Математическая биология и биоинформатика. 2012. Т. 7. № 2. С. 425-443.
http://www.matbio.org/2012/Avilov_7_425.pdf

==========МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ==========

УДК:004.94:519.7

Агентные модели: анализ подходов и возможности приложения к эпидемиологии

©2012 Авилов К.К.[1a], Соловей О.Ю.[2b]

[1]*Институт вычислительной математики РАН, Россия, Москва, 119991, ул.Губкина, д.8*
[2]*Московский физико-технический институт (государственный университет), Россия, Москва, 117303, ул. Керченская, дом 1А, корпус 1*

Аннотация. Проведена оценка возможностей ряда пакетов для агентного моделирования (SWARM, RePast-J, NetLogo и др.). На примере NetLogo изучены особенности построения моделей и эффективности их выполнения. Из-за недостаточной производительности NetLogo на эпидемиологических моделях произведен эксперимент по реализации агентных моделей на реляционной базе данных Oracle. Модели на основе Orcale оказались эффективнее моделей на NetLogo только в случае отсутствия у NetLogo оптимизаций под данный класс моделей. Тестовая эпидемиологическая модель на Orcale была на 0,5-3 порядка эффективнее в зависимости от структуры популяции. Проведено тестирование возможностей подхода на основе Oracle для моделирования эпидемических процессов.

Ключевые слова: *агентное моделирование, математическая эпидемиология, средства разработки, СУРБД, NetLogo, Oracle.*

[a]avilov@inm.ras.ru
[b]tungusus@gmail.com

1. ВВЕДЕНИЕ

Одной из сфер активного применения математического моделирования является прогнозирование распространения различных инфекций как среди людей, так и в популяциях других видов. Традиционный инструмент описания таких процессов – компартментные модели, опирающиеся на дифференциальные уравнения и представление эпидемического процесса как изменения численности организмов-хозяев, находящихся в каждом из нескольких дискретных состояний (например, “здоров”, “болен”, “излечен”). Однако с ростом количества учитываемых в модели факторов число таких состояний возрастает экспоненциально, что чрезмерно усложняет модели и делает их неидентифицируемыми.

Решением этой проблемы могут служить так называемые агентные (“имитационные”, “индивидуально-ориентированные”) модели, использующие сравнительно простые правила поведения каждого отдельного организма-хозяина и производящие прямые стохастические (“монте-карловские”) розыгрыши процесса взаимодействия между ними. Таким образом, ход эпидемического процесса описывается как сумма действий и взаимодействий всех индивидов, образующих рассматриваемую популяцию и называемых в данном случае “агентами”. Поскольку расчет действий агентов происходит, как правило, независимо, агентные модели позволяют с легкостью учитывать сложнейшие виды неоднородности популяции.

Для агентных моделей в эпидемиологии характерно большое количество парных взаимодействий, отражающих, прежде всего, процессы передачи инфекции между организмами-хозяевами. С модельной точки зрения, такие взаимодействия представляют собой поиск группы агентов по некоторому достаточно сложному условию (территориальная близость, близость в динамической сети социальных контактов и т.п.), причем такой поиск надо производить на каждом шаге по времени для каждого агента. Очевидно, сложность этой процедуры выше, чем линейная от числа агентов в модели (обычно $O(n^2)$ или $O(n*ln(n))$, где n - количество агентов). Поэтому численная реализация эпидемиологических агентных моделей может иметь большую вычислительную сложность.

В данной статье мы рассматриваем возможные подходы к численной реализации агентных моделей в эпидемиологии и исследуем эффективность подхода на основе системы реляционных баз данных.

2. СРАВНЕНИЕ СУЩЕСТВУЮЩИХ СРЕД ДЛЯ АГЕНТНОГО МОДЕЛИРОВАНИЯ

В начале исследования были изучены возможности реализации агентных моделей на основе распространенных универсальных сред для агентного моделирования. В качестве примеров таких сред были рассмотрены свободно распространяемые пакеты SWARM [1], RePast-J [2] и NetLogo [3]. Из рассмотрения были исключены коммерческие среды (например, AnyLogic [4]) и среды, имеющие узкую специализацию на каком-либо типе моделей или типе вычислительных систем (например, EpiGrass [5] и F.L.A.M.E. [6]).

Рассмотрим подробнее каждый из упомянутых выше пакетов:

- NetLogo – программный пакет, разрабатываемый и поддерживаемый Northwestern University (Center for Connected Learning and Computer Based Modeling). Изначально NetLogo разрабатывался в качестве учебной системы агентного моделирования, но затем его возможности были значительно расширены, что дало возможность создания достаточно сложных практических моделей на его основе.
- SWARM – программный пакет для моделирования сложных адаптивных систем. SWARM был разработан в институте Санта-Фе (Santa Fe Institute) в середине

1990-ых, а с 1999 года разрабатывается и поддерживается “Группой Разработки SWARM” (Swarm Development Group). SWARM является первым массовым пакетом для агентного моделирования и многие современные пакеты основаны на принципах, изначально заложенных в SWARM. На данный момент SWARM является во многом устаревшим, а потому малоиспользуемым пакетом.

- RePast – программный пакет для агентного моделирования сложных систем. Существует несколько версий RePast, основанных на разных языках программирования – Java, C#, Managed C++ и др. В данной статье рассматривался RePast-J, оснванный на Java. RePast был изначально разработан в University of Chicago (группа Social Science Reserch Computing), в данный же момент разработку продолжает “Организация Repast по архитектуре и разработке” (Repast Oraganization for Architecture and Development) . RePast – один из наиболее распространенных пакетов для создания агентных моделей в области социальных наук. В целом подобен SWARM, однако имеет большое количество оптимизаций и усовершенствований.

Характеристики SWARM, RePast-J и NetLogo, полученные на основании нашего собственного сравнения и выводов обзорной статьи [7], представлены в таблице 1.

По нашему мнению, одними из важнейших характеристик пакета для агентного моделирования с точки зрения прикладного использования являются простота освоения и доступность информативной документации. Поэтому, несмотря на потенциально бóльшую вычислительную производительность пакета RePast [8, 9], мы приняли решение продолжить исследование возможностей широкораспространенных пакетов для агентного моделирования на примере пакета NetLogo.

3. ТЕСТОВЫЕ МОДЕЛИ

В качестве тестовых мы использовали три агентные модели, отражающие основную особенность эпидемиологических агентных моделей – большое количество парных взаимодействий. Первая из тестовых моделей представляла собой усложненный вариант стандартной SIR-модели [10] распространения инфекции в популяции организмов-хозяев с явным моделированием схемы контактов. Две другие тестовые модели были взяты из библиотеки стандартных моделей пакета NetLogo – модель формирования стаи птиц и модель распространения вируса по сети.

В силу того, что данные модели являются тестовыми и не претендуют на количественное соответствие реальным процессам, большинство их параметров и деталей законов поведения были выбраны произвольно с учетом только требования “правдоподобности”. Для оценки производительности пакетов агентного моделирования использование таких тестовых моделей представляется оправданным.

