

УДК 681.3

**А.Я. Анопrienко, А.П. Коноплева, А.Ю. Василенко**  
Донецкий национальный технический университет, г. Донецк  
кафедра электронных вычислительных машин  
E-mail: [anoprien@cs.dgtu.donetsk.ua](mailto:anoprien@cs.dgtu.donetsk.ua)

## ОЦЕНКА ПРОИЗВОДИТЕЛЬНОСТИ ПРИ МОДЕЛИРОВАНИИ ПОСТБИНАРНЫХ КЛЕТОЧНЫХ АВТОМАТОВ И СПОСОБЫ ЕЕ ПОВЫШЕНИЯ

### *Abstract*

*Anoprienko A., Konoplova A., Vasilenko A. Productivity of the postbinary cellular automaton's simulation and the methods of its increase Postbinary cellular automata are automata with using of hypercodes and hyperlogic. Two realizations of simulation systems for such automata are analyzed: one on the base of Action Script and second on the base of C++. Some algorithms and methods for increase of productivity of such simulation are described.*

**Keywords:** cellular automata, hyperlogic, hypercode, post-binary cellular automata, fast-acting, algorithm.

### *Анотація*

*Анопrienко О.Я., Коноплева Г.П., Василенко А.В. Оцінка продуктивності при моделюванні постбінарних кліткових автоматів і засоби її підвищення Постбінарні кліткові автомати працюють на основі гіперлогіки і гіперкодів. У статті проаналізовано два варіанти реалізації моделюючих систем для таких автоматів, одна з яких є написаною на ActionScript, а інша на C++. Запропоновано алгоритми і методи для підвищення продуктивності таких автоматів.*

**Ключові слова:** клітковий автомат, гіперлогіка, гіперкод, постбінарний клітковий автомат, швидкодія, алгоритм.

### *Аннотация*

*Анопrienко А.Я., Коноплева А.П., Василенко А.В. Оценка производительности при моделировании постбинарных клеточных автоматов и способы ее повышения Постбинарные клеточные автоматы работают на основе гиперлогик и гиперкодов. В статье проанализированы два варианта реализации моделирующих систем для таких автоматов, одна из которых написана на ActionScript, а вторая на C++. Предложены алгоритмы и методы для повышения производительности таких автоматов.*

**Ключевые слова:** клеточный автомат, гиперлогіка, гиперкод, постбинарный клеточный автомат, быстродействие, алгоритм.

В общем случае клеточные автоматы (КА) — это системы с дискретным пространством и временем. Поведение клеточных автоматов полностью определяется локальными зависимостями. Но при этом, используя довольно простые правила, можно моделировать весьма сложные процессы, что определило актуальность данного направления исследований в компьютерных науках.

Существует много разновидностей КА. В данной работе рассматривается модифицированный клеточный автомат Конвея (игры «Жизнь») [1], в котором исходные комбинации задаются с помощью гиперкодов, а при задании состояний клеток и локальных зависимостей используются элементы гиперлогик [2]. Для данной и подобных ей модификаций КА предлагается использовать обобщающее название **постбинарные клеточные автоматы (ПКА)**.

Ранее в ряде публикаций [3, 4] авторами были достаточно детально описаны варианты реализации ПКА. При этом предполагалось, что основное отличие от традиционных реализаций заключается в том, что входные значения ПКА задаются набором из двух гиперкодовых переменных X и Y, хранящих значения координат множества клеток первой генерации КА.

### Оценка быстродействия исходного алгоритма работы программы моделирования ПКА

В предыдущих работах авторов данной статьи рассматривалось поле клеток размерностью не более 64x64, т.к. при попытке расширения размерности клеточного поля возникала проблема существенного снижения быстродействия выполнения программы из-за значительного возрастания количества вычислительных операций [3, 4]. В связи с этим возникла задача увеличения быстродействие работы программы моделирования ПКА на полях большой размерности.

К настоящему времени реализованы варианты программ моделирования ПКА в среде Adobe Flash на языке Action Script и в среде Visual Studio 2008 на языках C++ и C#, интерфейсы которых представлены на рисунках 1 и 2 соответственно.

Быстродействие работы различных вариантов реализации моделирования можно вычислить путем проведения автоматических замеров времени в стадии исполнения кода программ моделирования.

Если сравнивать быстродействие выполнения операций при выполнении реализованного алгоритма моделирования (рис. 3) в одной и той же среде при различных входных условия, то результаты будут существенно отличаться.

