

ОПТИМИЗАЦИЯ СИСТЕМ МАССОВОГО ОБСЛУЖИВАНИЯ МЕТОДОМ СТАТИСТИЧЕСКИХ ИСПЫТАНИЙ

В.Г. Хлопьяк (Москва)

Введение

В реальном мире существует множество систем, математические модели которых могут быть представлены в виде систем массового обслуживания.

В то же время, применение аналитических методов для оптимизации данных систем в силу различных обстоятельств (дискретность параметров, целевая функция не дифференцируема и т.п.) не представляется возможным.

В свою очередь, применение статистических методов оптимизации ограничивается сложностью разработки и верификации имитационных моделей, а также их вычислительной эффективностью.

Поэтому рассмотрение вопросов оптимизации систем массового обслуживания методами статистических испытаний и вопросов разработки быстродействующих моделирующих алгоритмов систем является весьма актуальным.

Постановка задачи

В качестве примера, рассматривается однофазная многоканальная система массового обслуживания (СМО) с ограниченной входной очередью заявок.

На вход системы поступает простейший поток заявок с известной интенсивностью λ . Поступившая заявка либо попадает в очередь, либо, если все места в очереди заняты, сразу покидает систему.

Время обслуживания заявки $t_{обсл}$ – случайная величина с известным законом распределения.

Известна некоторая дисциплина (FIFO, LIFO и т.п.) обслуживания заявок и некоторая дисциплина («Первый свободный», «Равномерная загрузка» и т.п.) загрузки каналов.

Все заявки и каналы являются однотипными.

Известны затраты (в единицу времени) на обеспечение функционирования одного канала – c_k и одного места очереди – c_o .

Требуется найти оптимальное количество каналов обслуживания и оптимальное количество мест очереди, если доход от обслуживания одной заявки в единицу времени равен d_3 .

Для оценки эффективности функционирования СМО в установившемся режиме целевую функцию зададим в виде:

$$W(n, m) = d_3 N_{cp} - c_k n - c_o m \quad (1)$$

где W – чистый доход СМО в единицу времени;

$N_{cp} = f(n, m)$ – среднее число обслуживаемых заявок в единицу времени;

n – число каналов СМО;

m – число мест в очереди.

Таким образом, по результатам статистических испытаний необходимо найти оптимальное значение вектора параметров системы x^* , при которых достигается максимум целевой функции (1):

$$\begin{aligned} x^* &= (n^*, m^*); \\ W^* &= W(x^*) = \max_x W(x) \end{aligned} \quad (2)$$

При условии: $d_3 = const$ $c_k = const$ $c_o = const$

Алгоритм оптимизации СМО

Для нахождения оптимальных значений параметров СМО может быть использован любой из известных методов статистического поиска [1, 2]. При этом, значение целевой функции на каждом шаге оптимизации вычисляется статистическим моделированием СМО.

Алгоритм оптимизации состоит из двух частей:

- алгоритма случайного поиска, задающего начальные значения оптимизируемых параметров СМО и их изменение на следующем шаге поиска, исходя из значения целевой функции, полученной в результате статистического моделирования;
- моделирующего алгоритма имитационной модели, с помощью которого на каждом шаге поиска вычисляется значение заданной целевой функции.

Алгоритм случайного поиска

Для примера, выбран алгоритм поиска с наказанием случайность, являющийся аналогом метода наискорейшего спуска.

Суть рассматриваемого алгоритма поиска состоит в следующем:

1. Случайным образом с учетом области допустимых значений параметров системы ($dom(x)$) необходимо выбрать начальное значение вектора параметров системы x_0 и вычислить (методом статистических испытаний) начальное значение целевой функции $W(0) = W(x_0)$.

Принять: $W^* = W(x_0)$, $x^* = x_0$.

Задать начальные значения: $j = 0$; $b = 1$, где

j – номер направления поиска в точке локального экстремума;

b – масштаб изменения параметров.

