

Санкт-Петербургский государственный университет

*Мишуров Илья Дмитриевич*  
Выпускная квалификационная работа

*Песочные модели на ориентированных  
графах*

Уровень образования: магистратура  
Направление 01.04.01 “Математика”  
Основная образовательная программа ВМ.5832.2022 ”Современная  
математика”

Научный руководитель:  
профессор,  
Факультет математики  
и компьютерных наук СПбГУ,  
д.ф.-м.н.,  
Петров Фёдор Владимирович

Рецензент:  
Доцент,  
Гуандун-Технион Израильский  
институт технологии (Китай),  
PhD,  
Калинин Никита Сергеевич

Санкт-Петербург  
2024

# Оглавление

<b>1. Введение и основные определения</b>	<b>3</b>
<b>2. Песочная динамика на ориентированной решетке</b>	<b>6</b>
2.1. Псевдо-Манхеттенская решетка . . . . .	6
2.2. Песочная динамика . . . . .	7
2.3. Вычисление размера лавины . . . . .	7
<b>3. Единица песочной группы</b>	<b>9</b>
3.1. Псевдо-Манхеттенская решетка и граничные условия . .	9
3.2. Единица Крейца . . . . .	9
3.3. Единица через максимально стабильное состояние . . . .	11
<b>4. Выводы</b>	<b>19</b>
<b>Список литературы</b>	<b>20</b>
<b>А. Приложение</b>	<b>21</b>

# 1. Введение и основные определения

В данной работе исследуются песочные модели на ориентированных целочисленных решетках. В некотором смысле это является упрощением стандартной неориентированной модели, однако и в этом случае песочная модель обладает многими такими же свойствами. Более того, состояние единицы группы для различных решеток и граничных условий так же обладает регулярной структурой. Целью данной работы было изучение некоторых свойств песочной модели на ориентированной целочисленной решетке. В рамках работы было показано что состояние  $n\beta$ , где  $\beta$  - единица Крейца, не будет являться единицей группы для таких моделей, а так же изучены свойства другого состояния, через которое вычисляется единица. В этой главе введем основные определения и обозначения для дальнейшей работы.

Пусть  $G = (V, E)$  ориентированный граф без петель и кратных ребер. Выделим одну вершину  $s$ , и назовем её стоком. Сделаем это так чтобы от любой вершины графа до неё был ориентированный путь. Песочная модель  $\Gamma$  определяется этими двумя параметрами.

**Определение.**  $C : V \setminus s \rightarrow \mathbb{Z}$  называется состоянием песочной модели  $\Gamma$

**Определение.** Состояние  $C$  называется стабильным, если  $\forall v \in V, C(v) < outdeg(v)$

Для не стабильных вершин определим оператор обвала  $T(v)$ . Если вершина не стабильна, то он вычитает из ее состояния  $outdeg(v)$  песчинок, а всем вершинам, к которым есть ориентированное ребро из  $v$  добавляет по одной. Отметим, что оператор обвала не обваливает вершину сток, но в нее может приходиться песок из других вершин.

Так же, для не стабильных состояний определим оператор релаксации. Не трудно заметить, что если состояние не стабильно, то за конечное число обвалов оно станет стабильным, а так же что итоговое состояние не зависит от порядка обвалов. Таким образом, оператор

релаксации делает обвалы пока это возможно и однозначно получает стабильное состояние  $C^\circ$ .

Определим сложение двух состояний как  $C \oplus C' = (C + C')^\circ$ , где  $+$  это поточечное сложение состояний.

**Определение.** Для натурального  $k$  и состояния  $C$  определим

$$kC = \underbrace{C \oplus \dots \oplus C}_k$$

**Определение.** Состояние  $C$  называется возвратным, если

- $\forall v \in V, C(v) \geq 0$
- $C$  стабильно
- $\forall a \exists b \geq 0 : a \oplus b = C$

**Утверждение.** Количество возвратных состояний равно числу ориентированных в сток остовных деревьев в  $G$

Данное утверждение доказано в [3], аналогичное утверждение верно и в неориентированном случае.

Возвратные состояния образуют группу относительно операции  $\oplus$ . Более подробное построение группы, а так же доказательство этого можно найти в [3]. Таким образом, можно вычислять порядок песочной группы зная число ориентированных деревьев на данном графе.

Единицей песочной группы называется такое единственное возвратное состояние  $Id$ , что для любого возвратного состояния  $C$  верно  $C \oplus Id = C$ .

**Определение.** Определим максимально стабильное состояние  $C_{max}(v) = outdeg(v) - 1$

Не сложно заметить что это возвратное состояние. А так же что любое возвратное состояние можно получить из максимально стабильного, то есть верно следующее утверждение.

