Proceedings of the Eighteenth International Symposium on Combinatorial Search (SoCS 2025)

Object Packing and Scheduling for Sequential 3D Printing: A Linear Arithmetic Model and a CEGAR-inspired Optimal Solver (Extended Abstract)

Pavel Surynek[1], **Vojtěch Bubník**[2], **Lukáš Matěna**[2], **Petr Kubiš**[2]
[1]Faculty of Information Technology, Czech Technical University in Prague, Thákurova 9, 160 00 Praha 6, Czechia
pavel.surynek@fit.cvut.cz
[2]Prusa Research, Partyzánská 188/7a, 170 00 Praha 7 - Holešovice, Czechia
{vojtech.bubnik, lukas.matena, petr.kubis}@prusa3d.cz

Abstract

We address the problem of object arrangement and scheduling for sequential 3D printing. Unlike the standard 3D printing, where all objects are printed slice by slice, in sequential 3D printing, objects are completed one after another. In the sequential case, it is necessary to ensure that the moving parts of the printer do not collide with previously printed objects. We propose to express the problem of sequential printing as a linear arithmetic formula, which is then solved using a solver for satisfiability modulo theories (SMT) combined with counterexample guided abstraction refinement (CEGAR).

Introduction

Additive manufacturing, i.e. 3D printing, is an increasingly important alternative to traditional manufacturing processes (Gao et al. 2015).

A standard Cartesian fused deposition modeling (FDM) 3D printer creates objects on a typically rectangular printing plate (usually heated) by gradually drawing individual slices of printed objects, where these slices are very thin, approximately tenths of a millimeter. Printing is performed using a print head with an extruder, which applies material through a narrow nozzle. The movement of the print head is ensured by a printer mechanism that allows the head to move in all x, y, and z coordinates.

One of the important tasks of combinatorial optimization in 3D printing is the arrangement of printed objects on the printing plate so that the space of the place is used effectively (Edelkamp and Wichern 2015).

In this work, however, we go further, we deal with the task of the so-called *sequential 3D printing*, where we will not print all objects slice-by-slice at once, but will complete the objects one after another, with individual objects being printed in the standard slice-by-slice manner. The problem is particularly challenging because the objects need to be arranged on the printing plate in such a way that the print head and other mechanical parts, such as the gantry on which the print head is mounted, avoid previously printed objects (see Figure 1 for illustrations). Moreover, sequential printing does not only mean spatial arrangement of objects on a printing plate, but also determining the order in which the objects are printed.

Sequential printing has far-reaching significance for modern 3D printing, it can help to tackle the following challenges:

(i) increasing the robustness of the printing process against errors (in case of failure, we do not have to repeat the entire print, but only the unfinished objects)
(ii) reduces stringing (sequential printing eliminates frequent movements of the print head between objects)
(iii) minimize the number of time consuming color changes during multi-color printing (a typical example is printing each object in a different color).

We suggest to model the problem as a linear arithmetic formula (Kroening and Strichman 2016), which we then solve using an off-the-shelf solver for *satisfiability modulo theories* (SMT). However, straightforward modeling the problem as a formula seems to be inefficient. Therefore, we propose a technique inspired by *counterexample guided abstraction refinement* (CEGAR) (Clarke et al. 2003) to achieve more practical efficiency.

This is a short version of (Surynek et al. 2025).

Towards a Linear Arithmetic Model

The problem of *3D sequential object arrangement and scheduling* (or "object packing and scheduling" in short, SEQ-PACK+S) is a task of determining the positions and order of 3D objects so that the objects can be printed by the 3D printer in a determined order at the determined positions sequentially. Unlike standard 3D printing, where all objects are printed at once in individual slices, in sequential printing, objects are completed individually. That is, when the next object is printed, the previously printed objects are still present on the printing plate. It is therefore necessary to ensure that the print head and other mechanical parts of the 3D printer do not collide with previously printed objects when printing the next object.

From the mathematical point of view, we can look at printing an object in such a way that it is necessary to create every point of the printed object, that is, to touch every point of the printed object with the extruder, or more precisely, with a selected point of the extruder that corresponds to the nozzle opening.

