j}^R + \kappa^e k_{i,j}^S \qquad \forall\ i,j \in \overline{\mathcal{V}}. \tag{31e}$$

where $M \in \mathbb{R}$ is a constant to scale the weights, $k_{i,j}^R \in \mathbb{R}$ and $k_{i,j}^S \in \mathbb{R}$ are the elements of matrices K^R and K^S, respectively. Due to constraint (31c), model (31) is nonlinear, thus the absolute value function in the objective function (31a) can remain. Its optimal solution was computed using a Trust-region method [10] implemented in the Python package Scipy 1.3.1 [61]. The obtained optimal values for the parameters are $\varphi^w = 1763^\circ\text{C}$, $\kappa^e = 0.1705$, and $\varphi^{add} = 428.65^\circ\text{C}$.

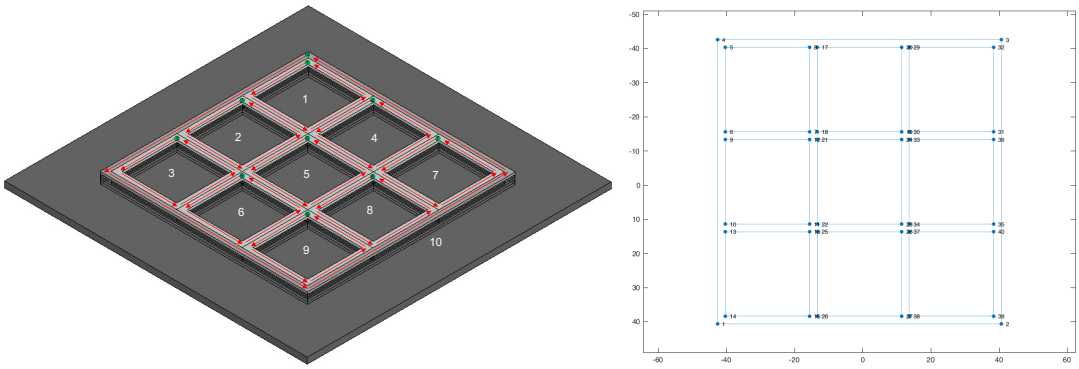

Figure 12: Example geometry with numbered components and a single layer of it.

3.4 Computational Results

The MILP consisting of constraints (13)–(21), (24)–(27), and (30), and objective function (29) was implemented in AMPL. The velocities of the heat source were set to $v^w = 6.66\frac{\text{mm}}{\text{s}}$, $v^m = 30\frac{\text{mm}}{\text{s}}$, and the time step length is $\Delta t = 1\text{s}$. The thermal diffusivity is again $\alpha = 15.6\frac{\text{W}}{\text{mm }^\circ\text{C}}$, the results of Section 3.3 are applied, and the parameters for the heat source are taken according to [4]. As initial temperature we choose $\theta_i^{init} = 500^\circ\text{C}$ for all nodes $i \in \overline{\mathcal{V}}$ and the target temperature is fixed to $\theta^{tar} = 700^\circ\text{C}$ for all nodes and time steps. They were chosen to achieve a constant temperature within the workpiece over the whole processing time reducing thermal stresses by uniform cooling behavior. To illustrate the advantage of trajectory optimization, we consider again the geometry of Section 3.3 and compare the solution of the optimization model to 100 random generated sequences. The geometry contains ten components, nine smaller squares and one surrounding square, displayed in Figure 12. To reduce the model complexity, we assume that any of the ten squares of the geometry must be processed completely before the next square can be chosen and within a square, the edges are welded counterclockwise starting in the upper left corner. Thus, it remains to find the optimal sequence of the ten squares, leading to $3,628,800$ possible trajectories and simulating of all their temperature distributions is ineffective.