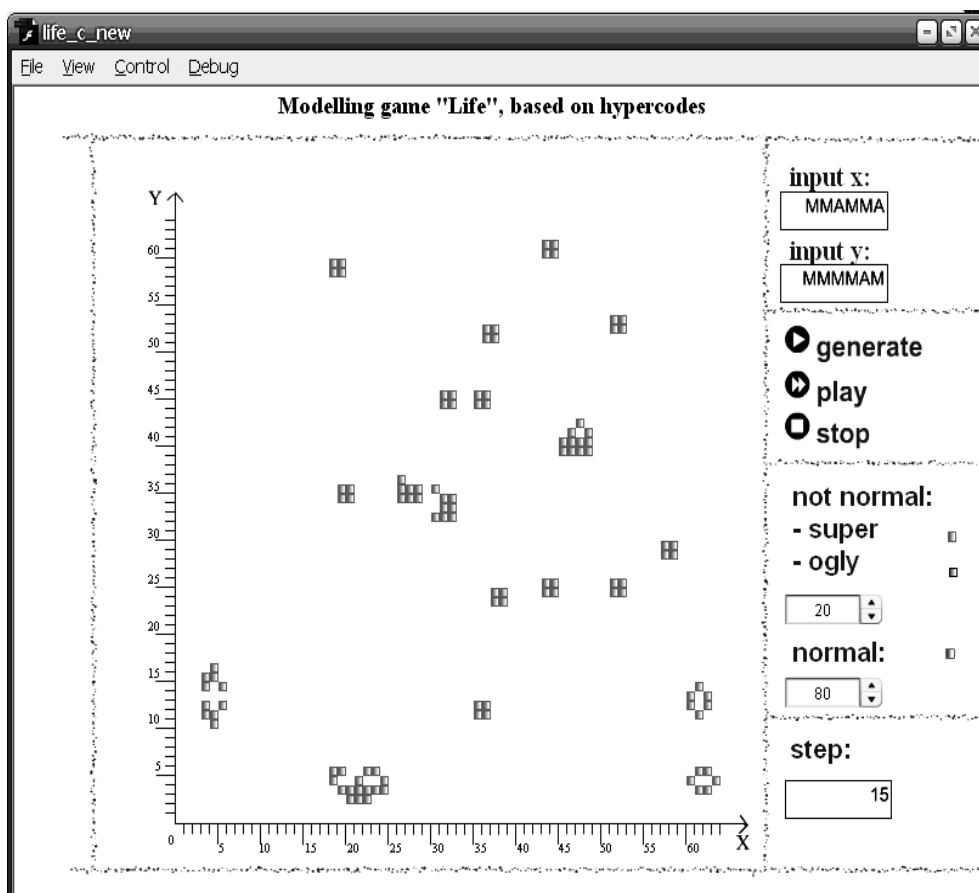


Рисунок 1 — Интерфейс приложения для моделирования ПКА в среде Adobe Flash

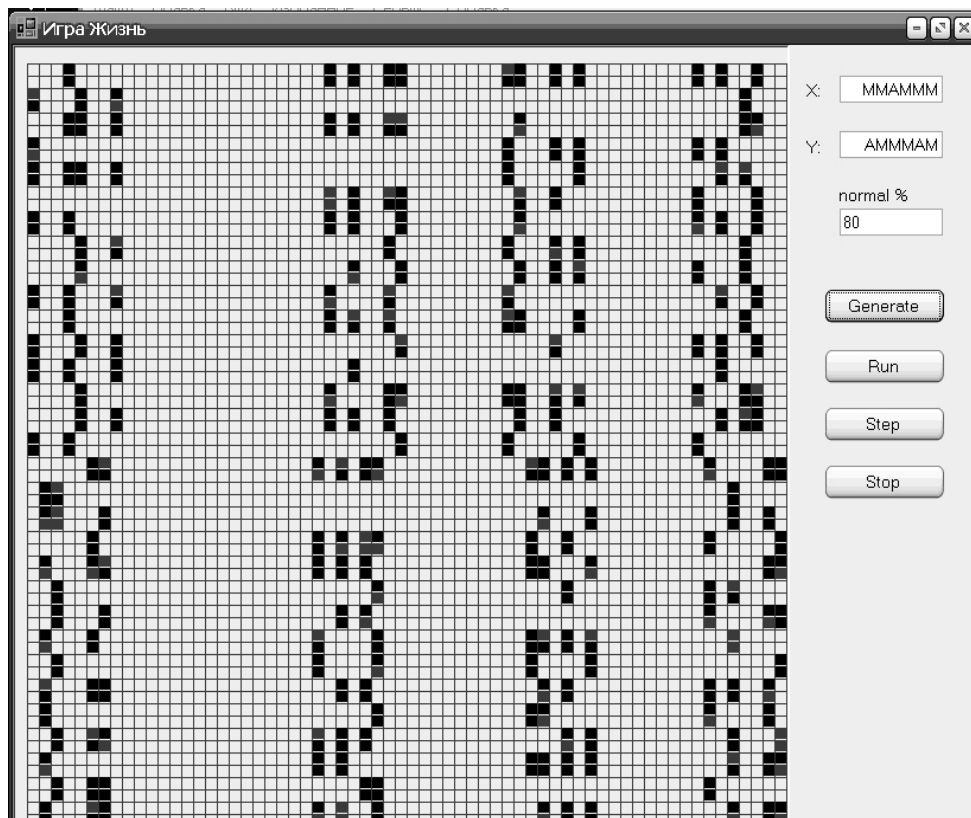


Рисунок 2 — Интерфейс приложения для моделирование ПКА на языке С++

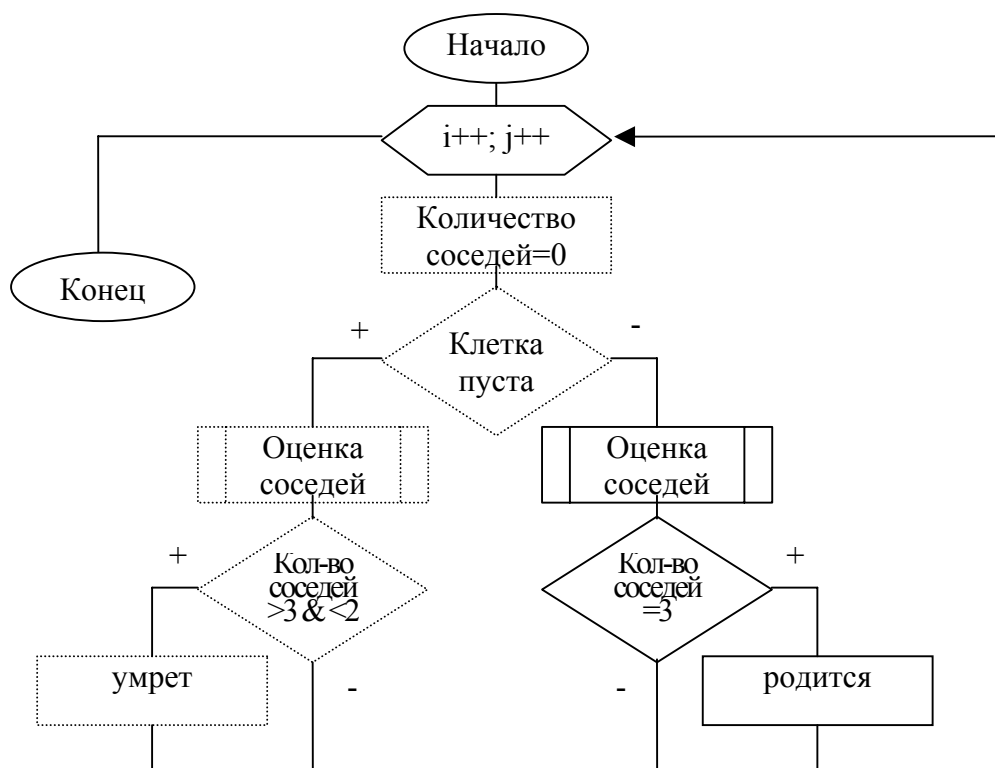


Рисунок 3 — Алгоритм реализованной игры «Жизнь» Конвея (алгоритм 1)

В таблице 1, в частности, представлены затраты времени на один шаг моделирования в среде Adobe Flash при генерации массива клеток размером 256 (2<sup>8</sup>), 512 (2<sup>9</sup>), 1024 (2<sup>10</sup>) и 2048 (2<sup>11</sup>)

элементов (при общем числе клеток 4096). Из таблицы видно, что, например, при 256-ти непустых клетках первой генерации среднее время вычисления одного шага составляет чуть более 0,2 с (при моделировании использовался компьютер на базе одноядерного микропроцессора Intel Pentium IV с тактовой частотой 1,6 ГГц), а при количестве непустых клеток 1024 среднее время вычисления шага составляет уже почти 0,4 с (рис. 4).

Очевидно, что и при увеличении размеров поля, и при увеличении количества сгенерированных клеток первой генерации быстродействие программы в целом будет существенно снижаться.

