

АКАДЕМИЯ НАУК СССР
ОРДЕНА ЛЕНИНА СИБИРСКОЕ ОТДЕЛЕНИЕ
Вычислительный центр (г. Красноярск)



ВЦ

Препринт № 6

Шарый С. П.

О НЕКОТОРЫХ МЕТОДАХ РЕШЕНИЯ
ЛИНЕЙНОЙ ЗАДАЧИ О ДОПУСКАХ

Красноярск — 1989

Шарый С.П. О некоторых методах решения линейной задачи о допусках. Препринт ВЦ СО АН СССР № 6, Красноярск, 1989, 45 стр., 2 рис., 17 библ.

Работа посвящена исследованию линейной задачи о допусках и построению численных алгоритмов ее решения. Показано, что исследование разрешимости линейной задачи о допусках сводится к выявлению знака безусловного максимума некоторого негладкого вогнутого функционала. В случае разрешимости одновременно получаем точку, вокруг которой можно построить интервальное решение задачи. Описаны алгоритмы ВЕГРА и ВЕГРАМ, осуществляющие построение интервального решения вокруг известного центра.

Рецензент - В.В.Шайдуров

Ответственный за выпуск - С.П.Шарый

© ВЦ СО АН СССР, г.Красноярск, 1989

ВВЕДЕНИЕ

Настоящая работа является продолжением препринта [6] и также посвящена исследованию линейной задачи о допусках и построению численных алгоритмов ее решения. Библиография работ, посвященных этой задаче (называемой иногда также "внутренней задачей" для системы интервальных линейных уравнений) невелика, почти все иностранные работы указаны в [4]. Первой советской публикацией на эту тему была, по-видимому, статья Н.А.Хлебалина [3]. В предлагаемой вниманию работе так же, как и в [6], развивается подход, который можно назвать "центровым": сначала ищется какая-нибудь точка, удовлетворяющая условиям задачи, — "центр", а затем уже, в случае успеха, вокруг нее строится интервальное решение. Иной, "алгебраический" подход к задаче о допусках сформулирован в препринте А.В.Захарова и Ю.И.Шокина [5].

Ниже, в §1 вводятся необходимые обозначения и кратко обсуждается постановка задачи. §2 посвящен построению критерия разрешимости задачи о допусках. Показано, что исследование разрешимости сводится к безусловной максимизации некоторого негладкого вогнутого функционала и последующему определению знака этого максимума. Получаемая в процессе максимизации информация может быть использована при более детальном исследовании задачи. Важно отметить, что в случае разрешимости одновременно находится точка, вокруг которой далее строится интервальное решение.

Описание алгоритма БЕГРА, осуществляющего это построение вокруг известного центра, составляет содержание § 3.

Трудоёмкость этого алгоритма в худшем случае экспоненциальная, т.е. такая же, как и переборных методов из [6]. Однако, как показывают численные эксперименты, в среднем (для достижения заданной точности) он требует существенно меньшего времени. Кроме того, БЕГРА отличается от переборных методов более гибкой вычислительной схемой, позволяющей в любой момент прекратить процесс последовательного уточнения оценки искомой величины. В § 4 рассматривается специальный прием, позволяющий при определенных условиях существенно упростить оптимизацию рациональных выражений на прямоугольниках. Развита при этом техника используется для построения алгоритма БЕГРАМ, — модифицированной версии алгоритма

БЕГГА - отличающегося гарантированным оцениванием точности результатов и меньшим требуемым объемом оперативной памяти.

§ 1. ОБОЗНАЧЕНИЯ. ПОСТАНОВКА ЗАДАЧИ.

В настоящей работе мы будем придерживаться следующих обозначений:

$\mathcal{R}$ - вещественная ось;

$\mathcal{R}^n$ - вещественное n -мерное пространство, причем из контекста будет ясно, идет речь о вектор-строках или вектор-столбцах;

$\mathcal{R}$ - множество вещественных интервалов, т.е. отрезков $[a; b]$, $a \leq b$ на вещественной оси;

$\mathcal{R}^n$ - множество интервальных n -мерных векторов;

$\mathcal{R}^{m \times n}$ - множество интервальных $m \times n$ -матриц.

Интервалы и интервальные величины (векторы, матрицы) будут обозначаться буквами двойного шрифта, например, $\mathcal{A}, \mathcal{B}, \mathcal{C}, \dots$.

