Proceedings of the Fifth Canadian Conference on Computational Geometry, A. Lubiw and J. Urrutia Eds.,
University of Waterloo, Ontario, Canada, August 5–9, 1993, pp. 7–11.

The Complexity of the Compaction Problem

Zhenyu Li * Victor Milenkovic†
Aiken Computation Laboratory,
Harvard University,
Cambridge, MA 02138.

1 Introduction

We define the *compaction* problem as a coordinated motion planning problem for a set of objects such that the area of the bounding convex hull or the bounding rectangle is minimized.

Since we are only interested in establishing lower bounds, we concentrate on a special case of compaction: all the objects are confined to move within a bounding rectangle whose top, left and bottom sides are fixed and whose right side can only move inward. It is easy to show that we can reduce this special case to the more general case of compaction by introducing a properly shaped sleeve piece and a piston piece. Therefore the lower bound of the special case is also a lower bound of the general case.

The coordinated motion planning problem for a collection of rectangles inside a fixed rectangular box (the warehouseman problem) has been studied by Hopcroft, *et. al.* [2]. They obtained a well-known PSPACE-hardness result for the problem by a reduction from the PSPACE-hard symbol transposition problem. Spirakis and Yap [4] studied the coordinated motion of a collection of discs in a polygonal container and proved the strong NP-hardness of the problem. Since the possibility of PSPACE=P and NP=P has not been ruled out, the hardness results do not imply that it requires exponential time to decide the *existence* of a coordinated motion plan, even though the shortest plan might have an exponential number of moves. Chazelle *et. al.* [1] showed that a special case of motion planning requires exponential number of moves. But in their case, only one object is allowed to move at a time. There is no previously known explicit construction of a coordinated motion planning that requires an exponential number of moves.

This paper is organized as follows. In Section 2, we establish the PSPACE-hardness of the compaction problem, and we prove the existence of sets of rectangles which require and exponential number of moves to compact. In Section 3, we prove that if some simple nonrectangular objects are allowed, one can explicitly construct warehouseman and compaction problem which require a number of moves exponential in the number of edges in the input. In the last section, we show that if more complicated shapes are allowed, even finding a locally optimal solution to the compaction might require an exponential number of moves. This result establishes that the local compaction algorithm presented in [3] has an exponential time worst case.

*This research was funded by the Textile/Clothing Technology Corporation from funds awarded to them by the Alfred P. Sloan Foundation.

†This research was funded by the Textile/Clothing Technology Corporation from funds awarded to them by the Alfred P. Sloan Foundation and by NSF grants CCR-91-157993 and CCR-90-09272.

2 The PSPACE-hardness of compaction

Hopcroft *et. al.* [2] proved the PSPACE-hardness of the warehouseman problem by an ingenious reduction from the PSPACE-hard symbol transposition problem to the warehouseman problem. Here we will follow their approach and reduce the transposition problem to the compaction problem. Our reduction is built on top of their construction with some additional construction to force the warehouseman problem to reach the final state when compaction has a solution. In the next section, we review the reduction used in [2], with emphasis on the facts that are related to our modification.

2.1 Review of the warehouseman problem

The layout of rectangular objects in the reduction in [2] is shown in Figure 1. The bounding rectangle has dimension $8 \times 8k$ where k will be specified later (we followed the convention in [2] to represent dimension as height × length). The large blocks A through H are called *sliders*. Their dimensions are shown in the figure. The $2 \times 2k$ space between A and B is filled with 2×2 blocks called *dominoes* which play the role of symbols in the symbol transposition problem. The 3×2 blocks in the lower left corner are called *bricks*. The small $1 \times \frac{1}{2}$ blocks between D, E and F are called *fillers*. There