Applying the optimization model, the considered instance was solved using IBM ILOG CPLEX 20.1 [30] with a time limit of $190,800$ seconds and default settings on a Mac Pro with an Intel Xeon W running 32 threads parallel at 3.2 GHz clockspeed and 768 GB RAM. The best found solution with a remaining gap of 1.32% is displayed in Figure 13. For the random generated sequences, their objective function values were computed after $12,510$ seconds in total using the constraints (24)–(27) and (30) with objective function (29) for accordingly fixed binary variables w_{i,t_i,j,t_j}, $(i,t_i,j,t_j) \in \mathcal{W}^*$, and u_{i,t_i,j,t_j}, $(i,t_i,j,t_j) \in \mathcal{U}^*$. None of them is better than the objective function value of the optimized trajectory, as their distribution shows in Figure 14.

4 Conclusions and Future Work

In this work we presented a mixed linear integer programming approach towards the otimization of printing orders for WAAM an LPBD manufacturing. We constructed complete models using real-world parameters, which were tested for computational properties for several instances and objectives. The computed results were evaluated in part using simulation tools common in the field of additive manufacturing, showing promising results for the reduction of warpage, due to the production process.

In our future work, we accelerate the solution process of the WAAM optimization model

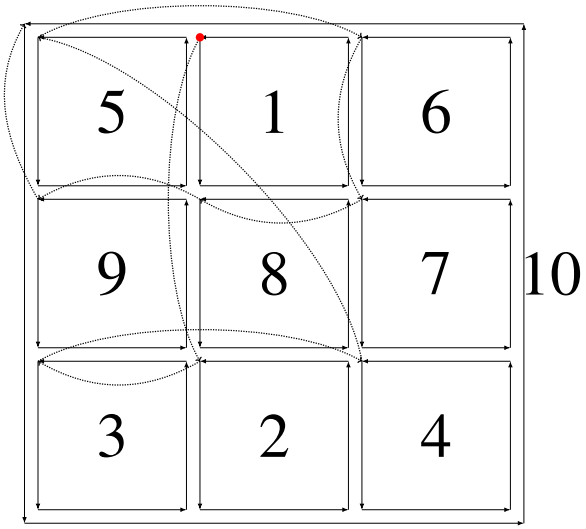

Figure 13: Best found welding sequence of the considered instance by the optimization model within a time limit of 190, 800 seconds. The red point is the starting node, dashed lines represent transition moves and the numbers give the sequence of the squares.

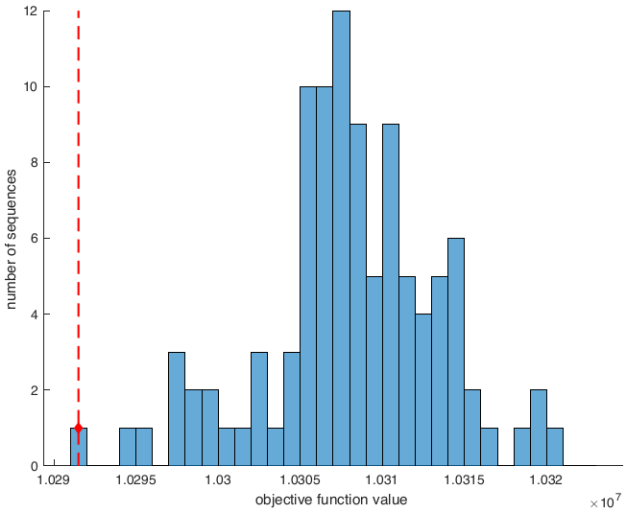

Figure 14: Distribution of the objective function values of the 100 random generated sequences and the solution found by optimization, marked by a red dotted line.

to reduce the computation time for finding optimal welding trajectories. Furthermore, the model is extended to geometries with arbitrary wall strengths and the objective function is reworked to achieve results easier to interpret.

