In *Information Processing Letters*, vol. 48, 5 (1993), 229–235

On the Complexity of Assembly Partitioning

Lydia Kavraki Jean-Claude Latombe Randall H. Wilson

Robotics Laboratory, Department of Computer Science,
Stanford University, Stanford, CA 94305, USA

Abstract

This paper studies the complexity of the *assembly partitioning* problem in the plane: given a collection of non-overlapping polygons, decide if there is a proper subcollection of them that can be removed as a rigid body without colliding with or disturbing the other parts of the assembly. It is shown that assembly partitioning is NP-complete. The result extends to several interesting variants of the problem.

Keywords: computational complexity, computational geometry, assembly planning, motion planning.

1 Introduction

This paper studies the *assembly partitioning* problem in the plane: given a collection of non-overlapping polygons, decide if there is a proper subcollection that can be removed as a rigid body without colliding with or disturbing the other parts of the assembly.

The partitioning problem arises in assembly planning, where a sequence of (possibly simultaneous) assembly motions are sought to bring separated parts into their relative goal positions in an assembly. Natarajan [7] showed that in its most general form, assembly planning is PSPACE-hard. In practice, however, most real assemblies can be constructed with *monotone two-handed* assembly plans [5]. Such plans consist of a sequence of operations, where each operation merges two rigid subassemblies to make a larger one that stays rigid for the rest of the plan. For instance, the assembly in figure 1a can be constructed with a monotone two-handed plan, while the assembly in figure 1b cannot.

Since assembly is the reverse of disassembly, a monotone two-handed assembly plan can be found by *partitioning* the assembly—removing a rigid subassembly—and then partitioning each of the two subassemblies, etc., and finally reversing the motions. Thus partitioning is the key to assembly planning in this case.

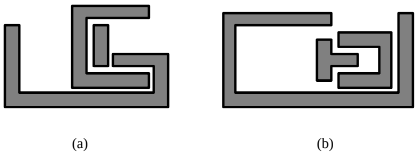

Figure 1: Assembly (a) admits a monotone two-handed plan, while (b) does not.

Recent work has presented polynomial-time partitioning algorithms in special cases. Arkin, Connelly and Mitchell [1] give an algorithm to partition assemblies of polygons if the separating motions are limited to single infinite translations. Wilson [13] presents algorithms to partition assemblies of polyhedra, where the separating motions are either infinite translations or infinitesimal rigid motions (the latter identifying a superset of the removable subassemblies for general separating motions). Interestingly, Snoeyink and Stolfi [11] present an assembly of convex polyhedra that cannot be partitioned. Other related geometric separation problems are studied in [3, 8, 9, 12].

In this paper we show that the partitioning problem for polygons in the plane is NP-complete. Our proof extends to some interesting variants of the problem, including the case where all motions are restricted to translations.

2 Complexity of Planar Partitioning

Let a *rigid motion* of a subassembly S be a set of simultaneous motions (translations and rotations) of the parts of S that preserve the relative positions of these parts throughout the motion. The subassembly S is then called a *rigid* subassembly.

Planar Partitioning (PP) *Given a set A of non-overlapping polygons in the plane, decide if there is proper subset S of A that can be separated from $A \setminus S$ by a collision-free rigid motion of S.*

We will show that PP is NP-complete. It is clearly in NP, since a nondeterministic algorithm can guess S and then invoke a path planner to find the path of S out of the assembly. Schwartz and Sharir [10] have shown that path planning can be done in polynomial time for the case considered here.

We show that PP is NP-hard by a reduction from 3-Satisfiability (3-SAT), a well known NP-complete problem [2]. An instance of 3-SAT is a set of clauses $C = \{c_1, c_2, \ldots, c_m\}$ on a set of boolean variables $U = \{u_1, u_2, \ldots, u_n\}$, where each clause is a disjunction of 3 terms. A term is either a variable u_i or the negation of a variable $\overline{u_i}$. The problem is to determine if there exists a truth assignment of the variables that satisfies the conjunction of the clauses.

