ՀԱՅԱՍՏԱՆԻ ԳԻՏՈՒԹՅՈՒՆՆԵՐԻ ԱԶԳԱՅԻՆ ԱԿԱԴԵՄԻԱ
НАЦИОНАЛЬНАЯ АКАДЕМИЯ НАУК АРМЕНИИ
NATIONAL ACADEMY OF SCIENCES OF ARMENIA
ДОКЛАДЫ ԶԵԿՈՒՅՑՆԵՐ REPORTS

107 2007 № 2

ИНФОРМАТИКА

УДК 616.65-006

В. С. Агабабов

# О выборе оптимальной структуры данных для представления опухоли при моделировании in vitro раковых заболеваний

(Представлено академиком С. А. Амбарцумяном 31/III 2007)

Ключевые слова: *структуры данных, алгоритмы, моделирование, рак, биология, биоинформатика*

1. Введение. Целью статьи является выбор структуры данных, которая наиболее оптимально подойдет к задаче моделирования роста, сжатия, а также лечения раковой опухоли. Особое внимание уделяется свободному росту опухоли и ее ответной реакции на радиотерапию, радиотерапию и гипертермию, а также цитокинетической модели. При исследовании моделей in vitro опухоль рассматривается на уровне отдельных клеток, а когда опухоль достигает макроскопических размеров, то применяется модель in vivo, в которой моделирование опухоли на уровне отдельных клеток уже выходит за рамки возможностей современных компьютеров. Большинство авторов рассматривают всего три состояния клетки: активное, спящее и некротическое. В нашей работе разделение более детальное, дающее возможность моделировать рост и сжатие опухоли более точно и соответственно процессам, происходящим в организме, и позволяет получить более корректные результаты.

2. Краткое описание клеточной модели раковой опухоли. Для эффективного и точного моделирования раковой опухоли необходимо иметь соответствующую модель отдельных клеток. В статье рассматривается цитокинетическая модель, обладающая следующими свойствами:

1. модель является детерминированной, с точки зрения последовательности переходов из одного состояния в другое;

2. модель является недетерминированной (случайной), с точки зрения

продолжительности времени, которое клетка проводит в каждом из состояний. В данной работе применяется нормальное распределение. Следует учесть, что для разных типов опухолей среднее время провождения клеток в каждом из состояний разное, что является одним из параметров моделирования;

3. выбор следующего состояния клетки зависит помимо состояния самой клетки и состояния клеток, находящихся по соседству, также от геометрического положения клетки в опухоли.

Клетки, которые моделируются в соответствии с данной моделью, имеют следующие фазы:

1. G0 (Gap 0) - гипоксичное спящее состояние. Клетка попадает в данное состояние после митоза, если питания недостаточно для последующего деления;

2. G1 (Gap 1) - спящее состояние, в которое клетка попадает после митоза, если питания хватает для следующего цикла деления;

3. S (Synthesis)- фаза синтеза ДНК, в результате которого в каждой хромосоме появляются две идентичные молекулы ДНК. Повреждения и мутации в ДНК обычно бывают именно во время этого состояния;

4. G2 (Gap 2) - спящее состояние, в которое клетка попадает после синтеза ДНК, перед митозом;

5. M (Mitosis) - митоз или деление клетки;

6. N (Necrosis) - некроз. Мертвое состояние, в которое попадают клетки после лечения. Основное отличие от апоптоза (A) определяется интервалом времени, которое необходимо организму на элиминацию погибшей клетки из организма. В случае некроза это время в несколько раз больше, что является одним из параметров модели;

7. A (Apoptosis) - апоптоз. Это состояние, в которое клетки попадают, когда умирают в результате программируемой гибели. Обычно в данном случае процесс лизиса (удаления клетки из организма) происходит достаточно быстро.

Состояния G1, S и G2 принадлежат так называемой интерфазе и в большинстве работ рассматриваются как одно состояние.

Автомат, описывающий состояния клеток и возможные переходы из состояния в состояние,показан на рис. 1. Штрихованными линиями показаны переходы в результате применения терапии.

