

Transporting Goods using Cellular Automata

Tycho Bismeijer¹ and Stefan Boronea¹

Vrije Universiteit Amsterdam

Abstract.

1 Introduction

Our goal within this paper is to show the applicability of the cellular automata model in transporting goods within a two dimensional environment. In this sense we have designed a cellular automata that can perform this task. The physical realism of the model was not be considered for the current model, since our goal is to show that a self-organizing system can achieve such behavior. In a nutshell we would like to describe how a cellular automata model can be used to transport objects over gaps within a world in which gravity affects the movements of objects. Our study considers variations of variables such as gap size, object weight and number of cellular automata bots initially placed within the system, which have a direct effect on how fast the object is transported to the final location.

2 Literature

[1]

3 Model

2D cellular automata with gravity. Bots, obstacles and boxes. Pushing.

3.1 The world

Falling

Pushing

3.2 Cellular Automata

The bots move according to cellular automata rules.

4 Implementation

The model and application we are studying are new to the literature and therefore our implementation of the model has also been built from scratch. For our simulations we have build a NetLogo application that allows the user to simulate the outcome of the previously described model. The interface also allows the creation of custom worlds and the simulation (in single run or batch mode) of different scenarios in which all or part of the four primary variables are modulated. A typical world view is presented in fig. 2.

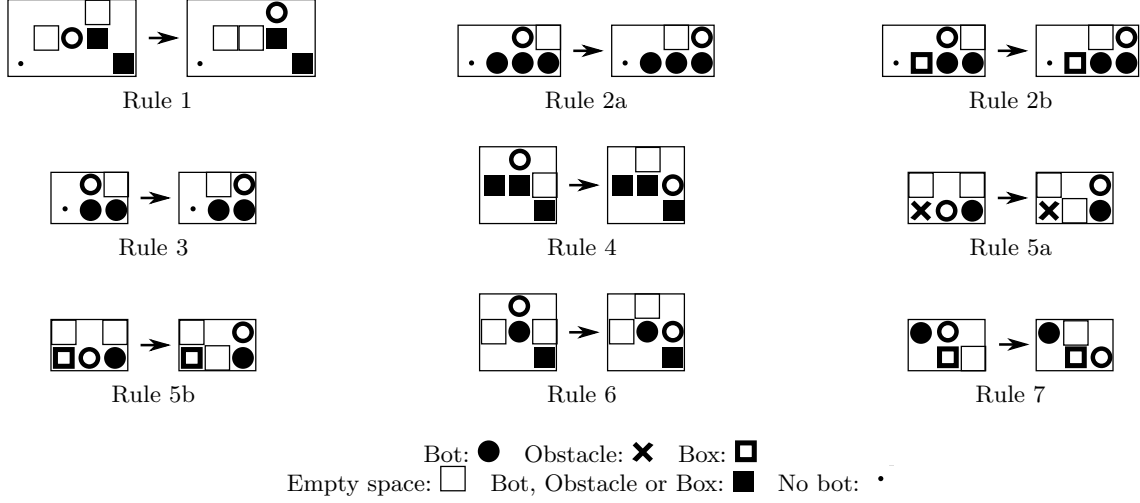


Fig. 1: Cellular automata rules for moving bots.

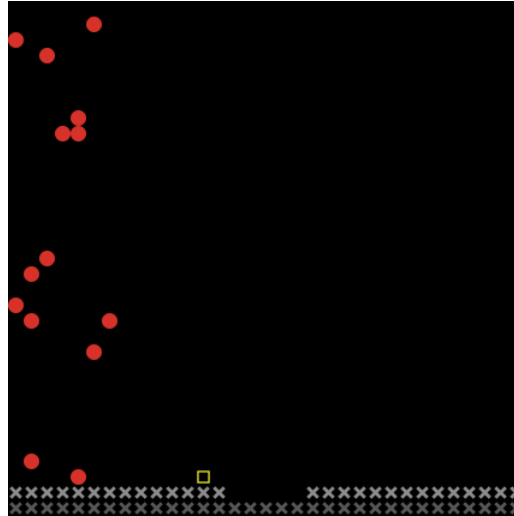


Fig. 2: An initial setting of the world in which the simulation takes place. The red dots denote randomly placed bots, the X marked places show obstacles and the yellow box is the object that needs to be transported to the other end of the world.