For the optimization of LPBD process many questions are still open. Our model can still be refined with regards to the temperature computation. Other numerical methods to solve the heat can still be tested with regards to the trade-off between accuracy and speed of computation in the integrated MILP model. Furthermore, as the direct simulation of the warpage is already computationally very expensive (the model used here needs several hours to compute one printing layer), the question still remains whether optimizing the heat distribution is the only way to approximate warpage efficiently. Our model can be adapted for a multitude of different objective functions, which could be compared to the ones used here. In a next step, experimental results can be used to further reinforce our results. This seems to be a challenge, as small differences during the printing process and in the material can sometimes lead to different warpage even for the same strategy. Another question is whether our model can be adapted to other related printing processes. For example, LPBF can also be executed with multiple lasers, both for welding and for heating the material under the melting point. This would lead to very different printing patterns and could have interesting effects on the optimization process.

5 Acknowledgments

This work is part of the project "MALEDIF: Maschinelles Lernen für die additive Fertigung", and was funded by the European Regional Development Fund (ERDF) within the program StaF (Stärkung der technologischen und anwendungsnahen Forschung an Wissenschaftseinrichtungen) under the grant number 85037495.

The authors would like thank Katharina Eissing, Qui Lam Nguyen, and Felix Jensch for their valuable input and fruitful discussions towards these results.

References

[1] D. Abramson and M. Randall. A simulated annealing code for general integer linear programs. *Annals of Operations Research*, 86:3–21, 1999.

[2] M. Ansari, E. Jabari, and E. Toyserkani. Opportunities and challenges in additive manufacturing of functionally graded metallic materials via powder-fed laser directed energy deposition: A review. *Journal of Materials Processing Technology*, 294:117117, aug 2021.

[3] Peyman Ansari and Metin U. Salamci. On the selective laser melting based additive manufacturing of AlSi10mg: The process parameter investigation through multiphysics simulation and experimental validation. *Journal of Alloys and Compounds*, 890:161873, jan 2022.

[4] Martin Bähr, Johannes Buhl, Georg Radow, Johannes Schmidt, Markus Bambach, Michael Breuß, and Armin Fügenschuh. Stable honeycomb structures and temperature based trajectory optimization for wire-arc additive manufacturing. *Optimization and Engineering*, 22(2):913–974, 2021.

[5] E. Balas, S. Ceria, G. Cornuéjols, and N. Natrjay. Gomory cuts revisited. *Operations Research Letters*, 19:1–9, 1996.

[6] Markus Bambach, Armin Fügenschuh, Johannes Buhl, Felix Jensch, and Johannes Schmidt. Mathematical modeling and optimization for powder-based additive manufacturing. *Procedia Manufacturing*, 47:1159–1163, 2020.

[7] Michel Bellet and Makhlouf Hamide. Direct modeling of material deposit and identification of energy transfer in gas metal arc welding. *International Journal of Numerical Methods for Heat & Fluid Flow*, 23(8):1340–1355, oct 2013.

[8] Johannes Buhl, Rameez Israr, and Markus Bambach. Modeling and convergence analysis of directed energy deposition simulations with hybrid implicit/explicit and implicit solutions. *Journal of Machine Engineering*, 2019.

[9] Michele Chiumenti, Miguel Cervera, Alessandro Salmi, Carlos Agelet de Saracibar, Narges Dialami, and Kazumi Matsui. Finite element modeling of multi-pass welding and shaped metal deposition processes. *Computer Methods in Applied Mechanics and Engineering*, 199(37-40):2343–2359, aug 2010.

[10] Andrew R Conn, Nicholas IM Gould, and Philippe L Toint. *Trust region methods*. SIAM, 2000.

[11] D. Connolly. General purpose simulated annealing. *Journal of the Operational Research Society*, 43:495–505, 1992.

[12] G. Cornuéjols. Revival of the Gomory Cuts in the 1990s. *Annals of Operations Research*, 149:63–66, 2007.

[13] K Dai and L Shaw. Distortion minimization of laser-processed components through control of laser scanning patterns. *Rapid Prototyping Journal*, 2002.

