Reduction Tools and Strategies
for the Simplified Piano Mover's Problem

JD Nir
15-455 Undergraduate Complexity Theory
Carnegie Mellon University

November 20, 2012

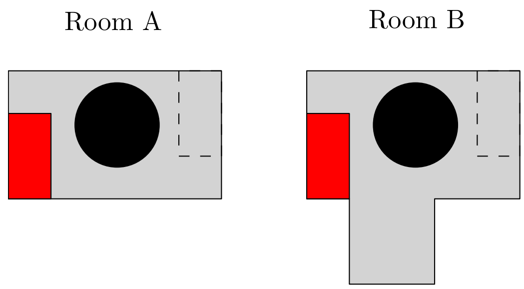
Abstract

Imagine a closed room without doors or windows containing a piano and other furniture. Is it possible to move the piano to a given position in the room by moving around the rest of the furniture? In this paper I give a simplified version of this problem (the SIMPLIEFIED PIANO MOVER PROBLEM) and prove it is NP-complete. I will give reductions from PARTITION, SET COVER, and SAT to SIMPLIFIED PIANO MOVER PROBLEM. I will identify useful tools used in the reductions and remark on other situations in which they could be used. Finally, I discuss how one could approach a Cook's Theorem-like proof that SIMPLIFIED PIANO MOVER PROBLEM is NP-hard by giving a reduction from a general nondeterministic Turing Machine paying special attention to areas in which the tools I've identified in other reductions could apply.

1 Introduction

Imagine a piano and some furniture locked in a room without doors or windows. Suppose you want to move the piano from its current position to some other fixed position. Does there exist a way to shift furniture to get the piano where you want it?

Consider two examples:



In room A, it is not possible to move the piano to the dotted region. In room B, however, by moving the circular table out of the way one could piano to it's final position.

We define the PIANO MOVER PROBLEM as follows:

INSTANCE: A closed region R, two sets $W = \{w_1, \ldots, w_m\}$ and $F = \{f_1, \ldots, f_n, P_i\}$ of closed regions contained in R such that no element in either set intersects with any element in either set except possibly at boundary points, and a closed region P_f isomorphic to P_i not intersecting any $w_i \in W$.

QUESTION: Keeping each w_i fixed (think of the w_i as internal walls or pillars), is there a sequence of translations and rotations of elements of F such that the region initially found at P_i is translated and rotated to P_f and at no point does any element of F intersect with any other element of F or W.

The general PIANO MOVER PROBLEM is very powerful, but I have not determined whether it is NP-complete, PSPACE-complete, or even in a different complexity class altogether. Instead, we will consider a simpler version of the problem.

Define a *simple* polygon to be a polygon that can be partitioned into 1 unit by 1 unit squares; in other words, polygons with integer side lengths and all right angles. The SIMPLIFIED PIANO MOVER PROBLEM is

INSTANCE: A simple polygon R, two sets $W = \{w_1, \ldots, w_m\}$ and $F = \{f_1, \ldots, f_n, P_i\}$ of simple polygons contained in R such that no element in either set interects with any element in either set except possibly at boundary points and each simple polygon has each edge parallel or perpindicular to the edges of R, and a closed region P_f isomorphic to P_i not intersecting any $w_i \in W$.

QUESTION: Keeping each w_i fixed, is there a sequence of translations (but not rotations) such that the region initially found at P_i is translated to P_f, at no point does any element of F intersect with any other element of F or W, and no peice of furniture is moved more than C times for some large constant C.

