

39. Новый метод утилизации изношенных шин. – Экспресс – информация. Шинная промышленность. – М., ЦНИИТЭНефтехим, 1989, № 8, с. 22 – 24.
40. Gummibereifung, 1985, Bd. 61, № 5, s. 28.
41. Довженко И.В., Регишевская И.Д. Термическое обезвреживание кислородсодержащих отходов, пути совершенствования, экологическая оценка метода. – Тезисы докладов совещания «Состояние и перспективы использования изношенных шин и полученных из них продуктов» (Чехов 1989).- М., ЦНИИТЭНефтехим, 1989, с. 28 – 30.
42. Макаров В.М., Дроздовский В.Ф. Использование амортизованных шин и отходов производства резиновых изделий. – Л., «Химия», Ленинградское отделение, 1986 249 с.

### **Модификация метода последовательного симплексного планирования и ее применение к решению задач оптимизации**

к.т.н. доц. Овсепян В.С., Дерцян А.С.

*Ванадзорский государственный педагогический институт им. Ованеса Туманяна  
(Армения)  
vardges1937@mail.ru*

*Аннотация.* Предложена модификация алгоритма Нелдера-Мида, при которой точка, определяющая направление отражения «наихудшей» вершины, выбирается с учетом значений минимизируемой функции в остальных вершинах симплекса. Исследована эффективность предложенной модификации на тестовых функциях.

*Ключевые слова:* симплекс, модификация, распределение, по времени, оптимизация.

#### **Предлагаемая модификация**

Метод симплексного планирования Нелдера-Мида является неградиентным методом нелинейной оптимизации для поиска минимума целевой функции  $f(\bar{x})$ ,  $\bar{x} \in R^n$ ,  $f: R^n \rightarrow R$ .

Симплексом в n-мерном евклидовом пространстве называется многогранник с количеством вершин n+1, не лежащих в одной гиперплоскости. Для случая n=2 – это треугольник, для случая n=3 – тетраэдр. Суть метода заключается в последовательном перемещении и деформировании симплекса вокруг точки экстремума с помощью процедур отражения, растяжения, сжатия и редукции симплекса [2-4]. Каждая итерация начинается с упорядочения вершин  $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_{n+1}$  таким образом, чтобы было выполнено условие:

$$f(\bar{x}_1) \leq f(\bar{x}_2) \leq \dots \leq f(\bar{x}_{n+1}). \quad (1)$$

С точки зрения минимизации  $\bar{x}_1$  считается наилучшей, а  $\bar{x}_{n+1}$  - наихудшей вершиной симплекса. Затем вычисляется центр тяжести  $\bar{x}_c$  первых n точек симплекса, где

$$\bar{x}_c = \frac{1}{n} \sum_{i=1}^n \bar{x}_i.$$

На линии  $\bar{x}_c + \alpha (\bar{x}_c - \bar{x}_{n+1})$ ,  $\alpha \in R$ , проходящий через точки  $\bar{x}_{n+1}$  и  $\bar{x}_c$ , рассматриваются точка отражения  $\bar{x}_r \in R^n$ , точки внутреннего и внешнего сжатия  $\bar{x}_{ic} \in R^n$  и  $\bar{x}_{oc} \in R^n$ , точка растяжения  $\bar{x}_e \in R^n$ . Исходя из соответствующих проверок,  $\bar{x}_{n+1}$  вершина симплекса заменяется одной из точек  $\bar{x}_r, \bar{x}_e, \bar{x}_{ic}, \bar{x}_{oc}$ , для которой целевая функция уменьшает свое значение (рисунок 1).

