

## РЕШЕНИЕ ПРЯМОЙ И ОБРАТНОЙ ЗАДАЧ В СИСТЕМЕ МОДЕЛИРОВАНИЯ

В. Д. Боев (Санкт-Петербург)

При имитационном моделировании с использованием специальных инструментальных средств, например, GPSS World, в общем случае решаются две задачи. Назовем их прямой и обратной. Прямая задача заключается в нахождении оценки математического ожидания какого-либо показателя моделируемой системы при заданном времени ее функционирования. Обратная задача состоит в определении оценки математического ожидания времени функционирования моделируемой системы, за которое какой-либо ее показатель достигает заданного значения.

Решение этих задач, особенно обратной задачи, имеет свои особенности, о которых не говорится даже в [1, 2], и лишь упоминается в [3]. Рассмотрим эти особенности на простом примере.

**Пример.** Сервер обрабатывает запросы, поступающие с автоматизированных рабочих мест (АРМ) с интервалами, распределенными по показательному закону со средним значением  $T1 = 2$  мин. Вычислительная сложность запросов распределена по нормальному закону с математическим ожиданием  $S1 = 6 \cdot 10^7$  оп и среднеквадратическим отклонением  $S2 = 2 \cdot 10^5$  оп. Производительность сервера  $Q = 6 \cdot 10^5$  оп/с. В случае занятости сервера поступающий запрос теряется.

*Прямая задача.* Построить имитационную модель для определения оценки математического ожидания количества запросов (далее – количества запросов), обработанных сервером за время функционирования  $T = 1$  час, и оценки математического ожидания вероятности потерь запросов (далее – вероятности потерь запросов).

*Обратная задача.* Построить имитационную модель для определения оценки математического ожидания времени (далее – времени обработки), за которое будет обработано сервером  $N$  запросов, и оценки математического ожидания вероятности потерь запросов.

Рассчитаем количество прогонов, которые нужно выполнить в каждом наблюдении, т. е. проведем так называемое *тактическое планирование* эксперимента. Пусть результаты моделирования (вероятность потерь запросов) нужно получить с доверительной вероятностью  $\alpha = 0,95$  и точностью  $\varepsilon = 0,01$ . При  $\alpha = 0,95$  аргумент функции Лапласа  $t_\alpha = 1,96$ . Расчет проведем для худшего случая, т. е. при  $p = 0,5$ :

$$N = t_\alpha^2 \cdot \frac{p \cdot (1-p)}{\varepsilon^2} = 1,96^2 \cdot \frac{0,5 \cdot (1-0,5)}{0,01^2} = 3,8416 \cdot \frac{0,25}{0,0001} = 9604.$$

Программа модели для решения прямой задачи приведена ниже. Для задания исходных данных использованы переменные пользователя. Им даны такие же имена, как и в постановке задачи, но добавлен знак подчеркивания.

```

; Прямая задача
; Задание исходных данных
T1_   EQU   120
S1_   EQU   60000000
S2_   EQU   200000
Q_     EQU   600000
VrMod EQU   3600           ; Время моделирования, 1 ед. мод. времени = 1 с.
; Сегмент имитации обработки запросов
GENERATE (Exponential(1,0,T1_)) ; Источник запросов
KolZap GATE NU Server,PotZap    ; Свободен ли сервер? Если да, то
SEIZE   Server                 ; занять сервер
ADVANCE ((Normal(2,S1_,S2_))/Q_) ; Имитация обработки запроса
RELEASE Server                 ; Освободить сервер

```

```

ObrZap  TERMINATE                                ; Обработанные запросы
PotZap  TERMINATE                                ; Потерянные запросы
; Сегмент задания времени моделирования и расчета результатов
GENERATE  VrMod
TEST L    X$Prog,TG1,Met1                        ; Если X$Prog < TG1, то
SAVEVALUE Prog,TG1                              ; то X$Prog = TG1
Met1     TEST E    TG1,1,Met2                    ; Если TG1 = 1, то
SAVEVALUE VerObr, (N$ObrZap/N$KolZap)           ; расчет вероятности обработки запросов
SAVEVALUE Res,(INT(N$ObrZap/X$Prog))            ; расчет количества обработанных запросов
Met2     TERMINATE 1
START    1000,NP                                ; Прогоны до установившегося режима
RESET    ; Сброс накопленной статистики
START    9604
    
```