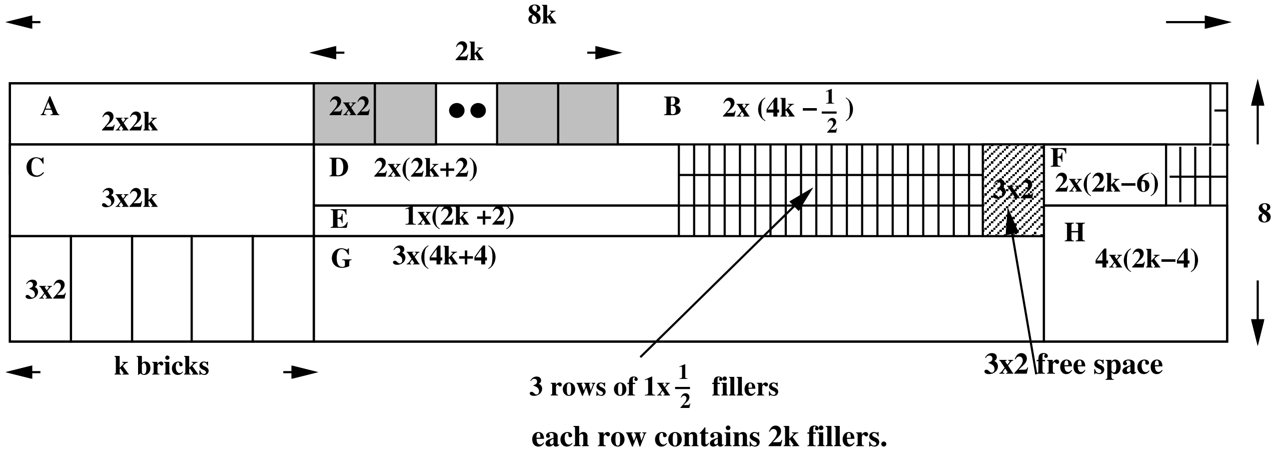


Figure 1: The reduction of warehouseman problem to symbol transposition problem

are some additional fillers between B, F and the right boundary. The 3×2 free space to the left of F is the only free space within the bounding box.

There are two meaningful motions allowed in this layout. First, C, D, E and the fillers to right of D, E can move to the right to fill the 3×2 empty space. One brick below C can move up and fill the space opened by C. G can subsequently move to the left and 12 fillers can drop down to fill the space left by G. Such a circular motion can continue until all the bricks move up to the middle row, or can be reversed to move in the opposite direction. Second, during any step of the circular motion, free space can be opened up between C and D to allow a domino to drop down. Then B can move to the left, the fillers to the right of F can move up and F can move next to the right wall. Eight fillers to the right of D and E can take the space to the left of F. A new 3×2 free space will be opened up to resume the circular motion which can ship the domino between C and D to the left or right and reinsert it back into the row of dominoes. This latter motion simulates the transposition of symbols.

To simulate the transformation rules in the symbol transposition problem, each domino is further partitioned into smaller rectangles as shown in Figure 3. The partition of a domino consists of a bottom rectangle, a thick vertical spine, and a left and right set of rectangles. The spines all have the same dimension. The rectangles in the left and right set have decreasing heights when counted from bottom to top. The lengths of the rectangles are varied slightly around a nominal length to form various "dents" and "tabs" which can be used to enforce symbol transformation rules that govern whether a pair of dominoes can be neighbors.

In addition, there are objects called *spacers* between the dominoes. A spacer is much longer than a domino so that it cannot be dropped down into the free space opened between C and D during the sequence of motion. A spacer can "transmit" the fit between a pair of dominoes on its two sides. The choices of the height of the rectangles in a domino together with the existence of the spacers ensure that (1) any pair of dominoes cannot be directly adjacent to each other, there must be at least one spacer between them (2) the rectangles of a domino cannot be rearranged within a domino and (3) the rectangles cannot be exchanged between two dominoes.

The amount k unspecified at the beginning can now be determined as the total length of all the symbols between block A and B (divided by 2).

Based on this construction, Hopcroft *et. al.* proved that the symbol transposition problem has a solution if and only if the dominoes in their initial configuration can reach a final configuration through a continuous coordinated motion.

2.2 Reduction of the compaction problem from the warehouseman problem

To reduce the symbol transposition problem to the compaction problem, we need to modify the construction so that the dominoes in their initial state, which represents an initial string in the transposition problem, can reach the final state if and only if the sliding wall in the compaction problem can move downward.

Let δ be the minimum height of the top rectangles in the left or right set of the dominoes. The height of the spine of a domino can be shortened by as much as $\epsilon = \frac{1}{5}\delta$ and still make the whole reduction work. This follows from the fact that the top rectangles in the left or right set of a domino have the smallest height among all the rectangles comprising a domino. Thus, the tiny free space on top of a spine cannot affect the two significant motions in the warehouseman problem, nor can it introduce new significant motions.