The extruder must therefore move to each point of the printed object. Once a point is printed, we must take into ac-

---

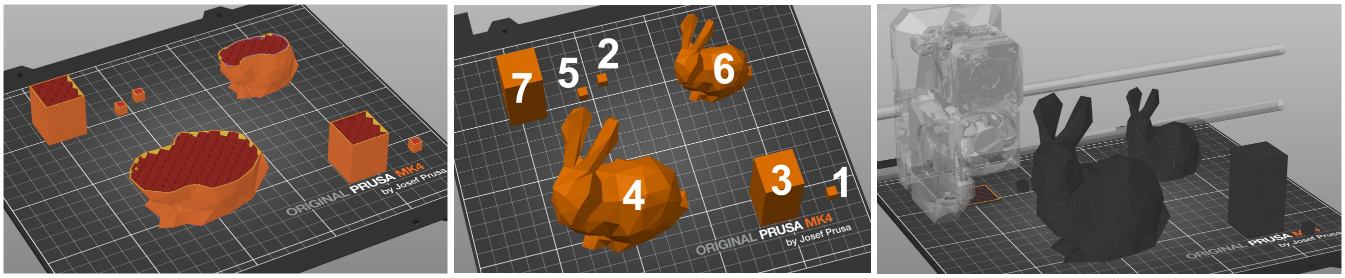

Figure 1: Standard 3D printing slice by slice and sequential 3D printing where objects are completed one by one shown in Prusa Slicer. The ordering of objects for sequential printing is shown by numbers. Printer extruder and gantry must avoid peviously printed objects in the sequential case (printing of the last object is shown).

count its presence for future movements of the extruder, that is, the extruder must not collide with the point in the future. This is not an issue in the standard 3D printing, however in the sequential case when the next object is printed the extruder needs to be lowered down to the printing plate again to start the next object which means potential collisions of the extruder with previous objects.

We first simplify the problem from 3D to 2D. The simplification consists in taking the convex hull of the maximum xy-projection of each printed object. Such convex hull represents an *envelope* of an object. If we ensure non-collision with the envelope then non-collision with the object itself is ensured too.

Movement of extruder visiting all points of the object is modeled by calculating Minkowski sum of the object envelope with an xy-projection of the extruder, which intuitively corresponds to the **inflation** of the object by the extruder. If we ensure a collision free placement of the envelope of one object and inflated envelope of the other object, then the second object can be sequentially printed after the first object without the extruder colliding with the first object. This requirement can be expressed as non-overlapping of two convex polygons which can be further expressed as a linear arithmetic formula.

More formally, we introduce rational-valued decision variables that model placements of individual objects, i.e. their convex envelopes. Non-overlapping requirement between convex envelopes can be expressed on top of these placement variables. However, whether an object envelope appears as inflated by the extruder or not depends on the temporal ordering of printing individual objects.

To determine temporal ordering we use another set of rational-valued decision variables. If say object O_j is to be printed after object O_i according to their temporal ordering variables, then the convex envelope of O_j appears to be inflated by the extruder with respect to the envelope of O_i that is not inflated.

The resulting linear arithmetic formula despite the 2D simplification is still relatively complicated for straightforward solving. Hence we took an inspiration from CEGAR. In our case, we initially abstract from certain type of constraints that are necessary to ensure non-overlapping of object envelopes (namely the requirement on non-intersection of edges of convex envelopes) while the other constraints remained (namely the requirement that all vertices of one convex envelope are outside of the other envelope and vice versa). The solution to the abstraction is refined only if a violation of abstracted constraints is detected. Often only a fraction of the constraints need to be considered before finding a solution which is the key to the efficiency of the algorithm.

Conclusion

In the full version of the paper we show a detailed formalization of the object arrangement (packing) and scheduling problem for sequential 3D printing and give a detailed formulation of the problem as a linear arithmetic formula. In addition to this, details of our CEGAR-inspired algorithm are given. The algorithm has been implemented using the Z3 SMT solver and integrated in Prusa Slicer, an open-source 3D printing software (The Prusa Slicer Team 2025) (since version 2.9.1).

References

Clarke, E. M.; Grumberg, O.; Jha, S.; Lu, Y.; and Veith, H. 2003. Counterexample-guided abstraction refinement for symbolic model checking. *J. ACM*, 50(5): 752–794.

Edelkamp, S.; and Wichern, P. 2015. Packing Irregular-Shaped Objects for 3D Printing. In *KI 2015: 38th Annual German Conference on AI*, volume 9324 of *Lecture Notes in Computer Science*, 45–58. Springer.

Gao, W.; Zhang, Y.; Ramanujan, D.; Ramani, K.; Chen, Y.; Williams, C. B.; Wang, C. C. L.; Shin, Y. C.; Zhang, S.; and Zavattieri, P. D. 2015. The status, challenges, and future of additive manufacturing in engineering. *Comput. Aided Des.*, 69: 65–89.

Kroening, D.; and Strichman, O. 2016. *Decision Procedures - An Algorithmic Point of View, Second Edition*. Texts in Theoretical Computer Science. An EATCS Series. Springer.

Surynek, P.; Bubnik, V.; Matena, L.; and Kubis, P. 2025. Object Packing and Scheduling for Sequential 3D Printing: a Linear Arithmetic Model and a CEGAR-inspired Optimal Solver. *CoRR*, abs/2503.05071.

The Prusa Slicer Team. 2025. Prusa Slicer 2.9.1. Available from `https://github.com/prusa3d/PrusaSlicer`.