Таблица 1 — Зависимость затрат времени (в мс) на моделирование одного шага при различном количестве непустых клеток на рабочем поле размерностью 64x64. Алгоритм 1

Кол-во клеток / № шага	2 <sup>8</sup>	2 <sup>9</sup>	2 <sup>10</sup>	2 <sup>11</sup>
1	316	339	495	607
2	214	261	368	585
3	224	256	385	560
...	...	...	...	...
13	232	264	384	480
14	219	272	408	496
15	236	272	408	465
<b>Среднее время вычисления одного шага (мс):</b>	<b>234</b>	<b>270</b>	<b>396</b>	<b>519</b>

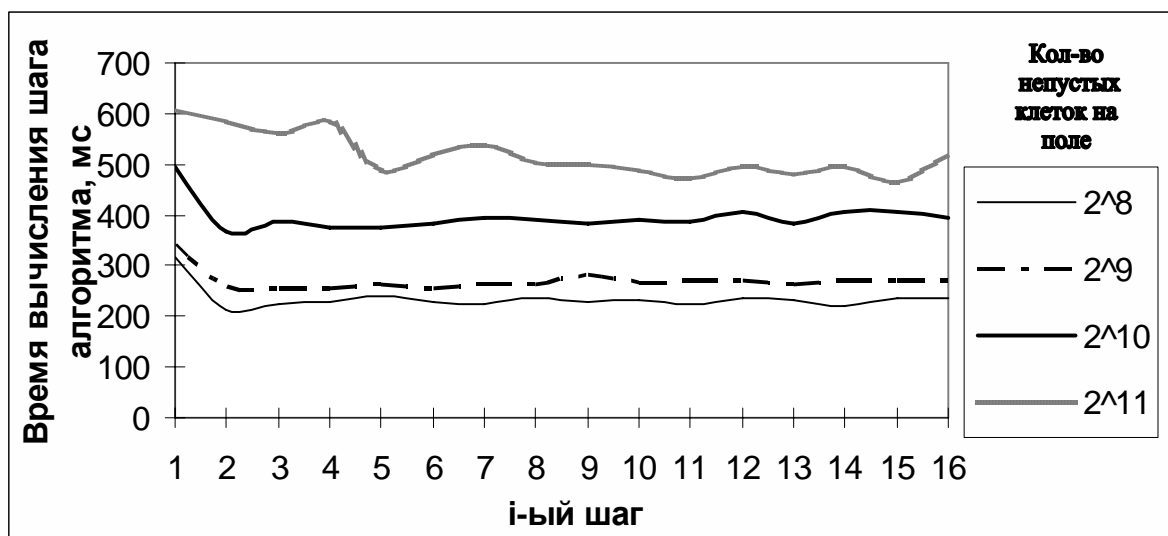


Рисунок 4 — Зависимость времени выполнения одного шага программы моделирования в зависимости от количества непустых клеток поля и номера шага (Алгоритм 1)

### Способы ускорения работы ПКА на уровне алгоритма

На рисунках 6 и 7 представлен реализованный алгоритм подпрограммы оценки соседей для каждой клетки, который является одним из наиболее трудоемких процессов алгоритма моделирования в целом. Каждая условная и операторная вершина в граф-схеме алгоритма — это, фактически, время, затрачиваемое процессором. Следовательно, чем меньше вершин, тем быстрее может осуществляться моделирование.

Первым и наиболее простым способом ускорения данного алгоритма является расширение объема памяти, выделяемой для хранения состояний клеток. Пусть, например, исходное поле имеет размерность 4x4. Тогда для того, чтобы снизить затраты

времени на контроль размерности поля (что даст возможность уменьшить количество условных вершин, а это может быть весьма существенно, учитывая количество итераций) можно, например, добавить граничные «пассивные клетки» (рис. 5). При этом объем памяти для хранения состояний ПКА увеличится, но размер рабочего поля КА останется прежним, а проверки в соответствии с алгоритмом будут осуществляться только для клеток рабочего поля.

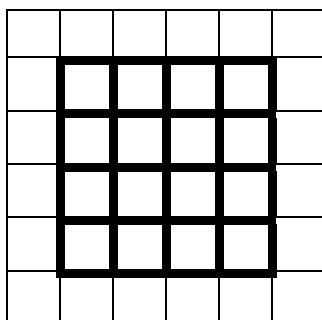


Рисунок 5 — Добавление граничных «пассивных клеток» — один из путей ускорения алгоритма путем расширения объемов памяти

Соответствующая модификация алгоритма представлена на рисунках 6 и 7.

Вторым способом ускорения алгоритма является уменьшение количества операторных вершин.

Для этого следует пересмотреть исходный алгоритм. При  $i$ -ом просмотре в нем осуществлялась проверка соседей для каждой клетки, вне зависимости от того, пустая она или нет. Это приводило к дополнительным затратам процессорного времени. В усовершенствованном алгоритме предлагается оценивать только соседей  $i$ -ой непустой клетки. А это существенно ускорит моделирование для некоторых частных случаев реализации. Граф-схема такого алгоритма приведена на рисунке 8.

Оценка затрат времени, требуемого на реализацию рассматриваемого алгоритма при различных условиях приведена на рисунке 9. Сравнительный анализ средних затрат времени для различных алгоритмов (рис. 4 и 8) позволяет сделать следующие выводы:

- алгоритм 1 дает преимущество по быстродействию для случаев, где количество клеток первой генерации  $\geq 2^{10}$ ;
- алгоритм 2 более эффективен по быстродействию для случаев, когда количество клеток первой генерации  $< 2^{10}$ .

