# The Box Mover Problem

Valentin Polishchuk*


## Abstract

We show that the *optimization* problem is NP-hard for a wide class of motion planning puzzles, including classical SOKOBAN. We investigate a new problem, the Box Mover Problem (BMP), in which the agent is allowed to lift and carry boxes on a rectilinear grid in order to rearrange them. Some classical motion planning puzzles are special cases of BMP. We also identify a natural class of BMP instances, for which optimization is in NP, making the optimization problems from the class NP-complete.


## 1 Introduction

There is a number of motion planning puzzles in which, given an arrangement of unit blocks (boxes) in the plane, one has to rearrange the boxes into another configuration by operating a robot which moves in the same plane amid the boxes. The classical example is SOKOBAN. ([8] provides a thorough description of the puzzles and corresponding algorithmic results.) The puzzles we consider here may be classified according to the following characteristics (the classification is adapted from [5]):

1. How powerful the robot is:
   - Can the robot push the boxes? How many at a time?
   - Can the robot pull the boxes? How many?
   - We introduce a "new dimension" for the robot: can the robot lift a box and put it to an adjacent position (including the position, occupied by the robot before the lift)? The new problem is dubbed the *Box Mover Problem* (BMP).
2. Box types
   - Are all boxes movable, or are some fixed to the plane? In other words, are we working on the infinite plane, with nothing else but the boxes on it, or are we constrained to a floor bound by rigid walls?
3. Robot path
   - Is the solution path required to have no self-intersections?
   - Are we looking for a closed path for the robot?
4. Boxes' IDs ("15"-style)
   - Are the boxes and the target positions labeled? This may be important with respect to the final configuration of the boxes; in SOKOBAN any box can occupy any target position.

Following notation in [5, 6, 9], we call BMP$(k, p, l)$ the Box Mover Problem for the robot capable of pushing $k$, pulling $p$ and lifting $l$ boxes at one time. If some boxes may be fixed to the plane, the problem is called BMP$(k, p, l)$-F. If only non-self-intersecting paths are allowed for the robot, the problem is called BMP$(k, p, l)$-X. We do not require the robot to return to its initial position; it can stop right after all the boxes are in their target positions. Finally, if the boxes and the target positions bear labels, the problem is called #BMP$(k, p, l)$. Thus, e.g., BMP$(1, 0, 0)$-F is the original SOKOBAN game, BMP$(\infty, \infty, \infty)$ is the Omnipotent Robot Problem (they also have problems), BMP$(k, 0, 0)$, BMP$(\infty, 0, 0)$ and BMP$(1, 0, 0)$-X are the Push-$k$, Push-* and Push-X versions of Push (see [8]).

To clarify rules for lifting, we emphasize that the robot can essentially "go under" a box: it can approach the box, swap positions with it and then put the box back. Such an operation requires 2 lifts. The robot can also carry a box to another location. We think of such carrying as a sequence of lifts; the number of necessary lifts equals the distance traveled by the robot with the box.

It is possible to come up with other rules for lifting. With some adjustment, our results remain valid for other rules as well.

### 1.1 Comparison with Previous Work

1. To our knowledge, previous research concentrated on investigating hardness of the *feasibility* problems, while in "reality" one would rather be interested in minimizing the amount of work to be done (i.e. in the *optimization*) when it is ensured that the problem is feasible. We define the cost of a solution to be the number of "loaded" moves (pushes, pulls, lifts); the unloaded motion of the robot is free.

   The only results on optimization of SOKOBAN can be found in [14]. We have taken the basic edge gadget from it. These results were never published and used third dimension to work or considered a slightly modified SOKOBAN problem [3].
2. Several attempts have been made to make the puzzles "more tractable" by limiting the robot's capabilities

*Department of Applied Math. & Statistics, Stony Brook University, valentin.polishchuk@stonybrook.edu

[4, 5, 7], i.e. by considering BMP$(k, p, l)$ with $l = 0$ and small values of $k, p$ satisfying $kp = 0$. The exact complexity of some of the problems is still unknown; others have been shown PSPACE-complete [1, 2, 11]. The only known "easier" (NP-complete) problem is a somewhat artificial Push-X version (see above), which restricts the robot's paths, rather than its power.

Our proof of NP-hardness of the optimization problem holds for an arbitrarily powerful robot. We also describe a natural class of *open* BMP instances, for which the optimization problem is in NP.