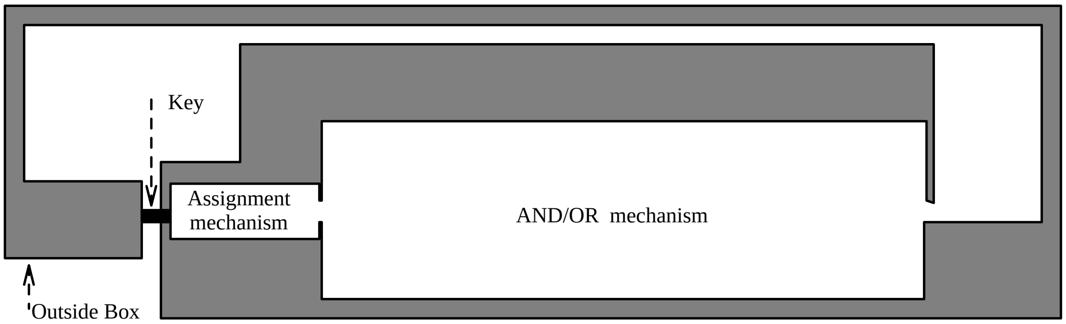

Figure 2: A sketch of the final assembly.

For any instance of 3-SAT, we construct in polynomial time an assembly of non-overlapping polygons that can be partitioned iff a satisfying truth assignment exists. Figure 2 shows the *outside box* and the *key* of the constructed assembly. Other parts are contained in the *assignment mechanism* and the *AND/OR mechanism*, detailed in figures 3 and 4 respectively. Our construction is summarized in the following:

- The part labeled as key in figure 2 must be removed before any other part is removed from the assembly. The reason is that the key blocks the only exit gate of the assembly.

- The key can be removed only through the assignment construct. For this to happen some other parts of the assignment construct must move rigidly with it. These parts represent a truth assignment for the variables of the 3-SAT instance.

- The subassembly can be removed only through the AND/OR mechanism. This mechanism enforces the clauses of the 3-SAT instance.

The assignment mechanism (figure 3) consists of (i) the walls of the assignment that are drawn in grey, (ii) the key which is a 3×1 rectangle drawn in black, and (iii) two 3×1 white rectangles for each of the variables of the 3-SAT instance. One of the two rectangles that correspond to the variable u_i is labeled with U_i, and the other with $\overline{U_i}$. Notice that U_i is placed always on top of $\overline{U_i}$ in the assignment construct. If U_i, ($\overline{U_i}$ resp.) is a member of S, we consider that the truth assignment *true* (*false* resp.), has been chosen for the variable u_i. We indicate with x_0 the initial position of the key and with x_i, the initial position of the assignment rectangles for the variable u_i, $i = 1, \ldots, n$. The choice of the x_i's is crucial and it is described below.

From figure 2 it is clear that the key initially can move only to the right. We observe that the key will not be able to pass through the assignment mechanism if no other parts of the mechanism are moved. In addition, the parts removed must translate rigidly.

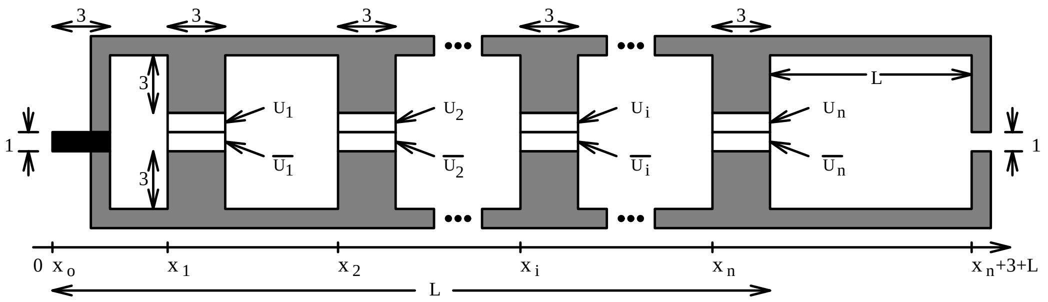

Figure 3: The assignment construct (not drawn to scale).

Hence, the only subassembly S that stands a chance to move out of the assignment mechanism, and eventually out of the total assembly, is a subassembly that consists of the key and at least one of U_i or $\overline{U_i}$, for each i. Since the exit gate of the assignment mechanism has a height of only 1, exactly one of U_i or $\overline{U_i}$, for each i, must be selected in S. Let L be the length of the moving subassembly S.

