Стратегическое и тактическое планирование модельного эксперимента при проведении оценки эффективности систем методом статистических испытаний в среде MATLAB

Цель работы: практическое изучение методов стратегического и тактического планирования модельного эксперимента, освоение навыков экспериментальных исследований при работе со статистическими имитационными моделями систем в ходе оценки их эффективности.

Работа выполняется в среде MATLAB и оформляется в виде m-файла сценария (script file), содержащего обращение к m-файлу функции, реализующей генерацию случайной величины, описывающую отклик системы в каждом эксперименте и имеющую определенный в конкретном задании вид плотности распределения вероятностей.

Работа состоит из двух частей: в первой части проводится ознакомление с возможностями стандартных функций, обеспечивающих разработку стратегического плана эксперимента и входящих в состав раздела Design of Experiments (планирование экспериментов) библиотеки Statistics Toolbox (набор инструментов статистического анализа) MATLAB. Во второй части осуществляется разработка и тестирование внешнего фрагмента имитационной модели, предназначенной для проведения оценки эффективности исследуемой системы по выбранному показателю методом статистических испытаний с оптимизацией объема испытаний в соответствии с основными соотношениями стратегического и тактического планирования. Для имитации функционирования системы разрабатывается отдельная m-функция, реализующая генерацию случайного отклика системы при каждом обращении в рамках проводимой совокупности испытаний.

Перед началом выполнения работы в соответствующем разделе создается рабочая папка. После запуска системы данная папка устанавливается в окне «Current Directory» путем выбора из списка рабочих папок файловой системы. Для этого используется кнопка «…», открывающая стандартное окно проводника файловой системы, в котором можно изменить текущий дисковый накопитель или раздел диска, а также войти в нужную директорию.

Для проведения моделирования в интересах отработки технологий оценки эффективности в рамках данной работы создается m-функция, реализующая имитацию статистического процесса функционирования исследуемой системы. В качестве подобного функционального эквивалента системы можно использовать генератор случайной величины с произвольным законом распределения. Параметры этого распределения выступают в роли факторов, а получаемая при обращении к функции случайная величина – в роли отклика системы в рамках единичного испытания прогона имитационной модели. В качестве примера можно рассмотреть m-функцию, реализующую генерацию логнормальной случайной величины с параметрами масштаба и формы a, b, которая имеет плотность распределения

$$f(u) = \frac{1}{ub(2\pi)^{1/2}} \exp\left[\frac{-\log(u/a)^2}{2b^2}\right],$$

причем математическое ожидание и дисперсия этой случайной величины равны

$$m = a\exp(0{,}5b^2), \quad D = a^2\exp(b^2)[\exp(b^2)-1]. \tag{6.1}$$

Определим m-функцию для генерации случайной величины в виде

```
function u=systemeqv(a,b);
%логнормальное распределение с параметрами масштаба и формы a, b
u=a*exp(b*randn);
```

Сохранив в рабочей папке соответствующий файл systemeqv.m, можно приступить к формированию m-файла сценария, реализующего выполнение сессии с целью отработки технологий стратегического и тактического планирования в интересах оценки эффективности моделируемой системы. Откроем этот файл под именем lab1.m.

Первоначально исследуем возможности стандартных m-функций раздела Dezign Experiment. Первой из них является функция fullfact, реализующая формирование полного многоуровневого факторного плана, исходя из количества факторов и количества уровней каждого фактора. Соответствующий фрагмент m-файла lab1.m, открывающий сессию, выглядит следующим образом:

```
%Стратегическое и тактическое планирование модельного эксперимента
clear all;
%задание количества факторов и диапазонов значений факторов
nf=3;
minf=[-5 -10 0]; maxf=[5 10 20];
%задание количества уровней каждого фактора
level=[3 3 2];
%формирование полного плана эксперимента
fullfact(level);
fulplan=ans
```

В результате в командном окне MATLAB появятся значения массива полного факторного плана

```
fulplan =

     1     1     1
     2     1     1
     3     1     1
     1     2     1
     2     2     1
     3     2     1
     1     3     1
     2     3     1
     3     3     1
     1     1     2
     2     1     2
     3     1     2
     1     2     2
     2     2     2
     3     2     2
     1     3     2
     2     3     2
     3     3     2
```

Здесь каждая строка отвечает одному эксперименту, в котором факторы принимают уровни, обозначенные номерами, определяемыми соответствующими элементами массива.

