

## **Аннотация:**

В настоящее время методы клеточно-автоматного моделирования активно развиваются. Клеточно-автоматные модели часто используются для исследования сложных систем в различных научных сферах, таких как физика, биология и химия для исследования следующих процессов: динамика газа и жидкости, распространение волн и формирование паттернов. В данной работе рассматривается процесс создания параллельной реализации библиотеки клеточно-автоматных топологий, разрабатываемой в ИВМиМГ СО РАН, для синхронных клеточных автоматов и тестирование данной реализации на практически значимой задаче.

**Тема работы:** Реализация параллельной версии библиотеки клеточно-автоматных топологий и ее применение для модели «Домино»

**Состав коллектива:** Трубицына Юлия Сергеевна, студентка НГУ; Медведев Юрий Геннадьевич, к.т.н, доцент кафедры ПВ ФИТ НГУ, старший научный сотрудник ИВМиМГ СО РАН, научный руководитель; Киреев Сергей Евгеньевич, ст. преп. кафедры ПВ ФИТ НГУ, научный сотрудник ИВМиМГ СО РАН, соруководитель

## **Научное содержание работы**

**Постановка задачи:** Разработка параллельной версии библиотеки клеточно-автоматных топологий для синхронных клеточных автоматов с возможностью их запуска на параллельных вычислительных системах с распределенной памятью, а также в тестировании разработанной версии на клеточном автомате «Домино».

## **Современное состояние проблемы:**

Клеточный автомат – это дискретная модель вычислений, которая нашла применение в различных исследованиях сложных химических, физических и биологических процессов.

На практике, как правило, клеточные автоматы имеют клеточные массивы больших размеров, из-за чего их последовательная программная реализация требует длительного времени для исполнения. Данная проблема решается с помощью создания параллельной реализации клеточного автомата.

Стоит сказать, что клеточный автомат обладает высокой степенью параллелизма, однако на практике за неимением специальной квалификации у программиста, часто создаются последовательные программы.

На данный момент существует множество систем клеточно-автоматного моделирования, которые упрощают работу с клеточными автоматами, однако, как показал анализ, большинство из них не могут предоставить нужную функциональность для тщательного исследования процессов и не поддерживают клеточные массивы больших размеров.

Вследствие этого появляется потребность в универсальной среде для разработки программных реализаций клеточных автоматов с поддержкой моделей больших размеров.

**Подробное описание работы, включая используемые алгоритмы**

### **1.1 Параллельная реализация библиотеки для синхронных клеточных автоматов**

Алгоритм работы клеточных автоматов в общем случае выглядит следующим образом: на каждом шаге эволюции происходит последовательный обход клеток клеточного массива, во время которого их состояние меняется в соответствии с заданной функцией переходов. Такой обход в программной реализации может создавать значительную вычислительную нагрузку, что будет приводить к длительному времени счета.

В библиотеке обходом клеточного массива занимаются функции симулятора, из-за чего было решено разработать их параллельную реализацию.

Для создания параллельной версии библиотеки был применен метод одномерной декомпозиции области моделирования

Основная идея использования одномерной декомпозиции для текущей задачи состоит в следующем: каждый процесс на начальном этапе работы получает часть поля, над которой он работает обособленно от других, периодически получая информацию о состоянии поля от соседних процессов. В конце работы главный процесс собирает результаты со всех процессов и записывает их в файл.

Функция `CAT_Iterate` создает наибольшую вычислительную нагрузку в системе, однако описанный выше алгоритм невозможно было бы разработать, затрагивая только данную функцию.

Было решено разделить этот алгоритм между тремя функциями симулятора: `CAT_InitSimulator`, `CAT_Iterate` и `CAT_FinalizeSimulator`.

`CAT_InitSimulator` был модифицирован следующим образом: главный процесс считывает всю информацию о клеточном автомате из файла и записывает ее в специальную структуру библиотеки. Данная структура раздается всем процессам. Затем, на основе данных клеточного автомата и количества процессов происходит расчет размеров частей клеточного массива. Стоит отметить, что каждый процесс хранит не только часть клеточного массива, но и состояния граничных клеток соседних процессов.