The collision-free motion of S out of the assignment construct is possible only if the x_i's are selected carefully. We present now our choice of x_i's and justify it. For the following assume that $S = \{R_0, R_1, \ldots, R_n\}$, where R_0 denotes the key and R_i denotes either U_i or $\overline{U_i}$, $1 \leq i \leq n$. Also, $pos(R_i)$ denotes the x-coordinate of the bottom-left vertex of the rectangle R_i.

Observation *Let $x_i = 10 \cdot a_i$, $i = 0, 1, \ldots, n$, where $a_0, a_1, \ldots, a_n$ is an integer sequence such that all pairwise differences $a_i - a_j$ are distinct when $i \neq j$. When $|x_p - pos(R_i)| < 4$, $p \neq i$, then for all R_j, $j \neq i$, we have that $|x_q - pos(R_j)| > 4$.*

Proof At some time during the motion of S, R_i is 4-close to the position x_p, that is $|x_p - pos(R_i)| < 4$. Consider now any R_j, $j \neq i$, and any position x_q. Let us bound the quantity $|x_q - pos(R_j)|$ from below:

$$|x_q - pos(R_j)| \geq |(x_p - pos(R_i)) - (x_q - pos(R_j))| - |x_p - pos(R_i)|$$

or equivalently,

$$|x_q - pos(R_j)| \geq |(x_p - x_q) + (pos(R_j) - pos(R_i))| - |x_p - pos(R_i)|$$

But since $|x_p - pos(R_i)| < 4$ and $pos(R_j) - pos(R_i) = x_j - x_i = 10 \cdot a_j - 10 \cdot a_i$ the above inequality becomes:

$$|x_q - pos(R_j)| > 10 \cdot |a_p - a_q - (a_j - a_i)| - 4.$$

From the hypothesis about a_n the quantity $|a_p - a_q + (a_j - a_i)| = |(a_j - a_q) - (a_i - a_p)|$ must be at least one, since $j \neq i$ and $i \neq p$. Thus we get:

$$|pos(R_j) - x_q| > 10 \cdot 1 - 4 > 4$$

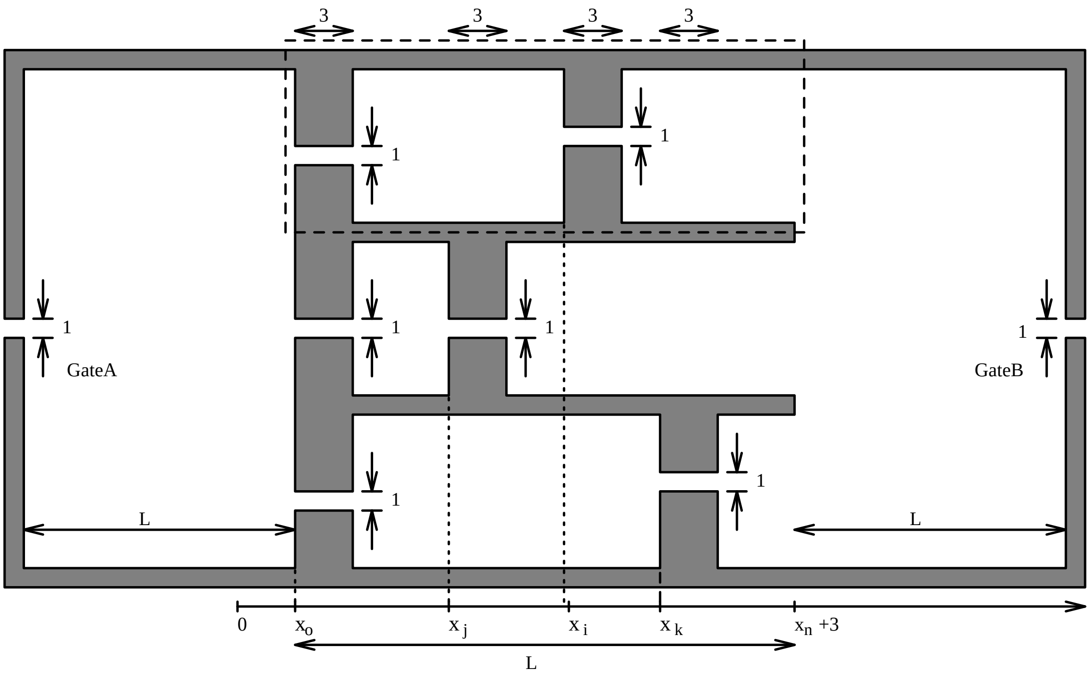

Figure 4: The OR gate for $c_l = u_i \vee \overline{u_j} \vee u_k$ (not drawn to scale).