3. Leakage is a major problem in proving hardness of puzzles with all blocks movable. To constrain the robot's motion certain configurations have to be used: e.g., if the robot can push up to $k$ boxes, a $(k+1) \times (k+1)$ square of boxes can be considered fixed to the floor, a wall of thickness more than $k$ can be considered rigid [4, 7], etc. Same configurations work for constraining of a pull-only robot - once disassembled, these configurations can never be put together. Yet, if the robot can both pull and push (or lift), then no obvious construction (if any at all!) is "heavy" enough to serve as an obstacle for the robot.

   In our proof the wall thickness is constant and the proof holds for an arbitrarily powerful robot.

4. In [4] and [7] the authors contrasted their work to "all previous approaches of building circuits based on graphs, which seem to inherently require [problematic] crossings." In fact, one of the first proofs of NP-hardness of SOKOBAN [9] was based on *Planar* 3-SAT problem and did not use any crossovers. Our construction does not require crossings either, since it is by reduction from HC for *planar* graphs.

## 2 The Reduction

The reduction is from the Hamiltonian cycle (HC) problem for planar directed graphs with each node $v$ satisfying $outdegree(v) + indegree(v) = 3$, which is NP-complete by [13]. Let $G = (N, A)$ be such a graph with $|N| = n$. We construct a BMP(1,0,0)-F instance from $G$ such that $G$ contains a HC iff the BMP instance is solvable in $3n - 2$ pushes.

First, embed $G$ in the plane in such a way that the edges of $G$ are drawn with vertical and horizontal segments (Figure 1, left and center). Such an embedding is possible and can be constructed from $G$ in polynomial time [12]. We then use the embedding as a "floor map" for constructing a BMP(1,0,0)-F instance. Each edge of $G$ becomes a corridor of width 1 and every node of $G$ becomes a "T-intersection" of 2 corridors (Figure 1, center and right). Next, we place a *node gadget* (Figure 2, left) in each node of $G$ and an *edge gadget* (Figure 2, right) in the middle of every edge to emulate the direction of the edge. (We may need to lengthen the corridors to

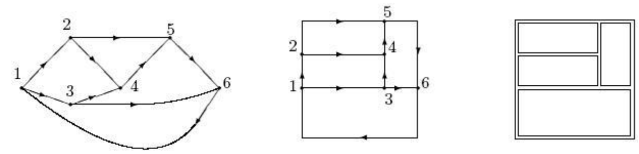

Figure 1: Planar embedding and floor map.

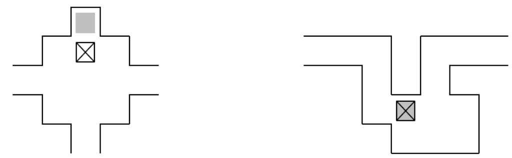

Figure 2: Node (left) and edge (right) gadgets. Boxes (in their initial positions) are marked with ⊠, boxes' target positions are marked with light grey

have enough space for inserting the edge gadgets.) Note that in the edge gadget, the initial and the target positions of the box coincide. It is easy to see that the edge gadget is passable in one direction only (with 2 pushes per each pass). The robot is initially placed inside a corridor. The robot's goal will be to put the boxes in the target positions.

If $G$ has an HC, then the constructed BMP(1,0,0)-F instance can be solved in $3n - 2$ pushes. Indeed, the robot will follow the HC in $G$, traveling along $n - 1$ edges (spending 2 pushes per edge) and pushing all the boxes in the node gadgets in the corresponding target positions (1 push per node). If $G$ is not Hamiltonian, then in order to visit all the nodes, the robot needs to travel twice along at least one edge of $G$, so the total number of pushes will be not less than $n \cdot 2 + n = 3n$. Thus, the constructed BMP(1,0,0)-F instance can be solved with $3n - 2$ pushes iff $G$ is Hamiltonian.

## 3 Results

From the preceding discussion follows

**Lemma 1** BMP$(1, 0, 0)$-F *is NP-hard.*

We shall now strengthen this result in several aspects. First, observe that if the robot is only allowed to *pull* 1 box (BMP$(0, 1, 0)$-F), the same edge gadget can be used to model the direction on an edge – the only difference is that the direction of the edge is now reversed. If, in addition, the initial and target positions of the boxes in node gadgets are swapped, the same reduction works for BMP$(0, 1, 0)$-F. Hence,