Далее происходит считывание поля главным процессом по частям таким образом, что первую часть он забирает себе, а последующие раздает оставшимся процессам (Рисунок 1). Затем происходит обмен границ (Рисунок 2).

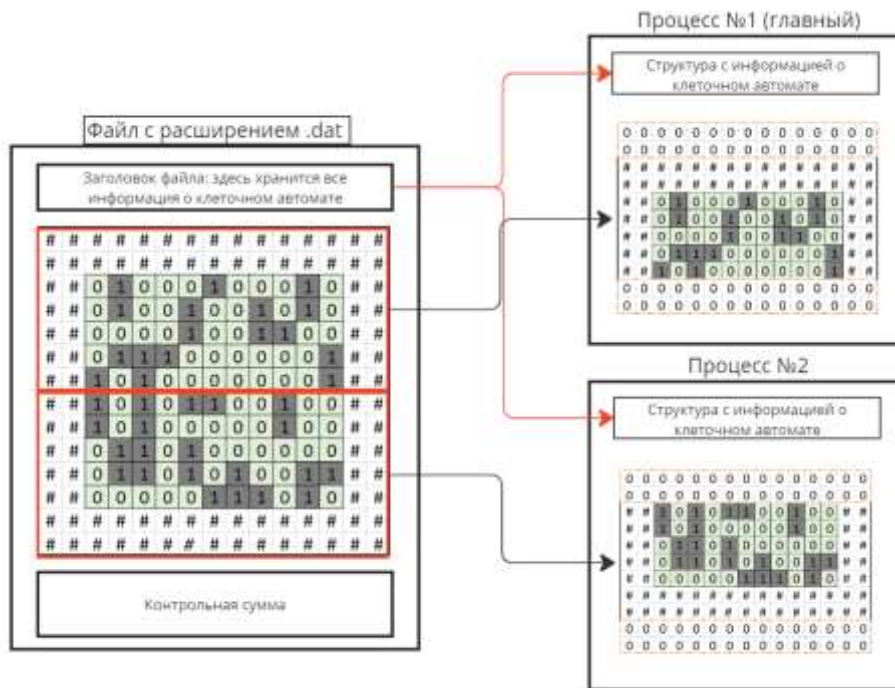


Рисунок 1 – Принцип работы функции `CAT_InitSimulator`

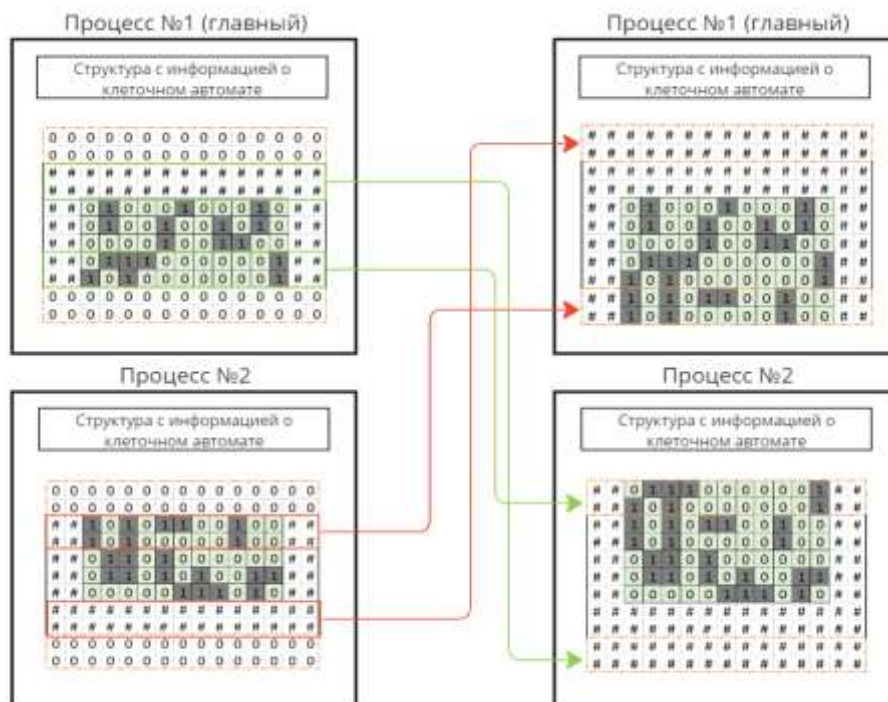


Рисунок 2 – Обмен границ между двумя процессами

В `CAT_Simulator` происходит последовательное прохождение по полю и применение функции переходов к каждой клетке. Изначально данная функция практически не претерпела изменений – в конце прохода каждого процесса по своей части клеточного массива был добавлен обмен границ с помощью блокирующих пересылок. Однако у данной реализации был выявлен один существенный недостаток. Обмен границ занимает

некоторое количество времени для исполнения, особенно в условиях нагруженной сети. В этот момент не производится никаких вычислений. Данный недостаток был решен при помощи неблокирующих пересылок, которые запускаются на фоне вычислений: в функции CAT\_Simulator происходит проход по границам части поля, затем иницируется их неблокирующая пересылка соседним процессам, после чего на фоне нее вычисляются клетки внутренней части поля. Разница в работе реализаций описывается в четвертой главе данной работы.

CAT\_FinalizeSimulator также претерпел изменения. Теперь главный процесс записывает всю информацию о клеточном автомате в файл, а затем начинает собирать с процессов части поля и в порядке очередности записывать их в тот же файл.