which completes the proof. □

In other words, for any k when R_k is close to position x_p, $p \neq k$, and needs to go through the hole that has been created at this position, all the other parts of S are in the wide free sections of the assignment mechanism and can follow the constrained motion of the part that is close to x_p. Hence, the assignment mechanism ensures independent selection of the variable assignments of the 3-SAT instance and allows the resulting subassembly to translate out of the mechanism.

The key element in the above proof is the property of the sequence a_i, namely that the pairwise differences of its terms are all distinct. The 10 is just a scaling factor that gives extra space for the width of the parts. It is not hard to choose the a_i's; a straightforward example is given by $a_i = 2^i$. Using a result of Erdös [4], it is possible to select the a_i's in such a way that $max\{a_0, \ldots, a_n\} = n^2$. The complexity of the assignment mechanism is measured in the number of vertices, and it is polynomial in n, no matter which of the above sequences is used. Using the result of Erdös the physical length of the assignment mechanism is $O(n^2)$.

Once S translates out of the exit gate of the assignment construct it must pass through the AND/OR mechanism. This mechanism is a sequence of OR gates, one for each clause of the 3-SAT instance. The OR gate for the clause $u_i \vee \overline{u_j} \vee u_k$ is shown in figure 4. We observe from the figure that there are three possible ways for S to go from

GateA to GateB. Each of these enforces a truth assignment for one of the terms of the clause. Suppose for example that S goes through the section enclosed in the dashed box in figure 4. This section enforces the truth assignment *true* for the variable u_i. Here is why: when the key is at x_0, the rectangle chosen for the truth assignment of u_i is at position x_i. Unless U_i is selected in S, it is impossible for S to go through the dashed box of figure 4. Notice however that the rest of S can be threaded through the gates at x_0 and x_i without problems: because of the property of the x_i's mentioned above, when a part of S needs to go through the above gates, none of the other parts of S is close to a narrow passage. Hence, S can follow the motion of its constrained part without being obstructed by the walls of the OR gate.

Suppose there exists a satisfying truth assignment for the 3-SAT instance. Let S be the subassembly that consists of the key and encodes this truth assignment. S can go through the AND/OR mechanism since it can translate through each of its OR gates. Then S can translate to the upper left corner of the assembly, rotate there by 90 degrees and exit through the 2-unit wide gate that was initially blocked by the key.

Conversely, assume that the assembly in figure 2 can be partitioned and let S be the subassembly that is removed from it. S clearly contains the key. As we argue above, the key can be removed only in a subassembly that contains a truth assignment for the variables of the 3-SAT instance. Since S can pass through the AND/OR mechanism, it represents a satisfying truth assignment for the 3-SAT instance. Finally, it can be shown easily that the reduction presented above is polynomial in the size of the 3-SAT problem. It follows that:

Theorem 1 *PP is NP-complete.*

3 Some variants

In this section we present some variants of the partitioning problem that are also NP-complete. A detailed discussion of these variants can be found in [6]. For a different proof of variant 2.3 see also [14].

3.1 Partitioning with Translations

In the construction of the previous section, the moving subassembly S rotates in the upper-left corner of the assembly before exiting. We can modify that construction to remove the rotation needed, thus showing that planar partitioning with translational paths is NP-complete. A more complicated exit gate allows S to pass through in translation, requiring small changes in the rest of the construction.