3.1. Тестовая эпидемиологическая модель

В качестве основы для данной модели была выбрана простейшая SIR-модель распространения инфекции в популяции организмов-хозяев [10], в которой каждый индивид популяции может находиться в одном из трех состояний:

- чувствителен к инфекции (S),
- заражен и инфекционен (I),
- иммунен к инфекции (R).

Скорость инфицирования индивидов (т.е. перехода из состояния S в состояние I) зависит от плотности индивидов в состоянии I. Скорость излечения с формированием иммунитета (т.е. перехода из стадии I в стадию R) осуществляется с постоянной удельной скоростью.

Таблица 1. Характеристики пакетов для агентного моделирования

Критерий	**REPAST-J**	**NetLogo**	**SWARM**
Тип лицензии	NewBSD (бесплатный)	GPL version 2 (бесплатный)	GNU GPL (бесплатный)
Язык описания моделей	Java	NetLogo – мультиагентное расширение языка Logo	Java или Objective C, в зависимости от версии
Тип исполнения кода	Компилятор	Интерпретатор, основанный на Java машине	Компилятор
Среда разработки	Набор библиотек	Интегрированная графическая среда	Набор библиотек
Дополнительные возможности	Набор узкоспециальных библиотек: поддержка ГИС, социальные сети, генетические алгоритмы, системная динамика	3D-визуализация, экспорт моделей на веб-страницы (Java-апплеты)	–
Оптимизация	Высокий уровень оптимизации узкоспециализированных библиотек	Оптимизация работы с агентами, движущимися на плоскости, и работы с сетями агентов	–
Параллелизация	Параллелизация в рамках одной модели; версия RePast-HPC для высокопроизводительных систем	Параллелизация по реализациям модели	Параллелизация отсутствует
Графика	Встроенная библиотека, присутствует специальная графическая среда для запуска моделей и отображения и изменения их параметров	Интегрированная графическая среда разработки и запуска, с привязкой графических объектов к написанному коду	Встроенная библиотека
Уровень поддержки авторами	высокий, но в основном для RePast-Symphony	высокий, частые обновления пакета и документации	низкий, практически не поддерживается
Уровень поддержки сообществом	высокий, присутствует большое сообщество, особенно в области социальных наук	средний, присутствует большое сообщество	низкий, ввиду общей устарелости пакета
Применение	Социальные науки. Моделирование сложных, адаптивных систем	Обучение, моделирование сетей	Моделирование сложных, адаптивных систем, устаревший
Сложность освоения	Высокая	Низкая	Средняя
Сложность установки	Низкая	Очень низкая	Высокая (крайне трудно установить на ОС Windows)
Официальная документация	Труднодоступна и малоинформативна	Доступна и информативна	Практически отсутствует

Как правило, в SIR-моделях на основе дифференциальных уравнений предполагается наличие однородной популяции, и эпидемиологические контакты между индивидами происходят на основе закона глобального перемешивания. Однако в нашей тестовой модели была учтена неоднородность популяции, имитирующая динамическую сеть социальных контактов, что приблизило ее к моделям, более адекватным для решения практических задач.

Социальная структура в нашей модели формируется за счет разделения популяции на "контактные группы", представляющие собой группы индивидов, находящихся в эпидемиологическом контакте, а потому способных передавать инфекцию друг другу. В нашей модели каждый индивид связан одновременно с тремя "контактными группами", но в каждый момент времени находится только в одной из них и перемещается между ними по определенному закону.

Каждый индивид популяции является отдельным агентом, имеющим следующие параметры:

- "SIR-статус" (т.е. состояние относительно инфекции),
- идентификаторы трех "контактных групп", с которыми связан индивид (условно "его семья", "его транспорт", "его коллеги"),
- идентификатор "контактной группы", в которой он находится,
- возраст,
- счетчик инфицирования.

Шаг времени модели – один час. В течение одного модельного дня каждый индивид, чей возраст находится в диапазоне от 7000 до 29000 модельных часов (условно "взрослый"), перемещается между тремя группами, с которыми он связан, по постоянному расписанию ("дом"→"транспорт"→"работа"→"транспорт"→"дом").

Моделирование передачи инфекции внутри "контактной группы" осуществляется за счет увеличения на единицу счетчика инфицирования у каждого чувствительного к инфекции индивида с вероятностью, пропорциональной плотности инфекционных индивидов в этой группе. При достижении счетчиком случайно изменяемого порога инфицирования индивид переходит в категорию инфекционных. Далее его счетчик инфицирования растет на единицу на каждом шаге модели, пока не превысит случайно изменяемый порог выздоровления. После этого индивид переводится в класс иммунных.

Изменчивость социальной структуры определяется тем, что каждый агент с вероятностью 10^{-5} может на каждом шаге модели сменить любую из трех своих "контактных групп" (в том числе и с формированием новой группы).

3.2. Тестовая модель формирования и поведения стаи птиц

Данная модель имитирует поведение стаи птиц в двух измерениях. Эта модель была выбрана нами в качестве тестовой, так как она является одной из распространенных агентных моделей (известна также как "модель Boids") и широко использует операцию поиска ближайших соседей (в данном случае – соседей на двумерной поверхности). Операция поиска ближайших соседей близка к задаче поиска эпидемиологических связей по сложному логическому условию в эпидемиологических моделях без явно описанной социальной структуры популяции.

Поверхность, на которой происходит движение "агентов-птиц", представляет из себя "закольцованный" квадрат (т.е. поверхность тора). Все "птицы" перемещаются с одной и той же постоянной по модулю скоростью.

На каждом шаге модели происходит изменение направления вектора скорости каждой из "птиц" в соответствии со следующими тремя правилами:

1) “Разделение”: при наличии ближайшего соседа на расстоянии меньшем определенного “радиуса опасности” каждая “птица” отворачивает от направления движения этого соседа с ограничением по углу поворота;
2) “Выстраивание”: каждая “птица” поворачивает с ограничением по углу поворота к среднему направлению движения ее ближайших соседей, находящихся в определенном “радиусе зрения” от нее;
3) “Сближение”: каждая “птица” поворачивает с ограничением по углу поворота к направлению на геометрический центр всех ее соседей в пределах “радиуса зрения”.

3.3. Тестовая модель “Вирус в сети”

Данная модель описывает распространение вируса в сети. Эта модель была выбрана нами в качестве тестовой ввиду того, что в ней описывается сравнительно распространенный в агентном моделировании тип взаимодействия между агентами – взаимодействие внутри постоянной сети. Такой тип взаимодействий часто используется в эпидемиологических моделях с жестко заданной социальной структурой популяции.

