виям оптимальности (4) компоненту Δ_j, $j \in J_н$. Такое изменение позволяет существенно облегчить процедуру проверки условий оптимальности, т.к. градиент будет вычисляться только частично.

Оценить теоретически влияние внесенных изменений на эффективность метода невозможно. Поэтому проведен численный эксперимент, в ходе которого проведено сравнение базовой версии алгоритма и описанной выше модификации.

4. Результаты численного эксперимента. Базовая версия алгоритма и его модификация реализованы программно на языке программирования C++. Проведен численный эксперимент, результаты которого приведены в таблице 1:

Таблица 1 – Результаты численного эксперимента

m	*n*	Базовый		Модифицированный		Небазисные элементы, %	Эффект., %
		Время	Итераций	Время	Итераций		
5	50	16026	91	3079	142	90	520,49
20	50	18417	92	8818	133	60	208,86
50	50	13492	46	7699	49	13	175,24
100	100	144844	87	68055	98	14	212,83

Здесь *m* – ранг матрицы *D*, *n* – количество переменных, время – количество тактов процессора.

Анализ результатов показал, что модифицированная версия алгоритма в 1,5-5 раз эффективнее базовой версии.

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Габасов, Р. К методам решения общей задачи выпуклого квадратичного программирования / Р. Габасов, Ф. М. Кириллова, В.М. Ракецкий: докл. АН СССР. – 1981. – Т.258, N6. – С. 1289-1293.

2. Ракецкий, В.М. Прямой опорный метод квадратичного программирования: Проблемы оптимального управления. – Минск: Наука и техника, 1981. – С. 318 – 335.

3. Габасов, Р. Конструктивные методы оптимизации / Р. Габасов, Ф.М. Кириллова, О.И. Костюкова, В.М. Ракецкий. – Мн.: Изд-во "Университетское", 1987. – Ч.4: Выпуклые задачи. – 223 с.

4. Фаддеев, Д.К. Вычислительные методы линейной алгебры / Д.К. Фаддеев, В.Н. Фаддеева. – М.: Физматгиз,1960. – 734 с.

УДК 681.3

Рыщук А.С.

Научный руководитель: проф. Муравьев Г.Л.

ТРУДОЁМКОСТЬ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ СТОХАСТИЧЕСКИХ СЕТЕЙ

Наряду с адекватностью, результативностью и другими характеристиками степень полезности и применимости моделей и в особенности имитационных существенно зависит от их сложности, трудоемкости. Характеристики моделей определяются в процессе их аттестации, используются для оценки собственных свойств моделей, анализа влияющих факторов, для сравнения с характеристиками аналогичных моделей [1, 2].

Для стохастических сетевых моделей (ССМ) [3] сложность определяется особенностями архитектуры модели – сложностью структуры сети узлов и потоков заявок, сложностью управления обслуживанием и может быть оценена по количеству и составу параметров модели, необходимых для ее спецификации.

Трудоемкость модели как сложность вычислений оценивается количеством вычислительной работы (времени, операций, команд и т.д.), требуемой для ее реализации с целью получения набора характеристик заданной полноты и точности. Трудоемкость модели связана со сложностью самой сети. Но при ее определении учитываются особенности реализации модели. Соответственно при оценке трудоемкости имитационного моделирования стохастических сетей следует говорить как о трудоемкости реализации используемого механизма, алгоритма имитации (включающего способ построения списков событий, продвижения модельного времени, метод организации квазипараллельностей [4], методы генерации случайных объектов, сбора данных, оценки характеристик и т.д.), так и о трудоемкости имитационного моделирования сети конкретной архитектуры.

Последняя для системы (алгоритма) моделирования представляет собой вероятностную вычислительную нагрузку, т.к. произвольная сеть функционирует случайным образом и соответственно задается параметрами, специфицирующими вероятностные процессы (входные, выходные, процессы изменения состояний).

Тем самым для оценки трудоемкости модели могут быть использованы подходы к оценке трудоемкости алгоритмов, обрабатывающих статистически устойчивую вычислительную нагрузку. Наиболее известны натурные, статистические методы оценки, экспертные методы, методы оценки трудоемкости на базе марковских цепей, методы на базе применения сетевых графиков и др. [3].

В работе рассмотрен подход к оценке трудоемкости имитационного моделирования на примере сетей массового обслуживания, когда модель реализуется на базе метода активностей [4], а способ продвижения времени – событийный. Соответственно модель строится как совокупность наборов активностей – процедур обработки событий, происходящих в сети, временных списков событий и других структур данных для отображения и прогнозирования состояний системы и сбора данных, образующих информационную базу (ИБ) модели и модулей управления активностями (МУ). Функционирование модели носит цикличный характер: по мере продвижения времени происходят события, создаются условия для запуска активностей; методы МУ анализируют списки событий ИБ и запускают активности; активности в нужной последовательности активизируют функции, методы участвующих в них объектов; функции (методы) меняют состояние сети (ИБ).