The new exit gate is depicted in figure 5(b). A series of vertical channels allow the key and assignment rectangles to move down through the exit gate, then exit to the

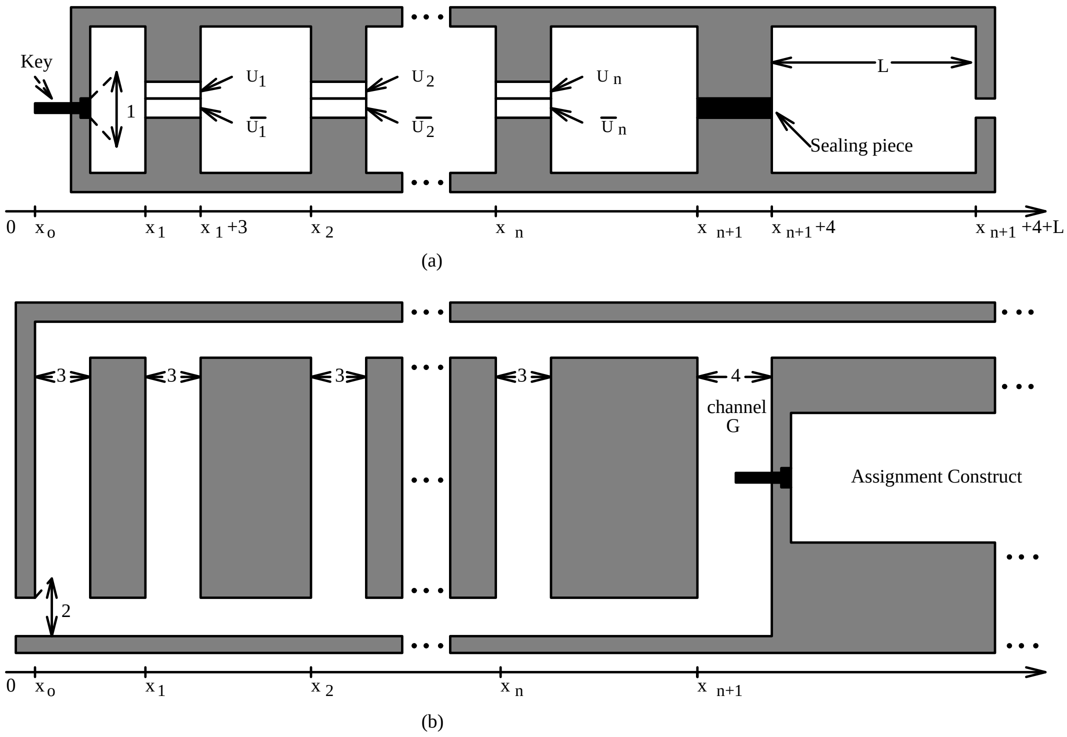

Figure 5: PPT: (a) The assignment mechanism (b) The exit mechanism.

left. However, with no other changes, rectangle U_n or $\overline{U_n}$ could be removed alone. To prevent this, a *sealing piece* is added to the assignment construct, to the right of all the assignment rectangles, at position x_{n+1} (see figure 5(a)); a corresponding channel (labeled G) is added to the exit gate. The sealing piece and channel G are both 1 unit wider than the other channels, and no rotations are allowed, so the sealing piece can only exit through gate G. Finally, since the key no longer touches the left side of G, its shape is changed to only allow movement to the right.

The sealing piece blocks the motion of the assignment rectangles, so it must be in S. The key blocks gate G, so it and exactly one of U_i or $\overline{U_i}$, for all i, must also be in S. Since the sealing piece is at position x_{n+1}, it does not affect the ability of S to move out of the assignment construct or through the AND/OR mechanism.

3.2 Partitioning on a Grid

Consider a polygonal assembly whose parts must (i) have their vertices on an $N \times N$ grid, (ii) have only right angles, and (iii) translate only in the vertical and horizontal directions by grid increments. N is considered the size of the problem. The planar partitioning problem for this assembly is called *partitioning on a grid* and is the most constrained NP-complete partitioning problem we have identified.

Partitioning on a grid is clearly in NP: we guess a subassembly and a path of N^2 unit steps on the grid, then check whether the path is collision-free and remove the subassembly.