Далее следует сформировать массив исходных данных – значений уровней факторов в диапазонах выбранных значений, которые реально будут использоваться в ходе эксперимента

```
N=3*3*2;
for i=1:nf,
for j=1:N,
fuleks(j,i)=minf(i)+(fulplan(j,i)-1)*(maxf(i)-minf(i))/(level(i)-1);
end;
end;
fuleks
```

В командном окне соответственно появится результат

```
fuleks =

    -5    -10     0
     0    -10     0
     5    -10     0
    -5      0     0
     0      0     0
     5      0     0
    -5     10     0
     0     10     0
     5     10     0
    -5    -10    20
     0    -10    20
     5    -10    20
    -5      0    20
     0      0    20
     5      0    20
    -5     10    20
     0     10    20
     5     10    20
```

Значения уровней факторов для проведения эксперимента в результате устанавливаются симметрично относительно срединной точки диапазона значений.

Еще одна из имеющихся стандартных функций ff2n реализует формирование полного двухуровневого факторного плана, в котором уровни факторов (минимальные и максимальные значения в заданных диапазонах) обозначаются как 0 и 1. Продолжая далее сессию, выполним обращение к данной функции при заданном количестве факторов nf, получив при этом план ff2nplan, а затем определим соответствующий массив исходных данных для проведения эксперимента в виде значений факторов в выбранных диапазонах значений

```
%формирование полного двухуровневого плана эксперимента
ff2n(nf);
N=2^nf;
ff2nplan=ans
for i=1:nf,
for j=1:N,
fuleks2n(j,i)=minf(i)+ff2nplan(j,i)*(maxf(i)-minf(i));
end;
end;
fuleks2n
```

В командном окне появится

```
ff2nplan =

     0     0     0
     0     0     1
     0     1     0
     0     1     1
     1     0     0
     1     0     1
     1     1     0
     1     1     1
```

fuleks2n =

-5	-10	0
-5	-10	20
-5	10	0
-5	10	20
5	-10	0
5	-10	20
5	10	0
5	10	20

Следующий вариант реализации этапа стратегического планирования эксперимента предоставляет функция fracfact, обеспечивающая формирование дробного факторного плана или плана для оценки взаимодействий факторов. Обращение к ней выглядит следующим образом:

```
%формирование дробного двухуровневого плана эксперимента
N=2^nf;
fracfact('a b c ab bc ac abc' );
fracplan=ans
```

Здесь количество уровней и учитываемых взаимодействий факторов указывается непосредственно при обращении к функции. Результат выдается в виде массива fracplan со значениями элементов –1 и 1

fracplan =

-1	-1	-1	1	1	1	-1
-1	-1	1	1	-1	-1	1
-1	1	-1	-1	-1	1	1
-1	1	1	-1	1	-1	-1
1	-1	-1	-1	1	-1	1
1	-1	1	-1	-1	1	-1
1	1	-1	1	-1	-1	-1
1	1	1	1	1	1	1

Для последующего вычисления коэффициентов линейной регрессии в соответствии с соотношениями (2.3), (2.4) раздела 2.4 сформируем матрицу X с добавлением столбца значений фиктивного фактора $x_0 \equiv 1$.

```
%формирование транспонированной матрицы плана с добавлением
%фиктивного фактора
fictfact=ones(N,1);
X=[fictfact ans]';
```

Наконец, сформируем исходные данные для проведения эксперимента

```
fraceks=zeros(N,nf);
for i=1:nf,
for j=1:N,
fraceks(j,i)=minf(i)+(fracplan(j,i)+1)*(maxf(i)-minf(i))/2;
end;
end;
fraceks
```

и получим при этом следующий результат:

```
fraceks =

    -5   -10     0
    -5   -10    20
    -5    10     0
    -5    10    20
     5   -10     0
     5   -10    20
     5    10     0
     5    10    20
```

Здесь в каждой j-ой строке представлены значения уровней факторов, которые фиксируют исходные данные в j-ом эксперименте.