Модель представляет собой набор агентов, именуемых “узлами”, между которыми установлена постоянная сеть “связей”. Распространение вируса описывается аналогом рассмотренной выше SIR-модели, где каждый из узлов может быть либо подвержен заражению, либо заражен, с возможностью инфицировать другие узлы, либо невосприимчив (иммунен) к заражению. Каждый зараженный узел на каждом шаге модели с определенной вероятностью инфицирует все узлы, имеющие с ним непосредственную связь и подверженные заражению. Инфицированный узел один раз в заданное количество шагов модели с определенной вероятностью “излечивается”, переходя либо в состояние подверженного заражению, либо в состояние невосприимчивости.

4. РЕАЛИЗАЦИЯ ТЕСТОВЫХ МОДЕЛЕЙ НА NETLOGO

В целях проверки применимости NetLogo к задачам эпидемиологии, а также оценки его производительности в таких задачах, мы провели тестирование на основе описанных выше моделей. Нами использовался пакет NetLogo v.4.1.1[1].

4.1. Реализация тестовой эпидемиологической модели на основе NetLogo

Максимально используя стандартные возможности пакета NetLogo, мы реализовали тестовую эпидемиологическую модель с социальной структурой, описанную в разделе 3.1.

Основную вычислительную нагрузку в данной модели создают операции формирования списка агентов, на данном шаге находящихся в заданной контактной группе, и подсчет их количества. Эти операции были реализованы с помощью операторов `turtles with` и `count` языка NetLogo. Выбор агентов для изменения параметров (например, при достижении счетчиком инфицирования порогового значения) также осуществлялся при помощи оператора `turtles with`.

Данные по времени исполнения тестовой эпидемиологической модели на NetLogo представлены в таблице 2. Нетрудно видеть, что время расчета оказывается достаточно велико[2] (учитывая, что шаг времени модели составляет 1 час), но оно приблизительно

[1]Отметим, что в феврале 2012 года был выпущен NetLogo 5 со значительно расширенными возможностями.

[2]Здесь и далее все замеры производительности были выполнены на ноутбуке Lenovo ThinkPad Edge со следующими характеристиками: процессор Intel Core i3 330M, 2.13Ghz, 2 ядра, HyperThreading; опе-

Таблица 2. Время расчета одного шага тестовой эпидемиологической модели (усредненное по 20 шагам) при 50 контактных группах. Столбец “NetLogo” – для реализации на пакете NetLogo, столбец “Oracle” – для реализации с использованием СУРБД Orcale

Число агентов	NetLogo, время на шаг, сек.	Oracle, время на шаг, сек.
500000	37,451	5,065
350000	20,787	3,061
200000	6,120	1,631
100000	3,025	0,569
50000	1,668	0,310
25000	0,839	0,141
10000	0,388	0,083
5000	0,135	0,041
1000	0,023	0,012
500	0,012	0,009

линейно зависит от количества агентов.

Зависимость времени исполнения одного шага от стартового количества контактных групп приведена в таблице 3. Легко показать, что для модели на NetLogo эта зависимость близка к линейной. Тем не менее, при использовании реалистичного количества групп время просчета одного шага модели становится неприемлемо большим.

4.2. Реализация модели формирования стаи и модели “вирус в сети” на NetLogo

Данные модели поставляются вместе со средой NetLogo в качестве стандартных примеров. Мы исследовали их производительность в немодифицированном виде, в связи с чем их реализация на NetLogo в данной статье не рассматривается. Каждая из этих моделей основана на встроенных в NetLogo оптимизированных алгоритмах работы с агентами: модель формирования стаи основана на инструментарии для работы с движением агентов на плоскости, а модель “вирус в сети” – на алгоритме работы с агентами, объединенными в фиксированную сеть.

Данные по производительности этих моделей приведены в таблицах 4 и 5.

4.3. Промежуточные выводы: NetLogo

NetLogo является весьма эффективным инструментом для создании простых моделей или моделей, для которых имеются оптимизированные алгоритмы – движение на плоскости и квазипостоянные сети агентов. Однако при попытке усложнить модель или значительно изменить ее структуру производительность получившейся модели существенно ухудшается. Такая особенность делает пакет NetLogo малопригодным для наших целей, поскольку эпидемиологические агентные модели отличаются достаточной сложностью и разнообразностью. В частности, NetLogo не позволяет достаточно эффективно решать задачу поиска агентов по произвольному условию, что необходимо для моделей, опирающихся на парные взаимодействия агентов.

ративная память DDR3, 4Гб, 1066 МГц; ОС Microsoft Windows 7; NetLogo v.4.1.1; СУРБД ORACLE 11.2 beta.

Таблица 3. Время расчета одного шага тестовой эпидемиологической модели (усредненное по 20 шагам). Столбец “NetLogo” – для реализации на пакете NetLogo, столбец “Oracle” – для реализации с использованием СУРБД Orcale

Число агентов	Начальное число контактных групп	NetLogo (время на шаг, секунд)	Oracle (время на шаг, секунд)
50000	500	26,641	0,304
	1000	53,943	0,307
	5000	288,593	0,333
	10000	586,303	0,339
100000	500	57,119	0,631
	1000	116,645	0,643
	5000	796,944	0,607
	10000	1218.384	0,845
200000	500	102,273	1,340
	1000	237,419	2,743
	5000	1079,273	1,667
	10000	2192,031	1,150

Таблица 4. Время расчета одного шага модели формирования стаи птиц (усредненное по 20 шагам). Столбец “NetLogo” – для реализации на пакете NetLogo, столбец “Oracle” – для реализации с использованием СУРБД Orcale

Число агентов	NetLogo (время на шаг, секунд)	Oracle (время на шаг, секунд)
5000	0,401	11,192
4000	0,248	6,627
3000	0,185	4,532
2000	0,102	2,536
1000	0,061	0,879
700	0,051	0,585
400	0,029	0,314
100	0,017	0,092

Таблица 5. Время расчета одного шага модели “вирус в сети” (усредненное по 20 шагам). Столбец “NetLogo” – для реализации на пакете NetLogo, столбец “Oracle” – для реализации с использованием СУРБД Orcale

Число агентов	NetLogo (время на шаг, секунд)	Oracle (время на шаг, секунд)
30000	0,059	0,062
24000	0,057	0,058
12000	0,047	0,050
6000	0,023	0,040
3000	0.080	0,039
600	0,025	0,021

5. АГЕНТНОЕ МОДЕЛИРОВАНИЕ НА СУРБД

Проведенное исследование возможностей пакета NetLogo показало его недостаточную эффективность для реализации эпидемиологических моделей. Причиной этого является прежде всего отсутствие эффективного механизма обработки парного взаимодействия агентов, не основанного на их пространственной близости или соседстве в квазипостоянной сети контактов.

Одним из эффективных инструментов для поиска записей (в данном случае – агентов) по сложному условию являются системы управления реляционными базами данных (СУРБД). Отталкиваясь от предшествующего опыта других авторов по использованию СУРБД для агентного моделирования[3] мы приняли решение реализовать наши тестовые модели на инструментарии Oracle PL/SQL[4].