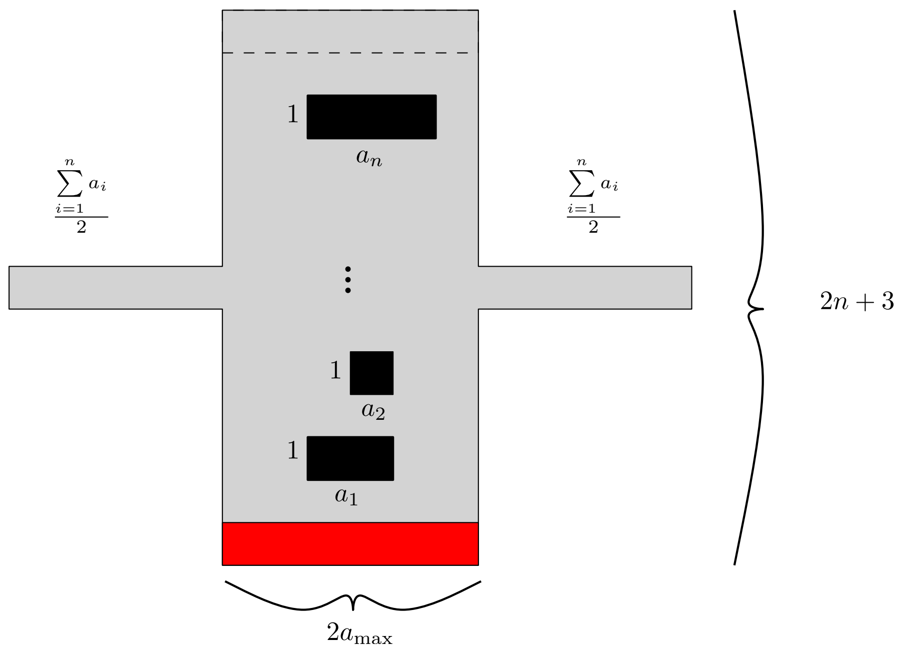
It is easy to see the SIMPLIFIED PIANO MOVER PROBLEM is in NP; a certificate is just a list of translations bounded above by $C|F|$ so verifying merely requires checking that no overlap occurs and the piano ends up in the correct position.

In the next section we prove SIMPLIFIED PIANO MOVER PROBLEM is NP-complete.

2 Reductions to SIMPLIFIED PIANO MOVER PROBLEM

In this section I give reductions from PARITION, SET COVER, and SAT to SIMPLIFIED PIANO MOVER PROBLEM.

First, consider an instance of partition with elements $a_1, \ldots, a_n$. Let $a_{\max} = \max_{1 \le i \le n} a_i$. Then consider the following instance of SIMPLIFIED PIANO MOVER PROBLEM:

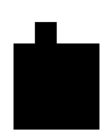


Once again, the piano is shown in red. Suppose there is a solution to the PARTITION instance. Then let one "closet" represent set A and the other represent set B. Using at most three translations per item, slide the table with width a_i into closet A if $a_i \in A$ and into B otherwise. Then the piano can slide through. Furthermore, if there is no solution to the PARTITION instance, then the furniture cannot be packed into the closets and will block the piano from crossing the room. Finally, assuming both the PARTITION instance and the dimensions of the objects are given in binary, the reduction is clearly polynomial in the lenghth of the input.

This completes the proof that SIMPLIFIED PIANO MOVER PROBLEM is NP-complete. Thus we can reduce any problem in NP to it. We now consider one such reduction from SET COVER.

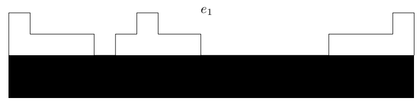
Let $\{S_1, \ldots, S_\ell\}$ and $\{e_1, \ldots, e_n\}$ be an instance of SET COVER with bound k. We construct a reduction as follows:

- Represent each S_i as an $\ell \times \ell$ square with one additional 1×1 square on the top edge i units from the left side. For instance, if $\ell = 4$, represent S_2 as

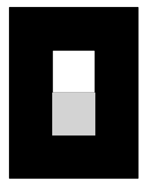
  Each set is initially in the main room.

- Represent each e_i as a set of closets that hold the furniture corresponding to those sets to which e_i belongs. For instance, if $\ell = 4$ and e_1 belongs to S_1, S_2 and S_4, but not S_3, then represent it by



The elements are lined up at the top of the main room.

- For each element, put a $2 \times (\ell^2 + (\ell - 1))$ "blocker" (represented as b_1 above) that blocks access to the closets of that element.
- Between every pair of blockers, put a "lock":

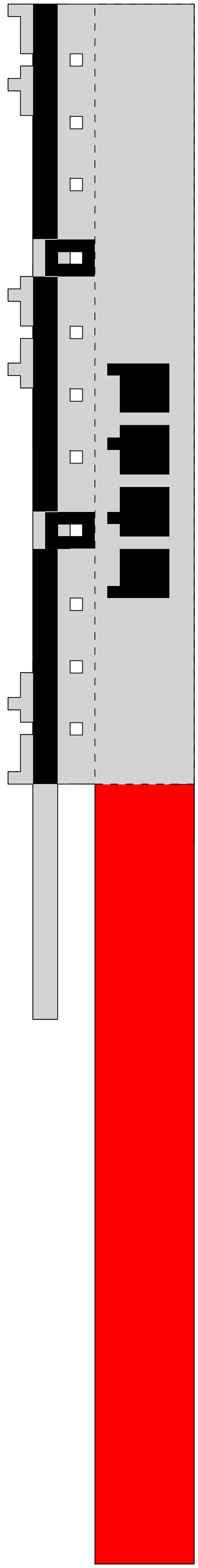


Note that there is a 1×1 column in the center of the lock; in essence it can slide up and down one unit.