Выполним теперь в рамках данной сессии стратегическое планирование для определения уравнения регрессии при оценке эффективности эквивалентной системы systemeqv с учетом взаимодействия двух факторов a и b. В качестве показателя эффективности будем рассматривать дисперсию отклонения выдаваемого в каждой реализации отклика системы по отношению к истинному значению.

```
clear all;
nf=2;
minf=[1 0.5];
maxf=[5 1];
%формирование дробного двухуровневого плана эксперимента
%для учета взаимодействий
fracfact('a b ab' );
N=2^nf;
fracplan=ans
fictfact=ones(N,1);
X=[fictfact ans]'
fraceks=zeros(N,nf);
for i=1:nf,
for j=1:N,
fraceks(j,i)=minf(i)+(fracplan(j,i)+1)*(maxf(i)-minf(i))/2;
end;
end;
fraceks
```

Получим при этом план для проведения исследования эффективности системы по выбранному показателю.

```
fracplan =

    -1    -1     1
    -1     1    -1
     1    -1    -1
     1     1     1

X =

     1     1     1     1
    -1    -1     1     1
    -1     1    -1     1
     1    -1    -1     1
```

fraceks =

```
    1.0000    0.5000
    1.0000    1.0000
    5.0000    0.5000
    5.0000    1.0000
```

После этого можно выполнить тактическое планирование эксперимента с использованием соотношений раздела 2.5 при заданном уровне относительной ошибки оценки показателя эффективности $d_\sigma = 0.1$ и уровне значимости $\alpha = 0.05$. Для определения $t_{кр}(\alpha)$ воспользуемся стандартной функцией norminv, реализующей вычисление величины

$$t_{кр}(\alpha) = \Phi^{-1}(1-\alpha/2), \quad \Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-t^2/2} dt.$$

Соответствующий фрагмент m-файла имеет вид

```
%тактическое планирование эксперимента
%задание доверительного интервала и уровня значимости
d_sigma=0.1;
alpha=0.05;
%определение t-критического
tkr_alpha=norminv(1-alpha/2);
%определение требуемого числа испытаний
NE=round(1+2*tkr_alpha^2/d_sigma^2)
```

В результате получаем значение

```
NE =
    769
```

Далее проводим полный набор экспериментов в соответствии с планом, используя функциональный эквивалент системы – функцию systemeqv

```
%цикл по совокупности экспериментов стратегического плана
for j=1:N,
   a=fraceks(j,1);
```

```
    b=fraceks(j,2);
    %цикл статистических испытаний
    for k=1:NE,
        %имитация функционирования системы
        u(k)=systemeqv(a,b);
    end;
    %оценка параметров (реакции) по выборке наблюдений
        mx=mean(u);
        DX=std(u)^2;
    Y(j)=DX;
    %формирование и отображение гистограммы с 12-ю интервалами
    figure;
    hist(u,12);
end;
```

Для визуального анализа характера распределения случайной величины дополнительно построим гистограммы в различных точках факторного пространства, графики которых могут быть просмотрены после завершения моделирования.

Далее в соответствии с основными соотношениями раздела 2.4 определяется вектор коэффициентов регрессии

```
%определение коэффициентов регрессии
C=X*X';
b_=inv(C)*X*Y'
```

с результатом

```
b_ =

26.6395
24.8441
21.6868
20.2838
```

После этого целесообразно осуществить отображение полученной в ходе эксперимента зависимости показателя эффективности от выбранных факторов с использованием средств трехмерной графики MATLAB. Для этого формируется массив значений поверхности реакции Yc с использованием преобразования масштаба исходных значений факторов в значения, лежащие в диапазоне от –1 до 1. Одновременно представляет интерес построение реальной (истинной) поверхности реакции с использованием соотношения (6.1) для величины D в виде массива значений Yo. Соответствующая последовательность операторов имеет вид

```
%формирование зависимости реакции системы на множестве
%реальных значений факторов
A=minf(1):0.1:maxf(1);
B=minf(2):0.1:maxf(2);
[k N1]=size(A);
[k N2]=size(B);
for i=1:N1,
    for j=1:N2,
    an(i)=2*(A(i)-minf(1))/(maxf(1)-minf(1))-1;
    bn(j)=2*(B(j)-minf(2))/(maxf(2)-minf(2))-1;
    %экспериментальная поверхность реакции
    Yc(j,i)=b_(1)+an(i)*b_(2)+bn(j)*b_(3)+an(i)*bn(j)*b_(4);
    %теоретическая поверхность реакции
    Yo(j,i)=B(j)^2;%(A(i)^2)*exp(B(j)^2)*(exp(B(j)^2)-1);
end;
end;
% отображение зависимостей в трехмерной графике
[x,y]=meshgrid(A,B);
figure;
subplot(1,2,1),plot3(x,y,Yc),
xlabel('fact a'),
ylabel('fact b'),
zlabel('Yc'),
title('System output'),
grid on,
subplot(1,2,2),plot3(x,y,Yo),
xlabel('fact a'),
ylabel('fact b'),
zlabel('Yo'),
title('System output'),
grid on;
```

В результате получим отображение результатов моделирования, представленное на рис. 6.2, где слева размещается экспериментальная, а справа – реальная (теоретически рассчитанная) зависимость дисперсии отклика от исследуемых факторов.

