

## THE USE OF MULTI-HEURISTIC APPROACH WHEN PROGRAMMING SOLUTIONS PUZZLES

© 2012

*A.A. Borgardt*, student

*A.M. Lysenko*, postgraduate student

*B.F. Melnikov*, doctor of physics and mathematics sciences, professor  
*Togliatti State University, Togliatti (Russia)*

*Keyword:* Bin packing problem; heuristics most touching; intractable problems; genetic algorithm

*Annotation:* The article discusses some solutions of discrete optimization based Multi-heuristic approach. This paper presents an approach to algorithmic Multi-heuristic solving puzzles. They have a finite number of situations and clearly defined rules. On the example of a three-dimensional Mahjong and Tetris.

УДК 004.456.45, 004.414.23

## МНОГОПОТОЧНЫЙ АЛГОРИТМ МОНТЕ-КАРЛО МОДЕЛИРОВАНИЯ РОСТА НАНОКРИСТАЛЛОВ

© 2012

*A.A. Borgardt*, студент

*Ю.С. Нагорнов*, кандидат физико-математических наук, старший научный сотрудник, доцент  
кафедры «Высшая математика»

*Тольяттинский государственный университет, Тольятти (Россия)*

*Ключевые слова:* моделирование методом Монте-Карло; рост нанокристаллов; многопоточный алгоритм; сетевое хранилище данных.

*Аннотация.* В статье рассмотрен однопоточный алгоритм на основе метода Монте-Карло, позволяющий моделировать образование нанокристаллов карбида кремния. С целью совершенствования процесса моделирования был разработан и апробирован алгоритм на основе принципа многопоточности. В работе также представлен алгоритм корректировки и синхронизации всех спорных точек поверхности.

### ВВЕДЕНИЕ

Для моделирования процессов, происходящих во время роста кристаллов, их отжига, выращивания различных пленок и соединений на кристаллах широко используют метод Монте-Карло (МК) [1 - 6]. Преимуществами метода МК являются простота получаемых моделей и высокая производительность алгоритмов, что в свою очередь позволяет исследовать значительные по размерам объекты.

Физическая модель кинетической вариации метода Монте-Карло роста кристалла сводится к введению понятий событий или элементарных процессов реализуемых в системе. Каждое событие в реальной системе происходит с заданной вероятностью, при этом величина, равная числу событий в единицу времени является темпом данного процесса. Пусть  $N$  возможное число всех событий в данной конфигурации системы  $C$  в мо-

мент времени  $t$ , а темпы процессов  $R_a$ , где  $a=1..N$ , тогда полный темп определяется выражением:

$$Q = Q(C) = \sum_{a=1}^N R_a V^a(C \rightarrow C'), \quad (1)$$

где  $V^a(C \rightarrow C')$  — стохастическая матрица, определяющая возможность перехода системы из состояния  $C$  в  $C'$  путем реализации события  $a$ . Соответственно с этим, вероятность изменения конфигурации системы путем реализации события  $a$  равно  $R_a/Q(C)$ . Таким образом, элементарная реализация метода Монте-Карло заключается в том, что в каждый момент времени рассчитывается темп каждого элементарного события в системе, определяется их вероятность, в соответствии с которыми и выбирается выполняемое событие.

Основными рассматриваемыми типами процессов в моделируемой системе являлись стандартные события

адсорбции на поверхность пористого кремния, поверхностной диффузии, в результате которого атом переходит на одно из свободных соседних мест, объемной диффузии, учитываемой по обменному механизму атома кремния с углеродом.

**Реализация алгоритма**

С целью реализации программы был выбран язык Java, что объясняется тем, что жёстко он не привязан ни к одной из современных платформ. Размерность поверхности задается выражением  $N*N$ , где  $N$  – количество кубиков по горизонтали решётки,  $N > 10$ .