Each of the entities involved (i.e. bots, objects and obstacles) are defined as agents within NetLogo. The objects have individual weights, the bots are considered to all have a pushing power of one.

At each time step, the operations in listing 1.1 are performed by the system. First, a bot is activated randomly and is allowed to perform a movement action based on the implemented rules described previously in section 3. Then all the bots are allowed to make a pushing action, where only the bots that have a box in the immediate vicinity are activated. These proceed to the pushing phase described in listing 1.2. Here the bot finds what the pushing power required for the next obstacle(s) and depending on how many bots can help push the object, it may decide to do so or wait for a new time to act (when perhaps it will be supported by the required number of bots). During the entire process both the bots and the boxes are affected by

gravity. This activity is performed at each time step and moves the bot to the lowest possible location to the ground that has the same x coordinate within the world.

Listing 1.1: The operations performed at each time step.

```
to make-single-run
  ; move bots
  ask one-of bots [
    bot-move
  ]

  ; apply gravity and push boxes
  ask bots [
    fall

    ; check if there are boxes to be moved
    push-box

    ; remove boxes at the storage area
    store-boxes
  ]

  ; apply gravity on boxes
  ask boxes [
    fall
  ]

  ;move to next time step
  tick
end
```

Listing 1.2: The actions performed when pushing a box.

```
to push-box
  ; check if there are any boxes to push
  if (any? boxes-at 1 0) ; a box next to the bot
  [
    if(not any? boxes-at 1 1) ;no boxes on top
    [
      let effective-weight [ compute-effective-weight ] of one-of boxes-at 1 0
      let pushing-power compute-pushing-power

      if effective-weight <= pushing-power
      [
        ask one-of boxes-at 1 0 [
          move-forward
        ]
      ]
    ]
  ]
end
```

5 Experiments

The experiments we will be presenting in the following sections take into account each of the hypotheses we previously mentioned. For this we have chosen four fundamental variables to study: the gap size, the number of bots used, the weight of a single box and the number of reconfigurations required to solve a specific setting. Out of these, the first three were considered to be independent variables, while the latter was the dependent variable.

We are primarily focusing on determining the following:

1. whether the number of extra robots needed to push over a gap is independent of the gap size;
2. whether the number of reconfigurations is linearly dependent on the gap size;
3. is there a linear correlation between the weight of the object that is transported and the minimum number of bots?

The following experiments were performed in the same world, with the characteristics presented in table 1. We can see that here the box is placed each time before the gap and the bots are placed behind it at random positions (on both the x and y axis).

Table 1: The characteristics of the simulation world.

Variable	Type	Interval
height	other	32
length	other	32

5.1 Experiment 1

5.2 Experiment 2

5.3 Experiment 3

In light of the implementation details provided in section 4, this experiment seeks to determine whether there is a linear correlation between the weight of the box and the minimum number of bots required for the task. The settings for this experiment are shown in table 2.

Table 2: The variables involved in experiment 3.

Variable	Type	Interval
gap size	independent	5
box weight	independent	[1, 50]
number of bots	independent	[4, 50] with step 5
minimum no. of bots	dependent	
runs per parameter set	other	10

Fig. 3 shows that the number of required bots to move an object of variable weight. Since the gap size is not actually important for this current experiment, it was set to 5 for all the runs concerning it. We can easily see that this minimum number of bots required increased with the weight of the object linearly,

independent of the gap size (the averaged results show the same linear dependency, as shown in fig. 4). This is an expected outcome that validates our initial hypothesis since a heavier object would require a higher pushing power that can only be achieved in our model using multiple bots aligned to create the required force. A different approach to the problem would be to modify a bot's pushing power, thus leading to a more rapid resolve using less bots. However, we will not consider this possibility during our paper since we have chosen to overlook the physical realism of the individual bots (such is the case in [2]).