Из приведенного рисунка видно, что полученная на основе линейной регрессии с учетом взаимодействия факторов зависимость приближенно отображает ход реальной зависимости. Подобное приближение будет тем хуже, чем больше диапазоны значений факторов. Этот вопрос может быть исследован самостоятельно.

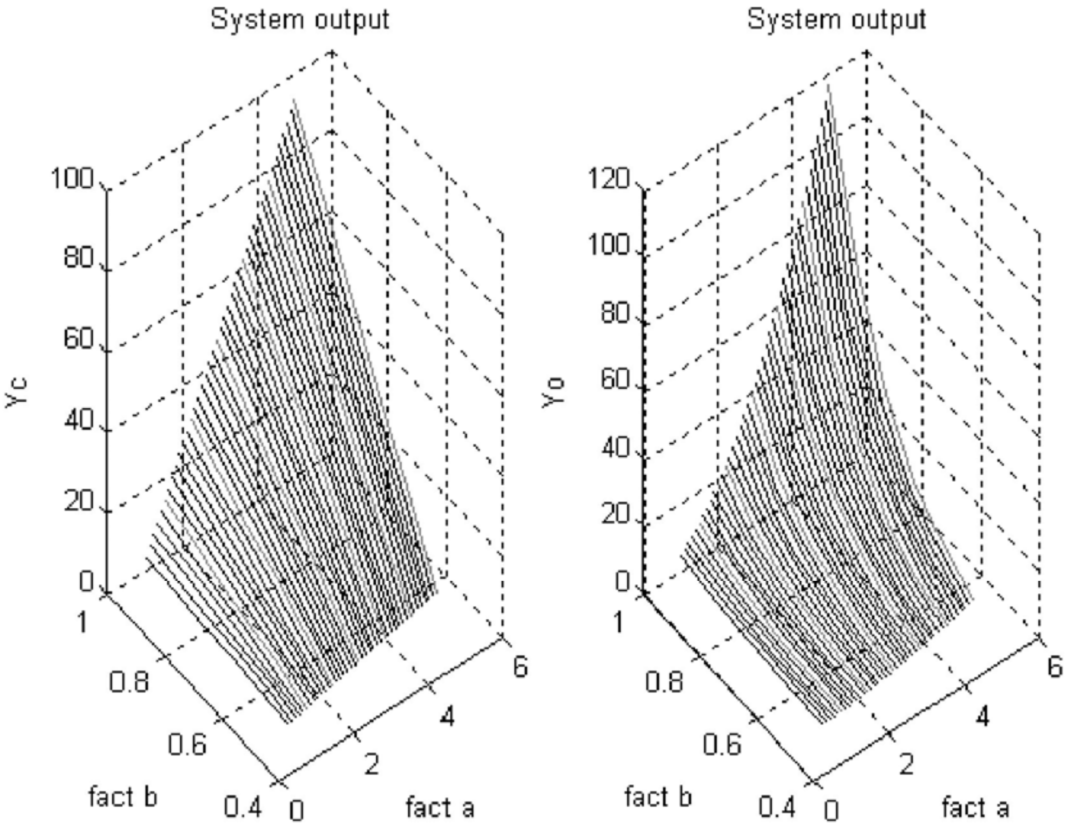

Рис. 6.2. Экспериментальная и реальная (теоретическая) зависимости реакции

Для сравнения на рис. 6.3 дан пример соответствующих зависимостей при использовании в качестве функции systemeqv генератора гауссовской случайной величины с параметрами $m = a$, $D = b^2$. Их анализ показывает, что в данном примере первый фактор не влияет на оцениваемый показатель эффективности и взаимодействие факторов отсутствует.