**Corollary 2** BMP$(0, 1, 0)$-F *is NP-hard.*

When the robot is allowed to *lift* a box, the directionality of the corridors (edges) is lost (the robot can travel in both directions). In fact, the edge gadget may now be simplified to just be a corridor with a box in target position. Still, the cost of traveling through an edge is 2. The BMP instance now models a planar undirected cubic graph. The HC problem for planar undirected 3-connected cubic graphs having no face with fewer than 5 edges is NP-complete [10]. Thus, our reduction is valid for BMP$(0,0,1)$-F as well:

**Corollary 3** BMP$(0,0,1)$-F *is NP-hard.*

Secondly, observe that if $G$ is Hamiltonian, the path of the robot in the proposed solution is non-self-intersecting. Thus,

**Corollary 4** BMP$(1,0,0)$-F-X, BMP$(0,1,0)$-F-X *and* BMP$(0,0,1)$-F-X *are NP-hard.*

Next, observe that we could have assigned numbers to the boxes and target locations. The boxes and the target locations in node gadgets could have been labeled 1 through $n$ and the boxes in the edge gadgets (they are already in target positions) – $n+1$ to $2n$. The reduction above would not change and thus

**Corollary 5** #BMP$(1,0,0)$-F, #BMP$(0,1,0)$-F *and* #BMP$(0,0,1)$-F *are NP-hard.*

Giving the robot the power to push, pull or lift an arbitrary number of boxes would not change the reduction (essentially, the edge gadget is just an "energy waster"). So,

**Corollary 6** BMP$(k,p,l)$-F *is NP-hard for any* $(k,p,l) \neq (0,0,0)$.

Since all of the above observations work independently of each other,

**Corollary 7** [#]BMP$(k,p,l)$-F[-X] *is NP-hard for any* $(k,p,l) \neq (0,0,0)$.

Finally, we can replace the rigid walls of the corridors by walls of boxes of thickness 2 and change the gadgets as shown in Figure 3. The reduction will still be in place. Indeed, even if the robot has enough power to break through a wall, it would not benefit from doing so, since it would still need to spend too much of a workload before getting to a node gadget. Thus, we have the main result:

**Theorem 8** *All variations of BMP are NP-hard, i.e.* [#]BMP$(k,p,l)$[-F][-X] *is NP-hard for any* $(k,p,l) \neq (0,0,0)$*, including infinite values of* $k,p,l$.

As mentioned above, if $G$ is Hamiltonian, the corresponding BMP(1,0,0)-F instance can be solved in $3n-2$ pushes, while if $G$ is non-Hamiltonian, the number of pushes, needed to solve the instance, is at least $3n$. This shows that (unless P=NP) there exist no Fully Polynomial Time Approximation Scheme (FPTAS) for the problem. Indeed, suppose, that there exists an algorithm, which, for any $\epsilon > 0$, finds a solution, requiring at most $1+\epsilon$ times the optimum pushes; and that such an algorithm runs in time, polynomial in $1/\epsilon$. Take $\epsilon < \frac{2}{3n-2}$. Then, the algorithm would output a solution of cost less than $3n$ iff $G$ is Hamiltonian. Since this argument works for all versions of the problem, we have

**Corollary 9** *Unless P=NP, there exists no FPTAS for* [#]BMP$(k,p,l)$[-F][-X].

Figure 3: Node (left) and edge (right) gadgets with all blocks movable.

## 4 A Realistic Assumption

To prove hardness of BMP, we have constructed some gadgets constraining the agent's motion. Moreover, to avoid leakage, the whole instance was "closed" – once inside the warehouse, the agent is constrained to stay there forever, never to be able to come out and report a solution to an NP-hard problem! Considering such a situation inhumane and unrealistic, we define the *Open Box Mover Problem* (OBMP) as BMP restricted to the instances in which the agent can escape to infinity from the initial position. OBMP retains all the notation introduced in BMP: # , $(k,p,l)$, -F, -X.

**Lemma 10** *OBMP is NP-hard for all formulations for which BMP is NP-hard.*

**Proof.** In the constructions used for proving hardness of BMP we could initially put the agent in the edge, adjacent to the unbounded face of the graph, and make a hole in the wall close to the agent's initial position. Having the ability to escape to infinity does not change the cost of a feasible solution (since we only count the workload, not the total travel of the agent). Thus, the reduction, which worked for a BMP formulation, also works for the corresponding version of OBMP. □

Although openness has no impact on optimality, it has drastic effect on feasibility: every instance of OBMP$(k,p,l)$ with $l > 0$ is feasible. Indeed, if the agent can lift and carry the boxes ($l > 0$), he can go to a far point ("infinity"), return to a box, carry it to infinity, return to another box, carry it to infinity and so on. Now, that he has all the boxes at infinity, he can start bringing the boxes back one by one to their target positions. If there are $N$ pixels in the floor map, there are no more than $N$ boxes in the instance. So, the point at the distance of $2N$ from the exit from the warehouse is far enough to be the "infinity" point, to which the agent can carry all the boxes one by one. Thus, any instance of OBMP$(k,p,l)$ with $l > 0$ is feasible. Moreover, it is solvable in at most $O(N^2)$ moves and therefore is in NP.