[14] R.J. Dakin. A tree-search algorithm for mixed integer programming problems. *The Computer Journal*, 8(3):250–255, 1965.

[15] G. Dantzig, R. Fulkerson, and S. Johnson. Solution of a Large-Scale Traveling-Salesman Problem. *Journal of the Operations Research Society of America*, 2(4):393–410, 1954.

[16] G. B. Dantzig. Linear Programming. In *Problems for the Numerical Analysis of the Future, Proceedings of the Symposium on Modern Calculating Machinery and Numerical Methods, UCLA (July 29-31, 1948)*, volume 15 of *Appl. Math.*, pages 18–21. National Bureau of Standards, 1951.

[17] R. Das and P.W. Cleary. Three-dimensional modelling of coupled flow dynamics, heat transfer and residual stress generation in arc welding processes using the mesh-free SPH method. *Journal of Computational Science*, 16:200–216, sep 2016.

[18] Donghong Ding, Zengxi Pan, Dominic Cuiuri, and Huijun Li. Wire-feed additive manufacturing of metal components: technologies, developments and future interests. *The International Journal of Advanced Manufacturing Technology*, 81(1-4):465–481, may 2015.

[19] J. Ding, P. Colegrove, J. Mehnen, S. Ganguly, P.M. Sequeira Almeida, F. Wang, and S. Williams. Thermo-mechanical analysis of wire and arc additive layer manufacturing process on large multi-layer parts. *Computational Materials Science*, 50(12):3315–3322, dec 2011.

[20] J. Edmonds and E. L. Johnson. Matching, Euler tours and the chinese postman. *Mathematical Programming*, 5:88–124, 1973.

[21] Fair Isaac Corporation. FICO Xpress Optimizer Reference Manual, 2021.

[22] J.B.J. Fourier. Solution d'une question particuliere du calcul des inegalities. In *Oeuvres de Fourier*, pages 317–319. G. Olms, Hildesheim, 1970. Reprinted of the original 1826 paper with an abstract of an 1824 paper.

[23] M. Frank, A. Fügenschuh, M. Herty, and L. Schewe. The Coolest Path Problem. *Networks and Heterogeneous Media*, 5(1):143–162, 2010.

[24] Armin Fügenschuh, Markus Bambach, and Johannes Buhl. Trajectory optimization for wire-arc additive manufacturing. In *Operations Research Proceedings*, pages 331–337. Springer International Publishing, 2019.

[25] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Co., 1979.

[26] F. Glover. Future Paths for Integer Programming and Links to Artificial Intelligence. *Computers and Operations Research*, 13(5):533–549, 1986.

[27] F. Gnegel, A. Fügenschuh, M. Hagel, S. Leyffer, and M. Stiemer. A solution framework for linear PDE-constrained mixed-integer problems. *Mathematical Programming, Series B*, 188:695–728, 2021.

[28] R.E. Gomory. An algorithm for the mixed integer problem. Technical report, RM-2597, The RAND Cooperation, 1960.

[29] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2021.

[30] International Business Machines Corporation. IBM ILOG CPLEX Optimizer Reference Manual, 2021.

[31] R Israr, J Buhl, J Elze, and M Bambach. Simulation of different path strategies for wire-arc additive manufacturing with Lagrangian finite element methods. In *LS-DYNA Forum*, 2018.

[32] M. Ito, Seiichiro Izawa, Yu Fukunishi, and Masaya Shigeta. Sph simulation of gas arc welding process. In *Proc. Seventh International Conference on Computational Fluid Dynamics (Hawaii, 2012), ICCFD7-3706*, 2011.

[33] A.M. Jaffe. The Millennium Grand Challenge in Mathematics. *Notices of the AMS*, 53(6):652–660, 2006.

[34] Jingchao Jiang and Yongsheng Ma. Path planning strategies to optimize accuracy, quality, build time and material use in additive manufacturing: a review. *Micromachines*, 11(7):633, 2020.