В цитокинетической модели мы исходим из следующих допущений:

1. Для того, чтобы клетка после митоза перешла в очередной цикл деления, необходимы питание и кислород. Достаточность питания и кислорода рассчитывается исходя из расстояния от точки нахождения клетки до

ближайшей кромки опухоли. Для разных типов опухолей этот параметр является разным и представляется как входной параметр модели. Стоит пояснить, что питание и кислород доставляются клеткам по кровеносным сосудам, которые, однако, сквозь опухоли не прорастают, или имеют хаотическую структуру и имеют множество тупиков, или "протекают", вследствие чего центральные части опухоли всегда "голодают".

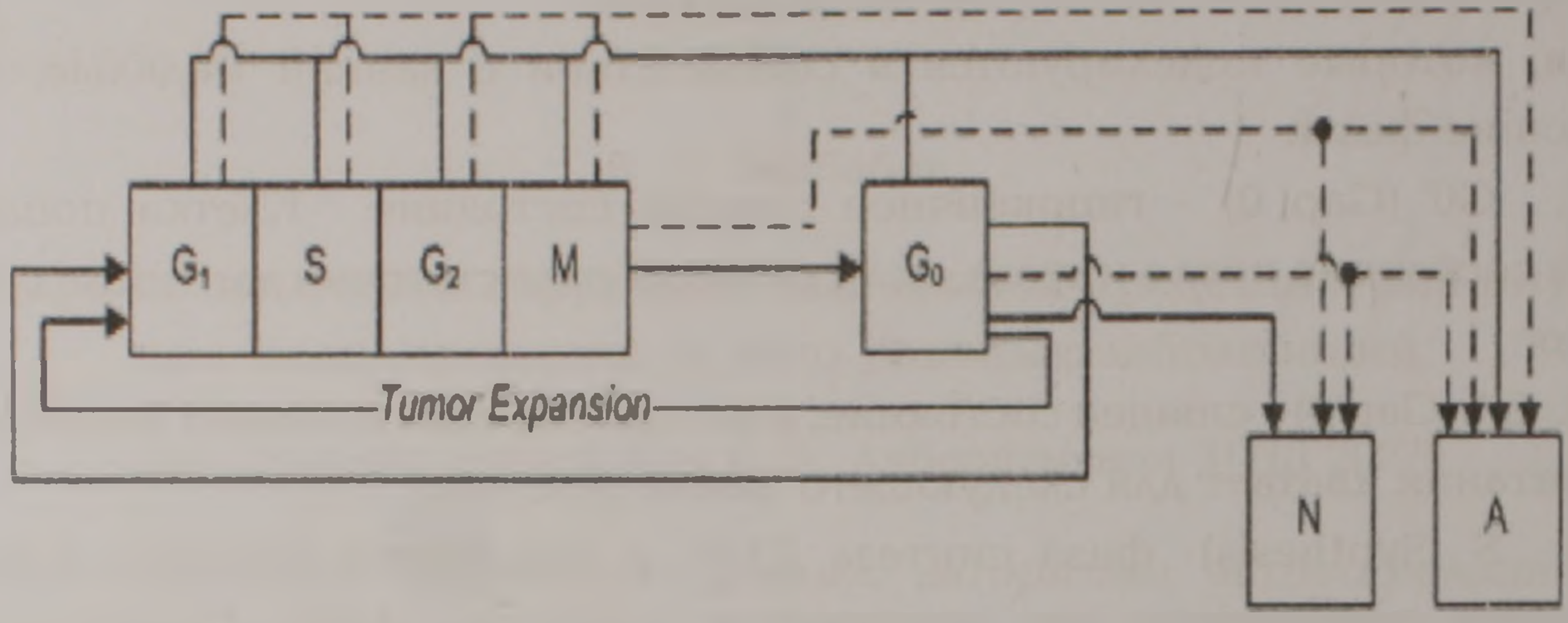


Рис.1.

2. Чувствительность клетки к различным способам лечения зависит от текущего состояния клетки. Например, при радиотерапии клетки в состоянии G0 являются наиболее резистентными, в то время как в случае гипертермии имеет место обратный эффект.

3. В любой заданный момент времени каждая из клеток может погибнуть с определенной вероятностью в результате программируемой смерти, старения и т.д.

4. Материнская и дочерняя клетки после митоза считаются абсолютно идентичными.

5. Каждая клетка обладает собственными часами для отсчета времени до следующего состояния.