## 1.2 Реализация клеточного автомата “Домино” с помощью библиотеки клеточно-автоматных топологий

Тестирование производилось практически значимой задаче – клеточно-автоматной модели “Домино” [Hoffmann, R., Désérable, D. & Sereďyński, F. A cellular automata rule placing a maximal number of dominoes in the square and diamond. J Supercomput 77, 9069–9087 (2021)]. Ее авторы, как следует из статьи, заинтересованы в развитии модели.

Модель “Домино” – это асинхронный клеточный автомат, представленный двумерной квадратной решеткой клеток с двумя базовыми состояниями "0" и "1" и состоянием “#” для обозначения граничных клеток поля. В результате работы клеточного автомата формируются плитки домино, которые покрывают клеточный массив. Для вычислений новых состояний клеток используется функция переходов, определенная в статье. Она пытается максимизировать количество плиток домино в клеточном массиве. Процесс формирования плиток домино можно увидеть на Рисунке 3.

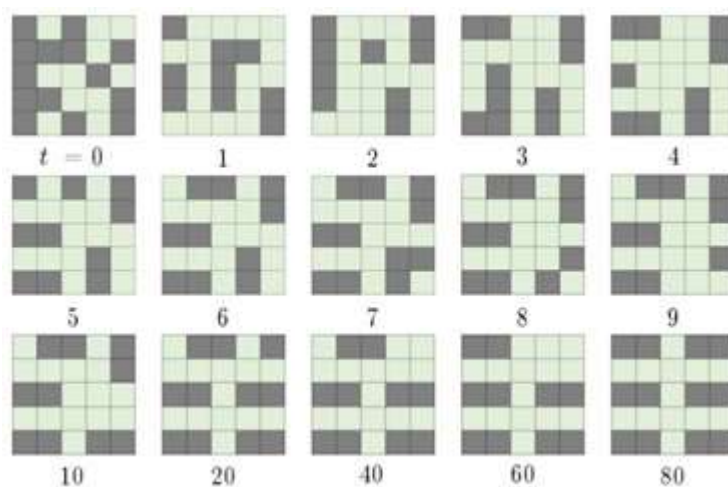


Рисунок 3 – Процесс формирования домино на клеточном массиве.  $t$  – представляет номер итерации

Для тестирования предварительно был совершен переход от ручной асинхронной реализации к синхронной реализации клеточного-автомата

“Домино” [1]. Во время этого перехода была разработана оптимизация “Мультишаблон” для функции переходов данного клеточного автомата. Далее была разработана синхронная версия клеточного автомата с помощью библиотеки.

Также были внесены некоторые дополнения в библиотеку для возможности реализации задачи с помощью нее, а именно была добавлена новая топология.

