УДК 512.152

В.А. ПОПОВ, А.К. КАЙДАЛОВ

Национальный аэрокосмический университет им. Н.Е. Жуковского "ХАИ", Украина

МЕТОД РЕШЕНИЯ СИСТЕМ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ ДЛЯ АНАЛИЗА НЕСТАЦИОНАРНЫХ ПРОЦЕССОВ В СИСТЕМАХ ОБСЛУЖИВАНИЯ

Предложен метод получения полуаналитического решения системы уравнений для СМО вида M|M|N|k|m|FIFO. Метод реализован в созданном программном продукте. В качестве результата выдаются полуаналитические выражения для вероятностей состояний, диаграмма переходов и графики зависимостей $P(t)$. Метод может быть использован для быстрого анализа СМО до $\max(N, k, m)=5$.

система массового обслуживания, граф состояний и переходов, вероятности состояния, полуаналитическое решение систем дифференциальных уравнений

Введение и постановка задачи

Системы массового обслуживания достаточно обширно распространены, они применяются в АТС (автоматических телефонных станциях), в сетях мобильной связи, в Интернете. Любой более или менее крупный сервер в Сети – это пример СМО. Поэтому применение СМО – актуальная задача.

Данная работа рассматривает пример разработки такого подхода программного продукта для решения систем обыкновенных дифференциальных уравнений (ОДУ) с постоянными коэффициентами, описывающих процессы в системах массового обслуживания [1, 2].

Используемые математические методы таковы: способ решения систем ОДУ с постоянными коэффициентами, расчет детерминанта по строке, метод прямого поиска для корней уравнения, метод Крамера для решения системы алгебраических уравнений.

1. Метод с традиционной точки зрения не-программиста

С точки зрения не-программиста метод решения рассматриваемых систем ОДУ можно описать следующим образом [1].

Сначала составляется диаграмма состояний для данной системы. Затем из этой диаграммы нужно составить систему обыкновенных однородных дифференциальных уравнений с постоянными коэффициентами. Для этой системы должна быть составлена матрица, которая не содержит переменных, обозначающих искомые функции (P_0, P_1 и т.п.), а содержит лишь коэффициенты при них. Для этого система ОДУ приводится к следующему виду (здесь символы штрихов – это производные):

$$\begin{aligned} P_0' &= a_{00}P_0 + a_{01}P_1 + a_{02}P_2 + \ldots + a_{0n}P_n, \\ P_1' &= a_{10}P_0 + a_{11}P_1 + a_{12}P_2 + \ldots + a_{1n}P_n, \\ &\ldots \\ P_n' &= a_{n0}P_0 + a_{n1}P_1 + a_{n2}P_2 + \ldots + a_{nn}P_n. \end{aligned} \tag{1}$$

Нужно положить $P_i = \alpha_i \cdot e^{\beta t}$, подставить эти выражения в (1), перенести все слагаемые в правую часть и сократить на $e^{\beta t}$, что дает:

$$\begin{aligned} 0 &= (a_{00} - \beta)\alpha_0 + a_{01}\alpha_1 + a_{02}\alpha_2 + \ldots + a_{0n}\alpha_n; \\ 0 &= a_{10}\alpha_0 + (a_{11} - \beta)\alpha_1 + a_{12}\alpha_2 + \ldots + a_{1n}\alpha_n; \\ &\ldots \\ 0 &= a_{n0}\alpha_0 + a_{n1}\alpha_1 + a_{n2}\alpha_2 + \ldots + (a_{nn} - \beta)\alpha_n. \end{aligned} \tag{2}$$

После этого производится образование матрицы из коэффициентов в правой части системы уравнений. Детерминант этой матрицы должен быть равен нулю:

$$\begin{vmatrix} a_{00}-\beta & a_{01} & a_{02}\ldots & a_{0n} \\ a_{10} & a_{11}-\beta & a_{12}\ldots & a_{1n} \\ & \ldots & & \\ a_{n0} & a_{n1} & a_{n2}\ldots & a_{nn}-\beta \end{vmatrix} = 0;$$