2. Определить случайное направление поиска, вычислив единичный случайный вектор F_j :

$j = j + 1$;

$F_j = (j_{1,j}, j_{2,j})$;

$j_{i,j} = w_{i,j} / |W|$, $|W| = \sqrt{w_{1,j}^2 + w_{2,j}^2}$;

$w_{i,j}$ – случайная величина, равномерно распределенная на интервале $[-1, 1]$.

3. Вычислить значение вектора параметров системы:

$$x = x^* + [b \ Dx \ F_j] \quad (3)$$

где $Dx = (Dn, Dm)$ – «рабочий» шаг поиска по параметрам;

$[y]$ – ближайшее к y целое число.

Для всех параметров системы, которые выходят за границы области допустимых значений, присвоить соответствующие граничные значения, таким образом, чтобы $x \in dom(x)$.

Если $x - x^* = 0 \ \forall j = j_{\max}$, то поиск завершен, решение найдено.

j_{\max} – максимальное (заданное) количество направлений поиска в точке локального экстремума.

Если $x - x^* = 0 \ \forall j < j_{\max}$, то в выбранном направлении (j) поиск закончен; сбросить счетчик направлений ($j = 0$) и перейти к определению следующего

направления поиска в той же точке локального экстремума (перейти к пункту 2 алгоритма).

4. Статистическим моделированием системы (экспериментальным путем) получить значение $W(x)$.

5. Если $|W(x) - W^* / W^*|^3 \leq \epsilon$, то шаг поиска удачен, зафиксировать найденную точку локального экстремума: ($W^* = W(x), x^* = x$).

Если масштаб шага поиска не изменялся ($b=1$), то продолжить поиск в выбранном направлении (перейти к пункту 3 алгоритма), иначе изменить направление поиска восстановив максимальное значение масштаба поиска (присвоить $b=1$ и перейти к пункт 2 алгоритма).

Если $|W(x) - W^* / W^*| < \epsilon$, то следует продолжить поиск в текущем направлении с меньшим шагом ($b = b/2$ и перейти к пункту 3).

Приведенный алгоритм поиска позволяет найти локальный экстремум целевой функции. Для «надежности» получаемого решения поиск можно повторить несколько раз из различных начальных точек, выбрав среди найденных решений наилучшее.

Моделирующий алгоритм имитационной модели

Суть моделирующего алгоритма состоит в следующем.

Моделирующий алгоритм создается на основе агрегативного подхода [3] и состоит из двух частей: системной и прикладной.

Прикладная часть является собственно моделью системы, которая представляет собой множество элементов, моделируемых в виде «X-агрегатов» [4].

Системная часть предназначена для управления модельным временем и имитации функционирования системы в соответствии с дискретно событийным принципом моделирования систем.

Под событием понимается то, что может произойти в некоторый момент времени, рассматривается как изменение состояния системы и имитируется изменением состояний одного или нескольких агрегатов. Согласованное, одновременное изменение состояний агрегатов в необходимой логической последовательности достигается с помощью механизма сообщений.

Под сообщением понимается некоторая информация, которая должна быть передана «X-агрегату» в момент наступления события (оповещение агрегата о происходящем событии).

Все сообщения, за исключением начальных (запускающих процесс моделирования), могут быть созданы только агрегатами системы.

«X-агрегат» является объектом, относящимся к классу конечных автоматов – имеет конечное множество состояний и может мгновенно переходить из одного состояния в другое.

Отличие «X-агрегата» от конечного автомата состоит в том, что он имеет некоторое множество других свойств и может выполнять некоторые действия, необходимые для имитирования его функционирования (поведения).

Функционирование «X-агрегата» рассматривается как последовательность смены состояний и выполнении определенных действий под воздействием входных сообщений. В один момент времени «X-агрегата» может принимать и обрабатывать только одно сообщение. До окончания обработки текущего сообщения поступление других сообщений считается недопустимым.

Обработка сообщения заключается в определении реакции агрегата на полученное сообщение. Реакция агрегата задается в виде таблицы переходов и состояний.

С каждым состоянием агрегата связано некоторое уникальное действие, которое выполняется при переходе в данное состояние. Действие представляет собой набор операций, в который могут входить операции создания сообщений (выходных сообщений), адресованных другим агрегатам системы или самому себе.