### **1.3 Тестирование и обсуждение результатов**

В ходе работы был разработан ряд тестов для исследования нефункциональных характеристик реализованных алгоритмов. Также были проведены исследования допустимого размера поля клеточного автомата, который может обработать библиотека на кластере, и максимума сформированных плиток домино на полях различных размеров.

#### **1.3.1 Тестирование разработанной оптимизации**

Тестирование проводилось для сравнения времени работы программной реализации с оптимизацией и реализации без нее. В ходе работы было измерено время работы нескольких реализаций клеточного автомата «Домино». Количество итераций  $T$  для каждого запуска равняется 40 000. Тестирование производилось на полях трех размеров:  $10 \times 10$ ,  $20 \times 20$  и  $40 \times 40$ . Результаты представлены на Рисунке 4.

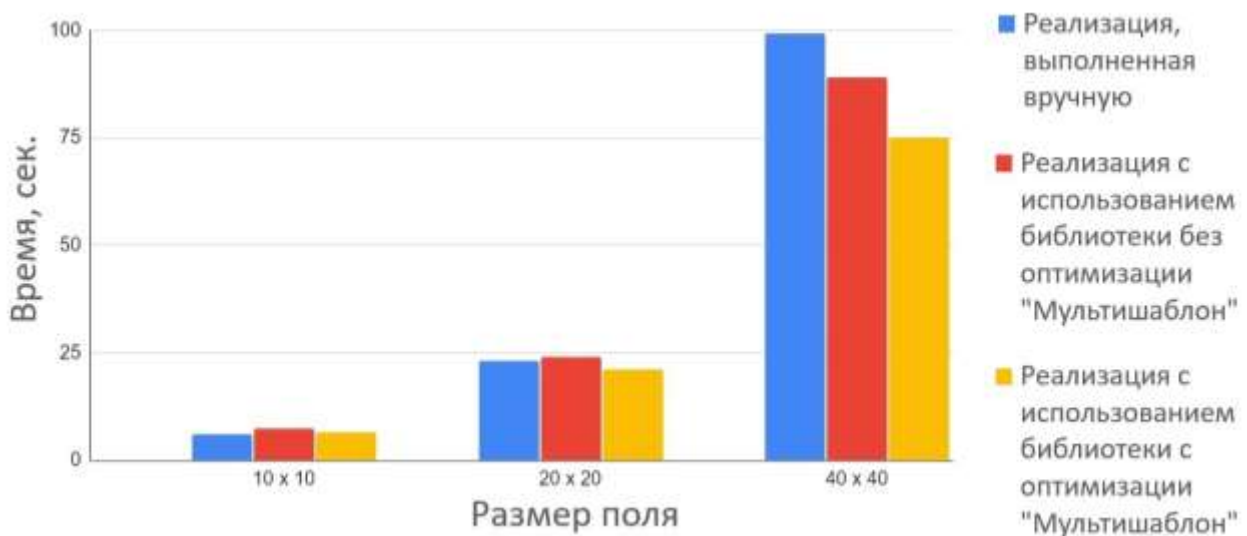


Рисунок 4 – Диаграмма с результатами тестирования разработанной оптимизации

Результаты тестирования показывают, что оптимизированная реализация на полях больших размеров работает на 25% быстрее, чем реализация, выполненная вручную. Также удалось выявить еще одну интересную особенность библиотеки: на больших размерах клеточного массива она показывает результаты лучше, чем реализация, выполненная вручную.

### 1.3.2 Тестирование эффективности реализаций библиотеки с блокирующими и неблокирующими пересылками

Данное тестирование проводилось для определения эффективности параллельных реализаций и их сравнения. Были выбраны следующие параметры клеточного автомата: размер поля  $10000 \times 100$ , количество итераций для каждого запуска  $T = 1000$ . Результаты представлены на Рисунке 5.