**Theorem 11** [#]OBMP$(k,p,l)$[-F][-X] *with* $l > 0$ *is NP-complete.*

## 5 Open Questions

1. A still-open question, proposed in [4] and [7], is to find an "interesting" tractable problem. Despite significant efforts, no such problem has been found in the "feasibility" direction. Maybe, investigating the optimization would bring a well-solved special case, in which feasibility is trivial, but optimization is still interesting to consider.

2. Our construction seems inapplicable for obtaining a hardness result for PushPush optimization problem (a variation in which a box, once pushed, slides to the maximal extent, until it collides with another box or a wall (see [8])). Possibly a modification of the construction would lead to establishing hardness for PushPush optimization problem or, maybe the optimization problem for PushPush (or its variation) is in P. The feasibility problem for PushPush is NP-hard due to [5], but its exact complexity is open.

3. It is essential for our reductions that the initial and target positions coincide for certain boxes. What if we restrict BMP to the instances where this is not true? What if the initial and target configurations, thought of as rigid bodies, may be pulled apart by a sequence of translations? The polynomial case of line-separability [15] is a special case of this general one.

## Acknowledgements

I would like to thank Joseph Mitchell for presenting the problem at the Stony Brook Algorithms Reading Group [15], where many inspiring discussions took place. I would also like to thank Joseph Culberson for information on [14]. In this final version of the paper, I have taken into account several very useful comments, provided by anonymous referees.

## References

[1] D. Bremner, J. O'Rourke, and T. Shermer. Motion planning amidst movable square blocks is PSPACE complete. Draft, June, 1994.

[2] J. Culberson. Sokoban is PSPACE-complete. In E. Lodi, L. Pagli and N. Santoro, editors, *Proc. Int. Conf. Fun with Algorithms* (1998), pages 65–76.

[3] J. Culberson. Private communication, 2003.

[4] E. Demaine, M. Demaine, M. Hoffmann, and J. O'Rourke. Pushing blocks is hard. *Comput. Geom. Theory Appl.*, 26(1):21–36, 2003. Special issue of selected papers from the 13th Canad. Conf. Comput. Geom., 2001.

[5] E. Demaine, M. Demaine, and J. O'Rourke. PushPush and Push-1 are NP-hard in 2D. In *Proc. 12th Canad. Conf. Comput. Geom.* (2000), pages 211–219.

[6] E. Demaine, R. Hearn, and M. Hoffmann. Push-2-F is PSPACE-complete. In *Proc. 14th Canad. Conf. Comput. Geom.* (2002), pages 31–35.

[7] E. Demaine and M. Hoffmann. Pushing blocks is NP-complete for noncrossing solution paths. In *Proc. 13th Canad. Conf. Comput. Geom.* (2001), pages 65–68.

[8] E. Demanie. http://theory.lcs.mit.edu/∼edemaine/ pushing-blocks/.

[9] D. Dor and U. Zwick. Sokoban and other motion planning problems. *Comput. Geom. Theory Appl.*, 13(4):215–228, 1999.

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

[11] R. Hearn and E. Demaine. The nondeterministic constraint logic model of computation: Reductions and applications. In *Proc. 29th Int. Colloq, on Automata, Languages and Programming* (2002), volume 2380 of *Lect. Notes Comp. Sci.*, pages 401–413.

[12] D. S. Johnson and C. H. Papadimitriou. Computational complexity and the traveling salesman problem. In E. Lawler, J. Lenstra, A. R. Kan, and D. Shmoys, editors, *The Traveling Salesman Problem*, pages 68–74. John Wiley & Sons, New York, 1985.

[13] J. Plesnik. The NP-completness of the Hamilton cycle problem for planar digraphs of degree bound two. *Inform. Process. Lett.*, 8(4):199–201, 1979.

[14] S. Sabey. On the complexity of SOKOBAN. Unpiblished, 1996.

[15] The Stony Brook Algorithms Reading Group, 2002.