Другие обозначения:

$\bar{A}, \bar{A}$ - левый и правый концы интервала A , (1)

$med A = (\bar{A} + \bar{A})/2$ - середина интервала A (медиана), (2)

$wid A = \bar{A} - \bar{A}$ - ширина интервала A , (3)

$|A| = \max\{|\bar{A}|, |\bar{A}|\}$ - абсолютная величина интервала. (4)

Если $A = (a_i)_{i=1}^n$ - интервальный вектор, то операции (1)-(4) будем применять к A покомпонентно, так что $med A$, например, - это вещественный вектор той же структуры, что и A , компонентами которого являются числа $med a_i$. То же самое относится к случаю, когда A - интервальная матрица.

Напомним постановку линейной задачи о допусках и некоторые связанные с ней понятия. Пусть $A \in \mathcal{R}^{n \times n}$, $b \in \mathcal{R}^n$.

Множество

$$X_* = \{x \in \mathcal{R}^n / (\forall A \in A)(Ax \in b)\}.$$

состоящее из векторов x , для которых при любой матрице

$A \in \mathcal{A}$ произведение (Ax) попадает в интервал правых частей $\mathcal{B}$, назовем допустимым множеством решений (ДМР) интервальной системы линейных алгебраических уравнений $Ax = \mathcal{B}$. X_* - многогранное множество и точное его описание в случае больших размерностей затруднительно и неудобно для дальнейшего использования. Поэтому имеет смысл искать интервальный вектор $u \in \mathcal{R}^n$, такой, что $u \in X_*$. Эту задачу мы и будем называть линейной задачей о допусках (ЛЗД).

Часто в приложениях постановка ЛЗД содержит фиксированное отношение длин отдельных компонент u :

$$\text{wid } u_i / \text{wid } u_j = d_i / d_j,$$

$$(d_1, d_2, \dots, d_n) = d, \quad d_k > 0.$$

Подобные ситуации легко сводятся к стандартной, когда $d = (1, 1, \dots, 1)$ и нам требуется вписать куб в соответствующим образом модифицированное множество X_* . В самом деле, пусть $D = \text{diag}\{d_1, d_2, \dots\}$, $\tilde{A} = AD$. Найдем вектор $\tilde{u}$ - решение ЛЗД с матрицей $\tilde{A}$ и правой частью $\mathcal{B}$. Тогда $D\tilde{u}$ - решение исходной задачи с нужным отношением допусков [6]. Поэтому далее решение ЛЗД будет состоять в поиске интервального вектора $u \in X_*$ с компонентами одинаковой ширины.

§ 2. РАЗРЕШИМОСТЬ ЗАДАЧИ О ДОПУСКАХ.

Вопрос выбора центра интервального решения ЛЗД тесно связан с разрешимостью задачи, т.к. *a priori* часто неясно, как выбирать хоть какую-нибудь точку из $\text{int } X_*$ и является ли множество X_* вообще непустым. Наиболее распространенная рекомендация [4,6] (брать в качестве центра u интервального решения ЛЗД вектор решений "средней" системы уравнений $\text{mid } A \cdot u = \text{mid } \mathcal{B}$) работает, если элементы матрицы A достаточно узки, но в общем случае её нельзя признать удовлетворительной. Это показывают следующие примеры.

Пусть $A = [-1; 3]$, $\theta = [-2; 8]$, тогда X_* - это отрезок $[-2/3; 2]$. А решение "средней" системы равно 3 и не лежит в X_* . Можно привести двумерный контрпример:

$$A = \begin{pmatrix} 3 & [1; 2] \\ [1; 2] & 3 \end{pmatrix}, \quad \theta = \begin{pmatrix} [5; 7] \\ [7; 9] \end{pmatrix}$$

Здесь множество X_* состоит из единственного вектора $(1, 2)^T$, тогда как решение "средней" системы - это $(8/9, 20/9)^T$. Последний пример замечателен тем, что интервальная матрица задачи содержит лишь невырожденные вещественные матрицы (и все элементы интервальной матрицы - строго положительные интервалы).

Разрешимость линейной задачи о допусках равносильна непустоте множества X_* , что в свою очередь означает существование вектора $x \in \mathcal{R}^n$ со свойством

$$\{Ax \mid A \in \mathcal{A}\} = Ax \subseteq \theta, \quad (5)$$

где под (Ax) понимается результат интервального умножения матрицы A на вектор x [1]. При этом критерий разрешимости ЛЗД может быть некоторым алгоритмом поиска векторов x , удовлетворяющих (5). Если же такие векторы не находятся, делается заключение о неразрешимости ЛЗД.