Основным достоинством использования СУРБД по сравнению со стандартными средами для агентного моделирования, на наш взгляд, является оптимизированность СУРБД для выполнения широкого спектра поисковых запросов и наличие простого в использовании механизма для ускорения выбранного типа запросов (механизм индексирования данных). Дополнительным преимуществом является высокая степень и “прозрачность” масштабирования современных СУРБД на многоядерные процессоры, многопроцессорные и кластерные компьютеры.

Языком для разработки приложений на основе СУРБД Oracle является PL/SQL (Procedural Language / Structured Query Language) – язык программирования, который является процедурным расширением языка SQL, разработанным корпорацией Oracle. PL/SQL базируется на языке Ада. Помимо этого, в СУРБД присутствует большое количество механизмов, оптимизации поиска, таких, как индексирование, партиционирование и т.д. Описание этих механизмов можно найти в документации СУРБД, поэтому в данной статье оно не приводится.

6. РЕАЛИЗАЦИЯ АГЕНТНЫХ МОДЕЛЕЙ НА ОСНОВЕ СУРБД ORACLE

Для сравнения эффективности предлагаемого подхода с эффективностью моделирования на NetLogo, мы реализовали на PL/SQL три тестовые модели, описанные в разделе 3.

6.1. Общие принципы реализации

Реализация всех трех описанных выше моделей на основе СУРБД имеет ряд общих черт. Каждый агент модели представляется в виде строки в “рабочей” таблице, а параметры агента соответствуют колонкам этой таблицы. Поиск, изменение и удаление агентов производятся стандартными средствами СУРБД для поиска, изменения и удаления строк, применяемыми к “рабочей” таблице.

Одним из главных инструментов для ускорения поиска данных в СУРБД являются индексы, то есть дополнительные объекты, хранящие информацию о структуре данных в таблице. Наиболее распространенным типом индексов являются дервовидные структуры (бинарные деревья и пр.). Индекс формируется либо для одного столбца таблицы,

[3]На основании частного сообщения. В открытой печати публикаций на эту тему нами не обнаружено.

[4]Oracle является коммерческой СУРБД фирмы Oracle Corporation. Нами использовалась версия Oracle 11.2 beta. На beta-версии СУРБД Oracle предоставляются бесплатные OTN-лицензии, достаточные для разработки и тестирования программ на их базе. Для обработки данных с помощью созданных программ или для использования полноценных версий СУРБД Oracle требуется приобретение полноценной лицензии. Однако производительность базовых механизмов работы с данными в beta-версиях Oracle обычно не отличается от полноценных версий, поскольку OTN-лицензии предоставляются в целях демонстрации производительности.

либо для группы столбцов, что ускоряет поиск по условиям, включающим значение соответствующего столбца или группы столбцов. При сложных поисковых запросах возможна ситуация, когда СУРБД использует индекс неэффективно (вплоть до перехода к прямому перебору). Основной недостаток использования индексов – необходимость перестройки индекса при каждом изменении данных в индексированной части таблицы.

При реализации агентных моделей на PL/SQL мы использовали индексирование по тем столбцам (т.е. параметрам агентов), по которым производились наиболее интенсивные поисковые операции.

Для моделей, в которых индексированные поля редко подвергаются изменению и поисковые запросы достаточно просты, более эффективной оказалась схема с одной рабочей таблицей и поиском всех пар взаимодействующих агентов при помощи одного SQL-запроса. Обновление состояния агентов производилось при помощи прямого обновления строк таблицы (оператор `UPDATE`), а расходы на обновление индексов оказывались сравнительно малы.

Однако для моделей с активным изменением индексированных полей и более сложными поисковыми запросами такая схема не подходит как из-за слишком больших накладных расходов на обновление индексов, так и из-за неэффективного использования индексов внутренним оптимизатором СУРБД Oracle. Для работы с такими моделями мы использовали следующую усложненную схему (см. рис. 1):

- используются две рабочих таблицы (текущий шаг и следующий шаг по времени),
- в “текущей” рабочей таблице формируются индексы по необходимым столбцам, а в рабочей таблице для следующего шага они отключаются,
- на каждом шаге перебираются все агенты и для каждого рассчитываются его парные взаимодействия с другими агентами и его состояние на следующем шаге, которое записывается в соответствующую таблицу,
- рабочие таблицы меняются местами: “таблица следующего шага” становится “текущей” (и в ней создаются индексы), а в бывшей “текущей” таблице индексы отключаются и она становится “таблицей следующего шага”.

Такая схема позволяет избежать обновления индексов при записи нового состояния каждого отдельного агента (за счет разделения таблиц), а также позволяет оптимизатору СУРБД Oracle эффективно использовать индексирование данных (за счет упрощения запросов при переходе к циклу по агентам).

Также при работе с статичной сетью агентов создается специальная таблица-ассоциатор, каждая строчка которой описывает связь между двумя агентами и содержит идентификаторы связанных агентов и, возможно, некоторые параметры, описывающие саму связь. Ввиду постоянства связей между агентами таблица-ассоциатор строится индекс-организованной, что значительно ускоряет поиск по ней.

При необходимости записи траектории модели необходимые данные записываются на каждом шаге в специальную таблицу и дополняются информацией о номере шага.

6.2. Реализация тестовой эпидемиологической модели на СУРБД

Для реализации на СУРБД тестовой эпидемиологической модели с усложненной социальной структурой, описанной в разделе 3.1, применялся подход с одной рабочей таблицей.

Подавляющее большинство поисковых операций в данной модели осуществляется по ключам принадлежности к определенной контактной группе, которые крайне редко изменяются. Поэтому, с одной стороны, по соответствующим колонкам рабочей таблицы

