



SAT Solving with Fragmented Hamiltonian Path Constraints for Wire Arc Additive Manufacturing

Rüdiger Ehlers¹(✉), Kai Treutler², and Volker Wesling²

¹ Institute of Software and Systems Engineering, Clausthal University of Technology, Clausthal-Zellerfeld, Germany

ruediger.ehlers@tu-clausthal.de

² Institute of Welding and Machining, Clausthal University of Technology, Clausthal-Zellerfeld, Germany

Abstract. In Wire Arc Additive Manufacturing (WAAM), an object is welded from scratch. Finding feasible welding paths that make use of the potential of the technology is a computationally complex problem as it requires planning paths in 3D. All parts of the object to be manufactured have to be visited in few welding paths. The search for such welding paths in 3D can be mapped to searching for a fragmented Hamiltonian path in a mathematical graph.

We propose a SAT-based approach to finding such fragmented Hamiltonian paths that is suitable for planning WAAM paths. We show how to encode the search for such paths as a mix of SAT clauses and one non-clausal constraint that can be integrated into the SAT solver itself. The reasoning power of the solver enables us to impose additional constraints coming from the application domain on the planned paths, and we show experimentally that in this way, we can find welding paths for relatively complex object geometries.

1 Introduction

Modern additive manufacturing approaches hold a potential to add substantial flexibility to manufacturing processes. In additive manufacturing, an object is built step by step and ground up from raw material. While for plastics, 3D printing is already established, for metal, additive manufacturing is more complex, leading to a plethora of manufacturing approaches with different properties.

A notable approach in this context is wire arc additive manufacturing (WAAM), where a metal object is welded from scratch onto a metal ground body using metal from a wire roll [5]. Utilizing an industrial welding robot, the approach enables the processing of relatively high volumes of metal in short time frames with a large number of possible materials (e.g. Titanium, Aluminum, Steels, Copper, . . .) [9, 14]. WAAM is based on arc welding, where an electric current induces the heat necessary to melt both the wire and the surface to which wire material gets attached. Different shielding gases are used to protect the

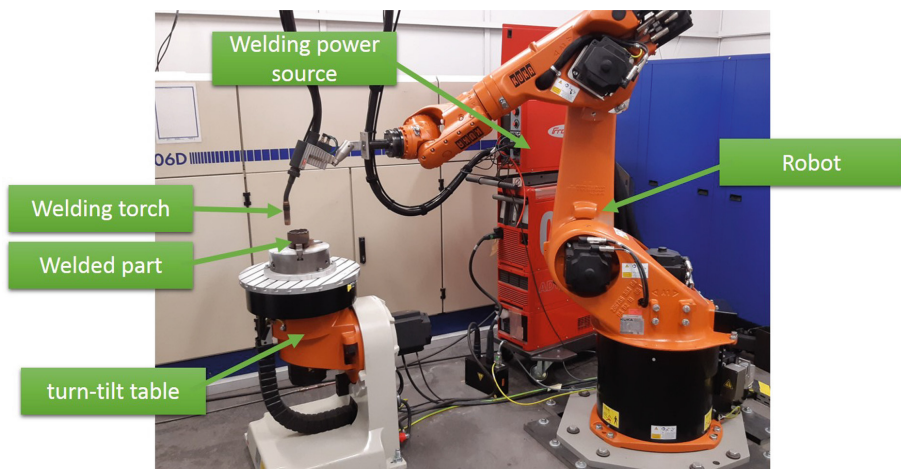


Fig. 1. Welding robot for WAAM (Institute of Welding and Machining - Clausthal University of Technology)

weld from interacting with the air. A typical manufacturing system can be seen in Fig. 1. The robot welds along a path that needs to be planned in advance. Traditionally, additive manufacturing is done layer by layer, just like for most 3D printers for plastics, so that these paths only need to be planned in 2D. The material properties of metals however depend on how quickly it cools down. For example, for low alloyed steel, the final material properties mainly depend on the time it takes for the material to cool down from $800\text{ }^{\circ}\text{C}$ to $500\text{ }^{\circ}\text{C}$ [4]. Shorter such so-called *t8/5 times* lead to a stronger and brittle material, while longer times lead to a tougher but weaker material with less residual stresses [4]. This can be exploited by making use of the possibility to stack up material locally during the welding process and to thus deviate from welding layer by layer, so that the material stays warm for a bit longer. The resulting local higher toughness can for instance be useful in the region of highly loaded notches [15]. On the other hand, far-sweeping welding paths speed up cooling, which is useful when a higher strength of the material is locally needed. Such material properties can at the same time be less important in other parts of the object. This gives the planning process some flexibility to attain local material properties, which is a clear advantage of additive manufacturing over traditional manufacturing processes. Planning in full 3D increases the size of the search space for possible welding paths dramatically, making the planning problem combinatorially complex. This observation asks for algorithmic support from computational engines such as SAT solvers. The planning problem has many side-constraints such as gravity (it is not possible to weld underneath a part of the object already welded), which can be encoded as clauses provided to the solver. The object to be welded can be composed of several welding paths, but the number of paths is typically low, as the disturbances caused by the ignition of the welding arc