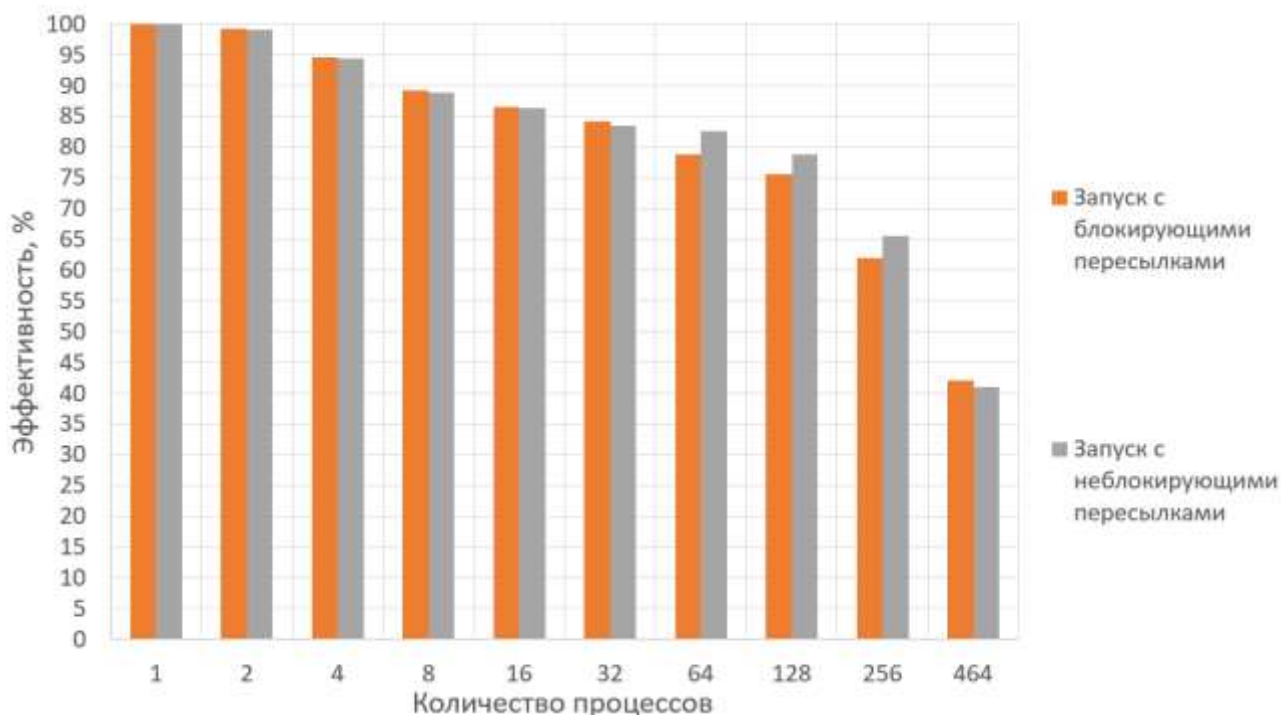


Рисунок 5 – График эффективности параллельных программных реализаций клеточного автомата «Домино» с использованием двух версий библиотеки

Общее время работы программ на одном процессе составило 483 секунды, а на 256 процессах запуск с неблокирующими пересылками показал результат уже в 3 секунды.

Было принято решение о внедрении реализации с неблокирующими пересылками в библиотеку.

### 1.3.3 Исследование среднего значения и дисперсии количества сформированных домино

Данное исследование проводилось для определения количества запусков для исследования максимума сформированных плиток домино на полях различных размеров. Так как функция переходов клеточного автомата «Домино» является вероятностной, то ее результаты могут отличаться от запуска к запуску. Для определения влияния вероятностей на результаты будущего исследования было проведено 10000 запусков клеточного автомата «Домино» на поле размером  $100 \times 100$ , количество итераций на каждом запуске  $T = 10000$ . Было вычислено среднее значение и дисперсия количества сформированных плиток на поле на каждой итерации. Результаты представлены на Рисунке 6.