Запишем условие (5) в следующей форме:

$$\sum_{j=1}^n a_{ij} x_j - \text{med } \theta_i \in \left[-\frac{1}{2} \text{ wid } \theta_i ; \frac{1}{2} \text{ wid } \theta_i \right], \\ i = 1, 2, \dots, n,$$

что означает

$$\left| \text{med } \mathcal{C}_i - \sum_{j=1}^n a_{ij} x_j \right| \leq \frac{1}{2} \text{wid } \mathcal{C}_i, \\ i = 1, 2, \dots, n$$

(вертикальными отрезками обозначена операция взятия абсолютно-го значения интервала $[I]$).

Следовательно, $x \in X_*$ тогда и только тогда, когда

$$J_{nn}(A, \mathcal{C}; x) = J_{nn}(x) = \min_{1 \leq i \leq n} \left\{ \frac{1}{2} \text{wid } \mathcal{C}_i - \left| \text{med } \mathcal{C}_i - \sum_{j=1}^n a_{ij} x_j \right| \right\} \geq 0,$$

а множество допустимых решений X_* - это лебегово множество $\{x \in \mathcal{R}^n \mid J_{nn}(x) \geq 0\}$ функционала $J_{nn} : \mathcal{R}^n \rightarrow \mathcal{R}$. Исследуем его свойства.

Лемма I. Функционал J_{nn} вогнутый.

Доказательство. J_{nn} - нижняя огибающая функционалов

$$\varphi_i(x) = \frac{1}{2} \text{wid } \mathcal{C}_i - \left| \text{med } \mathcal{C}_i - \sum_{j=1}^n a_{ij} x_j \right|. \\ i = 1, 2, \dots, n,$$

так что для доказательства леммы достаточно установить вогнутость всех φ_i .

Пусть $x, y \in \mathcal{R}^n$, $\lambda \in [0; 1]$, тогда в силу субдистрибутивности интервальной арифметики

$$\begin{aligned} \text{med } \mathcal{C}_i - \sum_{j=1}^n a_{ij} (\lambda x + (1-\lambda)y) &\leq \\ &\leq \lambda \cdot \left(\text{med } \mathcal{C}_i - \sum_{j=1}^n a_{ij} x_j \right) + (1-\lambda) \cdot \left(\text{med } \mathcal{C}_i - \sum_{j=1}^n a_{ij} y_j \right). \end{aligned}$$

Далее, из-за монотонности функции модуля интервала относительно стандартного порядка в $\mathcal{R}$ и порядка по включению в $\mathcal{R}$ [1] имеем

$$\begin{aligned} & \left| \text{med } \mathcal{B}_i - \sum_{j=1}^n a_{ij} (\lambda x_j + (1-\lambda) y_j) \right| \leq \\ & \leq \left| \lambda \cdot (\text{med } \mathcal{B}_i - \sum_{j=1}^n a_{ij} x_j) + (1-\lambda) \cdot (\text{med } \mathcal{B}_i - \sum_{j=1}^n a_{ij} y_j) \right| \leq \\ & \leq \lambda \left| \text{med } \mathcal{B}_i - \sum_{j=1}^n a_{ij} x_j \right| + (1-\lambda) \left| \text{med } \mathcal{B}_i - \sum_{j=1}^n a_{ij} y_j \right|, \end{aligned}$$

что и требовалось доказать.

Итак, подграфик $\text{hyp } \mathcal{I}nn = \{(x, y) \in \mathcal{R}^{n+1} / x \in \mathcal{R}^n, y \in \mathcal{R}, y \leq \mathcal{I}nn(x)\}$ отображения $\mathcal{I}nn : \mathcal{R}^n \rightarrow \mathcal{R}$ является выпуклым множеством. Покажем, что $\text{hyp } \mathcal{I}nn$ — многогранное множество, т.е. пересечение конечного числа полупространств в $\mathcal{R}^{n+1}$. Действительно, выражая абсолютную величину через функцию максимума, для каждого $i=1, 2, \dots, n$ получим