- At the end of the row of "blockers" is a hallway of width 2 and length $k(\ell^2 + (\ell - 1)$. (Recall k is the bound given for the SET COVER instance.)
- Finally, the piano is very large; it has width $2n - 2$ and length $n(\ell^2 + (\ell - 1)) + 3(n - 1)$. Initially it is in a closet. It's ultimate position takes up most of the main room.

On the next page we give an example of the reduction for the SET COVER instance with elements $\{1, 2, 3\}$, sets $\{\{1\}, \{1, 2\}, \{2, 3\}, \{3\}\}$ and $k = 1$.

If a solution to the SET COVER instance exists, then the appropriate elements can be unblocked and the sets fit into their closets. Note that in order for the piano to be in position, the locks must be in the "up" position as otherwise they occupy some of the space required by the piano. Thus the blockers must be perfectly between two locks or in the closet at the end. The selection of elements corresponds to those whose blockers are in the closet at the end. Thus if a solution to SIMPLIFIED PIANO MOVER PROBLEM exists, a solution to SET COVER must also exist. Finally, we note the entire reduction fits in a $4n \times n\ell^3$ rectangle and contains $O(\ell + n)$ objects and so is polynomial.



Finally, we give a third reduction, from SAT to SIMPLIFIED PIANO MOVER PROBLEM.

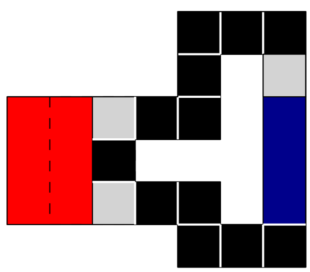
Let $\{x_1, \ldots, x_n\}$ be a set of variables and $\{C_1, \ldots, C_\ell\}$ be a set of disjunctive clauses containing those variables and their negations.

For each variable x_i, create a $(1 \times (2\ell - 1))$ piece of furniture with 1×1 "teeth" at $2i - 1$ on the top row if x_i appears in clause C_i nonnegated and on the bottom row if x_i appears negated in clause C_i. For instance, if the set of clauses is $\{x_1\}, \{x_2 \vee \neg x_3\}, \{x_1 \vee \neg x_1\}$, the piece of furniture corresponding to x_1 is

The piano similarly has teeth, in this case for each clause. The width of the tooth corresponding to clause C_i is $2|C_i| - 1$. For the clauses above, the piano looks like:

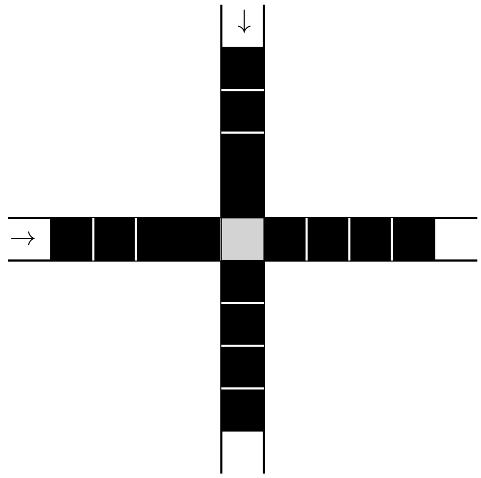
Now we come to the crucial idea of the reduction. The piano's final position is only one unit lower than it's current position. Blocking its progress are a sequence of 1×1 pieces of furniture representing a "pipe". The idea is for each clause, there is a pipe for each variable in that clause that "flows" to the corresponding tooth on the piece of furniture representing the variable. The variable has two positions, up and down, representing whether the variable has been assigned to true or false. When the variable is assigned to true, it creates a gap into which the 1×1 pieces of furniture can "flow" making room for the piano to move. As long as at least one variable per clause is satisfied by the assignment, the piano will have room to move.