В целом, наиболее эффективным по быстродействию будет вариант совмещения двух алгоритмов. Это легко достигается путем анализа входного значения, заданного в формате гиперкода. Для этого можно подсчитывать количество значений  $M$  в координатах  $X$  и  $Y$  первой генерации. Если оно превышает 9, то выполняется алгоритм 1, иначе — алгоритм 2. Это позволит сократить затраты процессорного времени в среднем в 1,5 раза.

#### **Ускорение моделирования ПКА на основе распараллеливания**

Еще одним способом решения задачи ускорения моделирования ПКА может быть распараллеливание. При этом возможно использование нескольких вариантов распараллеливания.

- распределение моделирования задач с различными входными данными на различные ресурсы;
- распределение одного процесса моделирования на различные ресурсы.

Во втором случае сложно обеспечить достаточную эффективность распараллеливания ввиду необходимости довольно интенсивного обмена информацией на каждом шаге моделирования. Поэтому на данном этапе наиболее перспективным представляется первый вариант распараллеливания.

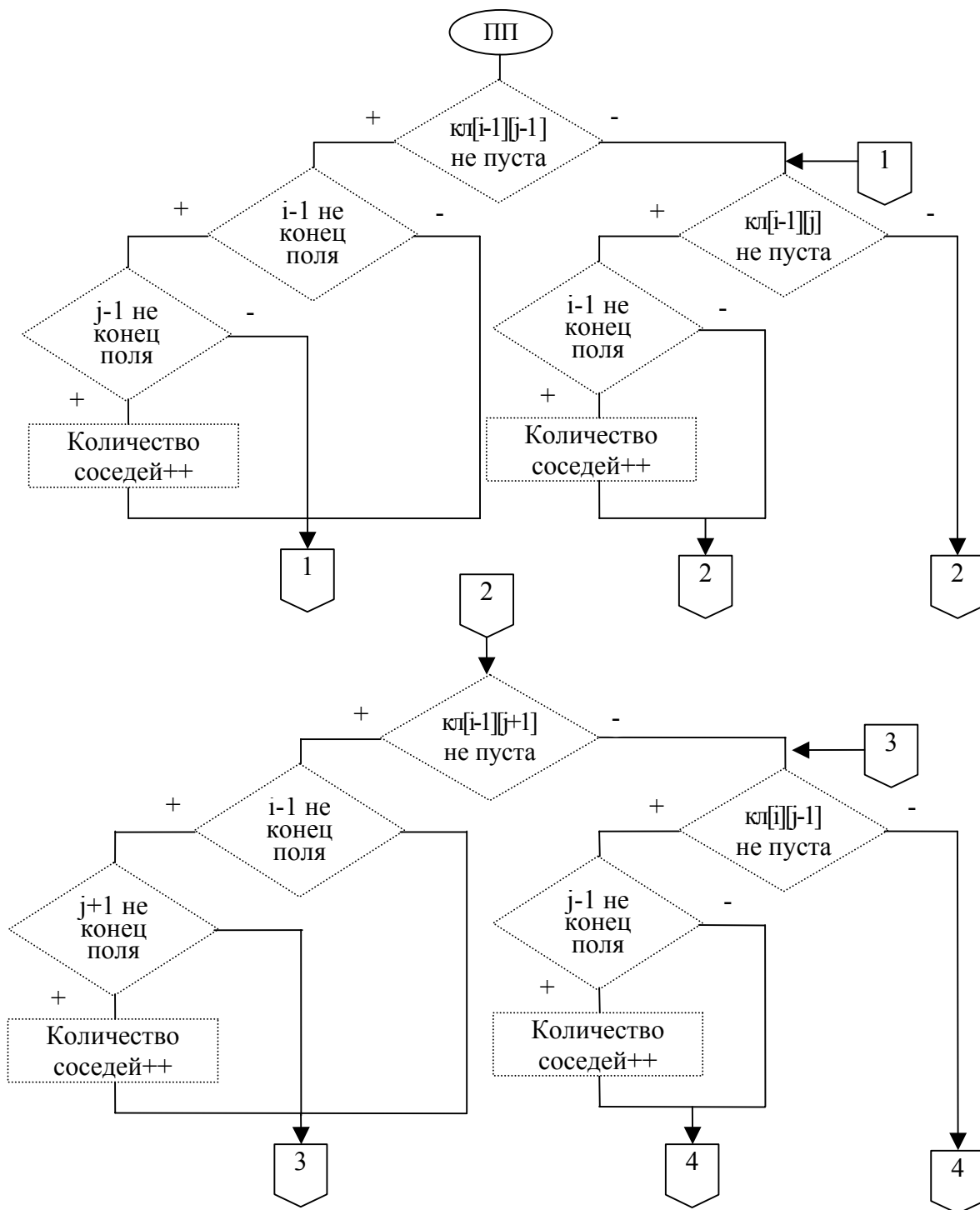


Рисунок 6 — Алгоритм подпрограммы оценки соседей

В процессе моделирования для игры «Жизнь» характерны следующие конечные ситуации:

- полное исчезновение клеток;
- установление статичных конфигураций;
- переход к динамическому состоянию от первоначального расположения клеток.