$$\begin{aligned} & \frac{1}{2} \text{wid } \mathcal{B}_i - \left| \text{med } \mathcal{B}_i - \sum_{j=1}^n a_{ij} x_j \right| = \\ & = \frac{1}{2} \text{wid } \mathcal{B}_i - \max_{\tilde{a}_{ij}} \left\{ \left| \text{med } \mathcal{B}_i - \sum_{j=1}^n \tilde{a}_{ij} x_j \right| \right\} = \\ & = \frac{1}{2} \text{wid } \mathcal{B}_i - \max_{\tilde{a}_{ij}} \left\{ \max \left\{ \text{med } \mathcal{B}_i - \sum_{j=1}^n \tilde{a}_{ij} x_j, -\text{med } \mathcal{B}_i + \sum_{j=1}^n \tilde{a}_{ij} x_j \right\} \right\} = \end{aligned}$$

$$= \min_{\tilde{a}_{ij}} \left\{ \min \left\{ \frac{1}{2} \text{wid } b_i - \text{med } b_i + \sum_{j=1}^n \tilde{a}_{ij} x_j, \frac{1}{2} \text{wid } b_i + \text{med } b_i - \sum_{j=1}^n \tilde{a}_{ij} x_j \right\} \right\},$$

где внешние максимумы и минимум берутся по всевозможным вершинам $(\tilde{a}_{i1}, \tilde{a}_{i2}, \dots, \tilde{a}_{in})$ прямоугольного параллелепипеда $a_{i1} \times a_{i2} \times \dots \times a_{in}$. Поэтому функционал J_{nn} является нижней огибающей семейства не более чем $n \cdot 2^{n+1}$ аффинных функционалов вида $\frac{1}{2} \text{wid } b_i \pm \text{med } b_i \mp \sum_{j=1}^n \tilde{a}_{ij} x_j$, а множество $\text{hyp } J_{nn}$ — пересечением подграфиков этих аффинных функционалов.

Лемма 2. Функционал J_{nn} достигает конечного максимума на всем пространстве $\mathcal{R}^n$.

Доказательство. Многогранное множество $\text{hyp } J_{nn}$ можно представить [7] в виде выпуклой комбинации крайних точек (s_j, t_j) ,

$s_j \in \mathcal{R}^n$, $t_j \in \mathcal{R}$, $j = 1, 2, \dots, k$ и крайних лучей с направляющими векторами (d_j, e_j) , $d_j \in \mathcal{R}^n$, $e_j \in \mathcal{R}$, $j = 1, 2, \dots, l$, так что

$$\text{hyp } J_{nn} = \left\{ \sum_{j=1}^k \lambda_j (s_j, t_j) + \sum_{j=1}^l \mu_j (d_j, e_j) \mid \sum_{j=1}^k \lambda_j = 1, \lambda_j \geq 0, \mu_j \geq 0 \right\}. \quad (6)$$

Заметим, что для любого $x \in \mathcal{R}^n$

$$J_{nn}(x) \leq M = \min_{1 \leq i \leq n} \left\{ \frac{1}{2} \text{wid } b_i \right\} \quad \text{и поэтому}$$

$\text{hyp } J_{nn}$ имеет пустое пересечение с полупространством в $\mathcal{R}^{n+1}$ тех векторов, последняя координата которых не меньше M . Отсюда вытекает, что $(n+1)$ -е компоненты у направляющих векторов крайних лучей множества $\text{hyp } J_{nn}$ должны быть по крайней

мере неположительны, т.е. $e_j \leq 0$ для $j=1, 2, \dots, l$ - поскольку в противном случае в множестве (6) окажутся точки со сколь угодно большой $(n+1)$ -й координатой. Из-за этого, желая отыскать $\max_{x^*} J_{nn} = \max \{ \text{последняя координата точек}$

из $\text{hyp } J_{nn} \} =$

$$= \max \left\{ \sum_{j=1}^n \lambda_j t_j + \sum_{j=1}^l \mu_j e_j \mid \sum_{j=1}^n \lambda_j = 1, \lambda_j \geq 0, \mu_j \geq 0 \right\}$$

мы должны положить $\mu_j = 0$ для $j=1, 2, \dots, l$ (или по крайней мере для тех j , для которых $e_j < 0$). Таким образом,

$$\max_{x^*} J_{nn}(x) = \max \left\{ \sum_{j=1}^n \lambda_j t_j \mid \sum_{j=1}^n \lambda_j = 1, \lambda_j \geq 0 \right\} = \max_{1 \leq j \leq n} t_j$$

и достигается в крайней точке подграфика $\text{hyp } J_{nn}$, имеющей наибольшую $(n+1)$ -ую координату.