**Утверждение.** *Состояние  $C$  возвратное тогда и только тогда, когда  $\exists b : C_{max} \oplus b = C$ .*

Данное утверждение, а так же другие свойства состояний песочной модели на ориентированном графе доказаны в работе [3]

## 2. Песочная динамика на ориентированной решетке

### 2.1. Псевдо-Манхеттенская решетка

Рассмотрим целочисленную решетку. Ориентируем ребра так, чтобы по горизонтали и по вертикали направления ребер чередовались. Ориентируем ее так, чтобы левый нижний квадрат был ориентирован против часовой стрелки, это понадобится для единства рассматриваемых решеток. Пример такой решетки представлен на (Рис. 1):

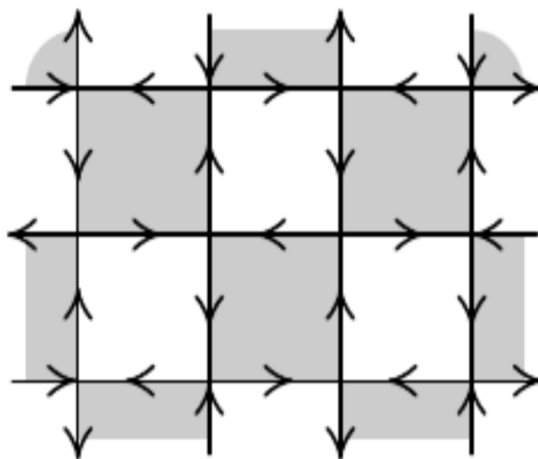


Рис. 1: Псевдо-Манхеттенская решетка

Рассмотрим такую решетку размера  $50 \times 50$ . Стоком будет всё что находится за решеткой в области  $\mathbb{Z}^2$

Заметим, что такой граф является сильно связным, действительно между любыми двумя вершинами есть ориентированный путь. Так же, можно задать такие граничные условия чтобы этот граф был Эйлеров. А именно, пустить с граничных вершин в сток ребро или ребро из стока так чтобы уравновесить входящую и исходящую степени.

## 2.2. Песочная динамика

Пусть  $y = ax^b$ , такая зависимость называется степенным законом. Песочные модели на неориентированной целочисленной решетке могут описывать такой закон путем случайного процесса. Более точно, рассмотрим модель на решетке размера  $50 \times 50$ . Пусть  $C_0$  - некоторое произвольное состояние. Возьмем случайную вершину  $p$  равновероятно из всего графа песочной модели, и добавим в нее одну песчинку, затем проведем релаксацию полученного состояния. Т.е. получается последовательность  $C_0 \rightarrow C_1 = (C_0 + \delta p_1) \rightarrow C_2 = (C_1 + \delta p_2) \dots$

Обозначим за  $h_k$  размер подножия лавины, вызванной добавлением  $k$ -ой песчинки, то есть число вершин, где случился хотя бы один обвал в процессе релаксации состояния  $C_k + \delta p_k$ . Для неориентированной решетки доказано, что последовательность  $h_k$  удовлетворяет степенному закону [1]. Объем лавины (то есть общее число обвалов во время лавины) тоже удовлетворяет степенному закону. Так же, в работе [4] было показано что похожие законы выполняются для Манхеттенской решетки (которая отличается ориентацией ребер) для других моделей самоорганизованной критичности.

Оказывается, если провести такое же на псевдо-Манхеттенской решетке, то объем лавины тоже будет удовлетворять степенному закону.

## 2.3. Вычисление размера лавины

Для вычисления лавины были проведены вычисления, с равной вероятностью добавлялась песчинка для графа размера  $50 \times 50$  и проводилась релаксация, в процессе релаксации подсчитывался объем лавины. Далее были построены графики на дважды логарифмической шкале. Численные эксперименты подтверждают, что песочные модели на псевдо-манхеттенской решетке обладают таким же свойством как и в неориентированном случае. Исходный код для вычисления предоставлен в приложении А.