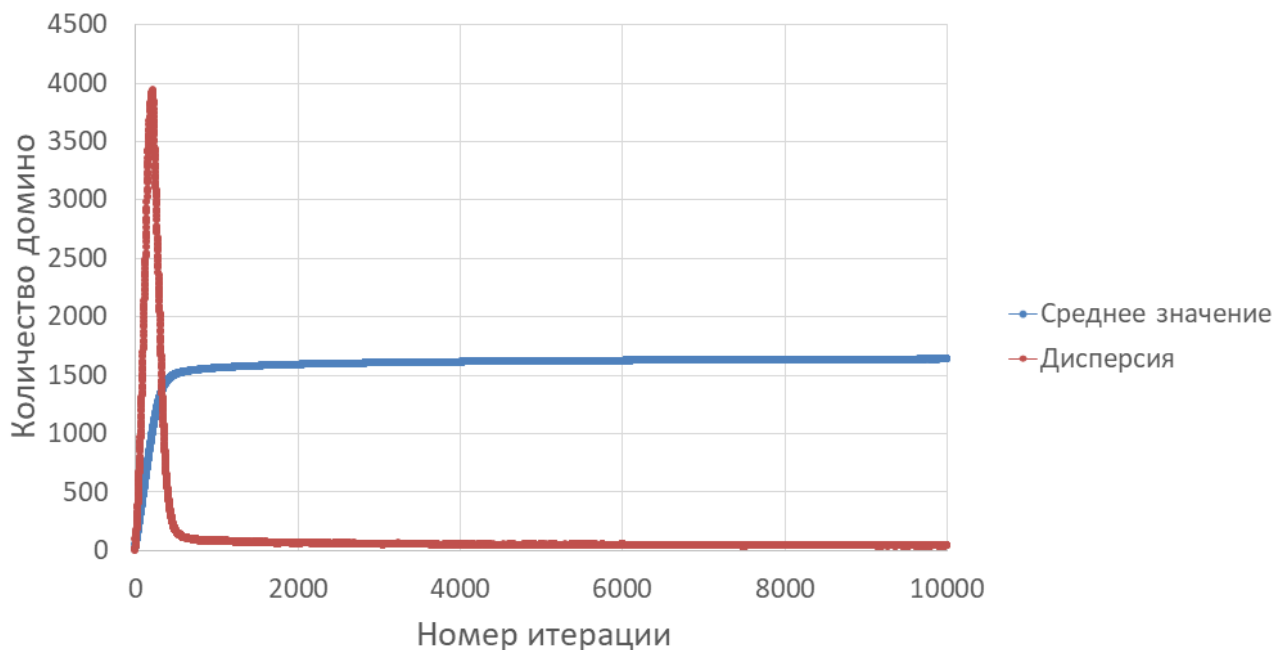


Рисунок 6 – График изменения среднего значения и дисперсии количества сформированных плиток домино на каждой итерации

По результатам четко прослеживается разброс величины примерно на первой тысяче итераций, однако затем отклонение дисперсии становится очень маленьким. Таким образом, для достижения максимального количества сформированных плиток домино на полях различных размеров достаточно производить по одному запуску.

### 1.3.4 Исследование максимума сформированных плиток домино на полях различных размеров и определение допустимого размера поля клеточного автомата

Тестирование производилось с целью изучить поведение формирования максимального покрытия поля плитками домино, а также для определения допустимого размера поля, который может обработать библиотека.

Для исследования было выведено ограничение максимального количества домино сверху следующим соотношением:

$$\left\lfloor \frac{(size + 1)^2}{6} \right\rfloor \quad (1)$$

, где *size* - размер стороны квадратного поля.

Плитки домино расположили таким образом, что первая плитка домино заняла 2 клетки, последующие в столбце – 3, а для дальнейшего расположения в строке понадобилось 4 и 6 клеток. Получилось одно из максимальных покрытий поля плитками домино. Площадь плиток домино равна 6 клеткам, кроме тех плиток, которые расположены около верхней и левой стороны поля (Рисунок 7).