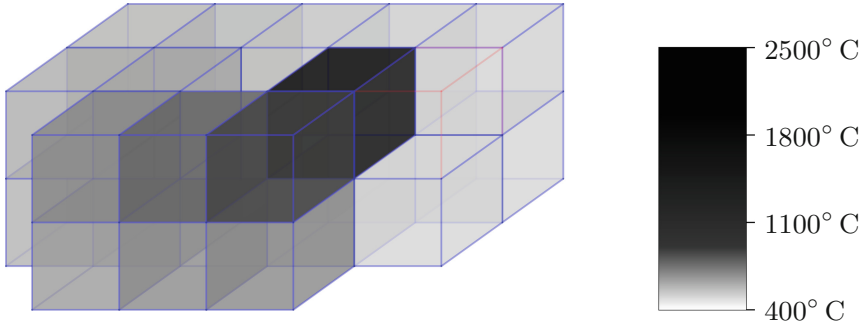


Fig. 2. Visualization of the block temperatures during a welding process. All blocks including those not yet welded are wire-framed, while blocks already welded are filled with a semi-transparent color representing their temperatures. (Color figure online)

cause the local material properties to be worse and moving the robot head while not welding leads to an additional loss of shielding gas and material.

We show in this paper how to search for welding paths with a satisfiability (SAT) solver. We discretize the object to be manufactured into blocks and represent the connections between these blocks in the form of a graph. Two blocks that can be manufactured in succession are connected by an edge in the graph. For this initial study of planning welding paths under cooling time constraints in 3D, we discretize the object parts into cubes, but the approach presented is not restricted to cubes. By searching for a *fragmented* Hamiltonian path in the graph, i.e., one that consists of multiple independent paths that together visit all vertices, we encode the search for welding paths that together implement the complete object manufacturing process. Additional constraints on the paths encode the process-induced requirements. In our approach, checking if the planned fragmented path satisfies the requested $t_{8/5}$ times is done after finding a satisfying assignment to all variables. A simplified simulator calculates the temperatures in all blocks during the welding process and implements abstract versions of heat loss due to thermal conduction and radiation. Once in any block with upper and lower $t_{8/5}$ time limits a $t_{8/5}$ time span has been observed that lies outside of the specified range, the simulation stops and a clause is generated that requires some part of the welding paths simulated until then to be different. To illustrate a simulation, Fig. 2 shows a visualization the simulator state in the middle of a welding process.

This paper is structured as follows: In the next section, we discuss the general approach to encoding the search for a fragmented Hamiltonian path into the SAT problem, including the special considerations from the application domain. Section 3 contains the architecture of our prototype solver, including details on the simulator for the evolution of the temperatures. Section 4 reports some experimental results, followed by a discussion of related work. We conclude with an outlook on future work and explain what role we expect SAT solving to play in additive manufacturing in the future.

2 Encoding Fragmented Hamiltonian Path Constraints for WAAM

We assume familiarity with the basics of satisfiability solving (see, e.g., [3] for an introduction) in the following. To describe the encoding of the welding path planning problem as a fragmented Hamiltonian path problem, we first need to define the latter.

Definition 1. Let $G = (V, E)$ be a directed graph. We say that a set of sequences $S \subseteq V^*$ is a fragmented Hamiltonian path with $k \in \mathbb{N}$ fragments if $|S| = k$, every sequence $(s_0, \dots, s_n) \in S$ is a path in G (i.e., such that for all $0 \leq i < n$, we have $(s_i, s_{i+1}) \in E$), and after adding at most $k-1$ edges to E , the sequences in S can be connected by the additional edges to a Hamiltonian path in G .

The fragmented Hamiltonian path problem is NP-hard since the Hamiltonian path problem (its special case for $k = 1$) is also NP-hard. We want to use a SAT solver to tackle this problem despite its NP-hardness. Since the fragments of a fragmented Hamiltonian path cannot share vertices, they cannot share edges. We exploit this by defining one Boolean variable x_e for every edge $e \in E$ to encode whether the edge is part of a fragment or not. The values of these variables can then together completely represent a fragmented Hamiltonian path.