Исходя из выше перечисленных допущений и постулатов мы можем сформулировать следующие требования к структуре данных с точки зрения клеточной модели:

1. возможность последовательного доступа ко всем клеткам для определения состояния клетки;

2. возможность удаления и добавления клетки к опухоли;

3. возможность расчета расстояния от точки нахождения клетки до ближайшего края опухоли, что необходимо для определения следующего состояния клетки после митоза.

Примененная терапия в нашей модели представляет из себя вычисление, которое производится на основе текущего состояния клетки, а также ее геометрического положения. В результате вычисления изменяется внутреннее состояние клетки. Например, она помечается как облученная. Для реализации данного процесса нам опять же необходим последовательный доступ к клеткам.

Дополнительными требованиями являются возможность сохранения структуры данных в постоянной памяти для дальнейшего анализа, а также возможность использования структуры для визуализации данных.

3. Анализ различных структур данных для моделирования раковых опухолей. Исходя из приведенных требований выбор структуры данных ограничивается следующими возможностями:

1. связанный список;
2. трехмерный статический массив;
3. трехмерный динамический массив.

Рассмотрим последовательно положительные и отрицательные стороны каждой из перечисленных структур. В анализе учтены следующие параметры: а) необходимая память, б) максимально возможный размер опухоли, в) скорость алгоритмов обхода, сжатия и расширения, г) скорость алгоритмов оценки геометрического положения клетки, д) удобство для сохранения на внешний носитель памяти и визуализации. Допускается, что опухоль имеет кубическую форму.

3.1. Связанные списки. Первой из возможностей является использование модифицированного связанного списка, который имел бы 6 указателей на своих соседей. Эта структура данных имеет то преимущество, что она не использует дополнительной памяти для несуществующих клеток, но проблемой является то, что при росте опухоли количество необходимой памяти для хранения связей, а именно расход памяти на цели, напрямую не связанных с моделированием опухоли, также растет пропорционально. Например, в нашей реализации память, необходимая для клетки, составляет 18 байт памяти (тут и дальше предполагается реализация на компьютерах с архитектурой IA-32). В то же время для хранения 6 указателей нам потребуется 24 байта памяти, в результате получается, что мы тратим больше памяти на указатели, чем на фактические данные. Итак, нам понадобится 42 байта для хранения одной клетки в памяти. В той же архитектуре IA-32 мы можем предоставить 2 гигабайта памяти под программу (при условии, что у нас есть столько памяти), что позволит нам одновременно хранить в памяти до 50 миллионов клеток, что представляет из себя куб с ребром в 370 клеток. У связанных списков есть также одно преимущество, которое в то

же самое время является слабым местом, - это возможность распределения клеток по непоследовательным адресам памяти. Преимущество заключается в том, что для расширения структуры нет необходимости в последовательном большом куске памяти, что требует массивы, а достаточно всего лишь 42 байта. Однако вспомним, что операционная система периодически выгружает часть оперативной памяти на жесткий диск (swapping), в результате чего после достаточно долгого моделирования указатели в узлах связанного списка будут указывать на разные области памяти, что резко повышает возможность того, что страница, на которой находится соседний узел, находится а) на жестком диске; б) в основной памяти, но отсутствует в кэш-памяти процессора. Данные проблемы резко замедляют время работы алгоритмов, которым необходим последовательный проход по клеткам опухоли, а также тех, которые анализируют соседство и геометрическое положение. Однако алгоритмы вставки и удаления клеток будут выполняться за константное O(1) время. Алгоритмы сохранения данных на внешнем носителе реализуются без особых затруднений в приведенной структуре данных, однако не все алгоритмы визуализации возможно эффективно реализовать в данной схеме - те алгоритмы, которые требуют произвольного доступа к элементам опухоли (random access), выполняются на связанных списках за время, пропорциональное $\sim O(N^{1/3})$, а именно длине ребра куба. Небольшим недостатком также является то, что программирование этой структуры данных является не самым тривиальным заданием и для корректной реализации требуется затратить немало усилий, чтобы все алгоритмы вставки, удаления и трансформации корректно изменяли указатели. Использование связанных списков также делает очень сложным, если не сказать невозможным, применение распределенных систем для моделирования опухоли. Определение геометрического положения клетки делается за время, пропорциональное $O(N^{1/3})$, где N количество клеток.