Assume there are n different types of dominoes. We shorten the height of the spine of a type i domino by $i\epsilon/n$. We introduce n types of *plug* blocks whose length equals that of the spine. A type i plug block has height $4 + i\epsilon/n$. Thus, when a type i plug is put immediately on top of the spine of a domino, the total height is 6 units.

Figure 2 shows the additional constructions we need for

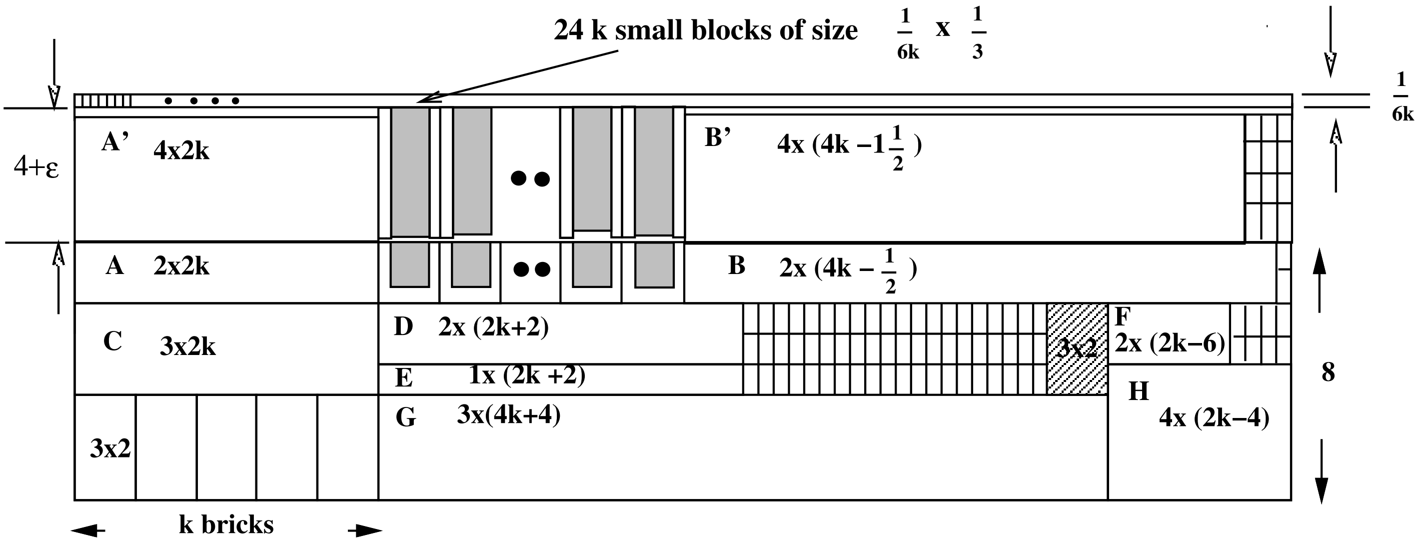


Figure 2: The reduction of compaction from the symbol transposition problem through the reduction of the warehouseman problem

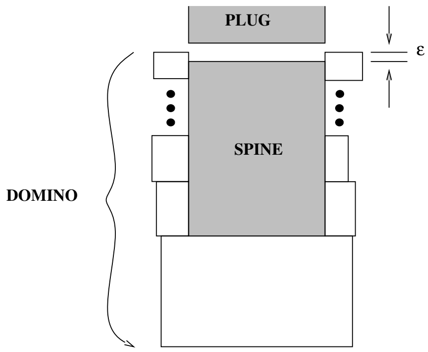


Figure 3: The reduction of compaction from symbol transposition problem through the reduction of the warehouseman problem

the reduction. Let R denote the bounding box of the original construction. We will first delete the top edge of R and extend the two side edges upwards. We arrange the plugs in such a way that a plug of type i is put above where a type i domino would be in the final configuration of the warehouseman problem. Between the plugs, there are *stuffer* blocks of height 4 and length equal to the nominal length of the rectangles in a left or right set. Once plugs and stuffers are put into place, it is not possible for them to exchange positions. Initially, the top edges of the plugs and stuffers are aligned along a horizontal line which is $4+\epsilon$ above the top edge of block A. The maximum height of free space between the plug row and the domino row is at most 2ϵ (e.g. when a type 1 plug is put on top of a type n domino), which is less than half of the smallest height among dominoes rectangles. Once again, the existence of these tiny free spaces will not change the two significant motions inside R.