[35] L.V. Kantorovich. Mathematical Methods of Organizing and Planning Production. *Management Science*, 6(4):366–422, 1960. English translation of the original Russian paper from 1939.

[36] Felix H Kim and Shawn P Moylan. Literature review of metal additive manufacturing defects. Technical report, U.S. Department of Commerce, may 2018.

[37] S. Kolossov, E. Boillat, R. Glardon, P. Fischer, and M. Locher. 3d fe simulation for temperature evolution in the selective laser sintering process. *International Journal of Machine Tools and Manufacture*, 44(2):117–123, 2004.

[38] T.C. Koopmans. Exchange Ratios between Cargoes on Various Routes (Non- Refrigerated Dry Cargoes). Technical report, Memorandum for the Combined Shipping Adjustment Board, Washington, D.C., 1942.

[39] H.R. Kotadia, G. Gibbons, A. Das, and P.D. Howes. A review of laser powder bed fusion additive manufacturing of aluminium alloys: Microstructure and properties. *Additive Manufacturing*, 46:102155, oct 2021.

[40] D. Kotzem, P. Dumke, P. Sepehri, J. Tenkamp, and F. Walther. Effect of miniaturization and surface roughness on the mechanical properties of the electron beam melted superalloy inconel®718. *Progress in Additive Manufacturing*, 5(3):267–276, oct 2019.

[41] Milan Kucharik, Richard Liska, Pavel Vachal, and Mikhail Shashkov. Arbitrary lagrangian-eulerian (ALE) methods in compressible fluid dynamics. *Programs and Algorithms of Numerical Mathematics*, 13:178–183, 2006.

[42] Carolin Körner, Andreas Bauereiß, and Elham Attar. Fundamental consolidation mechanisms during selective beam melting of powders. *Modelling and Simulation in Materials Science and Engineering*, 21(8):085011, nov 2013.

[43] A.H. Land and A.G. Doig. An automatic method for solving discrete programming problems. *Econometrica*, 28:497–520, 1960.

[44] S. Leyffer. Mixed-integer PDE-constrained optimization. In L. Liberti, S. Sager, and A. Wiegele, editors, *Mixed-integer Nonlinear Optimization: A Hatchery for Modern Mathematics*, volume 46, page 2738–2740. Mathematisches Forschungsinstitut Oberwolfach, 2015.

[45] Ming Liu, Louis N.S. Chiu, Chaitanya Vundru, Yang Liu, Aijun Huang, Chris Davies, Xinhua Wu, and Wenyi Yan. A characteristic time-based heat input model for simulating selective laser melting. *Additive Manufacturing*, 44:102026, aug 2021.

[46] T.S. Motzkin. *Beiträge zur Theorie der Linearen Ungleichungen*. PhD thesis, University of Zurich, 1936.

[47] G. Nemhauser and L. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, Inc., 1988.

[48] Lam Nguyen, Johannes Buhl, and Markus Bambach. Continuous eulerian tool path strategies for wire-arc additive manufacturing of rib-web structures with machine-learning-based adaptive void filling. *Additive Manufacturing*, 35:101265, oct 2020.

[49] Lam Nguyen, Johannes Buhl, Rameez Israr, and Markus Bambach. Analysis and compensation of shrinkage and distortion in wire-arc additive manufacturing of thin-walled curved hollow sections. *Additive Manufacturing*, 47:102365, nov 2021.

[50] M. Padberg and G. Rinaldi. A Branch-and-Cut Algorithm for the Resolution of Large-Scale Symmetric Traveling Salesman Problems. *SIAM Review*, 33(1):60–100, 1991.

[51] Emmanouil L. Papazoglou, Nikolaos E. Karkalos, Panagiotis Karmiris-Obratański, and Angelos P. Markopoulos. On the modeling and simulation of SLM and SLS for metal and polymer powders: A review. *Archives of Computational Methods in Engineering*, may 2021.