Рассмотрим однопоточный алгоритм моделирования роста нано кристалла:

1. Начальная инициализация системы. На этом этапе происходило создание кристалла кремния, установка параметров системы (температура, пористость, давление газа), составление списка событий, распределение событий по группам, подсчет суммарного темпа процессов.

2. Основной шаг моделирования. Выбор группы событий в соответствии со схемой Максима [7], равновероятный выбор конкретного реализуемого процесса, реализация процесса, пересчет вероятностей, сбор статистики о системе.

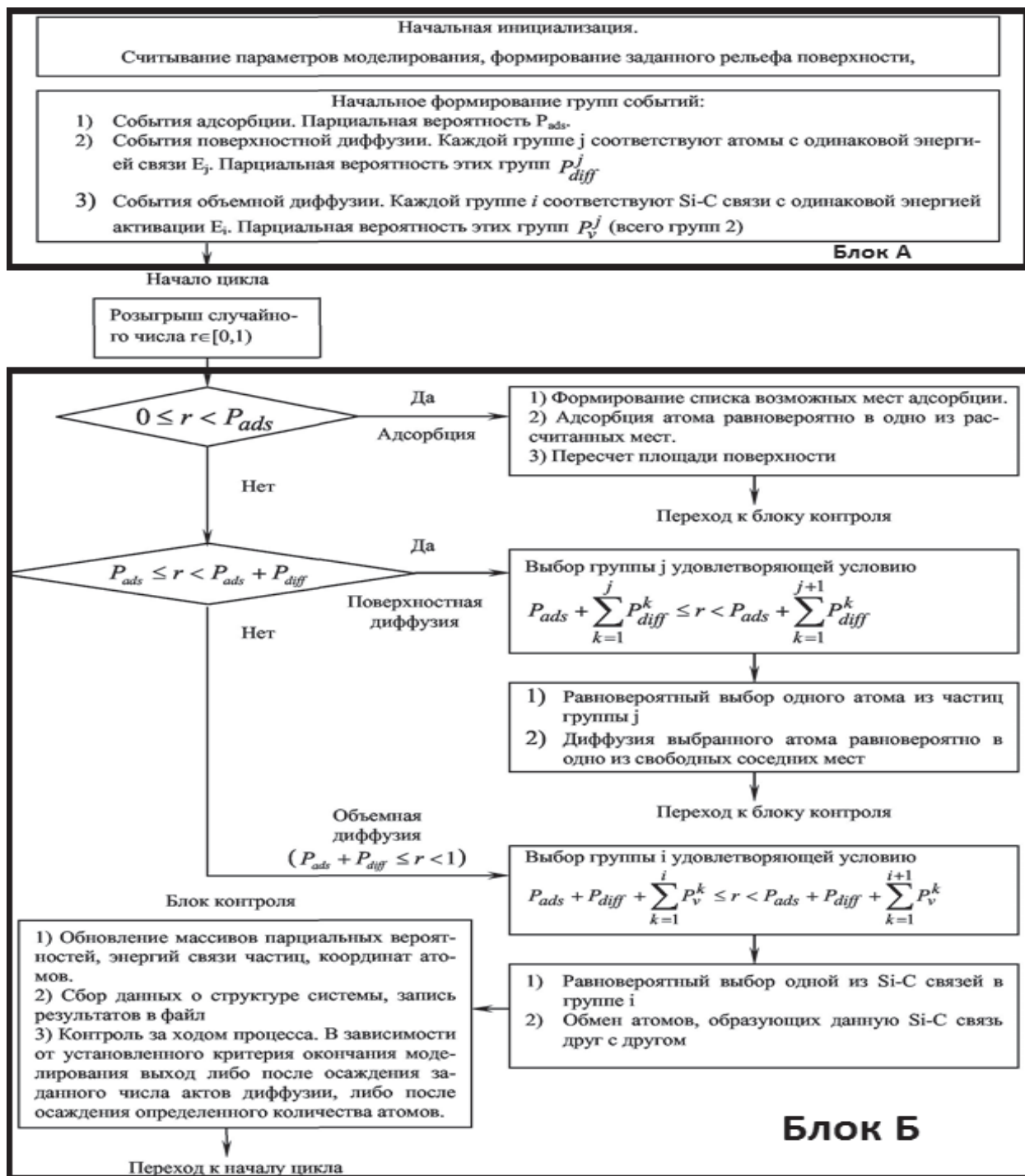


Рис. 1. Блок-схема однопоточного алгоритма моделирования процесса карбонизации методом Монте-Карло.