Задача заключается в том, чтобы определить, в какое конечное состояние установится КА. В простейшем случае задача решается путем полного перебора всех комбинаций (как NP-сложная задача).

Предполагается, что входными параметрами при моделировании являются числа в формате гиперкода (координаты X и Y). Для примера возьмем 6-ти разрядные гиперкоды. Координаты выражаются при этом следующим образом:

$$X=xxxxxx$$

$$Y=yyyyyy$$

где x и y могут принимать одно из 4 значений 0, 1, A, M.

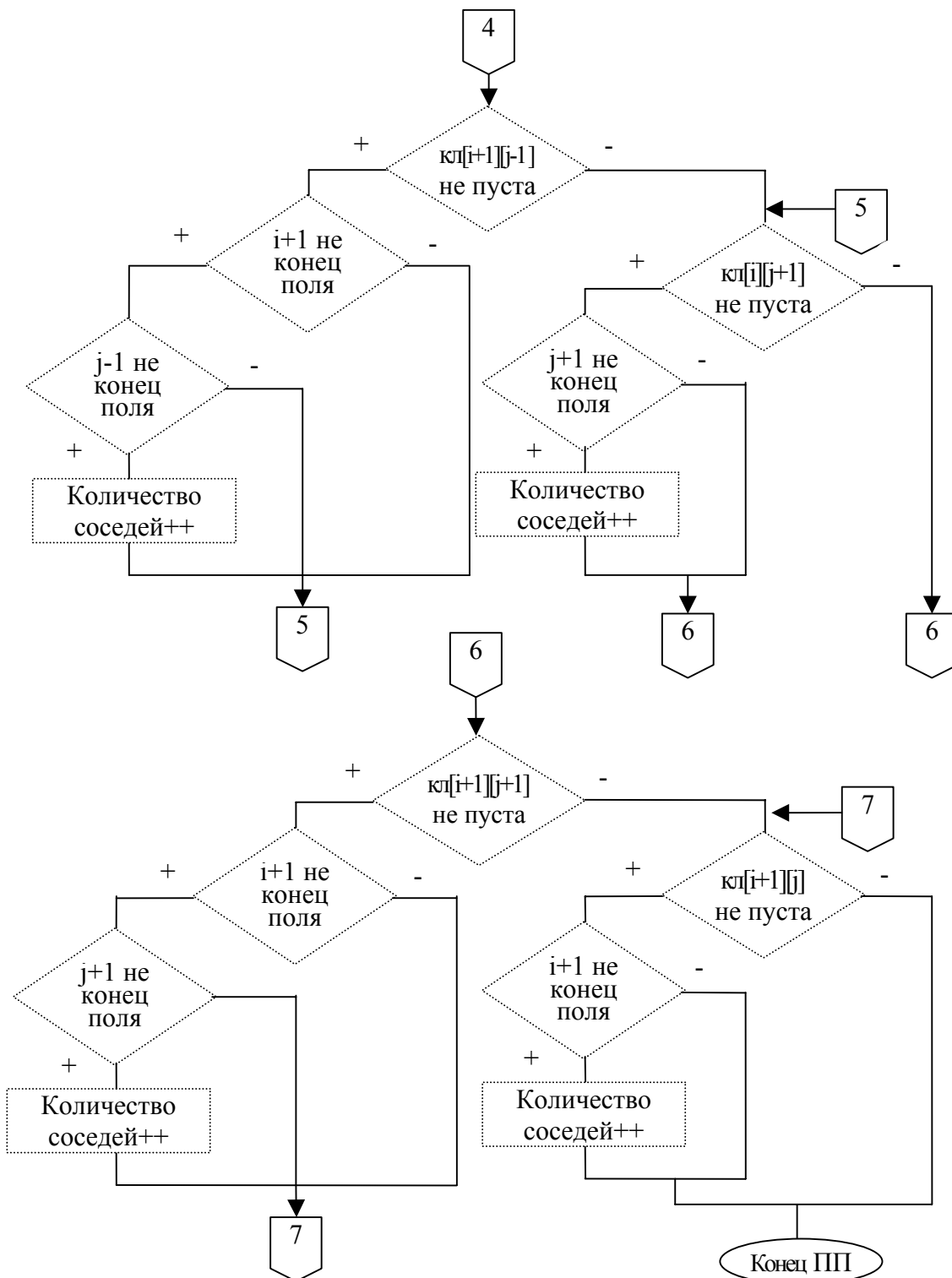


Рисунок 7 — Алгоритм подпрограммы оценки соседей (продолжение)