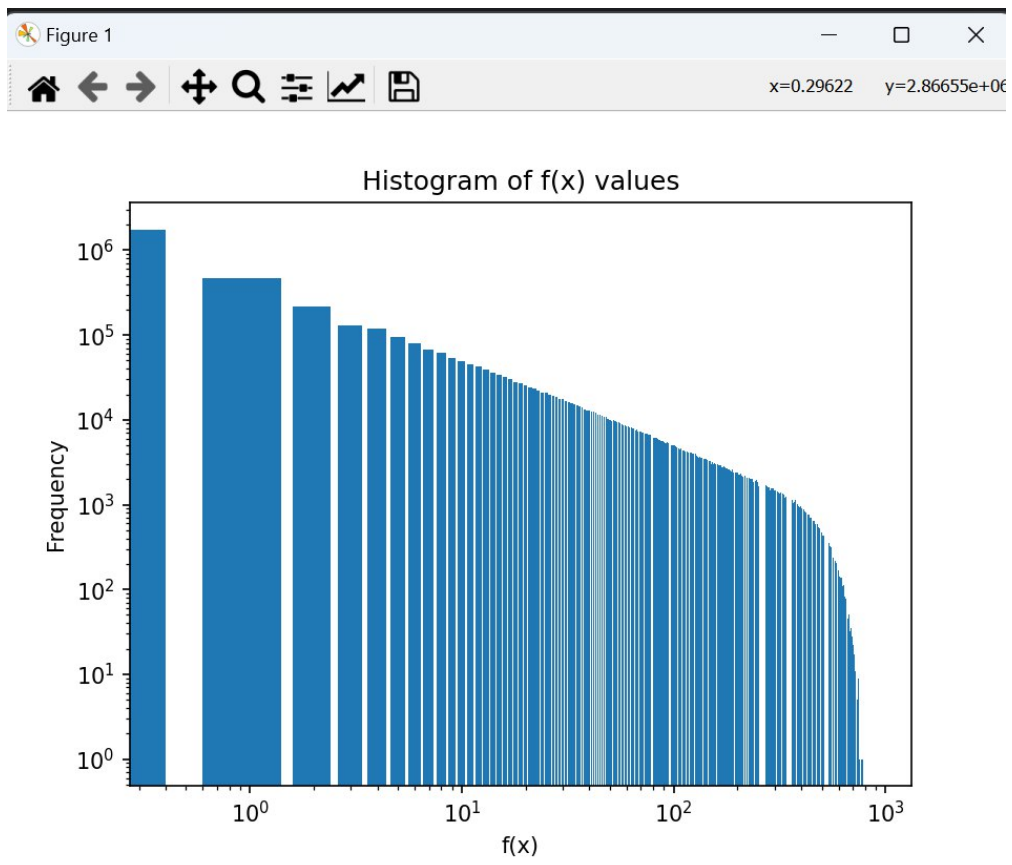


Рис. 2: Вычисление размера лавины для решетки  $50 \times 50$



## 3. Единица песочной группы

### 3.1. Псевдо-Манхеттенская решетка и граничные условия

В работе [2] было доказано, что единица псевдо-Манхеттенской решетки с открытыми границами имеет структуру самоподобия, и описывается через единицу решетки меньшего размера. Если рассмотреть псевдо-Манхеттенскую решетку с некоторыми другими граничными условиями, то единица тоже будет иметь некоторую регулярную структуру. Определим эти граничные условия более формально. Рассмотрим псевдо-манхеттенскую решетку с открытыми границами. Теперь, если  $j \neq 0$ , то уберем у вершины  $v(i, j)$  ребра которые ведут в сток или из стока. Ориентацию вершин в сток и из стока оставим такую же как и у решетки с открытыми границами. Заметим, что тогда у нас появятся вершины с  $\text{indeg}(v) \neq \text{outdeg}(v)$ .

Более формально,  $\delta\Gamma = \{v_{i,j} : j = 0\}$

Таким образом, граничными будут только вершины первой строки решетки. Очевидно, такой граф всё еще сильно связан и из любой вершины есть ориентированный путь в сток. Здесь и далее будем рассматривать псевдо-Манхеттенские решетки размера  $L \times L$ , подразумевая что  $L$  нечетно. Также зададим каждой вершине пару целочисленных координат. Начиная с самой нижней левой вершины,  $v_{i,j}$  есть вершина с координатами  $i, j$ , далее за позицию по горизонтали будет отвечать координата  $i$ , а за позицию по вертикали координата  $j$ . Пример такой решетки размера  $5 \times 5$  приведен на Рис. 3, в вершинах написаны координаты  $(i, j)$  данной вершины, или  $s$  если вершина сток.

### 3.2. Единица Крейца

Самым эффективным способом вычисления единицы песочной группы является единица Крейца. Рассмотрим вершины, в которые есть входящее ребро из стока. Рассмотрим состояние, равное 1 на этих вершинах, и 0 на всех остальных вершинах. Назовем это состояние  $\beta$ . За-

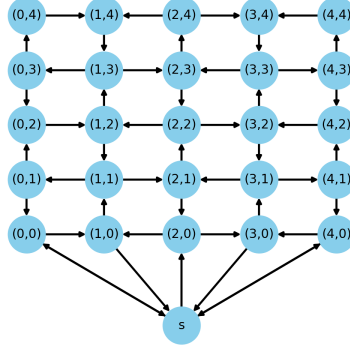


Рис. 3: Граф псевдо-манхеттенской решетки 5x5.

тем доказывается, что состояние  $\beta$  действует как единица на возвратных состояниях. Тогда рассмотрим состояния  $k\beta$ , где  $k \in \mathbb{N}$ . Начиная с некоторого числа  $n$  состояние  $n\beta$  станет возвратным. Это состояние и является единицей группы. Например, в статье [2] рассматривается такая конструкция, доказывается что она является единицей группы для Псевдо-Манхеттенской решетки с открытой границей, и доказывается что такое состояние имеет регулярную структуру, и описывается рекурсивно через решетки меньшего размера. Более того, доказана оценка на  $n$ , А именно  $n\beta$  будет возвратным состоянием для такой решетки размера  $2L \times 2L$ , если  $n = \frac{L(L+1)}{2}$ .

Однако, если рассмотреть граничные условия, заданные в пункте 3.1 окажется, что состояние  $n\beta$  не является единицей песочной группы для такой модели.

**Утверждение.** Пусть  $\Gamma = (G, s)$ , где  $G$  - псевдо-манхеттенская решетка размера  $L \times L$  с граничными условиями, описанными в пункте 3.1. Тогда состояние  $n\beta$  не является единицей песочной группы.

Для доказательства этого утверждения необходимо доказать следующую теорему.

**Теорема 1.** Пусть  $\Gamma = (G, s)$ , где  $G$  - псевдо-манхеттенская решетка размера  $L \times L$  с граничными условиями, описанными в пункте 3.1. Тогда для  $n \in \mathbb{N}_0$

$$C_{max} \oplus (2n + 1)\beta \neq C_{max}$$

*Доказательство.* Предположим обратное, пусть  $C_{max} \oplus (2n + 1)\beta = C_{max}$ . Так же рассмотрим функцию обвалов при релаксации  $(C_{max} + (2n + 1)\beta)$ , пусть  $g(i, j)$  есть число обвалов в вершине  $v_{i,j}$  при этой релаксации. Тогда:

$$\begin{aligned} \forall v \in E, C_{max} \oplus (2n + 1)\beta(v) &= outdeg(v) - 1 \Rightarrow \\ C_{max} \oplus (2n + 1)\beta(2, 0) &= 1 + 2n + 1 + g(2, 1) - 2g(2, 0) = 1 \Rightarrow \\ 1 + g(2, 1) &= 2g(2, 0) - 2n \end{aligned}$$

Таким образом,  $g(2, 1)$  нечетно. Более того, для всех  $k \in \mathbb{N} : 2k + 1 < L$  верно, что  $g(2, 2k + 1)$  нечетно. Докажем этот факт по индукции.