Since every node can only be visited once, for every node and every pair of incoming edges of the node, we use a SAT clause requiring one of the respective Boolean edge variables to have a value of **false**. If a fragment originates in a node, there does not actually need to be an active incoming edge. To encode that there are at most k such nodes, we first allocate variables y_v for every node $v \in V$ to represent whether a fragment originates from the node. Then, we add a clause $y_v \vee \bigvee_{(v',v) \in E} x_{(v',v)}$ for every vertex v to ensure that all vertices in which no fragment starts have incoming edges. Finally, we use some type of cardinality constraint [13] to ensure that at most k variables in $\{y_v\}_{v \in V}$ have values of **true**.

So far, such an encoding does not guarantee that the edges selected cannot contain cycles, which is disallowed by Definition 1. Pandey and Rintanen [12] showed that *acyclicity* can be efficiently taken into account by *non-clausal constraint* that the SAT solver evaluates before deciding next variable values (and after unit propagation), as a clausal encoding needs large numbers of clauses and variables. Following their observations, we use the same approach in this paper.

For the WAAM application, the graph represents the discretized blocks of the object to be welded and the order in which successive blocks can be welded. The fragments of a fragmented Hamiltonian path in the graph represent the welding paths that together build up the object. They cannot contain cycles since a block would then be welded twice. As the t8/5 times at the blocks depend on in which order the fragments are welded, we also need to encode this order. Using the assumption that k will normally be small in this application domain, we chose to do so using $|V| \times k$ many variables $\{z_{v,i}\}_{v \in V, 1 \leq i \leq k}$. A variable $z_{v,i}$ should have a value of **true** iff vertex v belongs to the i th path to be welded.

To ensure that these variables have meaningful values, we first encode that every vertex has to be part of at least one fragment number, using $|V|$

many clauses of length k . Then, we encode that all (successor) elements of a path are marked as belonging to the same order. We do so with the clauses $\bigwedge_{(v,v') \in E, 1 \leq i \leq k} \neg z_{i,v} \vee \neg x_{(v,v')} \vee z_{i,v'}$. Also, to exclude that all paths are marked as having the same index, we add the clauses $\bigwedge_{v,v' \in V, 1 \leq i \leq k} \neg y_v \vee \neg y_{v'} \vee \neg z_{v,i} \vee \neg z_{v',i}$ which make sure that all path starting points have different fragment numbers. This constraint will be the only one in our encoding that requires a number of clauses quadratic in the number of vertices.

The relatively variable-intensive encoding of which vertex belongs to which path number has the benefit that it enables us to concisely encode another important requirement for welding: vertices v' that lie above vertices v in the three-dimensional grid cannot be welded in earlier fragments. It suffices to add the clauses $\bigwedge_{v,v' \text{ s.t. } v \text{ is directly above } v', 1 \leq i < i' \leq k} \neg x_{i,v} \vee \neg x_{i',v'}$ to ensure this. We do not encode any downwards edges into E (and only the transitions back, front, left, right, and up from each block), so we can ignore the $i = i'$ case.

3 Solver and Simulator Engineering

To evaluate the encoding for fragmented Hamiltonian paths presented in the preceding section, we implemented the acyclicity non-clausal constraint on top of the SAT solver `Glucose 3.0` [2], which bases on `Minisat` [6]. The solver is called by a host tool that reads a geometry description of the object to be manufactured, constraints on t8/5 times for some blocks, and two constants (described below) detailing the cooling properties of the welding process. The tool then computes the underlying graph and performs the encoding presented above. For encoding the cardinality constraints, we use cardinality networks [1] as implemented in `PySAT` [10]. The SAT solver is used in incremental mode, so that after each found fragmented Hamiltonian path satisfying the encoded side constraints, we can run a simulator to check if welding the path would lead to the satisfaction of the t8/5 time constraints, and rule out this path afterwards.

The simulator tracks the temperatures of all blocks during the welding process. Initially, no block has been welded, meaning that no block has a temperature. When a block is welded, its temperature is initialized to 2500 °C. Between welding the blocks, the simulator calculates temperature changes due to two physical phenomena:

- Neighbouring blocks that have both been already welded exchange heat at a rate that is proportional to their temperature difference (and the surface area, which is however the same for all block faces in this work).
- Hot objects emit radiation from their outer surfaces (which can be detected by infrared heat cameras). The intensity of radiation is proportional to the difference between the fourth powers of the temperatures of the block and the environment.