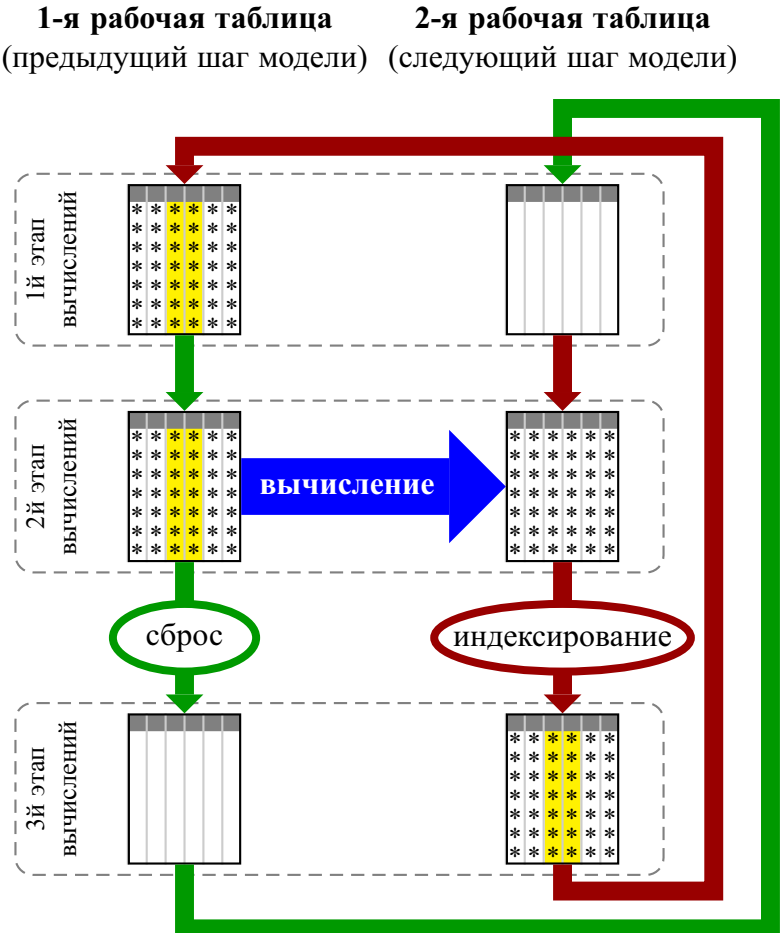

Рис. 1. Схема обработки моделей с частым изменением индексированных полей. Используются две рабочих таблицы, меняющиеся местами при переходе к обработке следующего шага модели. Символы “*” обозначают данные в рабочих таблицах СУРБД, а выделение столбцов желтым цветом – построение индексов по этим столбцам. Стрелка “вычисление” означает вычисление состояния модели на следующем шаге по времени.

был построен индекс, и, с другой стороны, затраты машинного времени на обновление индекса при переходе агентов из одной контактной группы в другую оказались малы. Также была проиндексирована колонка “возраст”, поскольку по этому параметру так же часто производится поиск.

Для расчета количества агентов и плотности зараженных применялись стандартные агрегатные функции СУРБД (`count`, `sum`), а также механизм группировки (оператор `GROUP BY`).

Время исполнения одного шага модели приведено в таблице 2. При реализации модели, сходной с NetLogo, время выполнения одного шага оказалось в 3-6 раз меньше.

Зависимость времени исполнения одного шага СУРБД-модели от начального числа контактных групп приведена в таблице 3. В отличие от реализации на основе NetLogo, в данной реализации не наблюдается явной зависимости времени выполнения одного шага модели от стартового числа контактных групп. Это в свою очередь приводит к тому, что при большом числе контактных групп реализация на СУРБД превосходит по производительности реализацию на NetLogo на два-три порядка.

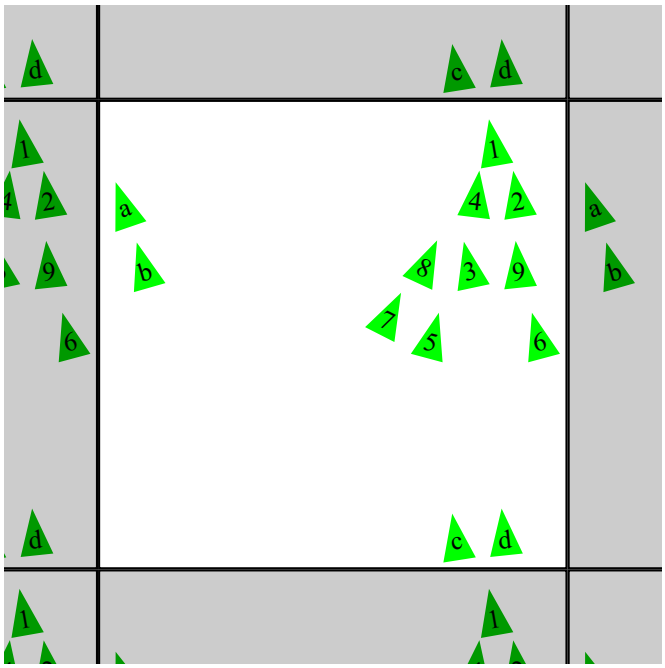

Рис. 2. Схема создания фиктивных агентов в модели формирования стаи птиц. Светлая область соответствует рабочей области модели, серая область – фиктивной области. Толщина фиктивной области равна “радиусу видимости” агентов. Агенты и их фиктивные образы имеют одинаковые номера.

6.3. Реализация модели формирования стаи на СУРБД

При реализации модели формирования стаи применялся подход с двумя рабочими таблицами (см. раздел 6.1 и рис. 1). Кроме того мы старались избегать конструирования узкоспециализированных алгоритмов, учитывающих двумерный характер движения агентов модели (методов кластеризации пространства и иных методов ускорения поиска точек на плоскости), и использовали только стандартные алгоритмы поиска, реализованные в СУРБД Oracle. Единственной вынужденной оптимизацией был отказ от использования тригонометрических функций, встроенных в СУРБД Orcale (из-за их крайне низкой производительности), и использование проекций скоростей агентов, а не направлений и модулей скоростей как в оригинальной модели из библиотеки NetLogo.

Все расстояния в модели определялись напрямую и в таком виде использовались в условиях поиска.

Закольцовывание рабочей области модели было сделано путем добавления на каждом шаге фиктивных агентов, создающих иллюзию продолжения рабочей области за ее границу. Такие фиктивные агенты формировались параллельным переносом агентов, находящихся на расстоянии менее “расстояния видимости” от противоположной грани рабочей области (см. рис. 2). Такой способ закольцовывания рабочей области был применен потому, что усложнение формул вычисления расстояния приводило к ухудшению работы индексов и, как следствие, к большему снижению производительности, чем добавление некоторого количества фиктивных агентов.

На каждом шаге модели перебирались все нефиктивные агенты, и для каждого средствами СУРБД производится отбор агентов, попавших в “радиус видимости” и в “опасный радиус”, поиск ближайшего агента, соответствующее изменение направления движения. Затем единой операцией для всех нефиктивных агентов производился расчет их

положения на следующем шаге.

Использование подхода с двумя рабочими таблицами позволило оптимизировать использование индексов. Мы применили цикл по агентам, так как при попытке обработки всех агентов в едином SQL-запросе оптимизатор запросов СУРБД Oracle неэффективно использовал индексы.

Результаты замеров производительности приведены в таблице 4. Несмотря на обширное применение индексов и длительную оптимизацию, мы не смогли достичь приемлемых результатов по производительности, хотя бы сопоставимых с NetLogo.

6.4. Реализация модели “вирус в сети” на СУРБД

При реализации модели “вирус в сети”, описанной в разделе 3.3, применялся такой же подход с одной рабочей таблицей, как в тестовой эпидемиологической модели. Основным отличием было наличие таблицы-ассоциатора, описывающей структуру сети.

Основная таблица системы содержит информацию о текущем состоянии узлов сети. Таблица-ассоциатор содержит только две колонки – идентификаторы узлов, между которыми существует связь, и является индекс-организованной по этим колонкам. Таблица-ассоциатор заполняется один раз перед запуском модели и в течение выполнения модели не изменяется.