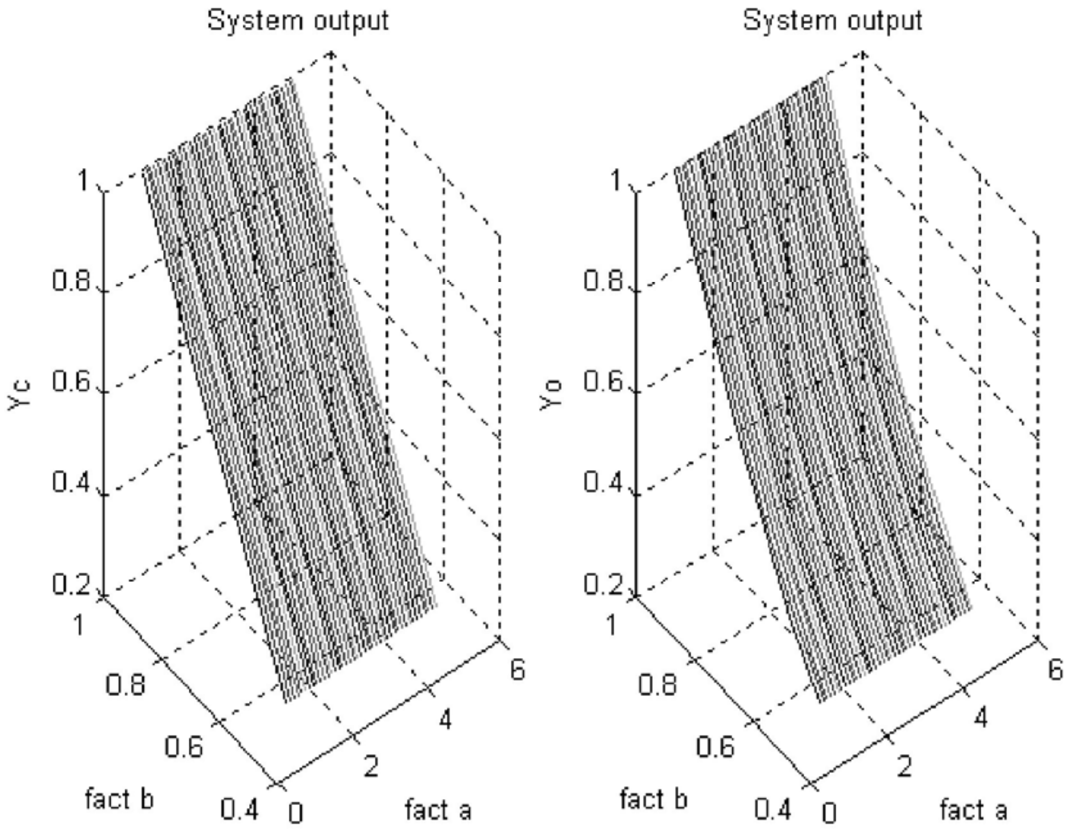

Рис. 6.3. Экспериментальная и реальная (теоретическая) зависимости реакции

Выполнение исследований в рамках данной работы может осуществляться с использованием распределений различных случайных величин в качестве датчиков отклика системы. В таблице 6.1 представлены наиболее распространенные распределения непрерывных случайных величин и алгоритмы их генерации с использованием стандартных датчиков равномерной и гауссовской случайных величин. Для имитации процесса функционирования системы с большим количеством факторов можно использовать различные композиции представленных случайных величин (суммы, произведения, отношения и т.п.).

Таблица 6.1. Алгоритмы генерации случайных величин

№ п/п	Наименование	Обозначение, параметры масштаба и формы	Плотность распределения вероятностей и моменты m и D	Алгоритм генерации
1	Равномерное распределение	$R: a, b$	$f(x)=\begin{cases}1/b, & a \le x \le a+b,\\ 0, & x<a, \quad x>a+b,\end{cases}$ $m = a + b/2$, $D = b^2/12$	$R \sim a + b[R^{(0)}: 0{,}1]$ $R^{(0)} = [R: 0{,}1]$ – стандартный датчик равновероятной случайной величины в диапазоне $[0, 1]$
2	Нормальное распределение	$N: \mu, \sigma$	$f(x)=\frac{1}{\sigma(2\pi)^{1/2}}\exp\left[\frac{-(x-\mu)^2}{2\sigma^2}\right]$, $m = \mu$, $D = \sigma^2$	$N: \mu, \sigma \sim (\sum_{i}^{12} R_i^{(0)} - 6)\sigma + \mu$.
3	Экспоненциальное распределение	$E: b$	$f(x)=\frac{1}{b}\exp(-\frac{x}{b})$, $m = b$, $D = b^2$	$E: b \sim -b\log R^{(0)}$