**База:**  $g(2, 1)$  нечетно

**Переход**  $g(2, j) \rightarrow g(2, j + 2)$ : Действительно, пусть  $g(2, j)$  нечетно, тогда:

$$\begin{aligned} C_{max} \oplus (2n + 1)\beta(2, j + 1) &= 1 + g(2, j) + g(2, j + 2) - 2g(2, j + 1) = 1 \Rightarrow \\ g(2, j) + g(2, j + 2) &= 2g(2, j + 1) \Rightarrow g(2, j + 2) = 2k + 1, k \in \mathbb{N}_0 \end{aligned}$$

Таким образом, в частности,  $g(2, L - 2)$  нечетно. Но, по предположению  $C_{max} \oplus (2n + 1)\beta = C_{max}$ , следовательно

$$\begin{aligned} 1 + g(2, L - 2) - 2g(2, L - 1) &= 1 \Rightarrow \\ g(2, L - 2) &= 2g(2, L - 1) \end{aligned}$$

Что приводит к противоречию, значит  $C_{max} \oplus (2n + 1)\beta \neq C_{max}$  □

Вычисления для серии решеток разных размеров также показали, что верно  $C_{max} \oplus (2n + 1)\beta \oplus \beta \neq C_{max}$ . Данный факт показывает, что для таких решеток не получится получить единицу группы через состояние  $\beta$ .

### 3.3. Единица через максимально стабильное состояние

Таким образом, в качестве единицы группы не получится рассматривать состояние  $n\beta$ . Однако есть еще один способ вычисления едини-

цы, не зависящий от строения графа.

**Утверждение.** *Состояние  $(2C_{max} - (2C_{max})^\circ)^\circ$  является единицей песочной группы.*

Действительно, это состояние является возвратным и эквивалентно нулю, а значит является единицей группы. Более подробное доказательство этого факта можно найти в [3]. Вычислим это состояние для песочной модели на решетках определенного размера  $L$ . В таблице ниже показаны примеры вычисления состояний  $(2C_{max} - (2C_{max})^\circ)^\circ$  и  $(2C_{max})^\circ$ . Черным цветом показано значение 0, серым цветом значение 1.

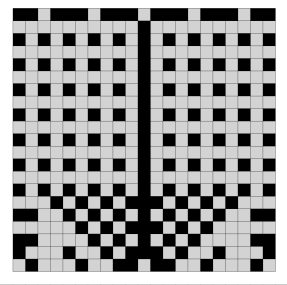
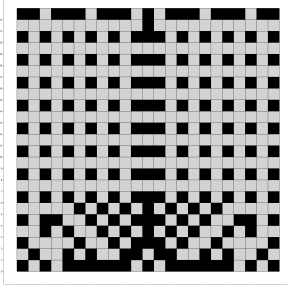
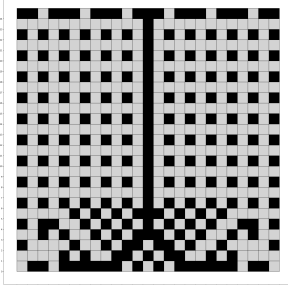
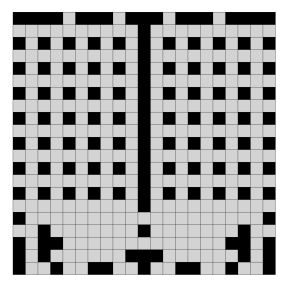
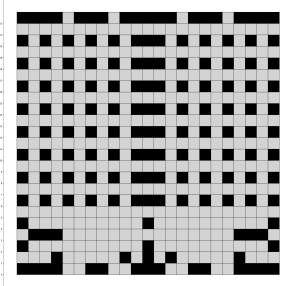
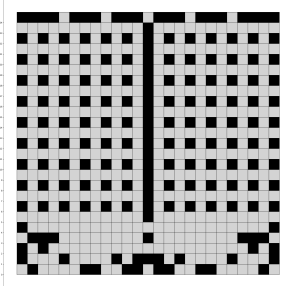
$(2C_{max} - (2C_{max})^\circ)^\circ, L = 21$	$(2C_{max} - (2C_{max})^\circ)^\circ, L = 23$	$(2C_{max} - (2C_{max})^\circ)^\circ, L = 25$
		
$(2C_{max})^\circ, L = 21$	$(2C_{max})^\circ, L = 23$	$(2C_{max})^\circ, L = 25$
		

Таблица 1: Состояния  $(2C_{max} - (2C_{max})^\circ)^\circ$  и  $(2C_{max})^\circ$

Заметим, что структура, возникающая в верхней части решетки, одинакова для не граничных вершин. С помощью вычислений данного состояния для решеток разных размеров была сформулирована гипотеза:

Пусть  $j > \frac{L}{2}$ , тогда паттерн

$$\begin{pmatrix} v_{i-1,j+1} & v_{i,j+1} & v_{i+1,j+1} \\ v_{i-1,j} & v_{i,j} & v_{i+1,j} \\ v_{i-1,j-1} & v_{i,j-1} & v_{i+1,j-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