На каждом шаге с использованием ассоциатора проводится выборка всех незараженных узлов, чьими соседями являются зараженные узлы, и для них с вероятностью, равной 0,1, производится смена значения “стадия заражения” с “подвержен заражению” на “инфекционен”. Также при наступлении времени проверки для каждого зараженного узла он с вероятностью 0,09 изменяет состояние с “заражен” на “подвержен заражению” и с вероятностью 0,01 – на “иммунен”, а параметр “время следующей проверки на зараженность” увеличивается на величину, равную параметру “период проверки на наличие вируса” для данного агента.

Сравнение времени выполнения одного шага для разного числа узлов сети приведено в таблице 5. Реализация модели на СУРБД позволила достичь показателей производительности, которые хотя и не превосходят показатели NetLogo, но сопоставимы с ними. Основной выигрыш в производительности достигнут за счет применения индекс-организованной таблицы-ассоциатора.

6.5. Промежуточные выводы: подход на основе СУРБД Oracle

Несмотря на то, что с моделями формирования стаи и вируса в сети, реализованными на Oracle, нам не удалось превзойти производительность пакета NetLogo, производительность тестовой эпидемиологической модели на Oracle оказалась больше производительности аналогичной модели на NetLogo на 0,5-3 порядка. Основной причиной такого различия, на наш взгляд, являются, с одной стороны, высокая оптимизированность пакета NetLogo под описание движения агентов на плоскости и взаимодействия в квазипостоянной сети и, с другой стороны, низкая оптимизированность для поиска агентов по произвольному логическому условию.

В нашей тестовой эпидемиологической модели на каждом шаге производится группировка и подсчет агентов по “контактным группам”. Пакет NetLogo при помощи оператора `turtles with` выполняет эту операцию, судя по всему, последовательно, что приводит к линейной зависимости времени выполнения шага модели от количества таких группировок на шаг (т.е. от количества “контактных гурпп”). Oracle же позволяет выполнить такие операции в поточно-параллельном режиме (прежде всего за счет оператора `GROUP BY`), в результате чего у него практически отсутсвует зависимость времени

выполнения шага модели от количества группировок.

Это наблюдение позволяет нам признать подход с реализацией агентных моделей на СУРБД (в частности – на Oracle) более универсальным и более производительным *в среднем*, чем использование пакета NetLogo и других подобных пакетов. Исключение составляют модели, существенно использующие узкие оптимизации, заложенные в специализированные пакеты для агентного моделирования.

Вместе с тем отметим, что мы намеренно применяли только базовые встроенные алгоритмы как СУРБД Oracle, так и пакета NetLogo. Использование дополнительных узкоспециализированных надстроек, вероятно, расширит спектр эффективно реализуемых моделей (прежде всего – для пакета NetLogo).

7. ПРИКЛАДНАЯ ЭПИДЕМИОЛОГИЧЕСКАЯ SEIRS-МОДЕЛЬ

Для тестирования возможностей предложенного подхода с использованием СУРБД в прикладных эпидемиологических исследованиях, нами была построена усложненная SEIRS-модель [10] распространения гриппа/ОРВИ в городской популяции, близкая по своим структуре и сложности к используемым в прикладных работах. Главной особенностью этой модели является явное описание достаточно сложной, неоднородной сети социальных контактов, приближенной к реальной. Несмотря на свое название, наша “прикладная” модель во многом является тестовой и не претендует на количественное соответствие реальным данным, а потому многие ее параметры были выбраны произвольно на основании соображений “правдоподобности”. Тем не менее, модель позволила качественно повторить профили эпидемий гриппа в Москве в советский период [11].

Данная прикладная SERIS-модель является развитием рассмотренной выше тестовой эпидемиологической SIR-модели. Каждый агент в ней представляет собой одного индивида популяции и обладает параметрами “возраст”, “социальный статус”, а также параметрами, описывающими течение болезни – “счетчик инфецирования” и “стадия болезни”.

7.1. Социальная структура

Социальная структура популяции задается при помощи разделения ее на “контактные группы”. Контактной группой называется однородная совокупность индивидов, в определенный момент времени имеющих вероятность вступить друг с другом в эпидемиологический контакт. Контактные группы моделируют такие коллективы людей как домохозяйства, рабочие коллективы и т.д. Всего присутствует три типа “контактных групп”: “домашняя”, “рабочая”, “транспортная”, при этом “рабочий” тип разделен на три подтипа “работа”, “школа” и “детский сад”.

Формирование “контактных групп” происходит на основе стратификации популяции по параметрам “возраст” и “социальный статус”. Различаются четыре схемы поведения агентов, зависящие от их возраста – “дошкольник”, “школьник”, “работающий” и “пенсионер”. Социальный статус влияет прежде всего на размер контактных групп: индивиды с более высоким социальным статусом имеют в среднем меньше эпидемиологических контактов за единицу времени.

Общая схема дневного цикла перемещения агента:

$$\text{дом } (t = 14\text{ч}) \rightarrow \text{транспорт } (t = 1\text{ч}) \rightarrow \text{работа } (t = 8\text{ч}) \rightarrow \text{транспорт } (t = 1\text{ч}) \rightarrow \text{дом},$$

где t – время пребывания агента в конкретной группе. Исключение из этой схемы составляют:

- все дошкольники – не участвуют в “транспорте”,

- дошкольники, не включенные в “детсадовские” группы – не посещают “работу (детсад)”,
- пенсионеры – не участвуют в утреннем “транспорте” и “работе”.

Процессы рождения, гибели, миграции и изменения социальной структуры в данной модели отсутствуют.

7.2. Эпидемиологические процессы

Агенты могут находиться в одном из трех основных эпидемиологических состояний – S, I, R. В состоянии S агент может быть как неинфицированным (значение счетчика инфицирования I_C равно 0), так и инфицированным, но не инфекционным ($0 < I_C < 1$). Контакты с инфекционными индивидами повышают значение счетчика I_C. Кроме того значение счетчика с течением времени затухает при значениях ниже 0.5 и растет при значениях выше 0.5. Первое моделирует самоизлечение после незначительных инфицирований, второе моделирует латентный период болезни (порядка 1-2 дней).

При достижении счетчиком I_C значения 1 индивид переходит в состояние инфекционности I. При этом ему случайным образом определяется длительность инфекционной стадии болезни (среднее – 6 дней), по истечении которой он излечивается и переходит в иммунное состояние R.

Инфекционные индивиды I могут инфицировать индивидов S, находящихся в той же контактной группе. Вероятность инфицирования чувствительного индивида в группе за один час задается как $p_I = 1-(1-\rho)^I$, где ρ – вероятность контакта в группе, зависящая от ее типа [12], а I – количество инфекционных индивидов в группе.

Иммунные индивиды с определенной вероятностью в единицу времени теряют иммунитет и переходят в состояние S. Удельная скорость потери иммунитета в нашей модели значительно превышает оцененную в выборочных исследованиях для гриппозной инфекции, что моделирует смену фенотипов инфекции при отсутствии полного перекрестного иммунитета. Это, в свою очередь, позволяет моделировать распространение гриппозной инфекции в замкнутой популяции как самоподдерживающийся процесс, а не как реакцию на занос инфекции извне.