Теперь нужно раскрыть детерминант (это достаточно трудоемкая работа, особенно для больших n), что приведет к алгебраическому уравнению степени $(n + 1)$ относительно β. Это уравнение имеет $(n + 1)$ корней, и выражение для P_i $(i = 0, \ldots, n)$ будет иметь $(n + 1)$ слагаемых, т.е. будет иметь вид

$$P_i = \sum \alpha_i \cdot e^{\beta t} = \alpha_i \cdot e^{\beta 0t} + \alpha_i' \cdot e^{\beta 1t} + \alpha_i'' \cdot e^{\beta 2t} + \ldots,$$

причем здесь символ-штрих – это не производная, а просто обозначение разных переменных. Учитывая, что рассматриваемые системы ОДУ описывают реальные процессы, а именно, переходные процессы в системах массового обслуживания, можно сделать следующие уточнения относительно корней:

1) все $(n + 1)$ корней являются вещественными;

2) один из корней равен нулю (слагаемое, соответствующее этому корню, отражает ход процесса при $t \to \infty$, т.е. стационарный процесс).

Используя утверждение (2), уравнение степени $(n + 1)$ можно превратить в уравнение степени n. В общем случае решение такого уравнения не может быть найдено аналитически, хотя при $n \le 4$ (т.е. вплоть до биквадратных уравнений), а также в ряде особых случаев это возможно. Для $n \le 4$ способы нахождения корней существуют для любых коэффициентов, под особыми же случаями имеются в виду случай с так называемыми "зеркальными" коэффициентами $(2n^4 + 5n^3 + 3n^2 + 5n + 2)$ и другие.

Если в ходе решения (после учитывания нулевого корня) получено уравнение общего вида (т.е. коэффициенты не обладают определенной закономерностью, которая могла бы дать возможность применить способы решения для особых случаев), и степень этого уравнения больше четвертой, решение не может быть найдено без применения численных методов. Этот случай не рассматривается в этом подразделе, он рассматривается ниже, с учетом автоматизации и исключения ручного счета. Здесь же далее полагается, что β_i найдены аналитически.

После нахождения β_i $(i = 0, \ldots, n)$, причем $\beta_0 = 0$, нужно вернуться к системе (2) и по очереди подставить в нее каждую из β. Так можно получить n систем уравнений (ведь всего имеется n корней β, вместе с нулевым), при помощи которых выразить все α через исходные коэффициенты a. А именно, подставляя очередное β_i в систему (2), мы получим очередные $\alpha_1,\ldots,\alpha_n$ (т.е. сначала α_i, потом α_i', α_i'' и т.д.). Для получения частного решения необходимо использовать начальные условия ($P_0(0) = 1$; $P_i(0) = 0$, $i \ne 0$) и условие нормировки ($\sum P_i = 1$ в любой момент времени t).

Действуя этим методом, получаем решение системы дифференциальных уравнений в виде:

$$\begin{aligned} P_0 &= \alpha_0 \cdot e^{\beta 0t} + \alpha_0' \cdot e^{\beta 1t} + \alpha_0'' \cdot e^{\beta 2t} + \ldots + \alpha_0'^{\ldots\prime} \cdot e^{\beta nt}; \\ P_1 &= \alpha_1 \cdot e^{\beta 0t} + \alpha_1' \cdot e^{\beta 1t} + \alpha_1'' \cdot e^{\beta 2t} + \ldots + \alpha_1'^{\ldots\prime} \cdot e^{\beta nt}; \\ &\ldots \\ P_n &= \alpha_n \cdot e^{\beta 0t} + \alpha_n' \cdot e^{\beta 1t} + \alpha_n'' \cdot e^{\beta 2t} + \ldots + \alpha_n'^{\ldots\prime} \cdot e^{\beta nt}. \end{aligned} \tag{3}$$

2. Метод с точки зрения программиста

Метод с точки зрения программиста во многом схож с методом, описанным выше. Однако у программиста есть неоспоримое преимущество, а именно, несвязанность аналитикой. К примеру, здесь все β находятся численно, и в этом случае уже не важно, какой степени получено уравнение. Далее будут в меньшей степени, чем в предыдущем методе, рассмотрены вопросы аналитической математики, но больше внимания будет уделено алгоритмизации специфических именно для этого варианта решения вопросов. Перейдем непосредственно к методу.