Апробация однопоточного алгоритма моделирования процесса карбонизации методом Монте-Карло[8] выявила существенные недостатки: значительные затраты времени на работу программы и потери данных в случае непредвиденных перебоев в сети. В связи с этим был разработан алгоритм на основе принципа многопоточности.

Ход программы:

1. Создание потоков, каждый из которых моделирует процесс карбонизации методом Монте-Карло.
2. Перенос информации из оперативной памяти базы данных Redis в базу данных Redis на жёстком диске.

Этапы реализации многопоточного алгоритма:

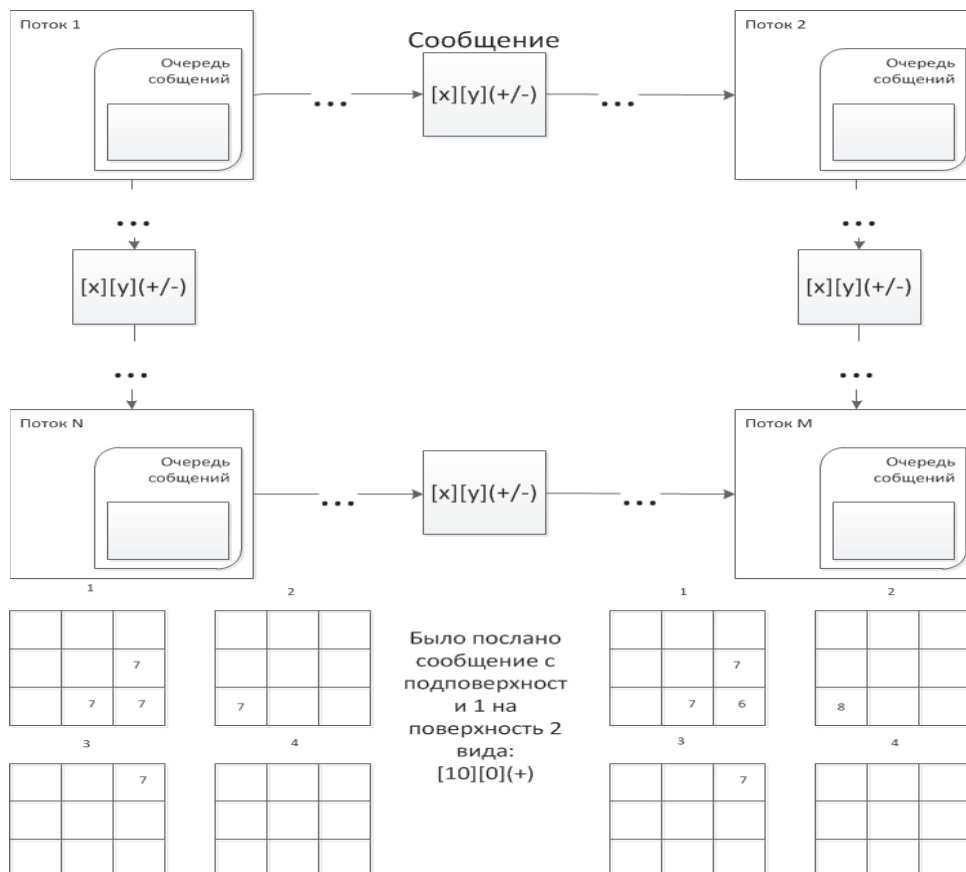
1. Начальная инициализация системы: создание кристалла кремния, установка параметров системы (температура, пористость, давление газа), составление списка событий, распределение событий по группам, подсчёт суммарного темпа процессов.
2. Оценка размера поверхности кристалла кремния вычисляется и переносится в БД. Происходит разбиение поверхности на подповерхности размером 10\*10, что упрощает вычисления. Так же основная поверхность сохраняется и принудительно выгружается из памяти в БД на жёстком диске.
3. В системе конфигурирует оптимальное число потоков, которые вычисляют все подповерхности. Число потоков ограничено верхним максимальным значением. Это значение равно числу ядер процессора, помноженное на два. Все под поверхности хранятся в записях БД.
4. Основной шаг моделирования. Выбор группы событий в соответствии со схемой Максима, равновероятный выбор конкретного реализуемого процесса, реализация процесса. Когда события происходили на стыке подповерхностей, отправляется сообщение,