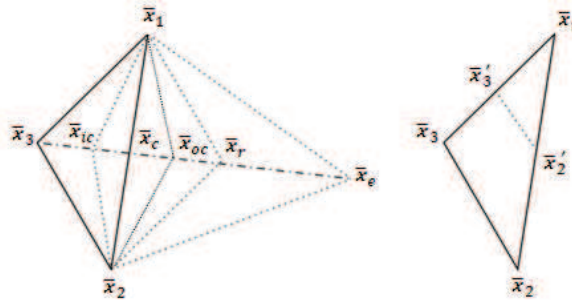


Рисунок 1

Если данный шаг проходит неудачно (точки с «лучшим» значением функции на линии отображения не найдено), симплекс редуцируется:  $\bar{x}_i = \bar{x}_1 + \sigma(\bar{x}_i - \bar{x}_1)$ ,  $\sigma = 0.5, i = 2, 3, \dots, n+1$ . Цикл поиска минимума целевой функции повторяется до тех пор, пока не выполняется критерий останова, например  $|f(\bar{x}_{n+1}) - f(\bar{x}_1)| < \varepsilon, \forall \varepsilon > 0$ . Если требуемая точность достигнута, выводится значение  $f(\bar{x}_1)$  как минимальное значение целевой функции.

В этой статье рассматривается модифицированный метод симплексного планирования, который для поиска минимума ряда тестовых функций требует меньше процессорного времени, чем исходной метод.

В методе Нелдера-Мида центр тяжести  $\bar{x}_c = \frac{1}{n} \sum_{i=1}^n \bar{x}_i$  равномерно удален от первых  $n$  вершин симплекса. Предлагаемая модификация основана на идее весовых коэффициентов. В этом случае по формуле (2) рассчитывается средневзвешенный центр грани, противостоящей «наихудшей» вершине.

$$\bar{x}'_c = \lambda_1 \bar{x}_1 + \lambda_2 \bar{x}_2 + \dots + \lambda_n \bar{x}_n, \quad (2)$$

где  $\sum_{i=1}^n \lambda_i = 1, \lambda_i > 0, i = \overline{1, n}$  (3)

Предлагается выбрать весовые коэффициенты  $\lambda_i, i = \overline{1, n}$  так, чтобы центр  $\bar{x}'_c$  был бы ближе к той вершине симплекса, по направлению к которой целевая функция быстрее убывает.

По формуле (4) сначала вычисляют весовые коэффициенты  $\mu_i, i = \overline{1, n}$ .

$$\mu_i = \frac{f(\bar{x}_{n+1}) - f(\bar{x}_i)}{\rho(\bar{x}_{n+1}, \bar{x}_i)}, \quad \text{где } \rho(\bar{x}_{n+1}, \bar{x}_i) = \sqrt{\sum_{j=1}^n (x_{n+1,j} - x_{i,j})^2} \quad (4)$$

Здесь  $\rho(\bar{x}_{n+1}, \bar{x}_i)$  – расстояние между вершинами  $\bar{x}_{n+1}$  и  $\bar{x}_i, i = \overline{1, n}$ , а числа  $(x_{i,1}, x_{i,2}, \dots, x_{i,n})$  – координаты  $\bar{x}_i, i = \overline{1, n+1}$ . Весовой коэффициент  $\mu_i$  есть относительное изменение целевой функции  $f(\bar{x})$ , на вершинах симплекса  $\bar{x}_{n+1}$  и  $\bar{x}_i$ . Чем больше это относительное изменение  $\mu_i$ , тем больше вероятность того, что значения целевой функции по направлению к вершине  $\bar{x}_i$  убывают быстрее. Целесообразно новую вершину искать по направлению к тем вершинам симплекса, для которых весовые коэффициенты  $\mu_i$  больше,  $i = \overline{1, n}$ . Отметим, что весовые коэффициенты  $\mu_i$  не удовлетворяют условию

$\sum_{i=1}^n \mu_i = 1$ , поэтому по формуле (5) рассчитывают из коэффициентов  $\mu_i$  новые нормированные весовые коэффициенты  $\lambda_i, i = \overline{1, n}$ .

$$\lambda_i = \mu_i / \mu, \quad i = \overline{1, n}; \quad \mu = \sum_{i=1}^n \mu_i. \quad (5)$$

Для всяких  $i, j$ , если  $\mu_i < \mu_j$ , то  $\lambda_i < \lambda_j$  и соотношение весовых коэффициентов не изменяется. Полученные коэффициенты  $\lambda_i$  полностью удовлетворяют условию (3). После вычисления весовых коэффициентов, определяется  $\bar{x}'_c$  средневзвешенный центр грани симплекса:

$$\bar{x}'_c = \lambda_1 \bar{x}_1 + \lambda_2 \bar{x}_2 + \dots + \lambda_n \bar{x}_n$$

Весовой центр  $\bar{x}'_c$  ближе к той вершине симплекса, для которой весовой коэффициент больше. Затем вершина  $\bar{x}_{n+1}$  заменяется точкой линии  $\bar{x}'_c + \alpha (\bar{x}'_c - \bar{x}_{n+1})$ ,  $\alpha \in \mathbb{R}$ , уменьшающей значение целевой функции. Данная модификация меняет направление поиска новой вершины симплекса, что повышает эффективность алгоритма. В качестве примера рассмотрим тестовую функцию “Trid”:  $f(\bar{x}) = f(x_1, x_2) = (x_1 - 1)^2 + (x_2 - 1)^2 - x_1 x_2$ .

Данная функция принимает свое минимальное значение  $f(\bar{x}^*) = -2$  на точке  $\bar{x}^* = (2; 2)$ . На рисунке 2 представлен симплекс с вершинами  $\bar{x}_1, \bar{x}_2, \bar{x}_3 \in \mathbb{R}^2$ , где  $\bar{x}_1 = (2.5; 0.3)$ ;  $\bar{x}_2 = (-1; 1.2)$ ;  $\bar{x}_3 = (0.6; -2.3)$ ;  $f(\bar{x}_1) = 1.99$ ;  $f(\bar{x}_2) = 5.24$ ;  $f(\bar{x}_3) = 12.43$ .

Поскольку  $f(\bar{x}_1) \leq f(\bar{x}_2) \leq f(\bar{x}_3)$ , то вершина  $\bar{x}_1$  считается наилучшей вершиной симплекса, а  $\bar{x}_3$  – наихудшей вершиной. Коэффициенты  $\mu_1, \mu_2$  будут рассчитываться следующим образом:

$$\mu_1 = \frac{f(\bar{x}_3) - f(\bar{x}_1)}{\rho(\bar{x}_3, \bar{x}_1)} \approx 3.24; \quad \mu_2 = \frac{f(\bar{x}_3) - f(\bar{x}_2)}{\rho(\bar{x}_3, \bar{x}_2)} \approx 1.87$$

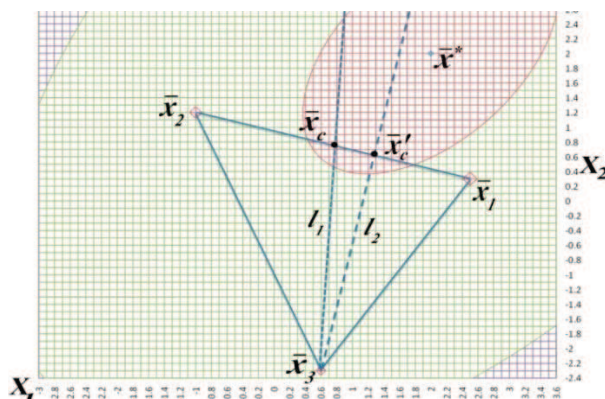


Рисунок 2

С помощью формулы (5) определяем коэффициенты  $\lambda_1, \lambda_2$ , для которых  $\lambda_1 + \lambda_2 = 1$ :

$$\lambda_1 = \mu_1 / (\mu_1 + \mu_2) = 3.24 / 5.11 \approx 0.634; \quad \lambda_2 = \mu_2 / (\mu_1 + \mu_2) = 1.87 / 5.11 \approx 0.366$$

Тогда  $\bar{x}'_c$  весовой центр симплекса будет определяться следующим образом:

$$\bar{\mathbf{x}}'_c = 0.634\bar{\mathbf{x}}_1 + 0.366\bar{\mathbf{x}}_2.$$

Из рисунка 2 видно, что полученный  $\bar{\mathbf{x}}'_c$  весовой центр симплекса находится ближе к наилучшей вершине  $\bar{\mathbf{x}}_1$  и к точке минимума целевой функции  $\bar{\mathbf{x}}^*$ , чем центр тяжести  $\bar{\mathbf{x}}_c$ , определяющийся методом Нелдера-Мида. Тем самым, с большой вероятностью можно сказать, что на точках линии  $l_2$  целевая функция примет меньшие значения, чем на точках линии  $l_1$ , (линия  $l_1$  проходит через точки  $\bar{\mathbf{x}}_c$  и  $\bar{\mathbf{x}}_3$ , а линия  $l_2$  - через точки  $\bar{\mathbf{x}}'_c$  и  $\bar{\mathbf{x}}_3$ ). Для достижения глобального минимума целевой функции нужно повторять процедуру поиска несколько раз.

### Описание алгоритма

**Шаг 1.1.** В области определения целевой функции  $f(\bar{\mathbf{x}})$ ,  $f: R^n \rightarrow R$  строят начальный регулярный симплекс  $S = \{\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, \dots, \bar{\mathbf{x}}_{n+1}\}$ . Для этого сначала случайным образом генерируют вершину  $\bar{\mathbf{x}}_1 \in R^n$ , а остальные вершины симплекса определяют следующим образом:

$$\bar{\mathbf{x}}_i = \bar{\mathbf{x}}_1 + h\bar{\mathbf{e}}_{i-1}, \quad i = \overline{2, n+1}, \quad h > 0,$$

где  $\bar{\mathbf{e}}_i \in R^n$ ,  $i = \overline{1, n}$  - единичные координатные векторы,  $h$  - ребро симплекса. Вычисляют значения целевой функции  $f(\bar{\mathbf{x}}_i)$ ,  $i = \overline{1, n+1}$ . Выбирают  $\varepsilon > 0$  число,  $\varepsilon = 10^{-8}$ .

**Шаг 1.2.** Вершины симплекса сортируют так, что  $f(\bar{\mathbf{x}}_1) \leq f(\bar{\mathbf{x}}_2) \leq \dots \leq f(\bar{\mathbf{x}}_{n+1})$ .

### Основной цикл

Пока  $|f(\bar{\mathbf{x}}_{n+1}) - f(\bar{\mathbf{x}}_1)| > \varepsilon$ , повторяется последовательность 2-6 шагов, в противном случае работа алгоритма завершается и выводится вершина симплекса  $\bar{\mathbf{x}}_1$  и значение  $f(\bar{\mathbf{x}}_1)$ .

**Шаг 2.** Определяют весовые коэффициенты  $\mu_i$ ,  $i = \overline{1, n}$ ,

$$\mu_i = \frac{f(\bar{\mathbf{x}}_{n+1}) - f(\bar{\mathbf{x}}_i)}{\rho(\bar{\mathbf{x}}_{n+1}, \bar{\mathbf{x}}_i)}, \quad \text{где } \rho(\bar{\mathbf{x}}_{n+1}, \bar{\mathbf{x}}_i) = \sqrt{\sum_{j=1}^n (x_{n+1,j} - x_{i,j})^2},$$

$\rho(\bar{\mathbf{x}}_{n+1}, \bar{\mathbf{x}}_i)$  - расстояние  $\bar{\mathbf{x}}_{n+1}$  и  $\bar{\mathbf{x}}_i$  вершин,  $(x_{i,1}, x_{i,2}, \dots, x_{i,n})$  - координаты вершин  $\bar{\mathbf{x}}_i$ ,  $i = \overline{1, n+1}$ .

**Шаг 3.** Нормированием коэффициентов  $\mu_i$  вычисляют новые весовые коэффициенты  $\lambda_i$ .

$$\lambda_i = \mu_i / \mu, \quad i = \overline{1, n}; \quad \mu = \sum_{i=1}^n \mu_i$$

Коэффициенты  $\lambda_i$  удовлетворяют условию  $\sum_{i=1}^n \lambda_i = 1$ ,  $\lambda_i > 0$ ,  $i = \overline{1, n}$ .

**Шаг 4.** Определяют  $\bar{\mathbf{x}}'_c \in R^n$  средневзвешенный центр первых  $n$  вершин симплекса.

$$\bar{\mathbf{x}}'_c = \lambda_1 \bar{\mathbf{x}}_1 + \lambda_2 \bar{\mathbf{x}}_2 + \dots + \lambda_n \bar{\mathbf{x}}_n$$

**Шаг 5.** На линии  $l_2 = \{\bar{\mathbf{x}}'_c + \alpha(\bar{\mathbf{x}}'_c - \bar{\mathbf{x}}_{n+1})\}$ ,  $\alpha \in R$ , определяют точку отражения  $\bar{\mathbf{x}}_r \in R^n$ .

$$\bar{\mathbf{x}}_r = \bar{\mathbf{x}}'_c + \alpha(\bar{\mathbf{x}}'_c - \bar{\mathbf{x}}_{n+1}), \quad \text{при } \alpha = 1.$$

Находят значение целевой функции  $f(\bar{\mathbf{x}}_r)$  на этой точке. Далее, как в методе Нелде-

ра-Мида, в зависимости от значения  $f(\bar{x}_r)$ , наихудшая вершина симплекса  $\bar{x}_{n+1}$  шагами 5.1-5.4 заменяется новой точкой, находящейся на линии  $l_2$ .

**Шаг 5.1.** Если  $f(\bar{x}_r) \in [f(\bar{x}_1), f(\bar{x}_n)]$ , то наихудшая вершина симплекса  $\bar{x}_{n+1}$  заменяется точкой отражения  $\bar{x}_r$ , делается переход на шаг 2.

**Шаг 5.2.** Если  $f(\bar{x}_r) < f(\bar{x}_1)$ , проводят операцию растяжения.

$$\bar{x}_e = \bar{x}'_c + \alpha(\bar{x}'_c - \bar{x}_{n+1}), \text{ при } \alpha = 2$$

Если  $f(\bar{x}_e) < f(\bar{x}_r)$ , то наихудшую вершину симплекса  $\bar{x}_{n+1}$  заменяют точкой  $\bar{x}_e$  и переходят на шаг 2. В противном случае вершину  $\bar{x}_{n+1}$  заменяют точкой  $\bar{x}_r$  и переходят на шаг 2.

**Шаг 5.3.** Если  $f(\bar{x}_r) \in [f(\bar{x}_n), f(\bar{x}_{n+1})]$ , делают операцию внешнего сжатия

$$\bar{x}_{oc} = \bar{x}'_c + \alpha(\bar{x}'_c - \bar{x}_{n+1}), \text{ при } \alpha = 0.5$$

Если точка  $\bar{x}_{oc} \in \mathbb{R}^n$  лучше, чем точка отражения, то есть  $f(\bar{x}_{oc}) \leq f(\bar{x}_r)$ , вершина  $\bar{x}_{n+1}$  заменяется точкой  $\bar{x}_{oc}$ , переходят на шаг 2. В противном случае переходят на шаг 6.

**Шаг 5.4.** Если  $f(\bar{x}_r) \geq f(\bar{x}_{n+1})$ , делается операция внутреннего сжатия

$$\bar{x}_{ic} = \bar{x}'_c + \alpha(\bar{x}'_c - \bar{x}_{n+1}), \text{ при } \alpha = -0.5$$

Если точка  $\bar{x}_{ic}$  лучше, чем вершина симплекса  $\bar{x}_{n+1}$ , то есть  $f(\bar{x}_{ic}) \leq f(\bar{x}_{n+1})$ , вершина  $\bar{x}_{n+1}$  заменяется точкой  $\bar{x}_{ic}$ , переходят на шаг 2. В противном случае переходят на шаг 6.

**Шаг 6.** Проводим операцию редукции симплекса.  $\bar{x}_i = \bar{x}_1 + y(\bar{x}_i - \bar{x}_1)$ ,  $y = 0.5$ ,  $i = 2, n+1$ .

### Тестирование предложенной модификации

Приведенные ниже тестовые функции оптимизации были протестированы методом Нелдера-Мида и предложенной модификацией весовых коэффициентов (ВК), каждый по 100 раз. Каждый раз первая вершина начального симплекса была выбрана из области поиска тестовой функции произвольным образом. В таблице 1 приведено сравнение методов Нелдера-Мида и предложенной модификации по среднему процессорному времени (CPU-сек.) и среднему числу расчета значений тестовой функции (вызовов функции) для одного теста. Тестирование проводилось на компьютере с 2,66 ГГц процессорной мощностью и 3 Gb оперативной памятью.

На рисунке 3 приведены данные о средних затратах процессорного времени для поиска минимума тестовых функций для обоих методов. После названия тестовой функции приведено число переменных данной функции. Для каждой тестовой функции отмечено, насколько сокращаются средние затраты процессорного времени при использовании модифицированного метода для поиска минимума целевой функции по сравнению с методом Нелдера-Мида.

Из рисунка 3 и таблицы 1 видно, что эффективность модифицированного метода возрастает при увеличении числа переменных тестовых функций. Например, для поиска минимального значения тестовой функции "Trid"  $\left( f(\vec{x}) = \sum_{i=1}^n (x_i - 1)^2 - \sum_{i=2}^n x_i x_{i-1} \right)$  средние затраты процессорного времени для  $n=2$  сокращаются на 2.08%-ов, для  $n=4$  – на 8.77%-ов, а для  $n=6$  – на 18.34%-ов.

Сравнение методов Нелдера-Мида и предложенной модификации (n – число переменных)

N	Название тестовой функции	n	Нелдер-Мид		Модификация ВК	
			CPU-сек.	Число выз. функции	CPU-сек.	Число выз. функции
1	Trid Function	2	0.048	101.58	0.047	102.25
2	Trid Function	4	0.171	285.69	0.156	253.26
3	Trid Function	6	0.529	567.08	0.432	471.5
4	Zakharov Function	2	0.048	109.69	0.046	108.75
5	Zakharov Function	4	0.181	291.62	0.163	263.14
6	Zakharov Function	6	0.583	619	0.479	515.71
7	Helical valley Function	3	0.131	255.49	0.141	281.52
8	Gaussian Function	3	0.123	183.54	0.113	177.2
9	Box 3 dim. Function	3	0.127	267.05	0.099	266.1
10	Colville Function	4	0.372	603.63	0.368	601.57
11	Branin Function	2	0.047	108.67	0.043	107.39
12	Sphere Function	3	0.081	165.87	0.076	159.01
13	Sphere Function	5	0.241	338.97	0.209	291.45
14	Sphere Function	10	2.347	1144.98	1.132	682.38
15	Sum Squares Function	3	0.087	177.16	0.079	168.29
16	Sum Squares Function	5	0.259	360.28	0.224	313.69
17	Sum Squares Function	10	2.939	1276.25	1.386	781.67
18	Rotated hyper-ellipsoid	3	0.106	198.44	0.105	193.21
19	Rotated hyper-ellipsoid	5	0.305	420.46	0.276	380.31

## Список тестовых функций

**1. Trid Function:**  $f(\vec{x}) = \sum_{i=1}^n (x_i - 1)^2 - \sum_{i=2}^n x_i x_{i-1}$ ,

Область поиска:  $-n^2 \leq x_i \leq n^2$ ,  $i = 1, 2, \dots, n$ .

Минимум функции ( $\vec{x}^*$  – точки минимума приведенных тестовых функций): при  $n = 2$ ,  $f(\vec{x}^*) = -2$ ; при  $n = 4$ ,  $f(\vec{x}^*) = -16$ ; при  $n = 6$ ,  $f(\vec{x}^*) = -50$

**2. Zakharov Function:**  $f(\vec{x}) = \sum_{i=1}^n x_i^2 + \left( \sum_{i=1}^n 0.5ix_i \right)^2 + \left( \sum_{i=1}^n 0.5ix_i \right)^4$ ,

Область поиска:  $-15 \leq x_i \leq 15$ ,  $i = \overline{1, n}$ . Минимум функции:  $f(\vec{x}^*) = 0$

**3. Helical valley Function:**  $f(\vec{x}) = 100[x_3 - 10\theta(x_1, x_2)]^2 + 100[\sqrt{x_1^2 + x_2^2} - 1]^2 + x_3^2$ ,

$$\theta(x_1, x_2) = \begin{cases} \frac{1}{2\pi} \arctg(x_2/x_1), & \text{когда } x_1 > 0 \\ \frac{1}{2\pi} \arctg(x_2/x_1) + 0.5, & \text{когда } x_1 < 0 \end{cases}$$

Область поиска:  $-10 \leq x_i \leq 10$ ,  $i = 1, 2, 3$ . Минимум функции:  $\vec{x}^* = (1, 0, 0)$ ,  $f(\vec{x}^*) = 0$





Рисунок 3. Сравнение средних затрат процессорного времени (сек.) для поиска минимума тестовых функций

4. Gaussian Function: 
$$f(\vec{x}) = \sum_{i=1}^m \left[ x_1 \exp\left(\frac{-x_2(\tau_i - x_3)^2}{2}\right) - y_i \right]^2,$$

$$\tau_i = \frac{8-i}{2}, \quad i = 1, \dots, m; m = 15$$

$$y = (0.0009, 0.0044, 0.0175, 0.054, 0.1295, 0.242, 0.3521, 0.3989, 0.3521, 0.242, 0.1295, 0.054, 0.0175, 0.0044, 0.0009);$$

Область поиска:  $-1.5 \leq x_i \leq 1.5, i=1, 2, 3$ . Минимум функции:  $f(\bar{x}^*) = 1.12793 \cdot 10^{-8}$

**5 Box 3 dimensional Function:**  $f(\bar{x}) = \sum_{i=1}^m [e^{-0.1ix_1} - e^{-0.1ix_2} - x_3(e^{-0.1i} - e^{-i})]^2$ ,

Область поиска:  $-50 \leq x_i \leq 50, i=1, 2, 3$ . Минимум функции:  $f(\bar{x}^*) = 0$

**6. Colville Function:**

$f(\bar{x}) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1)$

Область поиска:  $-10 \leq x_i \leq 10, i=1, 2, 3, 4$ . Минимум функции:  $f(\bar{x}^*) = 0$

**7. Branin Function:**  $f(\bar{x}) = \left(x_2 - \frac{5}{4p^2}x_1^2 + \frac{5}{p}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8p}\right)\cos(x_1) + 10$ ,

Область поиска:  $-5 \leq x_1 \leq 10, 0 \leq x_2 \leq 15$ . Минимум функции:  $f(\bar{x}^*) = 0.397887$

**8. Sphere Function:**  $f(\bar{x}) = \sum_{i=1}^n x_i^2$ ,

Область поиска:  $-2.56 \leq x_i \leq 5.12, i=1, 2, \dots, n$ . Минимум функции:  $f(\bar{x}^*) = 0$

**9. Sum Squares Function:**  $f(\bar{x}) = \sum_{i=1}^n ix_i^2$ ,

Область поиска:  $-5 \leq x_i \leq 10, i=1, 2, \dots, n$ . Минимум функции:  $f(\bar{x}^*) = 0$

**10. Rotated hyper-ellipsoid function:**  $f(\bar{x}) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j\right)^2$ ,

Область поиска:  $-65 \leq x_i \leq 65, i=1, \dots, n$ . Минимум функции:  $f(\bar{x}^*) = 0$

#### Применение метода симплексного планирования в задачах оптимизации

Предложенная модификация последовательного симплексного планирования может быть применена для решения задачи равномерного распределения ресурсов по времени. Пусть организация осуществляет комплексную программу, состоящую из  $n$  проектов. Известны функции  $f_i(t), t \in [0, d_i], i=1, \dots, n$ , характеризующие потребность ресурсов в момент времени  $t$  для реализации проекта  $i$ . Задана продолжительность каждого проекта -  $d_i$ , самый ранний момент начала -  $c_i$  и самый поздний момент окончания -  $v_i, i=1, \dots, n$ . Если реализация  $i$ -го проекта начнется с задержкой  $t_i \in [c_i, v_i - d_i]$ , распределение ресурсов будет соответствовать функции  $f_i(t - t_i)$ , где  $t \in [t_i, t_i + d_i], i=1, \dots, n$ . Суммарная потребность в ресурсах в момент времени  $t$  равна  $\sum_{i=1}^n f_i(t - t_i)$ . Требуется выбрать такие моменты начала

выполнения проектов  $t_1, t_2, \dots, t_n$ , чтобы распределение суммарных ресурсов по времени было наиболее равномерным. Под равномерным распределением ресурсов понимается такое распределение, когда значения суммарной функции как можно меньше отличаются от средней потребности  $M$ , где

$$M = \frac{1}{b-a} \int_a^b \sum_{i=1}^n f_i(t - t_i) dt,$$

$a = \min_i t_i = 0, b = \max_i t_i + d_i$ ,  $(b-a)$  – продолжительность комплексной программы.

В [1] в качестве оценки близости потребности ресурсов к среднему значению выбран квадратичный критерий. Тогда задача равномерного распределения ресурсов сходится к задаче



$$\min_{t_1, t_2, \dots, t_n} \sqrt{\frac{1}{b-a} \int_a^b \left[ \sum_{i=1}^n f_i(t-t_i) - M \right]^2 dt}.$$

### Литература

1. Цирлин А.М. Вариационные методы расчета химических аппаратов.-М.: «Машиностроение», 1978
2. Nelder J.A, Mead R. A simplex method for function minimization. *Computer J.* 7:308-313, 1965
3. Hedar A.R, Fukushima M. Simplex coding genetic algorithm for the global optimization of nonlinear functions, *Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan, 2002*
4. Овсепян В.С., Дерцяи А.С. Об одной модификации последовательного симплексного метода. Сборник трудов международной II конференции Горисского университета, Горис, 2011

### **Полимеризация порошковых красок с использованием ИК-излучателей с наноструктурированным керамическим покрытием**

Шкарин Н.Ю.<sup>1</sup>, д.т.н. Рахимов Р.Х.<sup>2</sup>, к.ф.-м.н. в.н.с. Казенин Д.А.<sup>1</sup>

<sup>1</sup> Университет машиностроения

<sup>2</sup> Институт материаловедения НПО «Физика-Солнца» АН РУз

8(488)267-16-69, n\_shkarin@mail.ru

Аннотация. Порошковые полимерные покрытия – одно из наиболее стремительно развивающихся направлений в покраске и антикоррозийной защите промышленных изделий, оборудования, деталей машин и агрегатов. Поэтому поиск новых энергосберегающих технологий в технологии порошковой окраски становится особенно актуален.

Ключевые слова: порошковые краски, ИК-излучатели, ИК-полимеризация, энергосбережение.

Машины и агрегаты автомобильного транспорта обычно эксплуатируются в сложных условиях. Из-за контакта с топливосмазочными материалами, химическими веществами, водой, негативного влияния переменных температурных режимов и ряда других факторов поверхности тракторов, автомобилей и сельскохозяйственных машин подвергаются воздействию агрессивных загрязнений.

Такие загрязнения уменьшают устойчивость защитно-декоративных покрытий, повышают скорость коррозионных процессов и в конечном итоге служат одной из причин, приводящей к снижению надежности машин и агрегатов.

По сравнению с традиционными лакокрасочными материалами порошковые краски обеспечивают практически безотходную технологию производства покрытий, также физико-механические свойства покрытий из порошковых красок по многим факторам превосходят покрытия из жидких лакокрасочных материалов [1].

Печи с инфракрасным нагревом хорошо показали себя при работе в автоматизированных линиях при окраске изделий простой формы.

Преимущество перед конвекционными печами обеспечивается за счёт непосредственной передачи энергии от нагревательного элемента к покрытию посредством селективного ИК-излучения, без участия промежуточного теплоносителя (воздух в конвекционных печах). За счёт снижения тепловых потерь и наличия селективных спектральных характеристик ИК-излучателей эффективность передачи тепла увеличивается в 3 и более раз.