Пусть имеем СМО по схеме Кендалла, λ и μ, отсюда получаем диаграмму переходов (для этого алгоритмизирован способ из [2]).

Потом происходит получение самой системы ОДУ, а именно, расчет коэффициентов a_{ij}:

$$P_0' = a_{00} \cdot P_0 + a_{01} \cdot P_1 + a_{02} \cdot P_2 + ... + a_{0n} \cdot P_n;$$
$$P_1' = a_{10} \cdot P_0 + a_{11} \cdot P_1 + a_{12} \cdot P_2 + ... + a_{1n} \cdot P_n;$$
$$\dots \qquad (4)$$
$$P_n' = a_{n0} \cdot P_0 + a_{n1} \cdot P_1 + a_{n2} \cdot P_2 + ... + a_{nn} \cdot P_n.$$

Отсюда программа, как и не-программист, получает матрицу. Но теперь нет необходимости раскрывать детерминант вручную, а затем считать корни получившегося уравнения.

В широком смысле любой численный метод нахождения корней основан на следующем принципе: есть некоторая функция $f(x)$ (пусть переменная, которую мы ищем, называется x), которая должна быть равна нулю в значениях-корнях, и метод последовательно подставляет в нее одиночные значения-кандидаты переменной x, определяя, какие из кандидатов дают достаточно близкое к нулю (с некоторой точностью) значение функции $f(x)$. Эти кандидаты и объявляются корнями. Так как такое широкое понимание не накладывает никаких ограничений на саму функцию (лишь бы было известно, как посчитать значение функции в точке), был использован следующий прием: функцией был объявлен сам детерминант:

$$f(\beta) = \begin{vmatrix} a_{00} - \beta & a_{01} & a_{02} \dots & a_{0n} \\ a_{10} & a_{11} - \beta & a_{12} \dots & a_{1n} \\ & \dots & & \\ a_{n0} & a_{n1} & a_{n2} \dots & a_{nn} - \beta \end{vmatrix} = 0. \qquad (5)$$

Затем был реализован расчет детерминанта для любого значения β, и использован один из простейших численных методов нахождения корней с функцией-детерминантом. В предыдущем методе после этого этапа следует решение $(n+1)$ систем алгебраических уравнений для определения коэффициентов $\alpha_0, \dots, \alpha_n$. Здесь же выбран другой путь: вместо $(n+1)$ систем уравнений составлена и решается одна, но с соответствующим размером матрицы $(n+1)^2 \times (n+1)^2$, т.е число переменных – $(n+1)^2$. Эти переменные – это $\alpha_0,\dots,\alpha_n$ без штрихов, α_0', …, α_n', α_0'', …, α_n'', и т.д., до n штрихов. Всего – $(n+1)$ раз по $(n+1)$ переменных – т.е. $(n+1)^2$.

3. Реализация

Расчет детерминанта. Детерминант для программы представлен как матрица некоторого размера $m \times m$, содержащая строки (тип TMatrix). Тип переменной "строка" выбран для того, чтобы расширить универсальность программы, а именно, чтобы на вход можно было подавать матрицу в виде

```
-2*l-b;m;0;
2*l;-l-b-m;2*m;
0;l;-2*m-b;
```

В этом примере (он описывает систему массового обслуживания M|M|2|2|2) размер матрицы 3×3, `l` обозначает λ (интенсивность поступления заявок в систему), `m` – μ (интенсивность обработки), а `b` – β (очередной корень). Точки с запятыми нужны для обозначения конца каждого элемента в каждой строке (однако точки с запятыми встречаются лишь на входе, потом программа разделяет элементы строк и их отбрасывает). Удобство состоит в том, что программа сама распознает эти величины и подставляет вместо них нужные числа.

В качестве элемента матрицы может выступать произвольное алгебраическое выражение со знаками +, –, *, /, скобками и символами `l`, `m`, `b`, используется синтаксический анализатор (о его реализации написано ниже).