сохраняется при переходе от состояния  $2C_{max}^\circ$  к  $(2C_{max} - (2C_{max})^\circ)^\circ$ . Таким образом, важно изучить свойства состояния  $2C_{max}^\circ$ . Для этого введем некоторые обозначения. Пусть  $C$  - состояние. Тогда определим:

$$C(i, j) = C(v_{i,j})$$

т.е. количество песка в вершине  $v_{i,j}$  при состоянии  $C$ . Так же определим

$$C^\circ(i, j) = C^\circ(v_{i,j})$$

т.е. количество песка в вершине  $v_{i,j}$  после релаксации состояния  $C$ . Еще определим функцию:  $g(i, j)$  - количество обвалов в вершине  $v_{i,j}$  при релаксации  $C$ . Тогда, очевидно:

$$C^\circ(i, j) = C(i, j) + \sum_{v_{k,l} \in V: (v_{k,l}, v_{i,j}) \in E} g(k, l) - outdeg(i, j) \cdot g(i, j)$$

Действительно, т.к. релаксация не зависит от порядка мы можем сделать обвалы во всех других вершинах. Тогда в вершине будет  $C(i, j) + \sum_{v_{k,l} \in V: (v_{k,l}, v_{i,j})} g(k, l)$  песка. Теперь, если сделать обвал вершины  $v_{i,j}$ , то из нее уйдет  $outdeg(v_{i,j}) \cdot g(i, j)$  единиц песка.

**Лемма 1.** Пусть  $v_{i,j} \in V \setminus \delta\Gamma$ ,  $C = 2C_{max}$ ,  $i + j \equiv 0 \pmod{2}$ ,  $0 < i, j < L - 1$ . Тогда

$$1. C^\circ(i, j) = 0 \Leftrightarrow g(i, j + 1) \equiv g(i, j - 1) \pmod{2}$$

$$2. C^\circ(i, j) = 1 \Leftrightarrow g(i, j + 1) \not\equiv g(i, j - 1) \pmod{2}$$

*Доказательство.*

Для всех вершин  $v_{i,j} \in V \setminus \delta\Gamma$  значение  $outdeg(i, j) = 2$ . Также, заметим что при  $i + j \equiv 0 \pmod{2}$  множество вершин, из которых есть исходящее

ребро в  $v_{i,j}$  описывается как  $\{v \in V : (v, w) \in E\} = \{v_{i,j+1}, v_{i,j-1}\}$ . Это следует из строения псевдо-манхеттенской решетки.

1. ( $\Rightarrow$ ) Пусть  $C^\circ(i, j) = 0$ , тогда

$$C(i, j) + g(i, j - 1) + g(i, j + 1) - 2g(i, j) = 0$$

Так как  $C = 2C_{\max}$ , то  $C(i, j) = 2$ , следовательно

$$\begin{aligned} 2 + g(i, j - 1) + g(i, j + 1) &= 2g(i, j) \Rightarrow \\ g(i, j - 1) + g(i, j + 1) &= 2(g(i, j) - 1) \Rightarrow \\ g(i, j + 1) &\equiv g(i, j - 1) \pmod{2} \end{aligned}$$

( $\Leftarrow$ ) Пусть  $g(i, j + 1) \equiv g(i, j - 1) \pmod{2}$ , тогда

$$\begin{aligned} g(i, j - 1) + g(i, j + 1) &= 2k, k \in \mathbb{N}_0 \Rightarrow \\ C^\circ(i, j) = 2 + 2k - 2g(i, j) &= 2(1 + k - g(i, j)) \Rightarrow \\ C^\circ(i, j) &\equiv 0 \pmod{2} \end{aligned}$$

Так как  $C^\circ(i, j)$  принадлежит стабильному состоянию, то

$$C^\circ(i, j) < \text{outdeg}(v_{i,j}) = 2 \Rightarrow C^\circ(i, j) = 0$$

2. ( $\Rightarrow$ ) Пусть  $C^\circ(i, j) = 1$ , тогда

$$\begin{aligned} 1 = 2 + g(i, j - 1) + g(i, j + 1) - 2g(i, j) &\Rightarrow \\ 2g(i, j) &= g(i, j - 1) + g(i, j + 1) + 1 \Rightarrow \\ g(i, j - 1) &\not\equiv g(i, j + 1) \pmod{2} \end{aligned}$$

( $\Leftarrow$ ) Пусть  $g(i, j - 1) \not\equiv g(i, j + 1) \pmod{2}$ , тогда

$$g(i, j - 1) + g(i, j + 1) = 2k + 1, k \in \mathbb{N}_0$$

Следовательно

$$\begin{aligned} C^\circ(i, j) = 2 + 2k + 1 - 2g(i, j) &= 2(1 + k - g(i, j)) + 1 \Rightarrow \\ C^\circ(i, j) &\not\equiv 0 \pmod{2} \end{aligned}$$