In order to force the dominoes to the desired final state, we must also take away the 3×2 free space inside rectangle R when the desired final configuration is reached. Otherwise, we can push D to the right by 2 units and drop a type 1 domino to the free space left between C and D. If that happens, there can be many possible arrangements for the rest of the dominoes such that the sliding wall can drop down by ϵ. The 12 additional $1 \times \frac{1}{2}$ *fillers* to the right of B' can serve the purpose of filling up the free space. We need an extra layer of $24k$ small blocks of size $\frac{1}{6k} \times \frac{1}{3}$ to fill the space left open by the 12 fillers to the right of B'. If we put the sliding wall horizontally at the position $4 + \frac{1}{6k} + \epsilon$ above the top edge of block A, then the sliding wall can drop down by $\frac{1}{6k} + \epsilon$ if and only if the dominoes are in their final configuration. It is easy to see that this reduction like the reduction in [2] takes polynomial time. Thus we have

Theorem 1 *Compaction is PSPACE-hard even for a collection of rectangles.*

3 Compaction in an exponential number of moves for non-rectangular objects

The PSPACE-hardness of the symbol transposition problem shows that it is possible to simulate a binary counter by moving rectangular objects inside a rectangular container such that reaching a goal state from an initial state requires an exponential number of moves. But such a simulation may be complicated to construct and to describe. Here we show that if we relax the restriction that all the objects must be rectangles, we can have an example which is small in size and easy to understand. The example shows that even when the compaction problem involves simple objects such as U-shaped objects, we still have to deal with an exponential number of moves.

Our construction is shown in Figure 4. We build a rectangular container of dimension $(6n+2) \times (4n+2)$. Two vertical bars of width 1 and height $3n+1$ divide the space within the container into three equal sized slots. We call them slot 1, 2 and 3, from left to right. The bottom half of each slot is taken by some rectangular blocks. There is one $2n \times 2n$ block in slot 1 and two $n \times 2n$ blocks on each of slots 2 and 3. Due to the vertical bars, these rectangular blocks cannot be moved from one slot to another. On the upper half of the third slot, there are some U-shaped objects: $U_1, U_2, \ldots, U_n$ stacked together vertically. The walls of these objects have width $\frac{1}{n}$. All of these objects have height n. The bottom of wall of U_i is $\frac{2n-2i}{n}$ long.

The height of the vertical bars ensures that only one U-shaped object can pass through the openings on top of the bars. Therefore, at any particular move, only the object on top of a stack can be moved. There is a $2n \times n$ block initially at the bottom of the left slot. This block can slide between the bottom of the slots and is called a *slider*.

It is easy to see that the right wall of the container can be moved to position $4n+2$ if and only if all the U-shaped objects are moved to the upper half of the first slot (see Figure 5 (a)).

It is easy to verify that to solve the compaction problem, a problem harder than the $n-3$ disc Tower of Hanoi problem has to be solved[1]. Thus the total motion required is at least $\Omega(2^{n-3})$ which is exponential in n.

Theorem 2 *Coordinated motion planning requires number of moves that is exponential in the number of edges in the input.*

Proof: Omitted. ■

4 Finding a local minimum requires an exponential number of moves

In this section, we first give another construction that involves more complicated shapes to show that compaction requires an exponential sized motion. Then, based on the construction, we show that even finding a local optimum, according to an algorithm presented in [3], requires an exponential number of steps.

Consider the device shown in Figure 6. The device is composed of four movable pieces and a static obstacle that also provides the top, left and bottom boundary. The obstacle is a single connected piece and has a row of k "teeth" on the top and a row of k "teeth" on the bottom. A tooth is one unit wide and one unit high. The teeth on the top are displaced a half unit to the right with respect to the teeth on the bottom. The four movable pieces are labeled **A, B, C** and **D** as shown in Figure 6. The goal is to move piece A to the left by k units. To this end, we need to move piece **B** through the two rows of teeth. During the motion, a total of k up and down motions will occur for piece **B**. **B** moves one unit vertically during each up or down motion. One unit of vertical motion of **B** will be transferred through piece **C** to piece **D**. The two slanted edges in piece **D** have slope $\frac{1}{k}$. Therefore, each up (down) motion of one unit in **B** will force **D** to move to the right (left) by k units.