Сам же детерминант считается обычным способом, как сумма произведений всех элементов первой строки на соответствующие им алгебраические дополнения (функция Determ), с оптимизацией для умножения на ноль. Для расчета каждого элемента (функция Cell) вызывается синтаксический анализатор (функция Value).

Синтаксический анализатор. Синтаксический анализатор (функция Value) принимает на входе строку-выражение и выдает одно вещественное число – значение этого выражения. Выражение может содержать знаки +, –, *, /, скобки и символы `l`, `m`, `b`, причем последние будут заменены значениями соответствующих глобальных переменных λ, μ, β.

Непосредственно расчет происходит так: сначала для всех пар скобок вычисляется рекурсивно выражение в них (скобки могут вкладываться друг в друга), и этим значением заменяется совокупность символов в скобках вместе со скобками, затем обрабатываются знаки * и /, каждая пара вида *a*b* или *a*/*b* рассчитывается и заменяется значением, после чего следует просмотр слева направо и подсчет суммированием оставшегося выражения вида *a*+*b*–*c*–*d*+*e*, получившаяся сумма и подается на выход.

Поиск корней. Так как неизвестно, в каком промежутке на оси β (система координат такова: абсциссы – β, ординаты – *f*(β)) находятся корни, ось абсцисс просматривается от нуля в направлении –∞ с некоторым шагом. Для каждого из значений β вычисляется функция (которая вычисляет для этой точки детерминант), и, если ее значение достаточно близко к нулю, программа делает вывод об еще одном найденном корне. Симметричный поиск в обе стороны продолжается, пока число найденных корней меньше *n* (невещественные корни отсутствуют в силу того, что система ОДУ описывает реальные процессы в системах массового обслуживания). Как только число корней достигло *n*, поиск прекращается, и корни передаются дальше в основную программу.

Решение системы алгебраических уравнений. Система алгебраических уравнений решается методом Крамера. Выбор обусловлен тем, что метод Крамера нетребователен к матрице в отношении разреженности (в отличие от метода Гаусса, например). В данном же случае в матрицу входит большое количество нулей, поэтому применение метода Крамера оказывается более оправданным.

Сначала рассчитывается детерминант *D* матрицы, составленной из коэффициентов при переменных, затем для каждой переменной α вычисляется детерминант D_α матрицы, составленной из коэффициентов при переменных, но столбец, соответствующий переменной α, заменен столбцом свободных членов. После этого переменная α вычисляется как D_α / D, и значения переменных α передаются в основную программу.

4. Пример использования программы для анализа СМО M|M|4|4|4|FIFO

Рассмотрим СМО: M|M|4|4|4 (рис. 1) при

$$\lambda = 2,\ \mu = 3$$

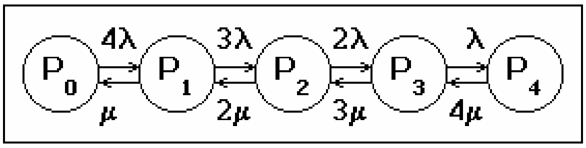

Рис. 1. Граф состояний системы M|M|4|4|FIFO

Исходная система дифференциальных уравнений:

$$P_0' = -8{,}000P_0 + 3{,}000P_1;$$

$$P_1' = 8{,}000P_0 - 9{,}000P_1 + 6{,}000P_2;$$

$$P_2' = 6{,}000P_1 - 10{,}000P_2 + 9{,}000P_3;$$

$$P_3' = 4{,}000P_2 - 11{,}000P_3 + 12{,}000P_4;$$

$$P_4' = 2{,}000P_3 - 12{,}000P_4.$$

Условие нормировки: сумма $P_i = 1$.

Начальные условия: $P_0(0) = 1$; $P_i(0) = 0$.