3.2. Статический массив данных. Второй структурой данных является статический трехмерный массив. Под статическим массивом подразумеваем последовательную единожды выделяемую область памяти, в которой размещаются клетки и существует фиксированный алгоритм адресации, позволяющий по индексу элемента получить его координату в трехмерном пространстве. Очевидным недостатком структуры является то, что необходим последовательный большой кусок памяти. Частичным решением проблемы является выделение большого массива в самом начале работы программы, пока адресное пространство, отведенное программе, еще не фрагментировано, но данное решение сильно уменьшает гибкость программы, увеличивает занимаемый объем в памяти (footprint), нерационально использует память,

а также заранее лимитирует возможный размер и форму опухоли по всем направлениям, что, например, не является проблемой в случае связанных списков, так как там есть ограничение на общее количество клеток, но нет никаких ограничений на распределение этих клеток по направлениям. Заметным преимуществом перед связанными списками является то, что практически нет дополнительных затрат памяти на данные, напрямую не связанные с процессом моделирования. Исходя из тех же посылок, что и в предыдущем случае, а также делая допущение, что существует последовательная область памяти в 2Gb, получаем, что максимально возможная опухоль может состоять из порядка 120 миллионов клеток, что соответствует кубу с размером ребра в 490 клеток. Данный подход позволяет моделировать опухоли на 135% больше, чем в предыдущем случае. Другими недостатками данного подхода являются алгоритмы вставки и удаления клеток. Если в случае связанных списков данные алгоритмы требовали O(1) количества операций, то тут потребуется $O(N^{1/3})$ операций копирования, где N - количество моделируемых клеток. Также данный подход позволяет использовать параллельное моделирование на системах с разделяемой памятью (SMP), где каждому процессору отводится часть куба опухоли; правда, требуется определенная синхронизация на границах кубов. Определение расстояния до грани опухоли, как и в случае со связанными списками, займет время, пропорциональное $O(N^{1/3})$. Наибольшим преимуществом данного подхода является то, что при анализе соседних клеток мы можем быть уверены, что в большинстве случаев они будут загружены в кэш-память процессора, что может очень сильно повлиять на время моделирования. Отметим, что в данной структуре отсутствует проблема с произвольным доступом к клеткам опухоли, так как данная операция реализуется за константное время.

3.3. Динамический массив данных. Третьей структурой данных является динамический трехмерный массив. Под этим мы понимаем структуру данных, которая является списком указателей на указатели высших размерностей; говоря языком C, мы используем тройные указатели. Данный метод также позволяет использовать более логичный метод для доступа к элементам массива через стандартный оператор взятия индекса в языках C/C++, в отличие от преобразующей функции, необходимой во втором методе, которая из трех индексов получала один индекс в линейном массиве.

Использование динамических массивов позволяет избавиться от определенных недостатков, присущих статическим массивам, таких как фиксированный размер опухоли по каждому из направлений. В этом случае мы можем выбирать размер динамически, также не требуется резервирования

большого линейного массива. За счет определенных потерь в памяти мы обходимся набором куда более меньших массивов, что с одной стороны является технически более трудным заданием для реализации, с другой - позволяет иметь фактически, а не только теоретически большие размеры моделируемых опухолей.

Для начала рассмотрим количество дополнительной, необходимой для поддержания рассматриваемой структуры данных. Не теряя общности, предположим, что опухоль имеет кубическую форму, откуда следует, что нам потребуется

$$M = 2 \times N^{1/3} \times 4 = 8 \times N^{1/3} \quad (1)$$

байта памяти под указатели, где 4 - размер указателя, 2 - количество размерностей, под которые необходимо отвести память. Это число во много раз меньше, чем в случае связанных списков, где требовалось по 24 байта дополнительной памяти на каждую из клеток. Исходя из формулы (1) мы можем найти максимальный размер опухоли в данном случае из следующего уравнения:

$$18 \times N + 8 \times N^{1/3} = MemAvail. \quad (2)$$

18 - необходимая память для хранения в памяти одной моделируемой клетки, MemAvail - общее количество доступной памяти. В наших примерах это число равно 2Gb. После преобразования получаем кубическое уравнение

$$x^3 + \frac{4}{9}x - 119304647 = 0. \quad (3)$$

Решая уравнение методом Кардано, получаем один действительный корень, равный после округления 489, что неудивительно, так как в предыдущем случае, когда у нас было 120 миллионов клеток, затраченная лишняя память составляла

$$18 \times N^{1/3} = 18 \times (120 \times 10^6)^{1/3} = 18 \times 100 \times 4.9 \approx 4\text{КБайт}, \quad (4)$$

что, конечно, в данном случае является несущественной затратой.