The construction is actually replicated n times as shown in Figure 7. The top part is connected to the bottom part ultimately at the far left. The pieces are organized into n groups with each group containing four pieces. There are overlap in components among the groups since the piece D in one group also serves as piece A in the next group. From left to right, the groups are ordered from 1 to n. By the analysis in the previous paragraph, the k units of horizontal motion of piece **A** in group 1 force k^2 units of horizontal motion of piece **D** in group 1 (which is the piece **A** of group 2) which in turn force k^3 unit horizontal motion of piece **A** of group 3. When the motion propagates to group n, piece **A, B** of group n has to move a total of k^n units horizontally. If we choose k as a small constant, then the total number of edges in the construction is linear in n. We have the following theorem.

Theorem 3 *The compaction problem shown in Figure 7 requires an exponential number of moves.*

We see that piece **B** of group n is the busiest piece. We divide the motion into basic steps according the motion of the piece. The motion between the moment that piece **B** of group n move from the highest vertical position to the lowest one (or vise versa) is called a *basic step.* At each basic step, only piece **B** in group n can move a full vertical and horizontal unit. The distance that can be moved by a piece in another group declines exponentially with group number. For example, the piece **B** in group 1 can only move $O(k^{-n})$ horizontal or vertical units in each basic step. At each basic step, a description of the state of the system still takes only polynomial space.

The motivation for the construction in Figure 7 is to show that the algorithm presented in [3] for finding a local minimum of the compaction problem needs an exponential number of iterations in the worst case. The algorithm uses a configuration space approach for finding a local optimum. For each neighboring pair of objects (represented as polygons), the Minkowski sum of the two objects is generated. A large convex subset of

[1] Discs are restricted to move to neighboring posts. This is still solvable with $\Theta(2^n)$ moves

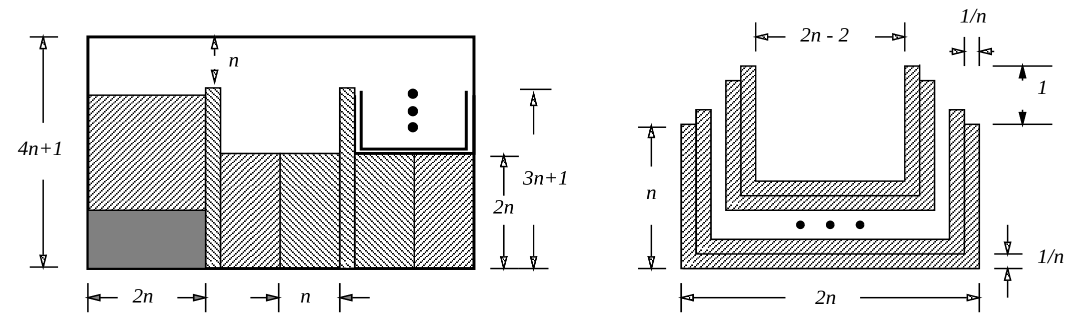
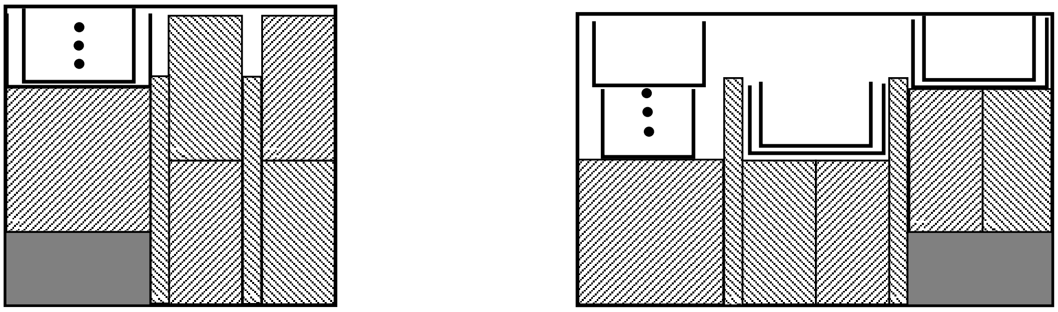


Figure 4: Construction and the initial state of a block puzzle



(a)

(b)

Figure 5: (a) The final state. (b) A dead-end situation

the exterior of the Minkowski sum is determined according to a locality heuristic. A set of linear inequalities are built using the boundary edges of the convex subset. The relative motion of the two pairs is constrained to stay inside the convex subset by the set of linear inequalities. A linear program is set up using the set of linear inequalities for every neighboring pair as constraints. The objective function maximizes the motion of an imaginary "piston" piece at the right boundary. The solution of linear program will give the new positions for all the objects. A new set of inequalities is set up if the convex subsets outside the Minkowski sum of a pair of objects changes. The algorithm iterates until there is no improvement to the objective function.