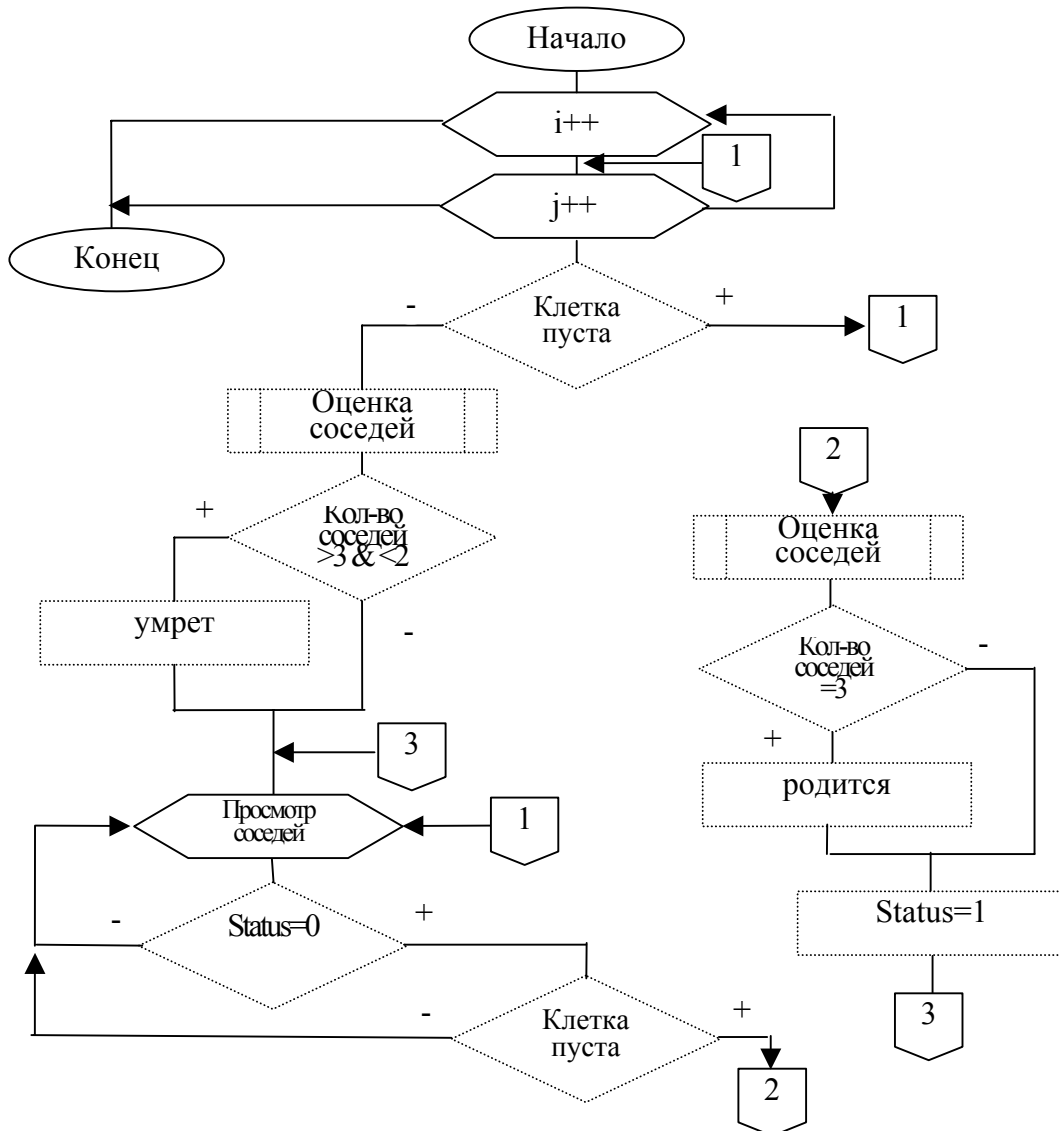


Рисунок 8 — Модифицированный алгоритм моделирования ПКА (алгоритм 2)