Так как  $C^\circ(i, j)$  принадлежит стабильному состоянию, то

$$C^\circ(i, j) < 2 \Rightarrow C^\circ(i, j) = 1$$

□

**Лемма 2.** Пусть  $v_{i,j} \in V \setminus \delta\Gamma$ ,  $C = 2C_{\max}$ ,  $i + j \not\equiv 0 \pmod{2}$ ,  $0 < i, j < L - 1$ . Тогда

$$1. C^\circ(i, j) = 0 \Leftrightarrow g(i + 1, j) \equiv g(i - 1, j) \pmod{2}$$

$$2. C^\circ(i, j) = 1 \Leftrightarrow g(i + 1, j) \not\equiv g(i - 1, j) \pmod{2}$$

*Доказательство.*

Для всех вершин  $v_{i,j} \in V \setminus \delta\Gamma$  значение  $\text{outdeg}(i, j) = 2$ . Также, заметим что при  $i + j \not\equiv 0 \pmod{2}$  множество вершин, из которых есть исходящее ребро в  $v_{i,j}$  описывается как  $\{v \in V : (v, w) \in E\} = \{v_{i+1,j}, v_{i-1,j}\}$ . Это следует из строения псевдо-манхеттенской решетки.

1. ( $\Rightarrow$ ) Пусть  $C^\circ(i, j) = 0$ , тогда

$$C(i, j) + g(i - 1, j) + g(i + 1, j) - 2g(i, j) = 0$$

Так как  $C = 2C_{\max}$ , то  $C(i, j) = 2$ , следовательно

$$\begin{aligned} 2 + g(i - 1, j) + g(i + 1, j) &= 2g(i, j) \Rightarrow \\ g(i - 1, j) + g(i + 1, j) &= 2(g(i, j) - 1) \Rightarrow \\ g(i + 1, j) &\equiv g(i - 1, j) \pmod{2} \end{aligned}$$

( $\Leftarrow$ ) Пусть  $g(i + 1, j) \equiv g(i - 1, j) \pmod{2}$ , тогда

$$\begin{aligned} g(i - 1, j) + g(i + 1, j) &= 2k, k \in \mathbb{N}_0 \Rightarrow \\ C^\circ(i, j) = 2 + 2k - 2g(i, j) &= 2(1 + k - g(i, j)) \Rightarrow \\ C^\circ(i, j) &\equiv 0 \pmod{2} \end{aligned}$$

Так как  $C^\circ(i, j)$  принадлежит стабильному состоянию, то

$$C^\circ(i, j) < \text{outdeg}(v_{i,j}) = 2 \Rightarrow C^\circ(i, j) = 0$$

2. ( $\Rightarrow$ ) Пусть  $C^\circ(i, j) = 1$ , тогда

$$\begin{aligned} 1 &= 2 + g(i-1, j) + g(i+1, j) - 2g(i, j) \Rightarrow \\ 2g(i, j) &= g(i-1, j) + g(i+1, j) + 1 \Rightarrow \\ g(i-1, j) &\not\equiv g(i+1, j) \pmod{2} \end{aligned}$$

( $\Leftarrow$ ) Пусть  $g(i-1, j) \not\equiv g(i+1, j) \pmod{2}$ , тогда

$$g(i-1, j) + g(i+1, j) = 2k + 1, k \in \mathbb{N}_0$$

Следовательно

$$\begin{aligned} C^\circ(i, j) &= 2 + 2k + 1 - 2g(i, j) = 2(1 + k - g(i, j)) + 1 \Rightarrow \\ C^\circ(i, j) &\not\equiv 0 \pmod{2} \end{aligned}$$

Так как  $C^\circ(i, j)$  принадлежит стабильному состоянию, то

$$C^\circ(i, j) < 2 \Rightarrow C^\circ(i, j) = 1$$

□

**Лемма 3.** Пусть  $v_{i,j} \in V \setminus \delta\Gamma$ ,  $C = 2C_{max}$ ,  $i = 0, 0 < j < L - 1$ . Тогда

1. При  $j \equiv 1 \pmod{2}$ ,  $C^\circ(i, j) \equiv g(i+1, j) \pmod{2}$
2. При  $j \equiv 0 \pmod{2}$ ,  $C^\circ(i, j) = 0$

*Доказательство.*

1. Действительно,

$$\begin{aligned} C(i, j) + g(i+1, j) - 2g(i, j) &= C^\circ(i, j) \Rightarrow \\ 2 + g(i+1, j) - 2g(i, j) &= C^\circ(i, j) \Rightarrow \\ C^\circ(i, j) &= 2(1 - g(i, j)) + g(i+1, j) \Rightarrow \\ C^\circ(i, j) &\equiv g(i+1, j) \pmod{2} \end{aligned}$$

2. В этом случае  $outdeg(i, j) = 1 \Rightarrow C^\circ(i, j) < outdeg(i, j) = 1 \Rightarrow C^\circ(i, j) = 0$ .



□

**Лемма 4.** Пусть  $v_{i,j} \in V \setminus \delta\Gamma$ ,  $C = 2C_{max}$ ,  $i = L - 1, 0 < j < L - 1$ . Тогда

1. При  $j \equiv 1 \pmod{2}$ ,  $C^\circ(i, j) \equiv g(i - 1, j) \pmod{2}$
2. При  $j \equiv 0 \pmod{2}$ ,  $C^\circ(i, j) = 0$

*Доказательство.*

1.

$$\begin{aligned} C(i, j) + g(i - 1, j) - 2g(i, j) &= C^\circ(i, j) \Rightarrow \\ 2 + g(i - 1, j) - 2g(i, j) &= C^\circ(i, j) \Rightarrow \\ C^\circ(i, j) &= 2(1 - g(i, j)) + g(i - 1, j) \Rightarrow \\ C^\circ(i, j) &\equiv g(i - 1, j) \pmod{2} \end{aligned}$$

2. В этом случае  $outdeg(i, j) = 1 \Rightarrow C^\circ(i, j) < outdeg(i, j) = 1 \Rightarrow C^\circ(i, j) = 0$ .