7.3. Моделирование процесса эпидемий

Для получения начального состояния модели генерировалась случайная популяция численностью около 11500 чел., все ее члены изначально находились в стадии S с счетчиком $I_C = 0$, а 100 случайных индивидов переводились в состояние инфекционных. Затем запускался расчет модели и продолжался до выхода на квазистационар.

Моделирование сезонных эпидемий опиралось на предположение о существовании климатического фактора, вносящего возмущение в параметры модели и тем самым отклоняющим ее от квазистационара. В качестве управляющего параметра была выбрана удельная скорость потери иммунитета.

Итоговые графики по динамике численностей эпидемиологических групп и количеству заболевших за неделю приведены на рисунках 3 и 4. График заболеваемости качественно повторяет траектории реальных эпидемий грипа в Москве в советский период [11], плохо описываемых стандартной SIR-моделью на основе обыкновенных дифференциальных уравнений.

7.4. Техническая реализация

Техническая реализация модели полностью эквивалентна описанной выше реализации тестовой SIR-модели на СУРБД с незначительными усложнениями. Замедление

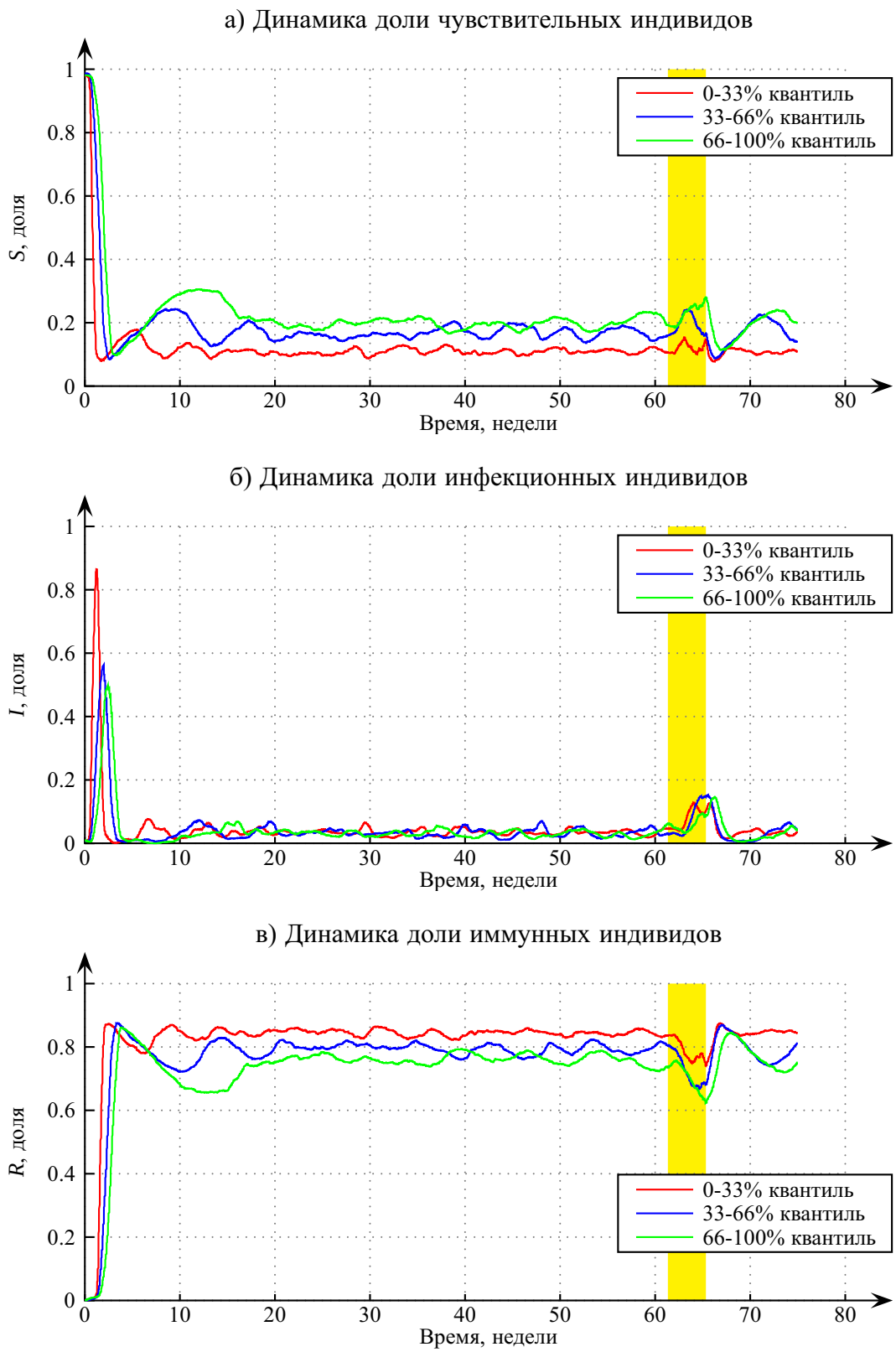

Рис. 3. Динамика количества чувствительных (S), инфекционных (I) и иммунных (R) индивидов при линейно нарастающей скорости потери иммунитета в течение 4 недель (желтым цветом обозначен период воздействия). Стратификация по 33-процентным квантилям по параметру “социальный статус”.

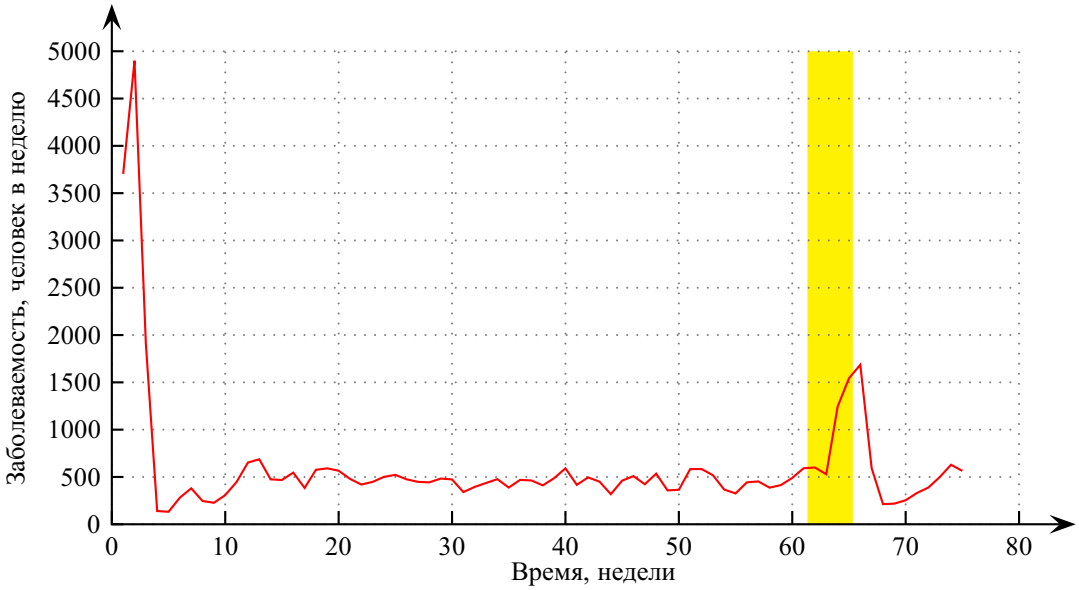

Рис. 4. Динамика суммарной недельной заболеваемости при линейно нарастающей скорости потери иммунитета в течение 4 недель (желтым цветом обозначен период воздействия).