Итак, третий случай имеет все преимущества второго метода, а именно быстрый доступ к элементам по индексам, что невозможно в первом случае, большие объемы моделируемых опухолей, относительно простую для программирования и манипуляции структуру данных, более быструю итерацию по всем клеткам опухоли (так как инкремент индекса быстрее перехода по указателю), а также неявные выгоды, возникающие в результате возможности лучшей оптимизации со стороны компилятора в виде загрузки соседних элементов в кэш-память процессора и т.п. Кроме того мы не теряем возможностей для реализации распределенного моделирования с помощью

метода, описанного в 3.2.

4. Выводы. В нашей программной реализации используется третий метод, единственным явным недостатком которого являются медленные операции вставки и удаления новых/погибших клеток. Этот недостаток во многих случаях удается исправить благодаря более интеллектуальным алгоритмам вставки и удаления, простейшим примером которых является следующая оптимизация: если исчезают и делятся соседние клетки, то новорожденная клетка помещается на место погибшей. Кроме того этот метод позволяет анализировать и моделировать опухоли, размер которых больше чем в два раза превышает максимальный размер, которого можно добиться в случае связанных списков.

Для более наглядного анализа данных мы свели их в таблицу. N - количество моделируемых клеток. Оценки размера приводятся для количества клеток, а также длины ребра соответствующего куба и радиуса сферы сферической опухоли.

Сравнительные характеристики структур данных

| СД/параметр | Время вставки | Время удаления | Перебор | Индексирование | Кэширование | Max размер опухоли |
|---|---|---|---|---|---|---|
| Список | O(1) | O(1) | O(1) | O(N1/3) | нет | $\sim 50 \cdot 10^6$ 370/228 |
| Стат. массив | O(N1/3), в идеале O(1) | O(N1/3), в идеале O(1) | O(1) | O(1) | да | $\sim 120 \cdot 10^6$ 490/305 |
| Дин. массив | O(N1/3), в идеале O(1) | O(N1/3), в идеале O(1) | O(1) | O(1) | да | $\sim 115 \cdot 10^6$ 489/301 |

Хотя перебор данных во всех случаях аппроксимируется константной функцией, на самом деле перебор всех элементов с использованием связанных списков будет во много раз медленнее, чем во втором и третьем случае.

Государственный инженерный университет Армении
vagababov@gmail.com

Վ. Ս. Աղաբաբով

**Քաղցկեղի in vitro մոդելավորման ժամանակ ուռուցքի ներկայացման համար օպտիմալ տվյալների կառուցվածքի ընտրության մասին**

Հոդվածում նկարագրվում են տարբեր մոտեցումներ, որոնց նպատակն է ընտրել օպտիմալ տվյալների կառուցվածք՝ համակարգչի հիշողության մեջ ուռուցքի ներկայացման համար: Ուսումնասիրության ժամանակ ուշադրություն է դարձվել հետևյալ հատկանիշներին. զբաղեցրած հիշողության ծավալը, տարրերին դիմելու ժամանակը և բարդությունը, ինչպես

նաև տվյալների կառուցվածքի մատչելիությունը մոդելավորող ալգորիթմների համար: Առանձին դիտարկվում է զուգահեռ կամ բաշխված համակարգերում ուռուցքների մոդելավորման հարցը:

V. S. Agababov

## On the Optimal Data Structure Selection for Tumor Representation for in vitro Cancer Simulation

The article describes different approaches aimed at data structure selection for optimal representation of a tumor in computer memory. During investigation the attention was paid to the following parameters such as memory consumption, element access complexity and suitability of the data structure for simulation algorithms. Suitability for parallel and distributed computing is also considered as one of the parameters.

### Литература

*1. Stamatakos G., Dionysiou D., Zacharaki E., Uzunoglu N.* — Proceedings of the IEEE. 2002. V. 90. N. 11. P. 525-540.

*2. Корн Г., Корн Т.* — Справочник по математике. М. Наука. 1973.

*3. Кормен Т., Лейзерстон Ч., Ривест Р., Штейн К.* — Алгоритмы: построение и анализ. М. Вильямс. 2005.