For very hot object parts (e.g., the weld), the temperature loss/exchange due to radiation typically dominates the former physical phenomenon. In both phenomena, temperature differences are driving the heat exchange, which means that the dynamics of the system are represented as differential equations. Our simulator

Table 1. Overview of the experimental results. computation times are taken on a Linux-based computer with a i5-4200U processor running at 1.6 GHz (8 GB RAM).

Example	Example 1	Example 2	Example 3
# Blocks to be welded	47	113	168
# t8/5 constraints	2	6	15
# Allowed path fragments	1	2	3
Comp. time to first solution	0.008 s	0.603 s	0.661 s
Comp. time enumerating all solutions	0.442 s	48.396 s	680.28 s
Comp. time enum. all frag. Ham. paths	0.228 s	26.552 s	422.40 s
# Solutions (satisfying t8/5 times)	64	256	2272
# Frag. Hamiltonian paths	728	31744	162624
Min/max/mean length of clauses	34/40/	32/116/	30/162/
added after t8/5 time violation	37.1084	55.1545	79.961

solves them approximately using the Euler method and 20 sub-steps per welded block. While the simulation is of relatively low precision, its computational cost is already substantial. During the simulation, we keep track of how quickly a block in the object to be welded cools down from 800 °C to 500 °C. Since research on the expected material properties in case of crossing the 500 °C barrier multiple times (without exceeding 800 °C) has not converged to a well-usable model yet, we require that for every crossing of the 500 °C limit, the time spent in temperatures between 800 °C and 500 °C is within the boundaries imposed in the object description. As soon as one boundary is crossed, the simulator computes a clause ruling out the choice of fragments already welded until that point (including their order). This clause is then added to the SAT solver’s clause database, so that all fragmented paths with the same simulated prefix are ruled out.

The simulator requires two constants that define the magnitude of the two physical phenomena causing temperature changes listed above. We chose those values so that a visualization of the temperature evolution shows reasonable behaviour and leads to non-trivial path planning problems that help us to evaluate our SAT-based approach. Similarly, the minimal and maximal t8/5 times are chosen to lead to computationally interesting path planning problems to help with our experimental evaluation in the next section. Fine-tuning the parameters to conform precisely to the actual welding process is left for future work after extensive experimentation with welding the planned paths.

4 Experiments

We evaluate the approach for encoding fragmented Hamiltonian path constraints on a few WAAM welding path planning problem instances. The aim of the experiments is to determine (a) how big the path-planned objects can be with our approach, i.e., scalability, (b) how much solving time is spent in the CDCL-part

of the SAT solver, the acyclicity constraint program code, and the simulator, and (c) how long the conflict clauses computed by the temperature evolution simulator are, i.e, how early in a fragmented path it can be detected that some t8/5 time is not in the allowed range. With the experimental evaluation, we want to shed light on the principal applicability of SAT solvers for the welding path planning problem and what the most pressing further algorithmic improvements necessary to support solving this problem for large objects to be manufactured are. Table 1 contains experimental results on three example objects of different sizes. To get an idea of how much computation time is spent in the core SAT solver, in the non-clausal constraint code, and in the simulator, we evaluated this on the second example using the profiling tool `callgrind/valgrind`. This yielded 36.94% of time for the core parts of the SAT solver, 5.02% of time for the non-clausal constraint, and 58.03% of time for the simulator. It can be seen that the SAT solving part (including the non-clausal constraint) is already quite efficient, and the simulator is the bottleneck. The clauses generated after a simulation are quite long, so that the t8/5 time constraints (currently) prune the search space only little.

5 Related Work

We are not aware of any published previous approach to encoding fragmented Hamiltonian paths in SAT. However, Hamiltonian circuit constraints have been considered in the context of constraint programming. One of the latest works in this area is the one by Francis et al. [7], who distill previous approaches to incorporating such constraints to a new strong technique with a very high *propagation strength*, i.e., such that for solving the problem instances, fewer decisions need to be made by the solver. While their approach can be extended to Hamiltonian paths using artificial cycle closing edges, we observed that propagation in their approach is mostly dependent on which of these artificial edges are chosen by the solver. This leads to unnecessary solver decisions.

Francis et al. [7] also argue that strong propagators based on computing flows in the graphs consisting of the edges not yet ruled out by the solver, as previously proposed and used for the *all-different* constraint in constraint programming [8], is too costly in an efficient solving approach. Since this may change when dealing with fragmented Hamiltonian paths, it makes sense to re-evaluate this statement in this context in the future.