Таблица 6.1. Продолжение

№ п/п	Наименование	Обозначение, параметры масштаба и формы	Плотность распределения вероятностей и моменты m и D	Алгоритм генерации
4	Логнормальное распределение	$L:\mu,\sigma$	$f(x)=\frac{1}{x\sigma(2\pi)^{1/2}}\exp\left\{\frac{-[\log(x/\mu)]^2}{2\sigma^2}\right\}$, $m=\mu\exp(\frac{1}{2}\sigma^2)$, $D=\mu^2\exp^2(\sigma^2)(\exp^2(\sigma^2)-1)$	$L:\mu,.\sigma\sim\mu\exp(\sigma[N:0,1])$
5	Гамма-распределение	$\gamma:b,c$	$f(x)=(x/b)^{c-1}[\exp(-x/b)]/b\Gamma(c)$, $\Gamma(c)=\int_0^\infty \exp(-u)u^{c-1}du$ – гамма-функция, $m=bc$, $D=b^2c$	$\gamma:b,c\sim -b\log(\prod_{i=1}^{c}R_i^{(0)})$
6	Бета-распределение	$\beta:v,w$	$f(x)=x^{v-1}(1-x)^{w-1}/B(v,w)$, $B(v,w)=\int_0^1 u^{v-1}(1-u)^{w-1}du$ – бета-функция, $m=v/(v+w)$, $D=vw/(v+w)^2(v+w+1)$	$\beta:v,w\sim\frac{S_1}{S_1+S_2}$, $S_1=\left(R_1^{(0)}\right)^{1/v}$, $S_2=\left(R_2^{(0)}\right)^{1/w}$, $S_1+S_2\le 1$

Таблица 6.1. Продолжение

№ п/п	Наименование	Обозначение, параметры масштаба и формы	Плотность распределения вероятностей и моменты m и D	Алгоритм генерации
7	Распределение экстремального значения	$Xe : a, b$	$f(x) = (1/b)\exp[(x-a)/b]\exp\{-\exp[(x-a)/b]\}$, $m = a + b\,(-0{,}57721)$, $D = b^2\pi^2/6$	$Xe : a, b \sim a + b\log\left\|\log R^{(0)}\right\|$
8	Логистическое распределение	$Xl : a, k$	$f(x) = \frac{\exp[(x-a)/k]}{k\{1+\exp[(x-a)/k]\}^2}$, $m = a$, $D = k^2\pi^2/3$	$Xl : a, k \sim a + k\log[R^{(0)}/(1-R^{(0)})]$
9	Распределение Парето	$X : c$	$f(x) = cx^{-c-1}$, $c > 2$, $m = c/(c-1)$, $D = c/(c-2) - [c/(c-1)]^2$	$X : c \sim (1/R^{(0)})^{1/c}$

Таблица 6.1. Окончание

№ п/п	Наименование	Обозначение, параметры масштаба и формы	Плотность распределения вероятностей и моменты m и D	Алгоритм генерации
10	Распределение Вейбулла	$W: b, c$	$f(x) = (cx^{c-1}/b^c)\exp[-(x/b)^c]$, $m = b\Gamma[(c+1)/c]$, $D = b^2\{\Gamma[(c+2)/c] - \Gamma^2[(c+1)/c]\}$	$W: b, c \sim b(-\log R^{(0)})^{1/c}$
11	Распределение Эрлангта	$Er: b,\ c$ (e – целое)	$f(x) = \dfrac{(x/b)^{c-1}\exp(-x/b)}{b(c-1)!}$ $m = bc$, $D = b^2 c$.	$Er: b,\ c \sim -b\ \log\left[\prod_{i=1}^{c} R_i^{(0)}\right]$
12	Распределение Коши	$C: b,\ c$	$f(x) = 1/\pi b\{[(x-a)/b]^2 + 1\}$	$C: 0{,}1 \sim tg(2\pi R^{(0)})$, $C: 0{,}1 \sim [N_1 : 0{,}1]/[N_2 : 0{,}1]$