□

**Лемма 5.** Пусть  $v_{i,j} \in V \setminus \delta\Gamma$ ,  $C = 2C_{max}$ ,  $j = L - 1, 0 < i < L - 1$ . Тогда

1. При  $i \equiv 0 \pmod{2}$ ,  $C^\circ(i, j) \equiv g(i, j - 1) \pmod{2}$
2. При  $i \equiv 1 \pmod{2}$ ,  $C^\circ(i, j) = 0$

*Доказательство.*

1. Действительно,

$$\begin{aligned} C(i, j) + g(i, j - 1) - 2g(i, j) &= C^\circ(i, j) \Rightarrow \\ 2 + g(i, j - 1) - 2g(i, j) &= C^\circ(i, j) \Rightarrow \\ C^\circ(i, j) &= 2(1 - g(i, j)) + g(i, j - 1) \Rightarrow \\ C^\circ(i, j) &\equiv g(i, j - 1) \pmod{2} \end{aligned}$$

2. В этом случае  $outdeg(i, j) = 1 \Rightarrow C^\circ(i, j) < outdeg(i, j) = 1 \Rightarrow C^\circ(i, j) = 0$ .

□

**Лемма 6.** Пусть  $v_{i,j} \in \delta\Gamma$ ,  $C = 2C_{max}$ ,  $j = 0$ . Тогда

1. При  $i \equiv 0 \pmod{2}$ ,  $C^\circ(i, j) \equiv g(i, j + 1) \pmod{2}$

2. При  $i \equiv 1 \pmod{2}$ ,  $C^\circ(i, j) = 0 \Leftrightarrow g(i - 1, j) \equiv g(i + 1, j) \pmod{2}$

*Доказательство.*

1. Действительно,

$$\begin{aligned} C(i, j) + g(i, j + 1) - 2g(i, j) &= C^\circ(i, j) \Rightarrow \\ C^\circ(i, j) &= 2(1 - 2g(i, j)) + g(i, j + 1) \Rightarrow \\ C^\circ(i, j) &\equiv g(i, j + 1) \pmod{2} \end{aligned}$$

2. В этом случае

$$\begin{aligned} C(i, j) + g(i - 1, j) + g(i + 1, j) - 2g(i, j) &= C^\circ(i, j) \Rightarrow \\ C^\circ(i, j) &= 2(1 - 2g(i, j)) + g(i - 1, j) + g(i + 1, j) \Rightarrow \\ C^\circ(i, j) = 0 &\Leftrightarrow g(i - 1, j) \equiv g(i + 1, j) \pmod{2} \end{aligned}$$

□

## 4. Выводы

В работе изучались песочные модели на ориентированных решетках. Была рассмотрена песочная динамика, численные эксперименты показали что размер лавины обладает теми же свойствами что и в неориентированном случае.

Так же было показано, что при некоторых граничных условиях для псевдо-Манхеттенской решетки, состояние  $n\beta$  не будет являться единицей группы. Далее было рассмотрено состояние, вычисляющее единицу через максимально стабильное состояние. Для него были доказаны леммы про зависимость значения песка в вершине после релаксации и значения функции обвалов для соседних вершин.

## Список литературы

- [1] Bhupatiraju Sandeep, Hanson Jack, and J'arai Antal A. Inequalities for critical exponents in d-dimensional sandpiles // arXiv: Probability. — 2016. — Access mode: <https://api.semanticscholar.org/CorpusID:119661665>.
- [2] Caracciolo Sergio, Paoletti Guglielmo, and Sportiello Andrea. Explicit characterization of the identity configuration in an Abelian sandpile model // Journal of Physics A: Mathematical and Theoretical. — 2008. — Vol. 41. — P. 495003. — Access mode: <https://api.semanticscholar.org/CorpusID:62834025>.
- [3] Corry Scott and Perkinson David. Divisors and Sandpiles: An Introduction to Chip-Firing. — 2018. — Access mode: <https://api.semanticscholar.org/CorpusID:226951772>.
- [4] Izmailian N. Sh., Papoyan Vl. V., Priezzhev V. B., and Hu Chin-Kun. Self-organizing behavior in a lattice model for co-evolution of virus and immune systems // Phys. Rev. E. — 2007. — Apr. — Vol. 75. — P. 041104. — Access mode: <https://link.aps.org/doi/10.1103/PhysRevE.75.041104>.