Так как для имитации события, как правило, необходимо одновременное изменение состояний нескольких агрегатов, то создается несколько сообщений, среди которых одно считается основным, а другие – вспомогательными.

В основном сообщении указывается момент времени, когда должно наступить событие. Во вспомогательных сообщениях время не задается, так как они связаны с текущим моментом времени. Например, для имитации функционирования рассматриваемой СМО достаточно имитировать наступление только двух событий: Событие 1 – «Пришла заявка» и Событие 2 – «Обработка заявки завершена».

Создаваемые сообщения размещаются соответственно в стеке будущих сообщений и в стеке текущих сообщений.

Принципиально выделение вспомогательных сообщений не является обязательным, однако их наличие существенно уменьшает сложность алгоритма и увеличивает его вычислительную эффективность, так как сообщения в стеке будущих сообщений должны быть отсортированы по времени наступления событий, а в стеке текущих сообщений нет.

Для управления модельным временем и реализации механизма передачи сообщений в имитационную модель включен специальный объект - диспетчер сообщений.

Алгоритм работы диспетчера сообщений заключается в следующем:

Если текущее модельное время меньше заданного времени моделирования, то выполнить действие 1, иначе – действие 5.

Действие 1. Проверить стек текущих сообщений на наличие в нем сообщений. Если сообщения есть, то перейти к действию 2, иначе – к действию 3.

Действие 2. Выбрать сообщение из стека текущих сообщений. Отправить сообщение адресату, указанному в нем, и получить ответ от адресата. Удалить выбранное сообщение из стека текущих сообщений. Перейти к действию 1.

Действие 3. Проверить стек будущих сообщений на наличие в нем сообщений. Если сообщения есть, то перейти к действию 4, иначе – к действию 5.

Действие 4. В стек будущих сообщений найти и выбрать сообщение с ближайшим событием. Перевести модельное время вперед на момент времени наступления события, указанного в сообщении. Отправить сообщение адресату, указанному в сообщении, и получить ответ от него. Удалить выбранное сообщение из стека будущих сообщений. Перейти к действию 1.

Действие 5. Моделирование завершено.

Реализация моделирующего алгоритма в виде программы для ЭВМ по сути является имитационной моделью системы, с помощью которой можно проводить численные эксперименты.

Диаграмма классов имитационной модели, рассматриваемой СМО, приведена на рис. 1. На диаграмме показаны только основные свойства и методы классов. Классы «Диспетчер сообщений», «Отложенные сообщения», «Текущие сообщения», «Правила переходов» и «X-агрегаты» предназначены для реализации системной части моделирующего алгоритма, которая является общей для моделей систем любой физической природы.

Прикладная часть моделирующего алгоритма реализована в виде классов «Источники заявок», «Заявки», «Каналы», «Диспетчеры очереди» и «Наблюдатели». Объекты прикладных классов являются наследниками объектов базового класса «X-агрегаты», в котором реализован весь функционал объектов, за исключением действий, выполняемых наследуемым объектом при смене состояния.