Лемма 3. Если интервальная матрица задачи A такова, что в любой ее строке имеются ненулевые элементы, то из

$$x \in \text{int } X_* \neq \emptyset \quad \text{следует} \quad J_{nn}(x) > 0$$

Доказательство.

Пусть $X_* \neq \emptyset$ и $\max_{x^*} J_{nn}$ достигается в некоторой точке $m \in X_*$. В случае $x \in \text{int } X_*$ точка x является внутренней для некоторого отрезка $[m; y] \subseteq X_*$, т.е. $x = \lambda m + (1-\lambda)y$, $\lambda \in (0; 1)$. Поскольку функционал J_{nn} вогнутый, имеем

$$J_{nn}(x) \geq \lambda \cdot J_{nn}(m) + (1-\lambda) \cdot J_{nn}(y).$$

Предположим, что $J_{nn}(x) = 0$. Тогда выписанное выше неравенство будет справедливым лишь при $J_{nn}(m) = J_{nn}(y) = 0$

и функционал J_{nn} должен быть нулевым на всем множестве X_* .

Пусть, далее, $R^n = \bigcup_{1 \leq k \leq n} \Omega_k$ - объединение множеств Ω_k , таких что

$$J_{nn}(x) = \frac{1}{2} \text{wid } \mathcal{C}_k - \left| \text{med } \mathcal{C}_k - \sum_{j=1}^n a_{kj} x_j \right| \quad \forall (x_1, x_2, \dots, x_n) \in \Omega_k.$$

Ясно, что $X_* = \bigcup_{1 \leq k \leq n} (X_* \cap \Omega_k)$, причем множества $(X_* \cap \Omega_k)$,

$k=1, 2, \dots, n$ - замкнутые. Если их внутренности пусты, то $(X_* \cap \Omega_k)$ - множество первой беровской категории в R^n .

Но в силу теоремы Бера [8] множество $X_* \in R^n$ с $\text{int } X_* \neq \emptyset$ является множеством второй беровской категории. Таким образом, хотя бы для одного $k=1, 2, \dots, n$ должно быть $\text{int}(X_* \cap \Omega_k) \neq \emptyset$, и тогда

$$\frac{1}{2} \text{wid } \mathcal{C}_k - \left| \text{med } \mathcal{C}_k - \sum_{j=1}^n a_{kj} x_j \right| = 0 = \text{const}$$

для всех $(x_1, x_2, \dots, x_n) \in \text{int}(X_* \cap \Omega_k)$.

Последнее возможно лишь в случае, когда коэффициенты $a_{kj}, j=1, 2, \dots$, все нулевые, что противоречит условиям на матрицу A .

Лемма 4. Если $J_{nn}(x) > 0$, то $x \in \text{int } X_*$.

Доказательство. Отображение $J_{nn} : R^n \rightarrow R$ - непрерывное, так что множество $Y = \{y \in R^n / J_{nn}(y) > 0\}$ - открытое и непустое, причем $x \in Y \subseteq X_*$. Следовательно,

$$\text{int } X_* \neq \emptyset \quad \text{и} \quad x \in \text{int } X_*.$$

Подытоживая сказанное, можно предложить следующий критерий разрешимости линейной задачи о допусках:

Решаем задачу безусловной максимизации функционала

$$J_{nn}(x) = \min_{1 \leq i \leq n} \left\{ \frac{1}{2} \text{wid } \theta_i - \left| \text{med } \theta_i - \sum_{j=1}^n a_{ij} x_j \right| \right\}.$$

Пусть $\max_{x \in \mathcal{R}^n} J_{nn}$ достигается в точке m , тогда

- а) если $\max_{x \in \mathcal{R}^n} J_{nn} \geq 0$, то ЛЗД разрешима и $m \in X_*$,
- а если $\max_{x \in \mathcal{R}^n} J_{nn} > 0$, то $m \in \text{int } X_*$;
- б) если $\max_{x \in \mathcal{R}^n} J_{nn} < 0$, то делаем заключение о неразрешимости ЛЗД.

Для отыскания максимума на $\mathcal{R}^n$ негладкого функционала J_{nn} можно применить методы, изложенные в [10 - 13].

Легко видеть, что леммы 1-4, а также критерий разрешимости останутся в силе, если функционал J_{nn} взять в виде

$$J_{nn\gamma}(x) = \min_{1 \leq i \leq n} \left\{ \gamma_i \left(\frac{1}{2} \text{wid } \theta_i - \left| \text{med } \theta_i - \sum_{j=1}^n a_{ij} x_j \right| \right) \right\},$$

где γ_i - положительные числа. Иногда применение такого функционала может оказаться более предпочтительным, чем исходного.