## А. Приложение

```
#include <iostream>
#include <vector>
#include <cstdlib>
#include <ctime>
#include <set>
#include <algorithm>
#include <numeric>
#include <random>
#include <fstream>

using namespace std;

int out_deg(const std::vector<std::vector<int>>& A, int i) {
    int sum = 0;
    for (int j = 0; j < A[i].size(); ++j) {
        if (i != j) {
            sum += A[i][j];
        }
    }
    return sum;
}

int in_deg(const std::vector<std::vector<int>>& A, int i) {
    int sum = 0;
    for (int j = 0; j < A.size(); ++j) {
        if (i != j) {
            sum += A[j][i];
        }
    }
    return sum;
}
```

```

std::vector<std::vector<int>> generate_full_pseudo_manhattan(
    int size = n * m + 1;
    std::vector<std::vector<int>> A(size, std::vector<int>(
        for (int i = 1; i < n - 1; ++i) {
            if (i % 2 == 0) {
                for (int j = 0; j < m; ++j) {
                    if (j == 0) {
                        A[i * m + j][i * m + j + 1] = 1;
                    } else {
                        if (j % 2 == 0) {
                            if (j != m - 1) {
                                A[i * m + j][i * m + j + 1] = 1;
                            }
                            A[i * m + j][i * m + j - 1] = 1;
                        } else {
                            A[i * m + j][m * (i - 1) + j] = 1;
                            A[i * m + j][m * (i + 1) + j] = 1;
                        }
                    }
                }
            }
        } else {
            for (int j = 0; j < m; ++j) {
                if (j % 2 == 0) {
                    A[i * m + j][m * (i - 1) + j] = 1;
                    A[i * m + j][m * (i + 1) + j] = 1;
                } else {
                    if (j != m - 1) {
                        A[i * m + j][i * m + j + 1] = 1;
                    }
                    A[i * m + j][i * m + j - 1] = 1;
                }
            }
        }
    }
}

```

```

    }
  }
}

for (int j = 0; j < m; ++j) {
  if (j == 0) {
    A[j][0 * m + j + 1] = 1;
  } else {
    if (j % 2 == 0) {
      if (j != m - 1) {
        A[j][j + 1] = 1;
      }
      A[j][j - 1] = 1;
    } else {
      A[j][m + j] = 1;
    }
  }
}

if ((n - 1) % 2 == 0) {
  for (int j = 0; j < m; ++j) {
    if (j == 0) {
      A[(n - 1) * m + j][(n - 1) * m + j + 1] = 1;
    } else {
      if (j % 2 == 0) {
        if (j != m - 1) {
          A[(n - 1) * m + j][(n - 1) * m + j - 1] = 1;
        }
        A[(n - 1) * m + j][(n - 1) * m + j + 1] = 1;
      } else {
        A[(n - 1) * m + j][m * ((n - 1) - 1) + j] = 1;
      }
    }
  }
}

```

```

    }
} else {
    for (int j = 0; j < m; ++j) {
        if (j % 2 == 0) {
            A[(n - 1) * m + j][m * ((n - 1) - 1) + j] = 1;
        } else {
            if (j != m - 1) {
                A[(n - 1) * m + j][(n - 1) * m + j + 1] = 1;
            }
            A[(n - 1) * m + j][(n - 1) * m + j - 1] = 1;
        }
    }
}

for (int i = 0; i < n * m; ++i) {
    if (out_deg(A, i) == in_deg(A, i)) {
        if (out_deg(A, i) == 1) {
            A[i][n * m] = 1;
            A[n * m][i] = 1;
        }
        continue;
    }
    if (out_deg(A, i) > in_deg(A, i)) {
        A[n * m][i] = 1;
    } else {
        A[i][n * m] = 1;
    }
}

return A;
}

```

```

bool not_stable(vector<vector<int>>& matrix_graph, vector<int> in_deg) {

```



```

vector<int> sum_(matrix_graph.size(), 0);
for (int i = 0; i < matrix_graph.size(); i++) {
    sum_[i] = accumulate(matrix_graph[i].begin(), matrix
}
for (int i = 0; i < pesok_array.size(); i++) {
    if (pesok_array[i] >= sum_[i]) {
        return true;
    }
}
return false;
}

```

```

void make_obval(vector<vector<int>>& matrix_graph, vector<int>
int counter = 0;
while (not_stable(matrix_graph, pesok_array)) {
    counter++;
    if(counter % 1024 == 0)
    {
        cout << counter << endl;
    }
}

```

```

vector<int> sum_(matrix_graph.size(), 0);
for (int i = 0; i < matrix_graph.size(); i++) {
    sum_[i] = accumulate(matrix_graph[i].begin(), m
}
for (int i = 0; i < pesok_array.size(); i++) {
    if (pesok_array[i] >= sum_[i]) {
        lavina_set.insert(i);
        for(int j = 0; j < matrix_graph[i].size();
        {
            if(j == (matrix_graph[i].size() - 1))
            {
                pesok_array[i]--;
            }
        }
    }
}

```

```

                continue;
            }
            if(matrix_graph[i][j] == 1)
            {
                pesok_array[i]--;
                pesok_array[j]++;
            }
        }
    }
}

```

```

vector<pair<int, int>> random_sand(vector<int>& pesok_array
set<int> lavina_set;
vector<pair<int, int>> posledovatelnost;
for (int i = 0; i < 5000000; i++) {
    if (i % 1000 == 0) {
        cout << i << endl;
    }
    int randint = rand() % pesok_array.size();
    pesok_array[randint] += 1;
    make_obval(matrix, pesok_array, lavina_set);
    posledovatelnost.push_back(make_pair(i, lavina_set));
    lavina_set.clear();
}
return posledovatelnost;
}

```

```

int main() {
    srand(time(NULL));
    int N = 50;
    int M = 50;

```

```
vector<vector<int>> A = generate_full_pseudo_manhattan(L, M);
vector<int> pesok;

for (int i = 0; i < N * M; ++i) {
    pesok.push_back(0);
}

auto answer = random_sand(pesok, A);

write_vector_to_file(answer, "output.txt");
return 0;
}
```