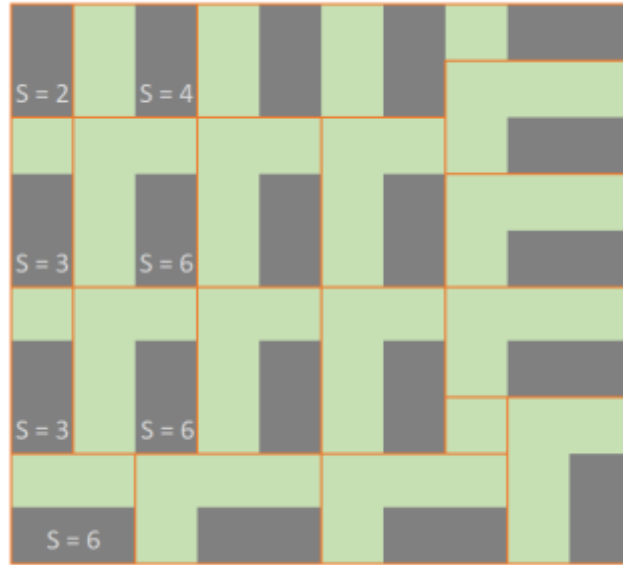


Рисунок 7 – Возможное расположение максимального количества плиток домино на поле размером  $10 \times 10$

Если к этим сторонам добавить по одному ряду (Рисунок 8), в который можно поставить только клетки с состоянием «0», то все плитки станут иметь площадь равную 6 клеткам. Таким образом, разделив всю площадь поля на площадь одной плитки домино, получаем результат, описанный в (1). Округление вниз в общем случае обусловлено оставшимися до 5 клеток включительно, на которые не поместилась плитка домино.

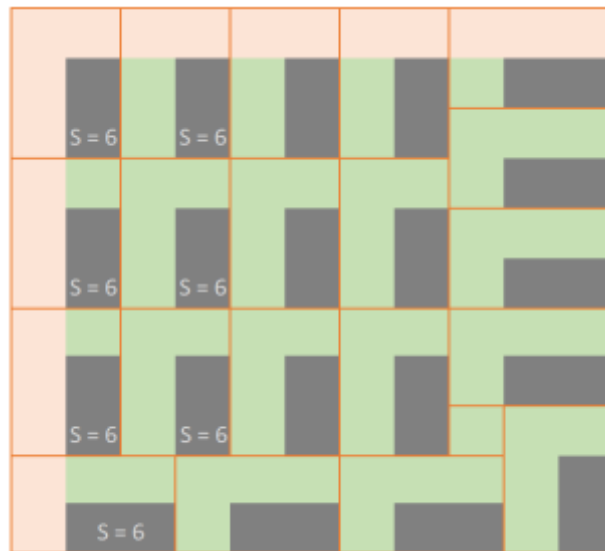


Рисунок 8 – Поле с добавленными пустыми рядами

Условием для остановки программы было достижение 95 % плиток домино от (1). Такой порог был выбран из-за того, что количество итераций, на которых достигается максимальное покрытие поля плитками домино, экспоненциально растет с ростом размера поля [1].

Время работы каждого запуска было ограничено 24 часами в том случае, если клеточный автомат не сможет сформировать 95% плиток домино от (1). Результаты тестирования представлены на Рисунке 9.