В заключение этого параграфа отметим следующее. Увеличение ширины всех компонент вектора θ (при неизменных A и $\text{med } \theta$) на одну и ту же величину $2c$, $c > 0$, равносильно добавлению к функционалу $J_{nn}(A, \theta; x)$ аддитивной константы c . Следовательно, при $E = ([-1; 1], \dots, [-1; 1])^T$ имеем

$$\max_{x \in \mathcal{R}^n} J_{nn}(A, \theta + 2cE; x) = c + \max_{x \in \mathcal{R}^n} J_{nn}(A, \theta; x).$$

Поэтому, если исходная задача о допусках была неразрешима, то путем уширения вектора правых частей на $2cE$, $c > |\max_{x \in \mathcal{R}^n} J_{nn}(A, \theta; x)|$

можно сделать ее разрешимой при неизменной матрице A .

Иногда на практике такое одновременное уширение всех компонент θ может быть неприемлемым в силу тех или иных соображений. Предположим поэтому, что задан положительный вектор $(\tau_1, \tau_2, \dots,$

..., τ_n), $\tau_i > 0$, такой, что увеличение ширины θ_i должно быть пропорциональным τ_i . Например, τ_i можно положить равными $|\theta_i|$ или $|\text{med } \theta_i|$, если требуется обеспечить одина-

ковое относительное ухирение компонент правой части. Далее найдем

$$\mathcal{W} = \max_{x \in \mathbb{R}^n} \text{Inn}_\tau(A, \theta; x), \text{ где}$$

$$\text{Inn}_\tau(A, \theta; x) = \min_{1 \leq i \leq n} \left\{ \frac{1}{\tau_i} \left(\frac{1}{2} \text{wid } \theta_i - |\text{med } \theta_i| - \sum_{j=1}^n a_{ij} x_j \right) \right\}.$$

Ясно, что если исходная ЛЗД была неразрешимой (т.е. $\mathcal{W} < 0$), то задача с той же матрицей A и интервалом правых частей

$(\theta_i + 2\tau_i c[-1; 1])_{i=1}^n$, сделается разрешимой при $c > |\mathcal{W}|$.

§ 3. АЛГОРИТМ ПОСТРОЕНИЯ ИНТЕРВАЛЬНОГО РЕШЕНИЯ ВОКРУГ ИЗВЕСТНОГО ЦЕНТРА.

После того, как найдена точка из внутренности множества X_* , можно принять ее за центр будущего интервального решения ЛЗД и приступить к построению вокруг него интервального решения. Мы не будем обсуждать здесь оптимизацию выбора центра, поскольку вопрос этот тесно связан с конкретными практическими постановками задачи о допусках.

В [6] показано, что если $u \in X_*$, то интервальный вектор с компонентами $[u_i - h; u_i + h]$, где

$$h = \min_{1 \leq i \leq n} \min_{A \in A} \left\{ \frac{\frac{1}{2} \text{wid } \theta_i - |\text{med } \theta_i| - \sum_{j=1}^n a_{ij} u_j}{\sum_{j=1}^n |a_{ij}|} \right\}, \quad (7)$$

целиком лежит в X_* , т.е. является интервальным решением ЛЗД. (Заметим, что без знаменателя выражение (7) совпадает с функцией Inn). В той же работе описаны несколько алгоритмов вычисления h . Простейший из них, в котором минимум по $A \in A$

доби в фигурных скобках (7) оценивается как нижний конец ее естественного интервального расширения, будем называть алгоритмом Ноймайера, т.к. по результатам он полностью совпадает (см. [6]) с алгоритмом, предложенным в [4], хотя и в несколько другом виде. Алгоритм Ноймайера быстр, имеет простую вычислительную схему, но "качество" даваемых им решений оставляет желать лучшего: получающийся интервальный вектор может иметь недопустимо малые размеры.