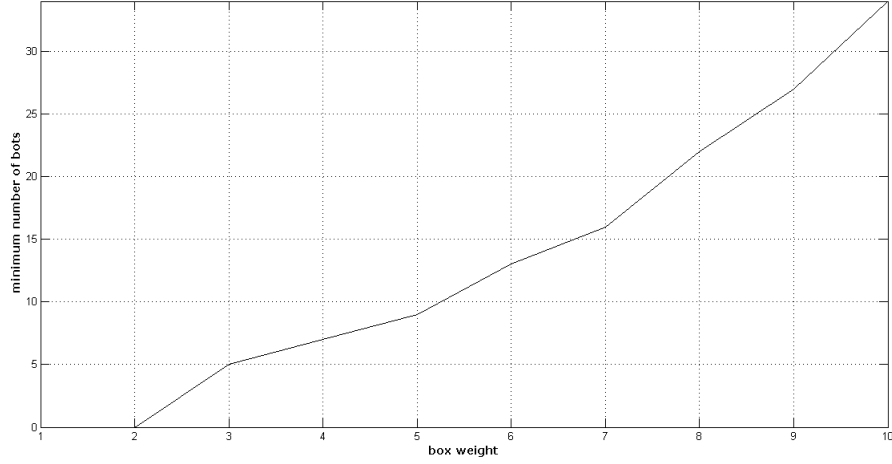


Fig. 3: Minimum number of bots required to push a box of a certain weight.

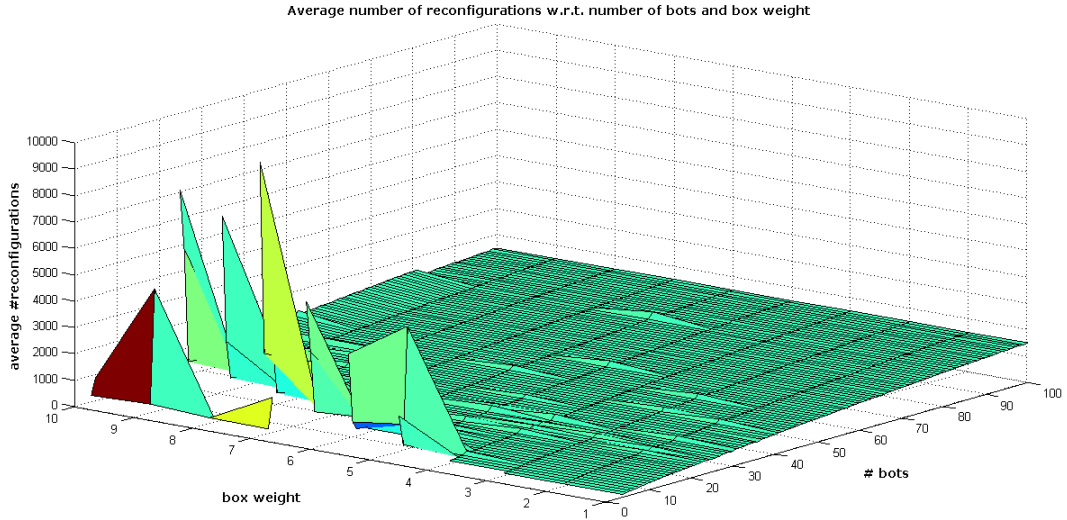


Fig. 4: The variation of the number of bots required to solve a specific configuration consisting of a single box. The results are averaged over the different gap sizes within the $[0, 10]$ interval.

6 Conclusions

The previous sections have shown how a cellular automata model could be used in the task of transporting a certain object within a gravity-affected world.

The current study focuses primarily on solving the somewhat simple task of transporting a single box over a single gap of height 1. However, the rules used in our model could be used in far more complex situations within various world configurations. For example, the current model can be used to transport a convoy of boxes aligned one after another, where multiple bots combine their pushing power to move the heavier load. Another interesting extension to the model would be to determine its use in placing certain objects at specific locations within the world so that a certain sequence of tasks is met.

References

1. Butler, Z., Kotay, K., Rus, D., Tomita, K.: Generic decentralized control for a class of self-reconfigurable robots. In: Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on. vol. 1, pp. 809 – 816 vol.1 (2002)
2. Yun, S.K., Rus, D.: Optimal self assembly of modular manipulators with active and passive modules. *Auton. Robots* 31, 183–207 (October 2011), <http://dx.doi.org/10.1007/s10514-011-9236-1>