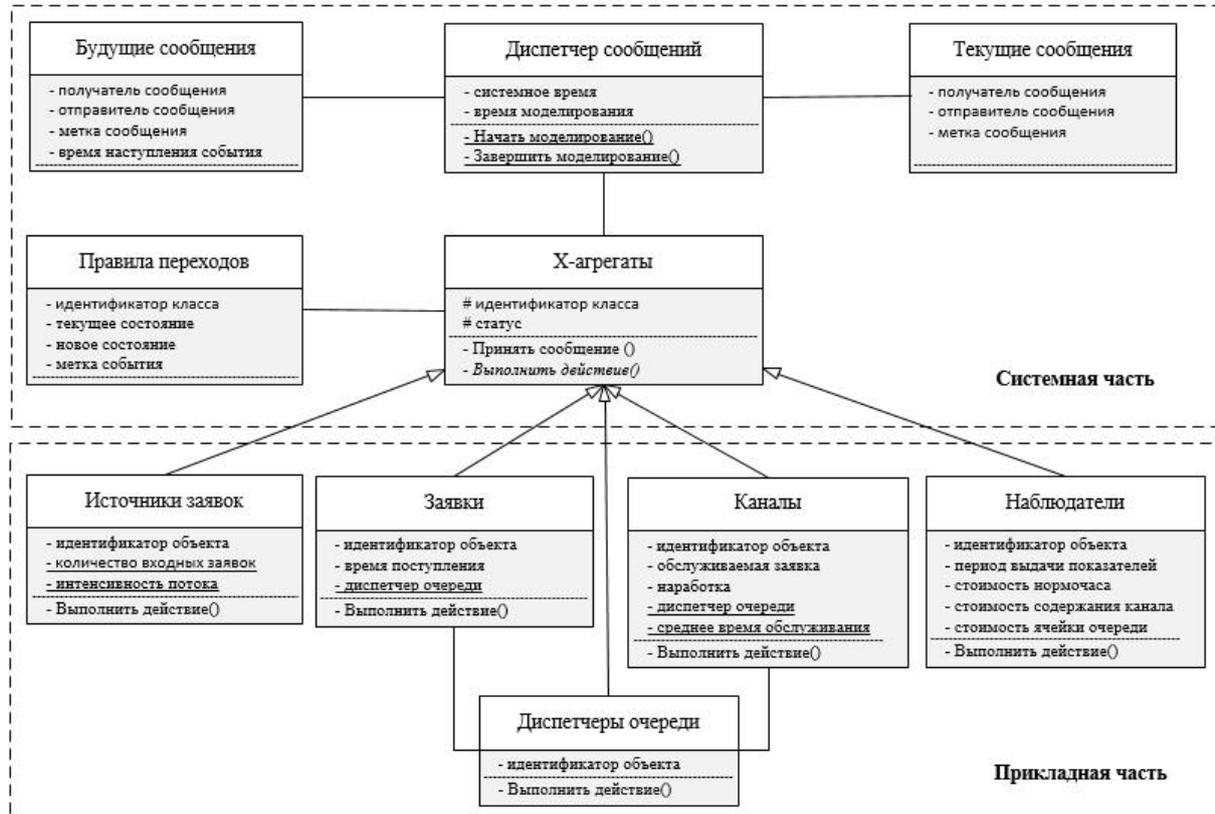


Рис. 1. Диаграмма классов имитационной модели

Объект класс «Источники заявок» предназначены для моделирования входного потока заявок. Объект класса через случайные интервалы времени получает основное сообщение «Создать заявку» и создает объекты класса «Заявки». Класс «Заявки» является единственным в данной модели, объекты которого создаются и уничтожаются в процессе моделирования.

Вычисление показателей функционирования системы, в том числе целевой функции, осуществляет объект класса «Наблюдатели».

Среднее число обслуживаемых заявок в единицу времени N_{cp} , необходимое для вычисления целевой функции рассчитывается по формуле:

$$N_{cp} = \dot{a} \sum_i^M t_{обс}(i) / T_{мод} = \dot{a} \sum_j^N t_{раб}(j) / T_{мод} \quad (4)$$

где M – число заявок, поступивших в систему;

$t_{обс}(i)$ – время обслуживания i -ой заявки;

N – число каналов обслуживания СМО;

$t_{раб}(j)$ – время наработки j -ого канала;

$T_{мод}$ – время моделирования СМО.

Подробно моделирующий алгоритм однофазной многоканальной системы с входной очередью заявок, разработанный на основе «X-агрегатов», приведен в работе [5].

Моделирующий алгоритм имеет высокую степень адекватности, так как достаточно просто может быть верифицирован [6].

Результаты статистической оптимизации СМО

Алгоритм поиска и моделирующий алгоритм могут быть реализованы с помощью любого из объектно-ориентированных языков программирования. По мнению автора, для этих целей целесообразно применять язык C++, как в целях решения прикладных задач, так и для обучения имитационному моделированию.