Точный алгоритм отыскания минимума (7) строится в [6] с учетом квазивогнутости функции в фигурных скобках (7) от аргументов a_{ij} . Из-за этого при каждом $i = 1, 2, \dots, n$ минимальные значения по $A \in A$ достигаются в вершинах параллелепипеда $a_{i1} \times a_{i2} \times \dots \times a_{in}$ и могут быть найдены прямым перебором, после чего берется минимум по i . Трудоемкость выполнения такого алгоритма, пропорциональную $n \cdot 2^n$, можно значительно уменьшить, если обход вершин осуществлять специальным образом, — переходя на каждом шаге к соседней вершине и рекуррентно перевычисляя суммы

$$\sigma_1 = \text{med } \beta_i - \sum_{j=1}^n a_{ij} u_j \quad \text{и} \quad \sigma_2 = \sum_{j=1}^n |a_{ij}|. \quad (8)$$

При этом выигрыш в трудоемкости тем значительней, чем больше размерность задачи. Но здесь имеется другая опасность. При переборе "с соседством" мы как бы отталкиваемся от начальных значений сумм σ_1 и σ_2 и потом уже больше не пользуемся "точными" формулами (8). Если число просматриваемых вершин велико, то ошибки округления, накапливаемые на каждом шаге, могут привести к значительному "отрыву" процесса от истинных значений σ_1 и σ_2 . Компромисс между точностью вычислений и их трудоемкостью достигается прерываниями через определенное количество вершин (например, через 2^{10} просмотренных вершин) процесса рекуррентного пересчета для σ_1 и σ_2 и вычислением их по формулам (8).

Ясно, что описанные в [6] переборные алгоритмы нахождения (7) непригодны для решения задач значительной размерности из-за

их экспоненциальной трудоемкости. Излагаемый ниже алгоритм ВЕГРА вычисления (7) основан на известном в комбинаторной оптимизации методе ветвей и границ. Он занимает промежуточное положение между алгоритмом А.Ноймайера и переборными алгоритмами из [6]. Трудоемкость алгоритма ВЕГРА в худшем случае экспоненциальная (как и для всех алгоритмов подобного рода), но на наш взгляд он может быть с успехом применен к задачам любой размерности, хотя точность вычислений будет лимитироваться ресурсами вычислительной системы.

Известно [2], что если рациональное выражение $f(x_1, \dots, x_n)$ содержит по одному вхождению каждой переменной в первой степени, то множество его значений на некотором прямоугольнике (прямоугольном параллелепипеде) $\Pi = [a_1; b_1] \times \dots \times [a_n; b_n] \subseteq \mathcal{R}^n$

совпадает с соответствующим естественным интервальным расширением f . Таким образом, левый конец естественного интервального расширения является в этом случае глобальным минимумом f на Π , а правый конец - глобальным максимумом f на Π . Мы можем узнать не только сами значения $\min_{\Pi} f(x)$ и $\max_{\Pi} f(x)$, но и то, какие аргументы их доставляют, т.е. множества $\text{Arg}_{\Pi} \min f(x)$ и $\text{Arg}_{\Pi} \max f(x)$ (они могут состоять более чем из одного элемента).

Действительно, если рациональное выражение f содержит лишь по одному вхождению каждой переменной в первой степени, то зависимость f от x_k , например, при любых фиксированных значениях остальных переменных выглядит следующим образом:

$$\text{это либо} \quad f(x_k) = \alpha x_k + \beta,$$

$$\text{либо} \quad f(x_k) = \frac{1}{\alpha x_k + \beta},$$

где α, β - вещественные числа, не зависящие от x_k . В любом случае f - монотонная функция от x_k (в отношении второй возможности это верно при $0 \notin \alpha[a_k; b_k] + \beta$), а поэтому при фиксированных значениях $x_1, \dots, x_{k-1}, x_{k+1}, \dots, x_n$ значения

$\min f(x)$ и $\max f(x)$ достигаются или на концах $[a_k; b_k]$, или же, если $\alpha = 0$, в любой точке из $[a_k; b_k]$. Поскольку

аналогичное рассуждение верно для любой переменной в f , то множества $\operatorname{Argmin}_{\Pi} f$ и $\operatorname{Argmax}_{\Pi} f$ — это либо вершины Π , либо

целиком его грани. Как их выявить?

При выполнении с интервалами любого из четырех арифметических действий — сложения, вычитания, умножения или деления, — одновременно с результирующим интервалом мы можем узнать, какие концы исходных интервалов складываясь (вычитаясь, умножаясь или делясь), дают тот или иной конец результирующего интервала. При вычитании, например, максимум разности (т.е. правый конец результирующего интервала) достигается, когда уменьшаемое принимает значение левого конца, а вычитаемое — правого конца соответствующих интервалов.