[52] Theo Persenot, Alexis Burr, Guilhem Martin, Jean-Yves Buffiere, Remy Dendievel, and Eric Maire. Effect of build orientation on the fatigue properties of as-built electron beam melted ti-6al-4v alloy. *International Journal of Fatigue*, 118:65–76, jan 2019.

[53] Patcharapit Promoppatum and Vitoon Uthaisangsuk. Part scale estimation of residual stress development in laser powder bed fusion additive manufacturing of inconel 718. *Finite Elements in Analysis and Design*, 189:103528, jul 2021.

[54] Davi Ramos, Fawzi Belblidia, and Johann Sienz. New scanning strategy to reduce warpage in additive manufacturing. *Additive Manufacturing*, 28:554–564, 2019.

[55] L.F. Richardson. The Approximate Arithmetical Solution by Finite Differences of Physical Problems Involving Differential Equations with an Application to the Stress in a Masonry Dam. *Philosophical Transactions of the Royal Society of London, Series A*, 210:307–357, 1911.

[56] E. Rothberg. An evolutionary algorithm for polishing mixed integer programming solutions. *Informs Journal on Computing*, 19:534–541, 2007.

[57] Fabio Scherillo. Chemical surface finishing of AlSi10mg components made by additive manufacturing. *Manufacturing Letters*, 19:5–9, jan 2019.

[58] Felix Stern, Jan Kleinhorst, Jochen Tenkamp, and Frank Walther. Investigation of the anisotropic cyclic damage behavior of selective laser melted AISI 316l stainless steel. *Fatigue & Fracture of Engineering Materials & Structures*, 42(11):2422–2430, may 2019.

[59] Blanka A. Szost, Sofiane Terzi, Filomeno Martina, Didier Boisselier, Anastasiia Prytuliak, Thilo Pirling, Michael Hofmann, and David J. Jarvis. A comparative study of additive manufacturing techniques: Residual stress and microstructural analysis of CLAD and WAAM printed ti–6al–4v components. *Materials & Design*, 89:559–567, jan 2016.

[60] Jan Taler and Paweł Ocłoń. Finite element method in steady-state and transient heat conduction. *Encyclopedia of thermal stresses*, 4:1604–1633, 2014.

[61] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[62] L. Wolsey. *Integer Programming, Second Edition.* John Wiley & Sons, Inc., 2020.

[63] Yancheng Zhang, Qiang Chen, Gildas Guillemot, Charles-André Gandin, and Michel Bellet. Numerical modelling of fluid and solid thermomechanics in additive manufacturing by powder-bed fusion: Continuum and level set formulation applied to track- and part-scale simulations. *Comptes Rendus Mécanique*, 346(11):1055–1071, nov 2018.

[64] Yupeng Zhang, Xizhang Chen, S. Jayalakshmi, R. Arvind Singh, Vladislav B. Deev, and Evgeny S. Prusov. Factors determining solid solution phase formation and stability in CoCrFeNiX0.4 (x=al, nb, ta) high entropy alloys fabricated by powder plasma arc additive manufacturing. *Journal of Alloys and Compounds*, 857:157625, mar 2021.

IMPRESSUM

Brandenburgische Technische Universität Cottbus-Senftenberg
Fakultät 1 | MINT - Mathematik, Informatik, Physik, Elektro- und Informationstechnik
Institut für Mathematik
Platz der Deutschen Einheit 1
D-03046 Cottbus

Professur für Ingenieurmathematik und Numerik der Optimierung
Professor Dr. rer. nat. Armin Fügenschuh

E fuegenschuh@b-tu.de
T +49 (0)355 69 3127
F +49 (0)355 69 2307

Cottbus Mathematical Preprints (COMP), ISSN (Print) 2627-4019
Cottbus Mathematical Preprints (COMP), ISSN (Online) 2627-6100

www.b-tu.de/cottbus-mathematical-preprints
cottbus-mathematical-preprints@b-tu.de
doi.org/10.26127/btuopen-5731