If we applied the algorithm to the compaction problem shown in Figure 7, it is easy to show that each iteration of the algorithm finds exactly the configuration after each basic step. The objective function can be improved until piece A of group 1 is moved to the left by k units. Because there are total $\Theta(k^n)$ basic steps, and each iteration of the algorithm corresponds to exactly one basic step, there will be a total of $\Theta(k^n)$ calls to the linear program.

To show that the algorithm can only find a local minimum, we will make two dents in piece **A** of group 1. The two dents have the same height h but different width d_1 and d_2. Assume that the lower dent has width d_1 and $d_1 < d_2$. We will make a rectangular object with dimension $d_2 \times h$. The rectangular object is originally placed at the lower dent. According to the heuristics used in the algorithm, the algorithm will never move the rectangular object from the lower dent to the higher dent. Hence, the algorithm does not find a global optimum when the algorithm terminates.

Theorem 4 *The compaction algorithm described in [3] requires an exponential number of iterations to find a local minimum.*

We would like to point out that the example shown in Figure 7 demonstrates the worst case behavior of the algorithm presented in [3]. In practice, the algorithm runs very fast. For the application of compacting garment layouts, the algorithm has always terminated in five iterations.

Acknowledgement: We would like to thank Karen Daniels for her careful reading and criticism on the early drafts of this paper.

References

[1] B. Chazelle, T. Ottmann, E. Soisalon-Soininen, and D. Wood. The complexity and decidability of SEPARATION$^{\mathrm{TM}}$. In *Proc. 11th Internat. Colloq. Automata Lang. Program.*, volume 172 of *Lecture Notes in Computer Science*, pages 119–127. Springer-Verlag, 1984.

[2] J. E. Hopcroft, J. T. Schwartz, and M. Sharir. On the complexity of motion planning for multiple independent objects: $PSPACE$-hardness of the "Ware-

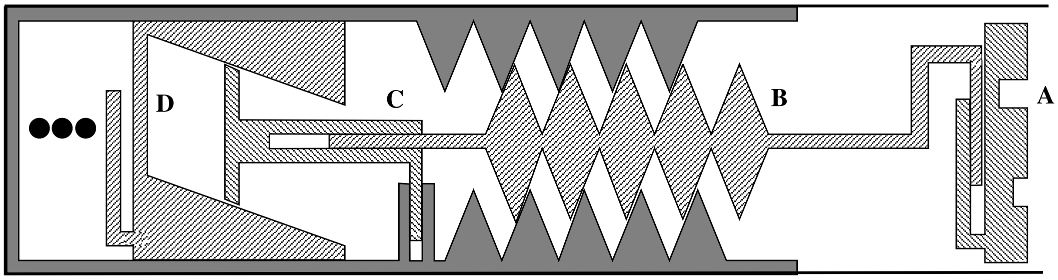


Figure 6:

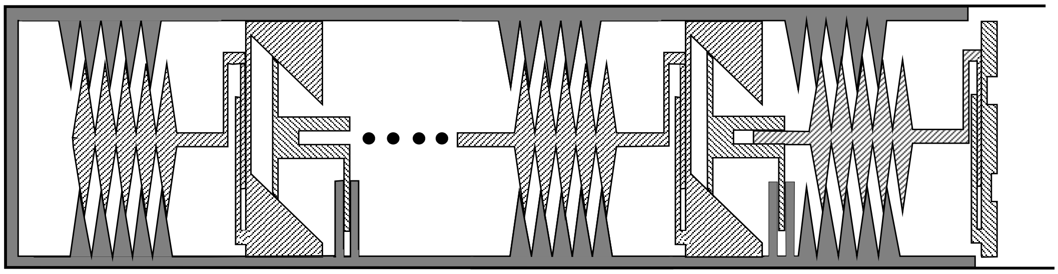


Figure 7:

houseman's Problem". *Internat. J. Robot. Res.*, 3(4):76–88, 1984.

[3] Z. Li and V. Milenkovic. A compaction algorithm for non-convex polygons and its application. In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 153–162, 1993.

[4] P. Spirakis and C. Yap. Strong NP-hardness of moving many discs. *Inform. Process. Lett.*, 19:55–59, 1984.