Функции вероятностей:

$$P_0 = 0{,}130 + 0{,}346e^{-5{,}00t} + 0{,}346e^{-10{,}00t} + 0{,}154e^{-15{,}00t} + 0{,}026e^{-20{,}00t};$$

$$P_1 = 0{,}346 + 0{,}346e^{-5{,}00t} - 0{,}231e^{-10{,}00t} - 0{,}358e^{-15{,}00t} - 0{,}102e^{-20{,}00t};$$

$$P_2 = 0{,}346 - 0{,}230e^{-5{,}00t} - 0{,}422e^{-10{,}00t} + 0{,}154e^{-15{,}00t} + 0{,}154e^{-20{,}00t}$$

$$P_3 = 0{,}154 - 0{,}358e^{-5{,}00t} + 0{,}154e^{-10{,}00t} + 0{,}154e^{-15{,}00t} - 0{,}102e^{-20{,}00t};$$

$$P_4 = 0{,}026 - 0{,}102e^{-5{,}00t} + 0{,}154e^{-10{,}00t} - 0{,}102e^{-15{,}00t} + 0{,}026e^{-20{,}00t}.$$

Графики зависимостей *P*(*t*) представлены на рис. 2.

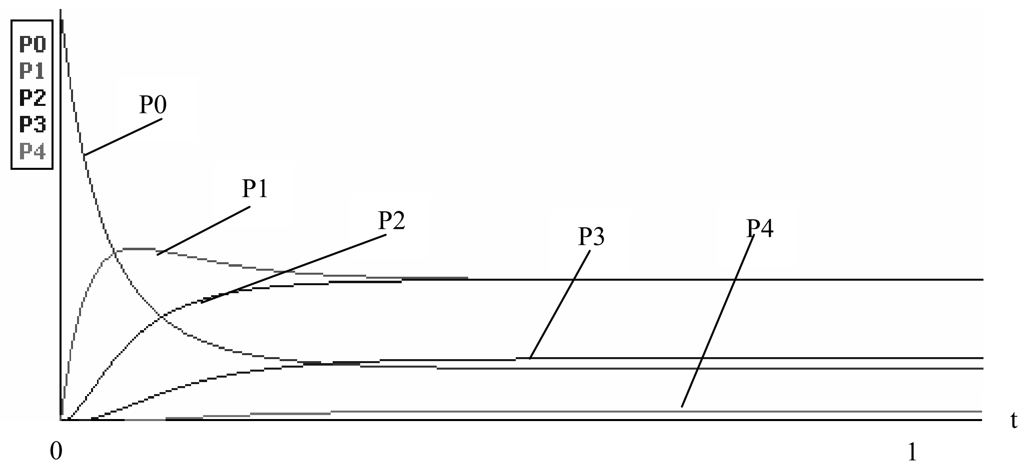

Рис. 2. График зависимостей вероятности состояний от времени

Заключение

Разработанный метод и приложение способны помочь специалисту в расчете характеристик СМО. Конечно, подразумевается компетентный специалист в этой области, знакомый с понятиями из области СМО и принципами их работы. В этом случае ему достаточно указать систему массового обслуживания по схеме Кендалла, а дальше программа рассчитает все сама.

Ограничения, накладываемые на область применения программы, состоят в невозможности расчета бесконечных систем дифференциальных уравнений, которые возникают при расчете СМО вида M|M|…|∞|∞ (т.е. когда буфер бесконечен и число источников также бесконечно). В остальных случаях расчет возможен и производится быстрее и эффективнее, чем это было бы сделано вручную. В перспективе возможно проведение оценок сложности алгоритма и проведение оптимизации.

Литература

1. Бочаров П.П., Печинкин А.В. Теория массового обслуживания. – М.: Изд-во Рос.ун-та Дружбы народов, 1995. – 236 с.

2. Лабскер Л.Г., Бабешко Л.О. Теория массового обслуживания экономической сферы. – М.: Банки и Биржи. Изд. объединения «Юнити», 1998. – 320 с.

3. Математические модели информационных процессов и управления / В.А. Попов, И.В. Дронова. Учеб. пособие. – Х.: Нац. аэрокосм. ун-т «ХАИ», 2001. – 59 с.

4. Основы теории алгоритмов и математических моделей / В.А. Попов, Н.В. Нечипорук, И.В. Дронова. – Х.: Нац. аэрокосм. ун-т «ХАИ», 2001. – 65 с.

Поступила в редакцию 31.05.2006

Рецензент: д-р техн. наук, проф. О.Е. Федорович, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков.