МИНИСТЕРСТВО НАУКИ И ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
КАФЕДРА ТПИ

Лабораторная работа 7,8

По курсу: Математические методы планирования эксперимента

Тема: прямая и двойственная градиентная процедура планирования D-оптимальных начальных условий

Факультет:	ПМИ
Группа:	ПММ-61
Студенты:	Холкин В. В.
	Кравченко О. П.
	Поверин Д. В.
Преподаватели:	Черникова О. С.
	Филиппова Е. В.

НОВОСИБИРСК
2017

Цель работы

Ознакомиться с прямой процедурой синтеза непрерывных планов эксперимента.

Теоретический материал

Градиентная процедура оптимизации

Оптимизационную задачу

$$\xi^* = \arg\min_{\xi\in\Omega_\xi} X[M(\xi)] \tag{1.25}$$

можно решать непосредственно (напрямую) с помощью общих методов численного поиска экстремума. Характерной особенностью этого подхода является большая размерность экстремальной задачи. Для выпуклого функционала $X[M(\xi)]$ имеем задачу выпуклого программирования, для решения которой предлагается следующий алгоритм:

Шаг 1. Зададим начальный невырожденный план

$$\xi_0 = \begin{Bmatrix} \alpha_1^0, \alpha_2^0, \ldots, \alpha_q^0, \\ p_1^0, p_2^0, \ldots, p_q^0 \end{Bmatrix}, \alpha_i^0 \in \Omega_\alpha, p_i^0 = \frac{1}{q}, i = 1,2,\ldots,q,$$

в котором $q = \frac{s(s+1)}{2} + 1$. Вычислим информационные матрицы $M\left(\alpha_i^0\right)$ одноточечных планов для $i = 1, 2, \ldots, q$ и по формуле (1.9) информационную матрицу всего плана ξ_0. Положим $l = 0$.

Шаг 2. Считая веса $p_1^l, p_2^l, \ldots, p_q^l$ фиксированными, решим оптимизационную задачу

$$X[M(\xi_l)] \to \min_{\alpha_1^l,\ldots,\alpha_q^l}, \ \alpha_i^l \in \Omega_\alpha, i = 1,2,\ldots,q.$$

Далее составим план

$$\breve{\xi}_l = \begin{Bmatrix} \alpha_1^{l+1}, \alpha_2^{l+1}, \ldots, \alpha_q^{l+1} \\ p_1^l, \ p_2^l, \ \ldots, p_q^l \end{Bmatrix},$$

где α_i^{l+1} - точки найденные на шаге 2. Вычислим $M\left(\alpha_i^{l+1}\right), i = 1,2,\ldots,q.$

Шаг 3. Зафиксируем точки спектра полученного плана и решим оптимизационную задачу

$$X\left[M\left(\breve{\xi}_l\right)\right] \to \min_{p_1^l, p_2^l, \ldots, p_q^l}, \ \sum_{i=1}^{q} p_i^l = 1, \ p_i^l \ge 0, \ i = 1,2,\ldots,q$$

Составим план

$$\xi_{l+1} = \begin{Bmatrix} \alpha_1^{l+1}, \alpha_2^{l+1}, \ldots, \alpha_q^{l+1} \\ p_1^{l+1}, p_2^{l+1}, \ldots, p_q^{l+1} \end{Bmatrix}.$$

Шаг 4. Если выполняется неравенство

$$\sum_{i=1}^{q} \left[\left\| \alpha_i^{l+1} - \alpha_i^l \right\|^2 + \left(p_i^{l+1} - p_i^l \right)^2 \right] \le \delta,$$

где δ - малое положительное число, перейдём на шаг 5. В противном случае для $l = l + 1$ повторим шаги 2 и 3.

Шаг 5. Проверим необходимое условие оптимальности плана.

$$\left|\mu\left(\alpha_i^{l+1}, \xi_{l+1}\right) - \eta\right| \le \delta,\ i = 1,2,\ldots,q.$$

Соответствие значений параметров $X[M(\xi)]$, $\mu(\alpha,\xi)$, η прямой процедуры критериям A- и D- оптимальности такое же, как в табл.1.1.

Если требуемое условие оптимальности выполняется, закончим процесс. В противном случае повторим всё сначала, скорректировав начальное приближение ξ_0.

Приведённый алгоритм требует вычисления градиентов критерия оптимальности $X[M(\xi)]$ по точкам спектра входного сигнала, а также по весам.

Вычисление градиентов критериев оптимальности

Ранее мы предложили две принципиально разные градиентные процедуры построения непрерывных оптимальных планов. Приведем вычисление градиентов критериев оптимальности. Начнем с прямой процедуры планирования входных сигналов. Она требует вычисления следующих градиентов:

$$\nabla_{\tilde{A}} X\left[M\left(\xi\right)\right] = \left\| \frac{\partial X\left[M\left(\xi\right)\right]}{\partial x_j^{(i)}\left(0\right)} \right\|;\ i = 1,\ldots,q; t = 0,\ldots,N-1; j = 1,\ldots,n,$$

i – номер наблюдения в плане, q – число точек в плане, j – номер компоненты вектора x(0),

и

$$\nabla_{\tilde{p}} X[M(\xi)] = \left\| \frac{\partial X[M(\xi)]}{\partial p_i} \right\|;\ i = 1,\ldots,q.$$

Начнем с критерия D – оптимальности. Для него

$$\frac{\partial X\left[M\left(\xi\right)\right]}{\partial x_j^{(i)}\left(0\right)} = -Sp\left[M^{-1}\left(\xi\right)\frac{\partial M\left(\xi\right)}{\partial x_j^{(i)}\left(0\right)}\right] =$$

$$-Sp\left[M^{-1}(\xi)p_i\frac{\partial M(U_i)}{\partial x_j^{(i)}(0)}\right] = -p_i Sp\left[M^{-1}(\xi)\frac{\partial M(U_i)}{\partial x_j^{(i)}(0)}\right].$$

Поскольку

$$\frac{\partial M\left(U\right)}{\partial x_j\left(0\right)} = \left\|\frac{\partial M_{\alpha\beta}\left(U\right)}{\partial x_j\left(0\right)}\right\|,\ \alpha,\beta = 1,\ldots,s$$

для вычисления производных по компонентам вектора начальных условий воспользуемся соотношением:

$$\frac{\partial \mathrm{M}_{\alpha\beta}(U;\Theta)}{\partial x_j(0)} =$$

$$= \sum_{t=0}^{N-1} \left\{ Sp\left[C_0 \left(\frac{\partial \overline{x}_A(t+1|t)}{\partial x_j(0)} \overline{x}_A^T(t+1|t) + \overline{x}_A(t+1|t) \frac{\partial \overline{x}_A^T(t+1|t)}{\partial x_j(0)} \right) C_0^T \frac{\partial \mathrm{H}^T}{\partial \theta_\beta} \mathrm{B}^{-1}(t+1) \frac{\partial \mathrm{H}}{\partial \theta_\alpha} \right] + \right.$$

$$+ Sp\left[C_0 \left(\frac{\partial \overline{x}_A(t+1|t)}{\partial x_j(0)} \overline{x}_A^T(t+1|t) + \overline{x}_A(t+1|t) \frac{\partial \overline{x}_A^T(t+1|t)}{\partial x_j(0)} \right) C_\beta^T \mathrm{H}^T \mathrm{B}^{-1}(t+1) \frac{\partial \mathrm{H}}{\partial \theta_\alpha} \right] +$$

$$+ Sp\left[C_\alpha \left(\frac{\partial \overline{x}_A(t+1|t)}{\partial x_j(0)} \overline{x}_A^T(t+1|t) + \overline{x}_A(t+1|t) \frac{\partial \overline{x}_A^T(t+1|t)}{\partial x_j(0)} \right) C_0^T \frac{\partial \mathrm{H}^T}{\partial \theta_\beta} \mathrm{B}^{-1}(t+1) \mathrm{H} \right] +$$

$$\left. + Sp\left[C_\alpha \left(\frac{\partial \overline{x}_A(t+1|t)}{\partial x_j(0)} \overline{x}_A^T(t+1|t) + \overline{x}_A(t+1|t) \frac{\partial \overline{x}_A^T(t+1|t)}{\partial x_j(0)} \right) C_\beta^T \mathrm{H}^T \mathrm{B}^{-1}(t+1) \mathrm{H} \right] \right\}.$$

Найдем производные $\dfrac{\partial \overline{x}_A(t+1|t)}{\partial x_j(0)}$:

$$\frac{\partial \overline{x}_A(t+1|t)}{\partial x_j(0)} = \begin{cases} j - \textit{му столбцу } \tilde{\Phi}_A, \quad \textit{если } t = 0, \\ \Phi_A(t+1,t) \dfrac{\partial \overline{x}_A(t|t-1)}{\partial x_j(0)}, \textit{ если } t > 0, \end{cases}$$

где

$$\tilde{\Phi}_A = \begin{bmatrix} \Phi \\ \dfrac{\partial \Phi}{\partial \theta_1} \\ \cdots \\ \dfrac{\partial \Phi}{\partial \theta_s} \end{bmatrix}$$

Алгоритм вычисления $\dfrac{\partial \mathrm{M}(U;\Theta)}{\partial x_j(0)}$ может быть следующим:

1^0. Для заданного θ найти:

$$\Phi,\ \frac{\partial\Phi}{\partial\theta_i},\ i=1,2,...,s; \qquad \Gamma, H,\ \frac{\partial H}{\partial\theta_i},\ i=1,2,...,s;$$

$$\Psi,\ \frac{\partial\Psi}{\partial\theta_i},\ i=1,2,...,s; \qquad Q;\ R;\ P(0), \bar{x}(0), \frac{\partial\bar{x}(0)}{\partial\theta_i},\ i=1,2,...,s.$$

Сформировать матрицы $\tilde{\Phi}_A$ и матрицу Ψ_A.

$$\tilde{\Phi}_A = \begin{bmatrix} \Phi \\ \dfrac{\partial\Phi}{\partial\theta_1} \\ \cdots \\ \dfrac{\partial\Phi}{\partial\theta_s} \end{bmatrix},$$

$$\Psi_A = \begin{bmatrix} \Psi \\ \dfrac{\partial\Psi}{\partial\theta_1} \\ \cdots \\ \dfrac{\partial\Psi}{\partial\theta_s} \end{bmatrix}.$$

2^0. Положить $\dfrac{\partial M(U;\Theta)}{\partial x_j(0)} = O$, $P(0|0) = P(0)$, t=0.

3^0. Вычислить $\dfrac{\partial \bar{x}_A(t+1|t)}{\partial x_j(0)}$ и $\bar{x}_A(t+1|t)$ по формуле

$$\bar{x}_A(t+1|t) = \begin{cases} \begin{bmatrix} \Phi \\ \dfrac{\partial\Phi}{\partial\theta_1} \\ \cdots \\ \dfrac{\partial\Phi}{\partial\theta_s} \end{bmatrix}\bar{x}(0) + \begin{bmatrix} 0 \\ \Phi\dfrac{\partial\bar{x}(0)}{\partial\theta_1} \\ \cdots \\ \Phi\dfrac{\partial\bar{x}(0)}{\partial\theta_s} \end{bmatrix} + \Psi_A u(0), & \text{если } t=0; \\ \Phi_A(t+1,t)\bar{x}(t|t-1) + \Psi_A u(t), & \text{если } t>0. \end{cases}$$

4^0. Найти $P(t+1|t), B(t+1)$, $K(t+1)$, $P(t+1|t+1), \tilde{K}(t+1)$, используя выражения фильтра Калмана.

5^0. Сформировать матрицу $\Phi_A(t+2, t+1)$ в соответствии с равенством

$$\Phi_A(t+2, t+1) =$$

$$= \begin{bmatrix} \Phi & O & . \quad . \quad . & O \\ \dfrac{\partial \Phi}{\partial \theta_1} - \tilde{K}(t+1)\dfrac{\partial H}{\partial \theta_1} & \Phi - \tilde{K}(t+1)H & . \quad . \quad . & O \\ . \quad . \quad . & . \quad . \quad . & . \quad . \quad . & . \quad . \quad . \\ \dfrac{\partial \Phi}{\partial \theta_s} - \tilde{K}(t+1)\dfrac{\partial H}{\partial \theta_s} & O & . \quad . \quad . & \Phi - \tilde{K}(t+1)H \\ & & & \end{bmatrix}.$$

6^0. Получить приращение $\Delta \dfrac{\partial M(U;\Theta)}{\partial x_j(0)}$, отвечающее текущему значению t.

7^0. Положить $\dfrac{\partial M(U;\Theta)}{\partial x_j(0)} = \dfrac{\partial M(U;\Theta)}{\partial x_j(0)} + \Delta \dfrac{\partial M(U;\Theta)}{\partial x_j(0)}$.

8^0. Увеличить t на единицу. Если $t \le N-1$, перейти на шаг 3^0. В противном случае закончить процесс.

Получить выражение для градиента по весам не составляет особого труда, поскольку

$$\frac{\partial X[M(\xi)]}{\partial p_i} = \frac{\partial[-\ln\det M(\xi)]}{\partial p_i} = -Sp\left[M^{-1}(\xi)\frac{\partial M(\xi)}{\partial p_i}\right] = -Sp\left[M^{-1}(\xi)M(U_i)\right],$$

$$i = 1,\ldots,q.$$

Двойственный алгоритм построения оптимальных сигналов требует вычисления градиента

$$\nabla_\alpha \mu(U,\xi) = \left\| \frac{\partial \mu(U,\xi)}{\partial x_j(0)} \right\|, t = 0,1,\ldots,N-1;\ j = 1,2,\ldots,n.$$

Для критерия D – оптимальности получаем:

$$\frac{\partial\mu(U,\xi)}{\partial x_j(0)}=\frac{\partial Sp\left[M^{-1}(\xi)M(U)\right]}{\partial x_j(0)}=Sp\left[M^{-1}(\xi)\frac{\partial M(U)}{\partial x_j(0)}\right];\ j=1,\ldots,n,$$

Критерий оптимальности

В данной работе рассматриваются D-оптимальные непрерывные планы.

Критерий	Параметры		
	$X[M(\xi)]$	$\mu[\alpha,\xi]$	η
D – оптимальности	$-\ln\det M(\xi)$	$Sp[M^{-1}(\xi)M(\alpha)]$	s

Исходная Модель

$$\begin{pmatrix} x_1(k+1) \\ x_2(k+1) \end{pmatrix}=\begin{pmatrix} \theta_1 & 1 \\ 0 & 0.5 \end{pmatrix}\begin{pmatrix} x_1(k) \\ x_2(k) \end{pmatrix}+\begin{pmatrix} \theta_2 \\ 1 \end{pmatrix}u(k)+\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} w_1(k) \\ w_2(k) \end{pmatrix}$$

$$y(k+1)=\begin{pmatrix} 1 & 0 \end{pmatrix}\begin{pmatrix} x_1(k+1) \\ x_2(k+1) \end{pmatrix}+\tau(k+1)$$

$$\begin{pmatrix} w_1(k) \\ w_2(k) \end{pmatrix}\succ N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix},\begin{pmatrix} 0.1 & 1 \\ 0 & 0.1 \end{pmatrix}\right)$$

$$\tau(k+1)\succ N(0,0.1)$$

$$\begin{pmatrix} x_1(0) \\ x_2(0) \end{pmatrix}\succ N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix},\begin{pmatrix} 0.1 & 1 \\ 0 & 0.1 \end{pmatrix}\right)$$

$$k=0,\ldots,N=30$$

Прямая процедура оптимизации

1 тест

Начальный план состоит из рандомизированных чисел на отрезке [-1,1].

$x0(k), k = \overline{1,n}$	p	X[M(ξ)]
0.4469	1	-13. 1696
0.4288		

Оптимальный план:

$x0(k), k = \overline{1,n}$	p	X[M(ξ)]
-5.0000	1	-13.6201
5.0000		

2 тест

Начальный план состоит из рандомизированных чисел на отрезке [-1,1].

$x0(k), k = \overline{1,n}$	p	X[M(ξ)]
0.6789	1	-13. 1696
0.2388		

Оптимальный план:

$x0(k), k = \overline{1,n}$	p	X[M(ξ)]
-5.0000	1	-13.6201
5.0000		

Двойственная процедура оптимизации

Начальный план состоит из рандомизированных чисел на отрезке [-1,1].

1 тест

Начальный план состоит из рандомизированных чисел на отрезке [-1,1].

<table>
<tr><th>$x0(k), k = 1, n$</th><th>p</th><th>X[M(ξ)]</th></tr>
<tr><td>0.5669</td><td rowspan="2">1</td><td rowspan="2">-13. 1696</td></tr>
<tr><td>0.8978</td></tr>
</table>

Оптимальный план:

<table>
<tr><th>$x0(k), k = 1, n$</th><th>p</th><th>X[M(ξ)]</th></tr>
<tr><td>-5.0000</td><td rowspan="2">1</td><td rowspan="2">-13.6201</td></tr>
<tr><td>5.0000</td></tr>
</table>

2 тест

<table>
<tr><th>$x0(k), k = 1, n$</th><th>p</th><th>X[M(ξ)]</th></tr>
<tr><td>0.4469</td><td rowspan="2">1</td><td rowspan="2">-13. 1696</td></tr>
<tr><td>0.4288</td></tr>
</table>

Оптимальный план:

<table>
<tr><th>$x0(k), k = 1, n$</th><th>p</th><th>X[M(ξ)]</th></tr>
<tr><td>-5.0000</td><td rowspan="2">1</td><td rowspan="2">-13.6201</td></tr>
<tr><td>5.0000</td></tr>
</table>

Выводы.

Программа работает правильно, что подтверждают результаты тестов.

Код программы.

```
global s_1 q u;
global e p n N s
delta = 0.01;
N=5;
n=2;
q=1;
delta = 0.01;
e = zeros(n,1);
u = zeros(N,1);
n=2;
% x_middle = [0; 0]
for i=1:n
    for j=1:size(e,2);
        e(i,j)=rand(1);
    end
end
x_middle = e;
for i=1:N
    for j=1:size(e,2);
        u(i,j)=5;%*rand(1);
    end
end

s_1 = size(e,1);
p = ones(1,size(e,2))/size(e,2);
options = optimset('Display','off');
delta = 0.01;
delta_2 = 0.0001;
Aeq = []; lb = []; ub = []; x0 = [];
q = size(e,2);
disp('Начальный  план');
disp(e);
disp(p);
N=N-1;
s_1 = size(e,1);
M = matrix_Fisher (e , p);
krA = -log(det(M));%trace(M^-1); % значение функционала начального плана
disp('значение функционала начального плана:'); disp(krA);

while 1
    % заполнение начального вектора и ограничений
    for i=1:q
       k = (i-1)*s_1+1;
       x0(:,k:k + (s_1-1)) = e(:,i)';
    end
    x0(:,s_1*q+1:q*(s_1+1))= p;
    %ограничения равенства
    Aeq = zeros(1, q*s_1); Aeq(:, q*s_1+1:q*(s_1+1)) = ones (1, q);
    beq = 1;
    %ограничения на x
    lb = -5 * ones(q*s_1,1); lb(q*s_1+1:q*(s_1+1), :) = zeros (q,1);
    ub = 5 * ones(q*s_1,1); ub(q*s_1+1:q*(s_1+1), :) = ones (q,1);
    [x,fval] = fmincon('fun',x0,[],[],Aeq,beq, lb,ub,[],options);
    disp('1')
    e0 = zeros(s_1,q);
    for i=1:q
        e0(:,i) =x((i-1)*s_1+1:i*s_1)';
    end
    p0=x(s_1*q+1:end);
    tmp_M = matrix_Fisher (e0 , p0); % вычисление матрицы Фишера от начального плана
    end_krA = -log(det(tmp_M));%;trace(tmp_M^-1); % значение функционала начального плана
    disp('значение функционала:'); disp(end_krA);

    M = matrix_Fisher (e , p);
    krA =  -log(det(M)); %trace(M^-1);
    nu = s_1;%trace(M^-1);
    e = e0; p = p0;
    %x_0 = [rand(1) rand(1) rand(1)]; %!
    x_0 = zeros(1, s_1); %!
    for i=1:s_1
        x_0(i) = rand(size(s_1));
    end
    lb1 = -5 * ones(s_1,1);
    ub1 = 5 * ones(s_1,1);
    [X_max,fl] = fmincon('fun_max', x_0,[],[],[],[],lb1, ub1,[],options);
    disp ('функция mu:'); disp (-fl);
    %disp ('n:'); disp (end_krA);
```

```
    if (abs (-fl - nu) < delta || abs (krA - end_krA) < delta_2  )
        break;
    end;
end;
[ e,p ] = cleaning(e,p, 0.001,0.001 );
disp('очистка плана');
M = matrix_Fisher (e , p);
krA =   -log(det(M)); %trace(M^-1);
disp('значение функционала после очистки:'); disp(end_krA);

disp('получение дискретного плана');
% disp(e);
% disp(p);
v = input('введите v:','s'); v = str2num(v);
dis = zeros (size(p));

k = around(v);
for i=1:(size(p,2))
    dis(i) = k(i)/v;
end
disp('Оптимальный план');
disp(e);
disp(dis);
```