Рассчитанное количество обработанных запросов заносится в сохраняемую ячейку с именем Res

```
SAVEVALUE Res,(INT(N$ObrZap/X$Prog))
```

При расчете количества обработанных запросов в арифметическом выражении (INT(N\$ObrZap/X\$Prog)) используется число прогонов. В арифметическом выражении указано не конкретное число прогонов, а ссылка на сохраняемую ячейку X\$Prog. Число прогонов заносится предварительно в эту ячейку по завершении первого прогона модели, но до того момента, когда из содержимого счетчика завершений TG1 (системный числовой атрибут) будет вычтена первая единица. В этом случае арифметическое выражение не зависит от числа прогонов, которое может меняться на различных этапах создания и эксплуатации модели, в том числе и в зависимости от исходных данных, а также от точности и достоверности результатов моделирования. Поскольку количество обработанных запросов может быть дробным числом, то для получения целого числа используется процедура INT из встроенной библиотеки GPSS World.

Для уменьшения машинного времени расчет искомых показателей производится не после каждого прогона, а после завершения последнего прогона, то есть когда содержимое счетчика завершений будет равно единице:

```
Met1     TEST E    TG1,1,Met2
```

В результате решения прямой задачи получим, что за один час сервером будет обработано в среднем 16 запросов, а вероятность обработки составит 0,546. Если не использовать процедуру INT – выделения целого числа с отбрасыванием дробной части, будет обработано 16,345 запроса.

Для решения обратной задачи возьмем  $N = 16$ . Программа модели приведена ниже.

```

; Обратная задача
; Задание исходных данных
T1_     EQU    120
S1_     EQU    6000000
S2_     EQU    200000
Q_      EQU    600000
N_      EQU    16                                ; Количество запросов
; Сегмент имитации обработки запросов
GENERATE (Exponential(1,0,T1_))                 ; Источник запросов
KolZap  GATE NU  Server,PotZap                   ; Свободен ли сервер? Если да, то
        SEIZE   Server                           ; занять сервер
        ADVANCE ((Normal(2,S1_,S2_))/Q_)         ; Имитация обработки запроса
        RELEASE Server                           ; Освободить сервер
        TRANSFER ,ObrZap
    
```

---

```

PotZap  TERMINATE                                ; Потерянные запросы
; Сегмент задания времени моделирования и расчета результатов
ObrZap  TEST L      X$Prog,TG1,Met1              ; Если X$Prog < TG1, то
        SAVEVALUE   Prog,TG1                    ; то X$Prog = TG1
        SAVEVALUE   NZap,0                       ; Обнуление счетчика обработанных запросов
Met1     SAVEVALUE   NZap+,1                     ; Счет количества обработанных запросов
        TEST E      X$NZap,N_,Ter1              ; Если X$NZap = N_, то
        TEST E      TG1,1,Met2                  ; если TG1 = 1, то
        SAVEVALUE   VerObr,(N$ObrZap/N$KolZap)   ; расчет вероятности обработки запросов
        SAVEVALUE   TimeNZap,((AC1-X$AC2)/X$Prog) ; расчет времени обработки запросов
        SAVEVALUE   AC2,AC1                      ; Запомнить абсолютное модельное время
Met2     SAVEVALUE   NZap,0                       ; Обнуление счетчика обработанных запросов
        TERMINATE   1
Ter1     TERMINATE
        START       1000,NP                      ; Прогоны до установившегося режима
        RESET       ; Сброс накопленной статистики
        START       9604

```

При решении обратной задачи один прогон определяется заданным количеством запросов  $N_$ , а не временем моделирования. Для этого организован счетчик обработанных запросов в виде сохраняемой ячейки  $NZap$ . Как только содержимое  $X$NZap = N_$ , он обнуляется

```
Met2  SAVEVALUE  NZap,0
```

из счетчика завершений вычитается единица.