6 Conclusion and Outlook

In this paper, we described how SAT solving can help with new additive manufacturing approaches such as Wire Arc Additive Manufacturing (WAAM). From an application point of view, the contribution of the paper is a way to handle the complex combinational search problem for feasible welding paths. From a SAT solving perspective, this paper describes a new encoding (using an acyclicity non-clausal constraint) for finding fragmented Hamiltonian paths. It should

be noted that there is an alternative to the non-clausal constraints employed in our approach. We also experimented with an adaptation of the linear feedback-shift register acyclicity SAT encoding by Johnson [11] to the case of fragmented Hamiltonian paths. Using it resulted in only slightly longer computation times. Adding further clausal constraints on welding paths may flip the advantage to the pure SAT encoding in the future.

The experiments performed with a first approximate simulator for the local temperatures during welding show two results. First of all, SAT solvers are a suitable computational engine for this application domain. Future side-constraints on feasible curve radius sizes or similar requirements on the welded paths that are yet to be identified in WAAM research can easily be integrated as clauses. Hence, SAT solvers can serve as a reasoning platform for such planning problems. Secondly, during simulation, the violation of t8/5 time constraints becomes apparent relatively late, leading to long added clauses.

To solve this, we plan to investigate whether for complex system dynamics, some sort of specialized simulation theory can be developed to exploit the fact that while temperatures evolve non-linearly, they still evolve monotonely with the heat applied in the system. At the same time, the fact that every block is visited by the welding torch exactly once can be exploited to detect conflicts earlier, suggesting that a full SMT approach with such a specialized theory solver should be evaluated.

References

1. Asín, R., Nieuwenhuis, R., Oliveras, A., Rodríguez-Carbonell, E.: Cardinality networks and their applications. In: Kullmann, O. (ed.) SAT 2009. LNCS, vol. 5584, pp. 167–180. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02777-2_18
2. Audemard, G., Simon, L.: Glucose and syrup in the SAT race 2015. In: Reports on the SAT 2015 Competition (2015)
3. Biere, A., Heule, M., van Maaren, H., Walsh, T. (eds.): Handbook of Satisfiability, Frontiers in Artificial Intelligence and Applications, vol. 185. IOS Press, Amsterdam (2009)
4. Schröpfer, D., Kromm, A., Hannemann, A., Kannengießer, T.: In-situ determination of critical welding stresses during assembly of thick-walled components made of high-strength steel. In: Materials Research Proceedings, vol. 6, pp. 191–196 (2018)
5. Dahat, S., Hurtig, K., Andersson, J., Scotti, A.: A methodology to parameterize wire + arc additive manufacturing: a case study for wall quality analysis. *J. Manuf. Mater. Process.* **4**(1), 14 (2020)
6. Eén, N., Sörensson, N.: An extensible SAT-solver. In: Giunchiglia, E., Tacchella, A. (eds.) SAT 2003. LNCS, vol. 2919, pp. 502–518. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24605-3_37
7. Francis, K.G., Stuckey, P.J.: Explaining circuit propagation. *Constraints* **19**(1), 1–29 (2013). <https://doi.org/10.1007/s10601-013-9148-0>
8. van Hoeve, W.: The all different constraint: a survey. In: Annual Workshop of the ERCIM Working Group on Constraints (2001)

9. A Hosseini, V., Högström, M., Hurtig, K., Valiente Bermejo, M.A., Stridh, L.-E., Karlsson, L.: Wire-arc additive manufacturing of a duplex stainless steel: thermal cycle analysis and microstructure characterization. *Weld. World* **63**(4), 975–987 (2019). <https://doi.org/10.1007/s40194-019-00735-y>
10. Ignatiev, A., Morgado, A., Marques-Silva, J.: PySAT: a python toolkit for prototyping with SAT oracles. In: Beyersdorff, O., Wintersteiger, C.M. (eds.) *SAT 2018*. LNCS, vol. 10929, pp. 428–437. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-94144-8_26
11. Johnson, A.: Stedman and Erin triples encoded as a SAT problem. In: *Federated Logic Conference 2018: Pragmatics of SAT*. Technical report (2018)
12. Pandey, B., Rintanen, J.: Planning for partial observability by SAT and graph constraints. In: *ICAPS*, pp. 190–198. AAAI Press (2018)
13. Roussel, O., Manquinho, V.: Pseudo-Boolean and Cardinality Constraints, chap. 22, pp. 695–733. Vol. 185 of Biere et al. [3] (2009)
14. Szost, B.A., et al.: A comparative study of additive manufacturing techniques: residual stress and microstructural analysis of CLAD and WAAM printed Ti-6Al-4V components. *Mater. Des.* **89**, 559–567 (2016)
15. Treutler, K., Kamper, S., Leicher, M., Bick, T., Wesling, V.: Multi-material design in welding arc additive manufacturing. *Metals* **9**(7), 809 (2019)