Efficient Packing of 3D-Polytopes into a Parallelepiped using an SMT-Solver*

Roberto Bruttomesso, Romeo Rizzi

Abstract

This report describes a method for solving the problem of polytope packing into a parallelepiped by means of an encoding into SMT. The encoding is simple and flexible, and it can easily handle both convex and non-convex polytopes. A simple experiment with a prototypcal implementation shows that this technique is competitive w.r.t. heuristic-based approaches in finding the minimal height of a feasible packing.

1 Introduction

Polytope Packing is the problem of placing a given set of polytopes into a parallelepiped of given length and width with the goal of finding the minimal possible height that avoids polytopes collision (polytopes may touch but they cannot compenetrate).

The problem has been studied for instance by Stoyan et. al in [5], which we use as a reference and comparison for this report. More related work can be found in the aforementioned paper and others by the same authors.

Our approach is a plain encoding into an SMT2 [2] formula. SMT, Satisfiability Modulo Theories, is an area of research that combines efficient SAT-Solving and domain-specific decision procedures to build efficient tools that could reason about, for instance, arbitraty boolean combinations of linear arithmetic costraints. Efficient SMT-Solvers are available off-the-shelf and under continuos improvement. Our encoding into SMT exploits the notion of Minkowski sum to formally describe concepts such as "polytope intersection".

Therefore, the approach can be summarized as follows: take a set of polytope descriptions, encode the problem into SMT2, execute an SMT-Solver to find a solution (if any exists), read the solution and translate it back to coordinates that describes the polytopes placement.

2 Notation

Throghout this paper we assume that we are working in a three-dimensional euclidian space. We shall use $P_1, P_2, \ldots$ to denote polytopes with points in $\mathbb{R}^3$, and $\mathbf{p_1}, \mathbf{p_2}, \ldots$ to denote points in $\mathbb{R}^3$, where in particular $\mathbf{0} = (0, 0, 0)$. For a $\mathbf{p_i}$ we indicate its three components with $(p_{i_x}, p_{i_y}, p_{i_z})$.

*This work dates back to 2005. We never had the chance to make it public before now.

3 Minkowski sum and difference

Given two points $\mathbf{p_1} = (x_1, y_1, z_1)$ and $\mathbf{p_2} = (x_2, y_2, z_2)$ we define their sum as usual as the point $\mathbf{p_1} + \mathbf{p_2} = (x_1 + x_2, y_1 + y_2, z_1 + z_2)$, and their difference as the point $\mathbf{p_1} - \mathbf{p_2} = (x_1 - x_2, y_1 - y_2, z_1 - z_2)$. Given two polytopes P_1 and P_2, the Minkowski sum is a polytope $P_1 \oplus P_2$ defined as

$$P_1 \oplus P_2 = \{\mathbf{p_1} + \mathbf{p_2} \mid \mathbf{p_1} \in P_1, \mathbf{p_2} \in P_2\}.$$

Similarly, the Minkowski difference is defined as

$$P_1 \ominus P_2 = \{\mathbf{p_1} - \mathbf{p_2} \mid \mathbf{p_1} \in P_1, \mathbf{p_2} \in P_2\}.$$

An obvious property of the Minkowski difference is the following.

Property 1. *Two polytopes P_1,P_2 are intersecting if and only if $\mathbf{0} \in (P_1 \ominus P_2)$.*

Consider now translating P_1 and P_2 by respectively vectors $\mathbf{v_1}$ and $\mathbf{v_2}$. We have that $\mathbf{v_1}+P_1$ and $\mathbf{v_2}+P_2$ intersect if and only if $\mathbf{0} \in ((\mathbf{v_1}+P_1)\ominus(\mathbf{v_2}+P_2))$. The following chain of relations hold

$$\begin{aligned}
& \mathbf{0} \in ((\mathbf{v_1} + P_1) \ominus (\mathbf{v_2} + P_2)) \\
\Leftrightarrow \quad & \mathbf{0} \in \{(\mathbf{p_1} + \mathbf{v_1}) - (\mathbf{p_2} + \mathbf{v_2}) \mid (\mathbf{p_1} + \mathbf{v_1}) \in P_1, (\mathbf{p_2} + \mathbf{v_2}) \in P_2\} \\
\Leftrightarrow \quad & -\mathbf{v_1} \in \{\mathbf{p_1} - (\mathbf{p_2} + \mathbf{v_2}) \mid \mathbf{p_1} \in P_1, (\mathbf{p_2} + \mathbf{v_2}) \in P_2\} \\
\Leftrightarrow \quad & (\mathbf{v_2} - \mathbf{v_1}) \in \{\mathbf{p_1} - \mathbf{p_2} \mid \mathbf{p_1} \in P_1, \mathbf{p_2} \in P_2\} \\
\Leftrightarrow \quad & (\mathbf{v_2} - \mathbf{v_1}) \in (P_1 \ominus P_2) \qquad (1)
\end{aligned}$$

We interpret (1) as follows: take two vectors $\mathbf{v_1}$, $\mathbf{v_2}$ for translating P_1 and P_2 respectively. The two translated polytopes intersect if and only if the difference vector is contained in the Minkowski difference.