работы модели по сравнению с тестовой SIR-моделью составило приблизительно полтора раза: например, для тестовой эпидемиологической модели при 11500 агентах и 1100 контактных группах время выполнения одного шага составляло 0,173 сек., а для прикладной модели при приблизительно равном количестве агентов и групп – 0,260 сек.

8. ВЫВОДЫ

В данном исследовании мы сравнили два подхода к реализации агентных моделей: на основе пакета NetLogo 4.1.1 и на основе СУРБД Oracle. Пакет NetLogo был выбран как наиболее удобный для освоения из широкораспространенных пакетов для агентного моделирования. Подход, основанный на программировании в среде Oracle на языке PL/SQL, представлялся перспективным из-за возможностей встроенных оптимизаций СУРБД Oracle. В качестве тестовых использовались три модели, обладающие характерными для эпидемиологических моделей свойствами: модель с поиском агентов по сложному условию (модель формирования стаи), модель взаимодействия в постоянной сети связей (модель вируса в сети) и собственно эпидемиологическая SIR-модель с явным (хотя и упрощенным) описанием социальной структуры.

По результатам проделанной работы можно сделать вывод о том, что ни один из исследованных подходов нельзя назвать “заведомо более эффективным” (см. таблицу 6). Удобство создания моделей и их вычислительная производительность радикальным образом зависят от того, насколько данная модель задействует заложенные в используемую среду высокооптимизированные механизмы. Вместе с тем, как показал эксперимент с прикладной эпидемиологической моделью, усложнение и расширение моделей на основе СУРБД оказались весьма просты в силу гибкости СУРБД Oracle.

Пакет NetLogo 4.1.1 отличается наглядностью, легкостью освоения на базовом уровне, и эффективно реализует агентные модели, опирающиеся на движение точечных агентов по двумерной плоскости и на квазипостоянные графы связей между агентами. Попытка реализовать на NetLogo модели других классов приводит к резкому падению производительности.

Таблица 6. Сравнение итоговых оценок качеств пакета NetLogo и СУРБД Orcale как средств реализации агентных моделей

Параметры сравнения		**Тестируемые пакеты**	
Категория параметров	Наименование параметра	*NetLogo*	*моделирование на базе СУРБД Oracle*
Производительность	Моделирование движения на плоскости	Встроенная оптимизация (агенты “turtle”, “patch” + эффективный поиск соседей)	Оптимизация сложна (GEOindex – эффективен только для очень сложных задач)
	Моделирование квазистационарных сетей	Встроенная оптимизация (агенты “link”)	Оптимизация крайне проста (вспомогательная таблица-ассоциатор)
	Моделирование сложных парных взаимодействий	Оптимизация сложна (поиск по условию малоэффективен)	Встроенная оптимизация поиска по условию
Удобство использования	Качество и доступность документации	Учебные курсы, полное описание, широкая поддержка сообществом	Учебные курсы, полное описание, широкая поддержка сообществом
	Сложность освоения	Малое время освоения (в т.ч. из-за ограниченного функционала)	Среднее или большое время освоения (в зависимости от объема освоения)
	Интегрированная среда разработки и запуска моделей	Графическая среда, динамическое отображение состояния модели, графики параметров	Консольный/SQL доступ, но существуют интерфейсные надстройки

Использование СУРБД Orcale в качестве средства реализации агентных моделей позволило воспользоваться ее встроенными оптимизациями широкого класса поисковых задач. При достаточном знакомстве с языками SQL и PL/SQL эти оптимизации позволяют легко создавать весьма производительные реализации широкого класса агентных моделей, но все же их производительность не превосходит производительности реализаций на подходящих узкоспециализированных пакетах для агентного моделирования.

Таким образом, подход на основе СУРБД можно рекомендовать в тех случаях, когда не существует или не доступен специализированный пакет агентного моделирования, оптимизированный для решения стоящей перед исследователем задачи.

СПИСОК ЛИТЕРАТУРЫ

1. Minar N., Burkhart R., Langton C., Askenazi M. The Swarm simulation system: a toolkit for building multi-agent simulations. *Working Paper 96-06-042, Santa Fe Institute, Santa Fe.* 1996.

2. North M.J., Collier N.T., Vos J.R. Experiences Creating Three Implementations of the Repast Agent Modeling Toolkit. *ACM Transactions on Modeling and Computer Simulation.* 2006. V. 16. Iss. 1. P. 1–25.

3. Alfaro J.F., Miller S.A. Planning the development of electricity grids in developing countries: An initial approach using Agent Based Models. In: *Proceedings of 2011 IEEE International Symposium on Sustainable Systems and Technology (ISSST) (May 16-18, 2011. Chicago, IL.)* 2011. P. 1–6.

4. Карпов Ю.Г. *Имитационное моделирование систем. Введение в моделирование с AnyLogic 5.* СПб: БХВ-Петербург, 2006. 400 с.

5. Coelho F.C., Cruz O.G., Codeço C.T. Epigrass: a tool to study disease spread in complex networks. *Source Code Biol. Med.* 2008. V. 3. № 3.

6. Holcombe M., Coakley S., Smallwood R. A General Framework for agent-based modelling of complex systems. *Proceedings of the European Conference on Complex Systems.* 2006.

7. Robertson D.A. Agent-Based Modeling Toolkits. *Academy of Management Learning and Education.* 2005. V. 4. № 4. P. 525–527.

8. Railsback S.F., Lytinen S.L., Jackson S.K. Agent-based Simulation Platforms: Review and Development Recommendations. *Simulation*. 2006. V. 82. № 9. P. 609–623.

9. Berryman M. Review of Software Platforms for Agent Based Models. *Land Operations Division Defence Science and Technology Organisation DSTO-GD-0532*. 2008.

10. Андерсон Р., Мэй Р. *Инфекционные болезни человека. Динамика и Контроль.* Пер. с англ. Москва: Мир, “Научный мир”, 2004. 784 с.

11. Романюха А.А., Санникова Т.Е., Дрынов И.Д. Возникновение эпидемий острых респираторных заболеваний. *Вестник Российской академии наук.* 2011. Т. 81. № 2. С. 122–126.

12. Перминов В.Д., Корнилина М.А. Индивидуум-ориентированная модель распространения эпидемии в городских условиях. *Матем. моделирование.* 2007. Т. 19. № 5. С. 116–127.

Материал поступил в редакцию 07.06.2012, опубликован 30.07.2012.