говорящее, что после завершения вычислений в данном секторе необходимо будет внести изменения. По необходимости запускается алгоритм Корректировка и синхронизация поверхности. На каждом шаге просматривается статистика о системе, эти сведения записываются в лог-файл базы данных.

5. Обновление данных. Все события базы просматриваются, вероятности пересчитываются, данные нарезаются в соответствие с расположением на поверхности и рассылаются по всей системе. Кроме этого посылаются случайное новое число  $t \in [0,1]$  одинаковое на всей системе.

После «Основного шага моделирования» на всех подповерхностях запускается алгоритм корректировки и синхронизации всех спорных точек поверхности, а затем уже и начинается следующая итерация. Изменение значения переменной в заданных координатах одной подповерхности, выходящее за ее границу, влечёт за собой формирование сообщения, содержащее необходимую информацию для изменения рельефа и массива энергии связей соседней подповерхности. Сообщения хранятся в очереди до конца выполнения «Основного шага моделирования» данной подповерхности. Перед началом следующей итерации вносятся все изменения, содержащиеся в сообщении. Этот способ достаточно прост и гибок, позволяет сократить время вычислений, которое в свою очередь зависит от размера и количества подповерхностей. Сообщение имеет вид:  $[x][y](-/+)$ , где  $[x][y]$  – место, в котором должно быть изменение: добавление или удаление высоты на массиве координат атомов.

На рис. 2 отражаются процессы взаимодействия потоков (подповерхностей) через сообщения и пример работы алгоритма.