4 Encoding polytope placement into SMT

Let's now focus on the task of placing polytopes into a parallelepiped. Informally, the placement is carried out by means of a vector for each polytope, which translates the polytope with respect to the origin. The polytope packing problem is then the problem of finding such placement vectors, which may or may not exist, depending on the particular instance to solve.

Formally, given a set of polytopes $P_1, \ldots, P_n$, we need to find placement vectors $\mathbf{v_1}, \ldots, \mathbf{v_n}$ such that for $0 \leq i < j \leq n$

$$(\mathbf{v_i} + P_i) \cap (\mathbf{v_j} + P_j) \neq \emptyset$$

which means that for $0 \leq i < j \leq n$

$$(\mathbf{v_j} - \mathbf{v_i}) \notin (P_i \ominus P_j).$$

The latter condition can be translated into a constraint satisfaction problem. In particular $P_i \ominus P_j$ is a polytope that can be defined by the intersection of a set of linear inequalities

$$\begin{aligned}
& c_{1_x}x + c_{1_y}y + c_{1_z}z \leq c_1 \\
\wedge \quad & c_{2_x}x + c_{2_y}y + c_{2_z}z \leq c_2 \\
\wedge \quad & \ldots \\
\wedge \quad & c_{n_x}x + c_{n_y}y + c_{n_z}z \leq c_n
\end{aligned}$$

each representing a faced of the polytope. The area “outside” $P_i \ominus P_j$ is therefore

$$\begin{array}{ll} & c_{1_x}x + c_{1_y}y + c_{1_z}z \geq c_1 \\ \vee & c_{2_x}x + c_{2_y}y + c_{2_z}z \geq c_2 \\ \vee & \dots \\ \vee & c_{n_x}x + c_{n_y}y + c_{n_z}z \geq c_n \end{array}$$

Technically we should use strict inequalities $>$, however we may allow polytopes to “touch” on a common point. At last we can specify that $\mathbf{v_j} - \mathbf{v_i}$ is in the area outside $P_i \ominus P_j$ with the following substitution

$$\begin{array}{ll} & c_{1_x}(v_{j_x} - v_{i_x}) + c_{1_y}(v_{j_y} - v_{i_y}) + c_{1_z}(v_{j_z} - v_{i_z}) \geq c_1 \\ \vee & c_{2_x}(v_{j_x} - v_{i_x}) + c_{2_y}(v_{j_y} - v_{i_y}) + c_{2_z}(v_{j_z} - v_{i_z}) \geq c_2 \\ \vee & \dots \\ \vee & c_{n_x}(v_{j_x} - v_{i_x}) + c_{n_y}(v_{j_y} - v_{i_y}) + c_{n_z}(v_{j_z} - v_{i_z}) \geq c_n \end{array} \tag{2}$$

In addition to the constraints above, we need to specify the “borders” of the parallelepiped. Suppose that the parallelepiped measures l,w,h of length, width, and height respectively, and let $x_\downarrow(P_i), x_\uparrow(P_i)$, the lowest and highest x coordinate of P_i, $y_\downarrow(P_i), y_\uparrow(P_i)$, the lowest and highest y coordinate of P_i, $z_\downarrow(P_i), z_\uparrow(P_i)$, the lowest and highest z coordinate of P_i. Then we need to encode for all i

$$\begin{array}{l} 0 \leq v_{i_x} + x_\downarrow(P_i) \wedge v_{i_x} + x_\uparrow(P_i) \leq l \\ 0 \leq v_{i_y} + y_\downarrow(P_i) \wedge v_{i_y} + y_\uparrow(P_i) \leq w \\ 0 \leq v_{i_z} + z_\downarrow(P_i) \wedge v_{i_z} + z_\uparrow(P_i) \leq h \end{array} \tag{3}$$

By encoding (2) and (3) into the SMT2 [2] language, we can find values of $\mathbf{v_i}$ for each P_i that represent a polytope placement such that the polytopes (may touch but) do not intersect and such that is contained in the given parallelepiped.