Для оценки трудоемкости использован модифицированный подход на базе марковских цепей [3], где в качестве вершин используются только функциональные операторы (ФО). Соответственно сетевая модель отображается “усреднённым” графом алгоритма, где все вершины представляют собой ФО, а дуги – переходы управления, соответствующие случайным событиям. Трудоёмкость ФО в свою очередь может быть рассчитана через марковские цепи или оценена натурно, имитационно, а вероятности переходов берутся из спецификации моделируемой стохастической сети заданной конфигурации.

Рассмотрены аналитические оценки, дающие верхнюю границу трудоемкости, позволяющие оценивать “удельную” трудоемкость моделирования сети заданной архитектуры как трудоемкость обработки в расчете на одно событие, "вычислительную" трудоемкость моделирования сети, в том числе с учетом требуемой точности вычислений.

Для получения статистических оценок, выявления структуры факторов, формирующих трудоемкость модели, необходимых исходных данных для формульной оценки трудоемкости выполнена реализация сетей средствами Visual Studio C++. Модель спроектирована в объектно-ориентированной технологии [5] в виде совокупности взаимодействующих объектов, адекватно отображающих структурные компоненты сети и специфику ее функционирования. Это обеспечило возможность организации эффективного таймирования, позволило на основе полученных данных оценивать структуру трудоемкости ЯВУ-моделей сетей массового обслуживания.

Оценки трудоемкости могут быть использованы для выявления структуры факторов, формирующих трудоемкость модели, определения наиболее затратных, ресурсоёмких фрагментов алгоритма имитации, неэффективно организованных функций модели, а также для выявления фрагментов, функций, процессов модели для последующего ускорения посредством распараллеливания.

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Советов, Б.Я. Моделирование систем / Б.Я.Советов, С.А. Яковлев. – М.: Высшая школа, 2005. – 343 с.
2. Рыжиков, Ю.И. Имитационное моделирование. Теория и технологии. – СПб.: КОРОНА, 2004. – 320 с.
3. Майоров, С.А. Основы теории вычислительных систем / С.А. Майоров, Г.И. Новиков, Т.И. Алиев. – М.: Высшая школа, 1978. – 320 с.
4. Максимей, И.В. Имитационное моделирование на ЭВМ. – М.: Радио и связь, 1988. – 232 с.
5. Труб, И.И. Объектно-ориентированное моделирование на С++. – СПб.: Питер, 2006. – 411 с.

УДК 510

Савченко Р.Я.

Научный руководитель: к.ф.-м.н., доцент Лебедь С.Ф.

ИНТЕГРАЛЫ ЭЙЛЕРА

Гамма-функция, или эйлеров интеграл второго рода, определяется формулой

$$\Gamma(p)=\int_0^\infty e^{-x}x^{p-1}dx \tag{1}$$

Этот интеграл является несобственным, так как верхний предел бесконечен. Кроме того, при *p-1<0* подынтегральная функция не ограничена в окрестности точки *x=0*. Интеграл (1) сходится при *p>0*. *Каждому положительному значению p соответствует* вполне определенное значение Г(р). Функция Г(р) не является элементарной.

С помощью метода интегрирования по частям докажем, что

$$\Gamma(p+1)=p\cdot\Gamma(p). \tag{2}$$

Имеем $\Gamma(p+1)=\int_0^\infty e^{-x}x^p dx=\begin{bmatrix} u=x^p & du=px^{p-1} \\ dv=e^{-x}dx & v=-e^{-x}\end{bmatrix}=-e^{-x}x^p\Big|_0^\infty+p\int_0^\infty e^{-x}x^{p-1}dx=p\Gamma(p)$,

так как при $p>0$ $\lim\limits_{x\to\infty}\frac{x^p}{e^x}=0$. Что и требовалось доказать.

При *p* = 1 интеграл находится непосредственно:

$$\Gamma(1)=\int_0^\infty e^{-x}dx=-e^{-x}\Big|_0^\infty=1.$$

Подставляя в формулу (2) значения *p* = 1, 2..., n, получаем Г(2) = 1·Г(1) = 1 = 1!, Г(3) = = 2Г(2) = 2·1 = 2!, Г(4) = 3·Г(3) = 3·2·1 = 3!,

$$\Gamma(n+1)=n\cdot(n-1)\ldots2\cdot1=n!. \tag{3}$$

Итак, при натуральных значениях аргумента гамма-функция совпадает с факториалом, т.е. с функцией $f(n)=n!$. Но гамма-функция определена не только при натуральных *n*, но и при любых положительных значениях аргумента. Из формулы (3) следует, что можно считать 0! = Г(1) = 1.

Гамма-функция определяется и при отрицательных значениях *p*. В этом случае необходимо применить формулу (3),переписав её в виде

$$\Gamma(p)=\frac{\Gamma(p+1)}{p}. \tag{4}$$

Если *-1<p<0* то *0<p+1<1*, поэтому правая часть формулы (4) имеет смысл, ею и определяется Г(*p*) при этих значениях р ; отметим, что в таком случае Г(*p*)<0. Продолжая аналогичные рассуждения, убеждаемся в том, что гамма-функция определена для всех отрицательных значениях *p*, *p* = -*k*, где *k* = 0, 1, 2,... и кроме *p* = 0.