**Рис. 2.** Алгоритм синхронизации подповерхностей и точек на краях.

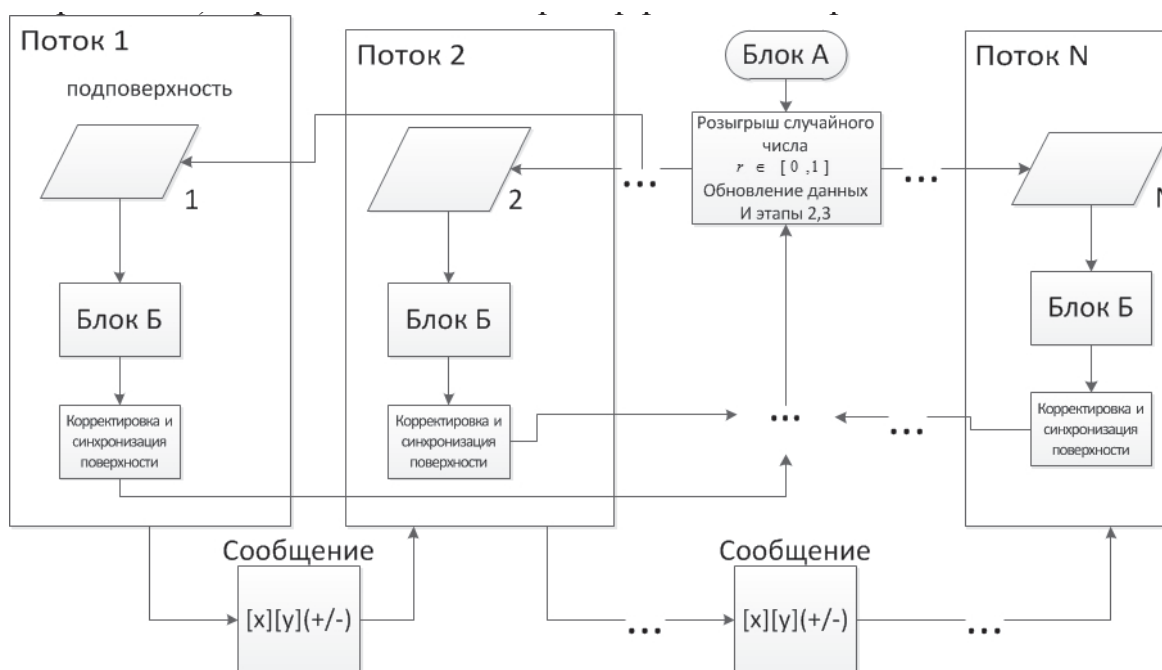


Рис. 3. Блок-схема многопоточного алгоритма моделирования процесса карбонизации методом Монте-Карло.

Базу данных можно представить как бесконечно расширяемую поверхность кристалла. Хранение данных происходит в базе данных Redis. Redis был собран и установлен по рекомендованным настройкам производителя. Redis – это документо-ориентированное сетевое хранилище данных типа «ключ-значение». В целях сохранения данных ведётся журнал, по которому их можно восстановить в случае повреждения оперативной памяти. Преимущества использования указанного способа хранения заключается в том, что он позволяет делать снимки мгновенно, передавать данные из оперативной памяти на диск асинхронно и обеспечивает полустойкость долговечного режима. Дабы избежать переполнения журнала, Redis добавляет новые файлы только в фоновом режиме. Каждая запись данных хранит в себе следующие поля:

1. температура;
2. пористость;
3. давление газа;
4. number – номер под поверхности нужен для дальнейшей склейки;
5. size – размер под поверхности;
6. under\_the\_surface – сама подповерхность.

Under\_the\_surface : массив координат атомов, массив парциальных вероятностей, массив энергий связи частиц.

**Апробация и проверка предположения**

Тестирование проводилось на материнской плате GIGABYTE GA-P55A-UD3 с Intel Core I 5 2,60 ГГц 4 Гб RAM. На операционной системе Ubuntu 12.04 server amd64. Java SE 7 update 7. Все тесты запускались на дефолтных настройках java машины с ключем –server. Redis был собран по конфигурации разработчиков.

Диаграмма на рис. 4 свидетельствует, что время работы многопоточного алгоритма моделирования процесса карбонизации методом Монте-Карло, по сравнению с однопоточным, сокращается примерно в 2–3 раза.

Объем оперативной памяти, используемой для работы многопоточного алгоритма, более чем в 2 раза превышает необходимый объем памяти для реализации однопоточного, что подтверждает диаграмма на рис. 5.

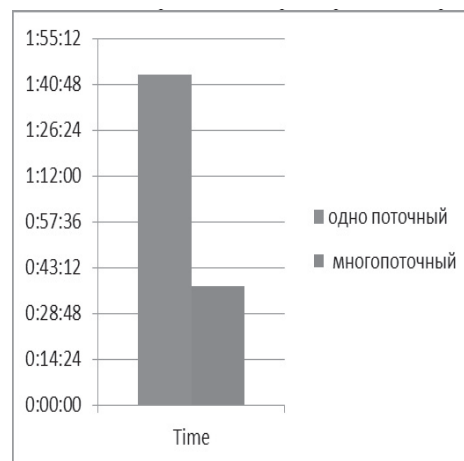


Рис. 4. Время работы однопоточного и многопоточного алгоритма моделирования процесса карбонизации