The reduction from 3-SAT is possible only because we can construct an assignment mechanism whose length is polynomial in the size of the 3-SAT instance (see section 2). This leads to an assembly of length $O(mn^2)$, where n is the number of variables and m is the number of clauses of the 3-SAT instance. Since the construction of section 3.1 satisfies the above restrictions on the assembly, and the grid rules out no interesting motions, partitioning on a grid is NP-complete.

3.3 Partitioning of Assemblies of Polyhedra

Each of the polygonal assemblies constructed in the previous sections can be made into an equivalent assembly of polyhedra by giving each polygonal part a thickness of one, and attaching plates on either side of the outside box that completely enclose the rest of the parts. The only possible motions for the inside parts then lie in the plane. Hence partitioning an assembly of polyhedra is also an NP-complete problem.

4 Concluding remarks

The reader may note that the assemblies above cannot be fully disassembled. Our construction can be modified to allow complete disassembly showing that assembly sequencing in itself is NP-complete [6, 14]. Space does not allow us to present this result here.

In experimental assembly sequencing, an additional constraint is often added, requiring both S and $A \setminus S$ to be connected. A subassembly is considered connected if the union of its parts is a connected set. This constraint is useful in practice, since connected assemblies are easier to grasp and manipulate. It is not hard to see that partitioning of assemblies of polyhedra into connected subassemblies is NP-complete: in the construction of section 3.3 a plate can be placed over the augmented parts, rigidly attached to the key and removed with S. However, it remains an open problem whether partitioning of assemblies of polygons under the connectedness constraint is NP-complete.

Acknowledgments The authors thank D. Halperin and L. Guibas for their comments. This work was supported by ARPA grant N00014-92-J-1809 and NSF grant IRI-9306544.

References

[1] E.M. Arkin, R. Connelly, J.S.B. Mitchell, "On monotone paths among obstacles with applications to planning assemblies", *Proc. of the 5th ACM Symposium on Computational Geometry*, pp. 334-343, 1989.

[2] M.R. Garey, D.S. Johnson, *Computers and Intractability, A Guide to the Theory of NP-Completeness*, Freeman, New York, 1979.

[3] L. Guibas, F. Yao, "On translating a set of rectangles", *Computational Geometry*, F. P. Preparata (ed.), series: Advances in Computing Research, Vol. 1, JAI Press, London, pp. 61-67, 1983.

[4] H. Halberstam, K. F. Roth, *Sequences*, Springer-Verlag NY, pp. 90, 1983.

[5] L.S. Homem de Mello, S. Lee editors, *Computer-Aided Mechanical Assembly Planning*, Kluwer Academic Publishers, Boston, 1991.

[6] L. Kavraki, J.-C. Latombe, "Complexity of partitioning a planar assembly", TR STAN-CS-93-1467, Dept. of Computer Science, Stanford Univ., 1993.

[7] B.K. Natarajan, "On planning assemblies", *Proc. of the 4th ACM Symposium on Computational Geometry*, pp. 299-308, 1988.

[8] D. Nussbaum, J. R. Sack, "Dissasembling Two-Dimensional Composite Parts Via Translations", *Inter. J. of Computational Geometry*, 3, pp. 71-84, 1993.

[9] R. Pollack, M. Sharir, S. Sifrony, "Separating two polygons by a sequence of translations", *Discrete and Computational Geometry*, 3, pp. 123-136, 1988.

[10] J.T. Schwartz and M. Sharir, "On the piano movers' problem: I. The case of a two-dimensional rigid polygonal body moving amidst polygonal barriers", *Communications on Pure and Applied Mathematics*, 36, pp. 345-398, 1983.

[11] J. Snoeyink and J. Stolfi, "Objects that cannot be taken apart with two hands", *Proc. of the 9th ACM Symp. on Computational Geometry*, pp. 247-256, 1993.

[12] G.T. Toussaint, "Movable separability of sets", *Computational Geometry*, Elsevier Science Publishers, North Holland, 1985.

[13] R. H. Wilson, *On Geometric Assembly Planning*, PhD Thesis, Stanford University, March 1992, Stanford Technical Report STAN-CS-92-1416.

[14] R. H. Wilson, J.C. Latombe, T. Lozano-Pérez, "On the complexity of partitioning an assembly", TR STAN-CS-92-1458, Dept. of Comp. Science, Stanford Univ., 1992.