Время обработки заданного количества запросов сохраняется в ячейке с именем  $TimeNZap$ :

```
SAVEVALUE  TimeNZap,((AC1-X$AC2)/X$Prog)
```

Для расчета времени обработки заданного количества запросов используется арифметическое выражение  $((AC1-X$AC2)/X$Prog)$ . В состав этого выражения входят системный числовой атрибут  $AC1$  – абсолютное модельное время и опять количество прогонов. Запоминается количество прогонов также как и при решении прямой задачи. Кроме этого, в арифметическом выражении есть ссылка на сохраняемую ячейку  $X$AC2$ . Дело в том, что команда  $RESET$  не влияет на абсолютное модельное время. Время же выполнения 1000 прогонов до установившегося режима не должно участвовать в расчете. Поэтому оно запоминается, а затем вычитается из абсолютного модельного времени выполнения  $1000 + 9604 = 10604$  прогонов. Количество прогонов до установившегося режима может быть и другим.

Однако арифметическое выражение для расчета времени обработки заданного количества запросов может быть упрощено:

```
SAVEVALUE  TimeNZap,(C1/X$Prog)
```

где  $C1$  – относительное модельное время с момента последней команды  $RESET$ .

В этом случае нет необходимости в запоминании абсолютного модельного времени прогонов до установившегося режима, т. е. в строке программы

```
SAVEVALUE  AC2,AC1
```

Если не использовать команду  $RESET$ , арифметическое выражение для расчета времени обработки заданного количества запросов имеет вид

```
SAVEVALUE  TimeNZap,(AC1/X$Prog)
```

В результате моделирования получим время обработки 16 запросов: 3523,68 с. А почему не 3600 с? Ведь это же время моделирования было задано при решении прямой задачи. Потому что мы отбросили дробную часть, т. е. взяли 16, а не 16,345. Как же поступить, чтобы учесть и отброшенную дробную часть? Ведь в счетчике фиксируются обработанные запросы только целиком, а не по частям.

Для учета десятых долей дробной части зададим  $N_ = 163$ , т. е. увеличим в 10 раз. Это нужно учесть и в арифметическом выражении. Следовательно, оно будет таким:

```
SAVEVALUE TimeNZap ((C1/(X$Prog#Koeff))
```

Переменной пользователя Koeff задается значение 10. По завершении моделирования получим 3586,504. Этот результат уже ближе к 3600.

Для учета сотых долей дробной части зададим  $N_ = 1634$ , а Koeff = 100. Получим 3595,409 с.

Вероятность обработки запросов во всех случаях практически одна и та же, т. е. 0,546. Однако время моделирования существенно возрастает: 10 с, 1 мин 42 с и 17 мин, т. е. в 10 и 100 раз соответственно.

Аналогичным образом учитываются и другие доли дробной части.

Моделирование проводилось на компьютере с процессором Intel Pentium III с тактовой частотой 668 МГц. Естественно, что тенденция подобного увеличения машинного времени сохранится и на персональных компьютерах с большей производительностью.

Обратим внимание еще на одну особенность, характерную для задач массового обслуживания. Вероятность обработки запросов составляет 0,546, в то время как коэффициент загрузки сервера 0,454. Увеличить вероятность обработки запросов и коэффициент загрузки сервера можно за счет создания очереди. Если в рассмотренном примере ввести неограниченную очередь, вероятность обработки запросов составит единицу, а коэффициент загрузки сервера будет равен 0,831.

### Литература

1. Руководство пользователя по GPSS World. /Перевод с английского/ – Казань: Мастер Лайн, 2002. – 384 с.
2. Учебное пособие по GPSS World. /Перевод с английского/ – Казань: Мастер Лайн, 2002. – 270 с.
3. Боев В. Д. Моделирование систем. Инструментальные средства GPSS World: Учеб. Пособие. – СПб.: БХВ-Петербург, 2004. – 368 с.