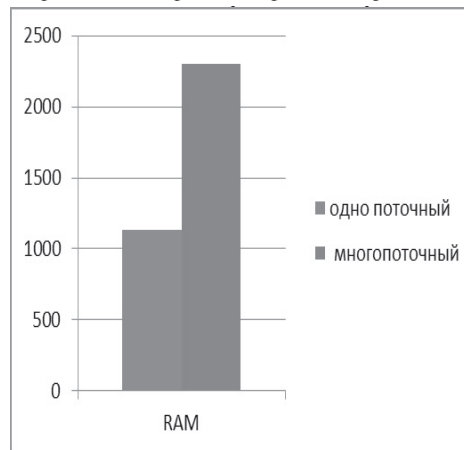


Рис. 5. Объем оперативной памяти однопоточного и многопоточного алгоритма моделирования процесса карбонизации.

#### ЗАКЛЮЧЕНИЕ

Таким образом, в данной работе был рассмотрен однопоточный алгоритм численного моделирования нанокристаллов карбида кремния методом Монте-Карло в процессе высокотемпературной карбонизации. Реализация этой программы позволила выяснить, что скорость вычисления не достаточно высокая. С целью устранения этого недостатка предложено использовать принцип многопоточности, что позволило ускорить реализацию алгоритма в 2–3 раза. С целью повышения безопасности системы рекомендовано применить Redis, документо-ориентированное хранилище данных.

*Работа выполнена при поддержке ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009–2013 годы (соглашение № 14.В37.21.0222).*

*Работа первого автора частично поддержана программой Министерства образования и науки в рамках Госзадания Тольяттинского государственного университета на 2012 год.*

*Работа второго автора частично поддержана Федеральной целевой программой «Научные и научно-педагогические кадры инновационной России» на 2009–2013 годы.*

#### СПИСОК ЛИТЕРАТУРЫ

1. Frenkel D. Understanding Molecular simulation from algorithms to applications. Academic Press. 2002.
2. Heerman D.W. Computer simulation methods in theoretical physics. Second edition. New York. 1990.
3. Fishman G.S. Monte Carlo : concepts, algorithms, and applications. NewYork. 1995.
4. Ke S.C., DeLucas L.J., Harrison J.G. Computer simulation of protein crystal growth using aggregates as the growth unit // J. Phys. D: Appl. Phys. 1998, V.31, P. 1064–1070.
5. Cavallotti C. et. al. Multiscale simulation of silicon film growth // Cryst.Res. Technol. 2005. V. 40. № 10–11. P. 958
6. Miller R.S. et. al. Monte Carlo simulation of three-dimensional noniso-thermal grain-microstructure evolution: application to LENS rapid fabrication // Journal of Materials Synthesis and Processing. 2001. V. 9. № 6. P. 329.
7. Maksym P.A. Monte Carlo simulation of III-V MBE growth // Semicond. Sci. Tech. 1998. No.3. 594-599.
8. Fishman G.S. Monte Carlo: concepts, algorithms, and applications. NewYork. 1995.

## MULTITHREADING MONTE CARLO ALGORITHM OF MODELING NANOCRYSTAL GROWTH

© 2012

*A.A. Borgardt*, student

*Yu.S. Nagornov*, candidate of physical and mathematical sciences, senior researcher,  
associate professor of the chair «Higher Mathematics»

*Togliatti State University, Togliatti (Russia)*

*Keyword:* Monte Carlo simulation; network database; nanocrystal growth; multithreading algorithm.

*Annotation:* The article presents nanocrystal growth using Monte Carlo single-flow method. The main goal was to improve process of modeling. For that purpose we split nanocrystal surface into peaces and make multithreading calculations. The computation of general points between peaces involves adjustment and synchronization algorithm. For the further research all data is stored in network database.