As a simple example, consider the set of clauses containing the single clause $\{x_1, \neg x_1\}$. The reduction looks like:



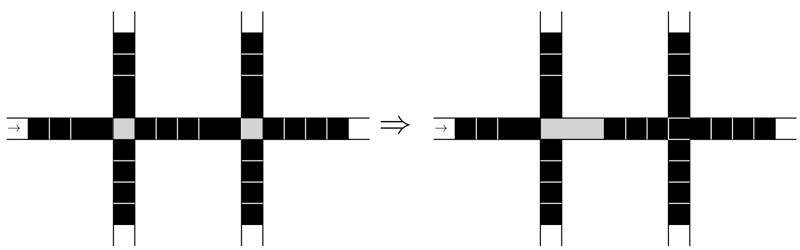
Note that we have colored the piece of furniture corresponding to x_1 dark blue to distinguish it. In this case, x_1 is in the down position (which we arbitrarily decide assigns x_1 to true). There is space for the x_1 "pipe" to flow, but not for the $\neg x_1$ pipe. Of course, we could have just as easily assigned x_1 to be false, put the corresponding piece of furniture in the up position, and the situation would be reversed.

Unfortunately, most reductions are not so simple. In general, we must allow for our pipes to cross one another. We introduce the following gadget to do so:



Note that the arrows indicate the intended direction of flow and that the square at the center of the cross is empty. The 2×1 pieces of furniture can make it through the crossing only if the 1×1 furniture on the other side can flow. Furthermore, the 2×1 furniture is stuck in its orientation, unlike the 1×1 squares which could flow into the wrong pipe.

Unfortunately, this gadget introduces a fair bit of complexity to our reduction because it requires an additional empty space. The first complication occurs when one pipe crosses two other pipes. Starting from the furniture representing the variable and working back towards the piano, the first crossing we encounter works, but the second does not, as is illustrated below:



The problem is that the 2×1 piece of furniture now slides all the way through, allowing the 1×1 piece of furniture to flow into the wrong pipe. However, we can fix this problem by changing that 1×1 piece of furniture into another 2×1 piece of furniture. In general, for the j^{th} crossing, again working backwards, we need j 2×1 pieces of furniture.

However, the extra space introduces an additional problem. We could "jam" an intersection clearing a space for a block to flow even though the corresponding variable is not set appropriately. To solve this problem, we will add more 1×1 squares to block the piano. Before we describe just how many, we need a lemma.

Lemma: For any reduction in which two "pipes" cross more than once, there is an equivalent reduction in which those pipes cross at most once.

Proof: It suffices to show if two pipes cross twice, there is an equivalent reduction in which they do not cross and no new crossings are introduced. Label the pipes A and B partition them into three parts, A_1 and B_1 that occur before the first crossing, A_2 and B_2 that occur between the crossings, and A_3 and B_3 that occur after the second crossing. In the equivalent redution, rather than crossing, pipe A simply consists of the segments A_1, B_2, and A_3, and pipe B consists of the segments B_1, A_2, and B_3.

Thus if we have n variables and ℓ clauses and each variable occurs at most twice (once negated and once not negated), then we have at most $2\ell n$ pipes and at most $\binom{2\ell n}{2}$ crossings. Thus each "tooth" of the piano is actually of length $\binom{2\ell n}{2}$ instead of 1 and there are $\binom{2\ell n}{2} + 1$ 1×1 squares blocking the tooth instead of one.

Of course, we must also extend the teeth on the furniture representing our variables to provide additional space for the new pieces of furniture. In this case we extend the width instead of the length; they are also of width $\binom{2\ell n}{2}+1$ so that if even one variable in a clause is satisfied, all of the blocking 1×1 squares can flow into appropriate pipes.

Formally proving that this reduction maps YES instances to YES instances and NO instances to NO instances is left to the reader, but I will show briefly that this reduction is polynomial. The piano has dimension at most $4n\ell \times 16n^2\ell^2$. Each of the n variables is represented by a piece of furniture of dimension at most $16n^2\ell^3 \times 3$. The number of pieces of furniture contained in the pipes is linear in the length of the pipe. There are $O(n^2\ell^2)$ many pipes, and no pipe need be longer than approximately $1000n^6\ell^6$ as all other objects defined thus far fit in a box of that size with enough room for all the other pipes, including crossings.