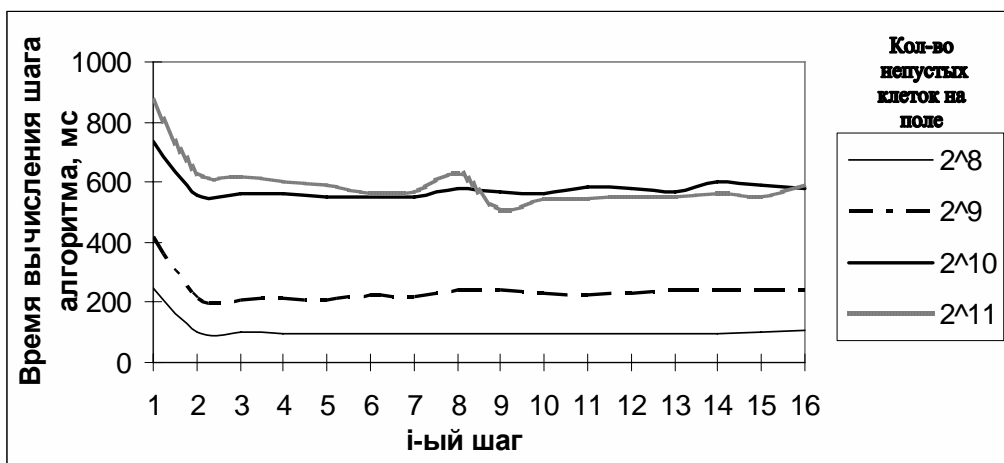


Рисунок 9 — Зависимость времени выполнения одного шага программы моделирования в зависимости от количества непустых клеток поля и номера шага (Алгоритм 2)



Общее число возможных комбинаций можно представить следующим образом:

$$N = 4^6 * 4^6 = 16\,777\,216 \text{ возможных значений.}$$

Общая формула имеет следующий вид:

$$N = 4^{N_x * N_y}$$

где,  $N_x$  и  $N_y$  — разрядность координаты  $X$  и  $Y$  соответственно.

Одной из нетривиальных задач является определение условий завершения моделирования, т.к. в некоторых случаях мы в итоге получаем динамическое состояние системы с довольно сложными периодически повторяющимися комбинациями. Это наиболее сложный вариант. В этом случае в качестве результата моделирования может рассматриваться количество шагов, необходимое для установления автомата в циклическое состояние, и описание конечного состояния ПКА. При этом наиболее целесообразным представляется описание состояния ПКА в виде комбинации гиперкодов, что позволит передавать информацию о результатах в достаточно компактном виде. Для этого необходимо выполнить преобразование состояния клеточного поля в описание в виде совокупности гиперкодов.

### Заключение

Повышение быстродействия моделирования ПКА на базе гиперкодов может быть достигнуто 2-мя основными способами: совершенствованием алгоритмов и распараллеливанием моделирования. В данной статье предложен усовершенствованный комбинированный алгоритм, который в определенных ситуациях использует преимущества алгоритма 2, суть которого изложена выше. Данный алгоритм дает улучшение показателей быстродействия в некоторых случаях (когда количество клеток на поле не превышает  $2^8$ ) более чем в 1,5 раза. Проанализированы также возможности распараллеливания моделирования ПКА, реализация которых планируется на последующих этапах проведения исследований.

### Литература

1. Жизнь (игра). Материал из Википедии — свободной энциклопедии ([http://ru.wikipedia.org/wiki/Жизнь\\_\(игра\)](http://ru.wikipedia.org/wiki/Жизнь_(игра)))
2. Аноприенко А.Я. Обобщенный кодо-логический базис в вычислительном моделировании и представлении знаний: эволюция идеи и перспективы развития // Научные труды Донецкого национального технического университета. Серия: Информатика, кибернетика и вычислительная техника (ИКВТ-2005) выпуск 93. — Донецк: ДонНТУ, 2005. — С. 289–316.
3. Аноприенко А.Я., Коноплева А.П. Опыт применения гиперкодов в моделировании клеточных автоматов // Научные труды Донецкого национального технического университета. Серия "Проблемы моделирования и автоматизации проектирования динамических систем" (МАП-2007). Выпуск 6 (127). — Донецк: ДонНТУ, 2007. — С. 220–227.
4. Аноприенко А.Я., Коноплева А.П. Развитие идеи применения гиперкодов в моделировании клеточных автоматов // Научные труды Донецкого национального технического университета. Серия: Информатика, кибернетика и вычислительная техника (ИКВТ-2008) выпуск 93. — Донецк: ДонНТУ, 2008. — С. 289–316.
5. Аноприенко А.Я., Святный В.А. Высокопроизводительные информационно-моделирующие среды для исследования, разработки и сопровождения сложных динамических систем // Научные труды Донецкого государственного технического университета. Выпуск 29. Серия "Проблемы моделирования и автоматизации проектирования динамических систем". — Севастополь: «Вебер». — 2001. — С. 346–367.

Здано в редакцію:  
12.03.2009р.

Рекомендовано до друку:  
д.т.н, проф. Башков Є.О.