5 Experiments

We have implemented a tool chain that takes as input a description of the polytopes in terms of vertices, and the parallelepiped boundaries, and it returns the placement vectors for the arrangement, if any exists.

In order to compute the facets of the polytopes we rely on the CGAL library [6]. We then solve the SMT problem with some efficient and open-source SMT-solver, such as CVC4 [1], OpenSMT [3], or Z3 [4]. We use CGAL again to export the solution returned by the SMT-Solver into the VRML graphical models that can be seen in this paper. The source code of our tool-chain can be downloaded from `https://github.com/bobosoft/polytopepacking`. At the same address it is possible to obtain the problem descriptions, the encoded smt2 files, and the 3D models of the results.

As an experiment to test our proof-of-concept tool-chain we have encoded and solved the first problem described in [5], where 7 polytopes are placed in a parallelepiped of length 12 and width 10. [5] shows a placement of height 27, which is a local optima for the approach of that paper. By running our tool-chain we were able to prove, in about 10 minutes, that a height of 23 is

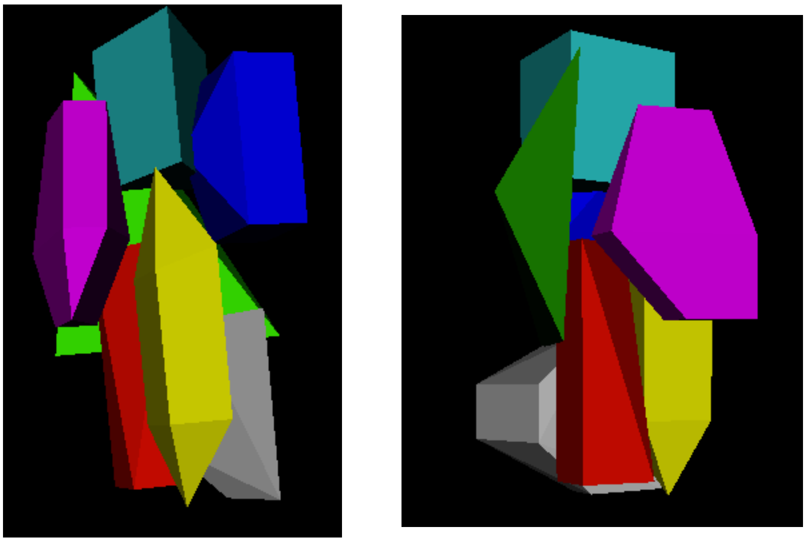

Fig. 1: Two views of the packing placement for 7 polytopes as per [5] for a height of 23.

actually sufficient. We were not able to obtain a model for height 22 within an hour, and the polytope placement is shown in Figure 1. In these trials we have increased and decreased the value for the height manually, but this process can be automated in a way that is standard among optimizing solvers.

A more automated implementation, and more exhaustive and detailed experimentation are left as future work.

6 Conclusion

We have presented a simple approach to solve the polytope packing problem that relies on an encoding into an SMT formula. The approach is flexible, straightforward, and it can be easily extended to support more features: for example, selected polytopes can be constrained to touch or to stick to a particular position, by adding more constraints to the problem. Fixed-radious rotations can also be considered as a further step. The availability and the efficiency of SMT-Solvers are key factors for making this approach feasible in practice.

References

[1] Clark Barrett, Christopher L. Conway, Morgan Deters, Liana Hadarean, Dejan Jovanovic, Tim King, Andrew Reynolds, and Cesare Tinelli. CVC4. In *CAV*, pages 171–177, 2011.

[2] Clark Barrett, Aaron Stump, and Cesare Tinelli. The Satisfiability Modulo Theories Library (SMT-LIB). `www.SMT-LIB.org`, 2010.

[3] Roberto Bruttomesso, Edgar Pek, Natasha Sharygina, and Aliaksei Tsitovich. The opensmt solver. In *TACAS*, pages 150–153, 2010.

[4] Leonardo De Moura and Nikolaj Bjørner. Z3: An efficient smt solver. TACAS, pages 337–340, 2008.

[5] Y. Stoyan, N. Gil, G. Scheithauer, A. Pankratov, and I.Magdalina. Packing of convex polytopes into a parallelepiped. 2003. Technical Report.

[6] The CGAL Project. *CGAL User and Reference Manual.* CGAL Editorial Board, 4.6 edition, 2015.