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

Roberto Bruttomesso, Romeo Rizzi

Abstract

This report describes an approach to the problem of polytope packing into a parallelepiped by means of an encoding into an SMT-Solver. The encoding is simple and flexible, and it can easily handle both convex and non-convex polytopes. A simple experiment with an unoptimized prototype shows that this technique may outperform heuristic-based approaches in finding the minimal height of the 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 [1]. We take this paper as a reference and comparison for this report. More related work can be found in the aforementioned paper.

Our approach is a plain encoding into an SMT 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. The encoding exploits the notion of Minkowski sum to formally describes concepts such as "polytope intersection".

Therefore, the approach can be summarized as follows: take a set of polytope descriptions, encode the problem into SMT, 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 $P_1, P_2, \dots$ to denote polytopes with points in $\mathbb{R}^3$, and $\mathbf{p_1}, \mathbf{p_2}, \dots$ 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

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

A useful and well-known 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

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

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 $i \neq j$

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

which means that $i \neq j$

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

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

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

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

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

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

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

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

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

By encoding (2) and (3) into the SMT 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

6 Conclusion

References

[1] et. al Y. Stoyan. Packing of convex polytopes into a parallelepiped. 2003.