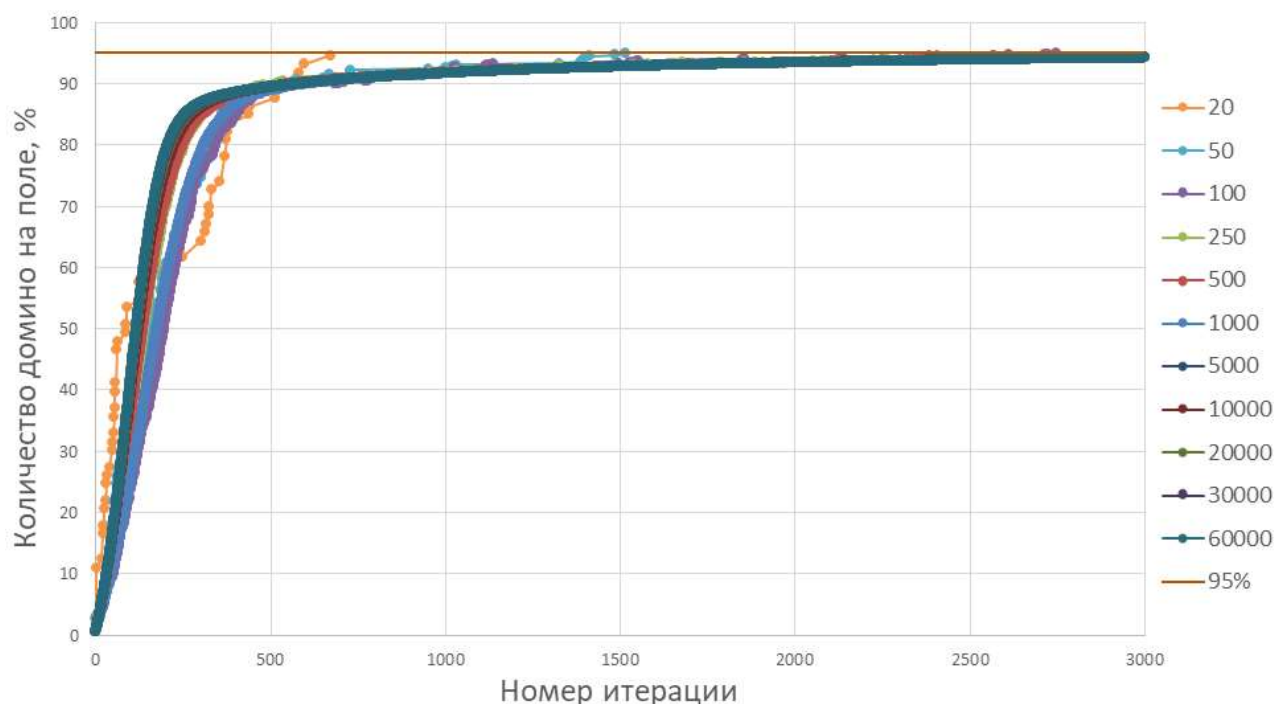


Рисунок 9 – График динамики роста максимального количества сформированных плиток домино на поле в зависимости от его размера.

Исследования показали, что клеточный автомат «Домино» с размерами поля от  $20 \times 20$  до  $60000 \times 60000$  сформировали 95 % плиток домино от рассчитанного ограничения сверху. Поведение клеточного автомата «Домино» при схождении к максимальному покрытию на клеточных массивах разных размерах оказалось схожим. На данный момент библиотека смогла обработать поле клеточного автомата «Домино» размером  $60000 \times 60000$ , в то время как в статье [Hoffmann, R., Désérable, D. & Seredyński, F. A cellular automata rule placing a maximal number of dominoes in the square and diamond. J Supercomput 77, 9069–9087 (2021)] максимальный протестированный размер поля авторами составляет  $18 \times 18$ .

#### **Эффект от использования кластера:**

Использование кластера ИВЦ НГУ позволило провести тестирование, описанное в пункте 1.3.1 данного отчета.

#### **Перечень публикаций, содержащих результат работы:**

- 1) Kireev S., Trubitsyna Yu. Software implementation of asynchronous and synchronous cellular automata with maximum domino tiles coverage // Bull. Nov. Comp. Center, Comp. Science. – 2022. – Issue No 46 (принято в печать).
- 2) Трубицына Ю.С. Оптимизация алгоритма работы клеточного автомата “Домино” и его реализация с помощью библиотеки клеточно-автоматных топологий. // Всероссийская летняя XXXVII молодежная Школа-конференция по параллельному программированию с

международным участием (г. Новосибирск, 5 – 16.07.2021 г.). Всероссийская летняя XXXIX молодежная Школа-конференция по параллельному программированию с международным участием (г. Новосибирск, 4 – 15.07.2022 г.): Тез. докл. / отв. ред. д-р техн. наук, проф. В. Э. Малышкин. Новосиб. гос. унт. – Новосибирск: ИПЦ НГУ (принято в печать).

3) Трубицына Ю.С. Разработка параллельной версии библиотеки клеточно-автоматных топологий «SATlib» // Параллельные вычисления: Материалы 61-й Междунар. науч. конф. 17–26 апреля 2023 г. / Новосиб. гос. ун-т–Новосибирск: ИПЦ НГУ (принято в печать).