Реализация моделирующего алгоритма на языке C++ позволяет создавать имитационные модели, обладающие высокой вычислительной эффективностью. Так, для получения статистических показателей рассматриваемой СМО на персональном компьютере, оснащенном 8 ядерным процессором Intel Core i7 с тактовой частотой 3.40 GHz, требуется не более 400 миллисекунд (при интенсивности входного потока заявок $\lambda = 3.5$ и среднем времени обслуживания $t_{обсл} = 10$). Протокол работы имитационной модели СМО приведен на рис. 2.

```

ls
Start program Thu Sep  2 12:55:13 2021
Начальная точка: 1 (n:38, m:8)
time_CPU: 39ms  A: 3.4171  m_ср: 1.2276  n_ср: 34.122  W: 21.82  N_заявок: 35070  t_мог: 10000
time_CPU: 77ms  A: 3.4224  m_ср: 1.2714  n_ср: 34.167  W: 21.9  N_заявок: 70354  t_мог: 20000
time_CPU: 115ms A: 3.4091  m_ср: 1.2633  n_ср: 34.077  W: 21.739  N_заявок: 105161  t_мог: 30000
time_CPU: 154ms A: 3.4075  m_ср: 1.2666  n_ср: 34.089  W: 21.761  N_заявок: 140137  t_мог: 40000
time_CPU: 192ms A: 3.4033  m_ср: 1.2481  n_ср: 34.046  W: 21.683  N_заявок: 174926  t_мог: 50000
time_CPU: 230ms A: 3.4037  m_ср: 1.2569  n_ср: 34.082  W: 21.747  N_заявок: 209997  t_мог: 60000
time_CPU: 268ms A: 3.4063  m_ср: 1.2667  n_ср: 34.091  W: 21.764  N_заявок: 245254  t_мог: 70000
time_CPU: 306ms A: 3.4039  m_ср: 1.2676  n_ср: 34.101  W: 21.782  N_заявок: 280205  t_мог: 80000
time_CPU: 343ms A: 3.400  m_ср: 1.2628  n_ср: 34.089  W: 21.76  N_заявок: 315166  t_мог: 90000
time_CPU: 381ms A: 3.4017  m_ср: 1.2662  n_ср: 34.07  W: 21.725  N_заявок: 350007  t_мог: 1e+05
End program Thu Sep  2 12:55:13 2021
Для закрытия данного окна нажмите <ВВОД>...

```

Рис. 2. Протокол работы имитационной модели СМО

Вид целевой функции (1), полученный в результате статистического моделирования, приведен на рис. 3.

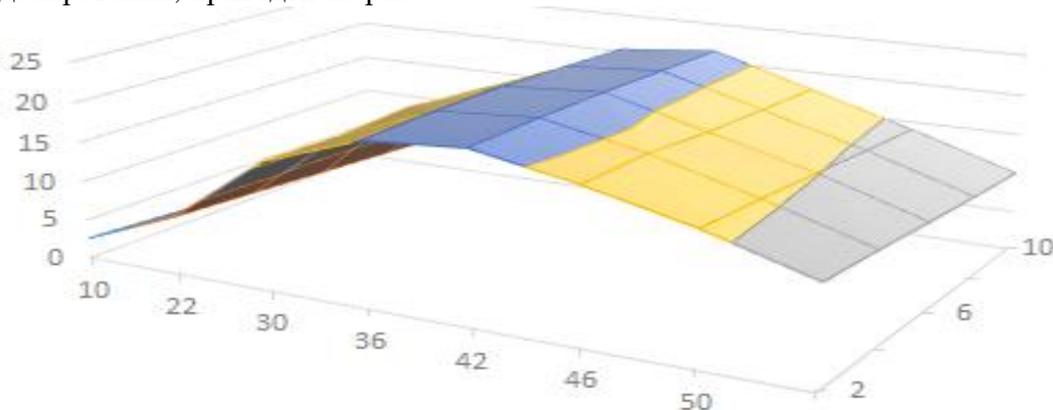


Рис. 3. Вид целевой функции

Для решения задачи оптимизации рассматриваемой СМО на основе приведенных выше алгоритмов требуется более 60 секунд процессорного времени. Протокол работы программы оптимизации (см. рис. 4) получен при следующих исходных данных:

- параметры целевой функции: $d_з = 1.8$; $c_к = 1.0$; $c_о = 0.2$;

- область допустимых значений параметров: $1 \leq n \leq 50$; $0 \leq m \leq 10$;
- «рабочий» шаг поиска по параметрам: $Dn = 12$, $Dm = 4$;
- критерий остановки поиска: $\epsilon = 0.005$;
- количество направлений поиска в точке: $J_{\max} = 8$;
- количество начальных точек поиска 3.