3 Thoughts on a General Reduction

In the previous section I gave three reductions to SIMPLIFIED PIANO MOVER PROBLEM. I now consider the structures in those reductions that could be aplicable in other reductions. In particular, though the three decision problems, PARTITION, SET COVER, and SAT, have very different structure, their reductions have some common elements.

The concept of a closet defined by some parameter of the original equation is present in the reductions of both PARTITION and SET COVER. In the latter, I also used distinctive notches to distinquish between distinguishable objects. The SET COVER reduction also introduces the concept of a locking mechanism to assure furniture can only be moved in discrete steps. Similar mechanisms were present in earlier designs of the SAT reduction, though they were found to be unnecessary. Finally, the SAT reduction introduced the concept of a flow which allows complex mechanisms to affect each other "at a distance" making life much easier for budding reduction designers.

Can we use these tools to create a reduction from an arbitrary problem in NP? Of course we can apply our SAT reduction to the SAT instance given in Cook's theorem, but the result would be both huge and unintuitive.

One possible avenue would be to try and simulate a nondeterministic Turing Machine. Characters in Γ could be represented using distinct notches, as in the SET COVER reduction. Locks could be used to assure the tape is in an appropriate position. The controller of the machine could use flows to pass control between various states.

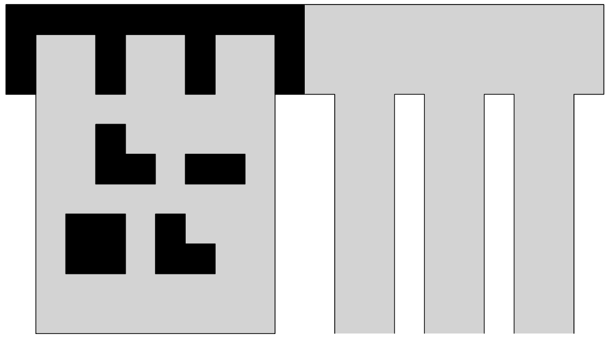
While this first idea shows some merit, we should also take pause. Recall that one of the restrictions on SIMPLIFIED PIANO MOVER PROBLEM is that no object may be moved more than C times. In the reductions above, it is relatively easy to prove that $C \approx 10$ suffices. In this general solver, however, particularly using flow to pass control between states, it may be impossible to give a constant bound on the number of times one piece of furniture is moved.

Returning to Cook's Theorem for inspiration, a second approach would be to model a tableau. Recall from Cook's proof the five necessary abstractions:

1. A way to represent the contents of cell i at timestep j,
2. a way to assure every cell has exactly on value at each timestep,
3. a way to assure the first row of the tableau is the starting configuration of our tableau N on input w,
4. a way to assure an accepting state occurs somewhere in the tableau, and
5. a way to verify that every row follows legally from the preceeding row.

I will consider points one, two, and four; the others are left as areas for further research.

As mentioned in the first attempt at a general reduction, it is possible to assign each element of Γ a distinct piece of furniture. To represent the state of the tape at timestep j, imagine two rooms mostly separated by a wall. One room contains many copies of the furniture representing each character. The other contains several corrals; one for each cell on the tape. The only way that furniture can get to the other side of the wall is by slotting into a transport piece of furniture with one closet per cell on the tape.



If there were some way to adapt our locking mechanism to assure the transport furniture does not move backwards, the i^{th} copy of this gadget could place exactly one character in a corral where they could be taken for further processing. This gadget covers point one, and assuming the rest of the problem is not possible without a piece of furniture in each corral, point two as well.

To cover point four, we assume there is a set of rooms corresponding to the state of the controller at each time step. Perhaps in each room is a "plug" that can only move out of the way if the computation has reached an accept state. When the plug is uncorked, furniture can flow into a pipe, clearing the way for a piano, located elsewhere, to move to its final position.

4 Conclusion

I have shown that the SIMPLIFIED PIANO MOVER PROBLEM is a very expressive language. Thinking about various reductions gives a very general framework for thinking about reductions of other NP-complete problems. While it is sometimes more difficult to create formal proofs that reductions are correct, the visual structure of SIMPLFIED PIANO MOVER PROBLEM makes reductions easy to explain and understand.

It is still unclear if there is a better general reduction from an aribitrary language in NP to SIMPLIFIED PIANO MOVER PROBLEM than applying the reduction from SAT to the SAT instance given in Cook's Theorem, but I have given some ideas for how parts of such a reduction would work.