При оптимизации систем с большим количеством параметров время решения задачи можно сократить путем распараллеливания вычислений, как по начальным точкам, так и по параметрам системы.

```

Start program Thu Sep  2 14:55:32 2021
Начальная точка: 1 (n:38, m:5)
Точка экстремума. time_CPU: 380ms n_extr: 38 m_extr: 5 W_extr: 21.664
Точка экстремума. time_CPU: 2283ms n_extr: 34 m_extr: 9 W_extr: 22.619
Начальная точка: 2 (n:7, m:6)
Точка экстремума. time_CPU: 12532ms n_extr: 7 m_extr: 6 W_extr: 4.3999
Точка экстремума. time_CPU: 12755ms n_extr: 19 m_extr: 6 W_extr: 13.956
Точка экстремума. time_CPU: 13099ms n_extr: 31 m_extr: 6 W_extr: 21.765
Точка экстремума. time_CPU: 14211ms n_extr: 34 m_extr: 6 W_extr: 22.236
Точка экстремума. time_CPU: 16091ms n_extr: 35 m_extr: 6 W_extr: 22.357
Начальная точка: 3 (n:5, m:9)
Точка экстремума. time_CPU: 24241ms n_extr: 5 m_extr: 9 W_extr: 2.1999
Точка экстремума. time_CPU: 24411ms n_extr: 14 m_extr: 7 W_extr: 9.7982
Точка экстремума. time_CPU: 24680ms n_extr: 23 m_extr: 5 W_extr: 17.124
Точка экстремума. time_CPU: 25018ms n_extr: 32 m_extr: 3 W_extr: 21.465
Точка экстремума. time_CPU: 26082ms n_extr: 34 m_extr: 3 W_extr: 21.775
Точка экстремума. time_CPU: 28687ms n_extr: 37 m_extr: 5 W_extr: 21.935
Точка экстремума. time_CPU: 30468ms n_extr: 35 m_extr: 5 W_extr: 22.251
Точка экстремума. time_CPU: 34809ms n_extr: 33 m_extr: 7 W_extr: 22.379
Точка экстремума. time_CPU: 38196ms n_extr: 34 m_extr: 7 W_extr: 22.528
Точка экстремума. time_CPU: 47967ms W_extr: 22.619 n_extr: 34 m_extr: 9
End program Thu Sep  2 14:56:21 2021
Для закрытия данного окна нажмите <ВВОД>...

```

Рис. 4. Протокол работы программы оптимизации СМО

Вывод

Применение концепции «X-агрегата» и метода сообщений позволяет получить высокоэффективный моделирующий алгоритм, что существенно расширяет возможности применения имитационного моделирования для решения задач оптимизации сложных систем.

Литература

1. Растрингин Л.А. Статистические методы поиска. М., «Наука», 1968.
2. Лившиц А.Л., Мальц Э.А. Статистическое моделирование систем массового обслуживания. М., «Советское радио», 1978.
3. Бусленко Н.П. Моделирование сложных систем, М., «Наука», 1968.
4. Хлопяк В.Г. Развитие теории агрегативных систем – каноническая форма имитационной модели. Материалы конференции «ИММОД-2005», СПб. ФГУП ЦНИИТС, 2005.
5. Хлопяк В.Г. Моделирование систем массового обслуживания на основе X-агрегатов. Материалы конференции «ИММОД-2009», СПб. ФГУП ЦНИИТС, 2009.
6. Хлопяк В.Г. Верификация агрегативных систем. Материалы конференции «ИММОД-2007», СПб. ФГУП ЦНИИТС, 2007.