Для вычисления произведения $[a_1; b_1] \cdot [a_2; b_2]$ нужно выполнить четыре умножения — найти $\{a_1 a_2, a_1 b_2, b_1 a_2, b_1 b_2\}$, а затем выделить среди этих чисел минимум и максимум. Пусть, например, это будут $a_1 b_2$ и $b_1 b_2$. Тем самым мы узнали, что

$$\operatorname{Argmin}_{\substack{x \in [a_1; b_1] \\ y \in [a_2; b_2]}} (xy) = \begin{pmatrix} a_1 \\ b_2 \end{pmatrix}, \quad \operatorname{Argmax}_{\substack{x \in [a_1; b_1] \\ y \in [a_2; b_2]}} (xy) = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}.$$

Если бы случилось, что минимальными из множества $\{a_1 a_2, a_1 b_2, a_2 b_1, b_1 b_2\}$ являются одновременно два произведения, — скажем, $a_1 b_2$ и $a_2 a_1$, то мы заключили бы следующее:

$$\operatorname{Argmin}_{\substack{x \in [a_1; b_1] \\ y \in [a_2; b_2]}} (xy) = \begin{pmatrix} a_1 \\ [a_2; b_2] \end{pmatrix},$$

т.е. множество Argmin является в этом случае целой гранью прямоугольника $[a_1; b_1] \times [a_2; b_2]$.

Далее, значение любого рационального выражения на интервальном векторе можно вычислить путем применения конечного числа интервальных сложений, вычитаний, умножений и делений. Если $f(x)$ содержит по одному вхождению каждой переменной в первой степени, то проследив рекуррентно эволюцию концов интервалов при каждой из арифметических операций, мы сможем узнать тот набор концов интервалов $[a_1; b_1]$, $[a_2; b_2]$, ..., $[a_n; b_n]$, в которых достигаются $\min_{\Pi} f(x)$ и $\max_{\Pi} f(x)$.

Легко понять, что многое из сказанного останется справедливым и в том случае, когда выражение $f(x)$ сконструировано не только с помощью четырех арифметических операций, а содержит вхождения $\Lambda_1, \Lambda_2, \dots, \Lambda_r$ каких-нибудь других функций, если при этом, переходя к "естественному" интервальному расширению f , мы заменим $\Lambda_1, \Lambda_2, \dots, \Lambda_r$ вхождениями соответствующих оптимальных интервальных расширений. Например, если $\mathbf{b}$ выражения

$$\rho(x) = w - \left| m - \sum_{j=1}^n x_j u_j \right|,$$

$$q(x) = \sum_{j=1}^n |x_j|$$

подставить вместо $x_1, x_2, \dots, x_n$ интервалы их возможных изменений $[a_1; b_1], [a_2; b_2], \dots, [a_n; b_n]$, а вместо

функции модуля вещественного числа взять ее оптимальное интервальное расширение

$$abs([a; b]) = [0, \text{если } ab \leq 0, \text{ иначе } \min\{|a|, |b|\}; \max\{|a|, |b|\}],$$

то получим точные области значений $\rho(x)$ и $q(x)$ на прямоугольнике $\Pi = [a_1; b_1] \times [a_2; b_2] \times \dots \times [a_n; b_n] \subseteq \mathbb{R}^n$.

Ниже нас будут интересовать множества $\text{Arg min}_{\Pi} \rho(x)$ и

$\text{Arg max}_{\Pi} q(x)$. Поскольку мы умеем находить Arg min и

Argmax для функции модуля, входящий в $p(x)$ и $q(x)$ помимо $\{+, -, \cdot, /\}$, то задача отыскания множеств $\text{Argmin}_{\Pi} p(x)$

и $\text{Argmax}_{\Pi} q(x)$ является не более сложной, чем для рациональных выражений. Она может быть решена с помощью той же самой методики "отслеживания концов". Например, записанный на неформальном АЛГОЛе алгоритм вычисления $\text{Argmax}_{\Pi} q(x)$ и $\max_{\Pi} q(x)$ выглядит следующим образом:

```

max := 0 ;
FOR i:=1 STEP 1 UNTIL n DO
  IF |ai| > |bi| THEN BEGIN max:=max+|ai|;
                           Argmaxi:= ai
                           END
  ELSE BEGIN max:=max+|bi|;
           Argmaxi:= bi
           END
END

```

Конечно, этот алгоритм в целях наглядности упрощен и не учитывает того факта, что строение множеств $\text{Argmin}_{\Pi} p(x)$ и $\text{Argmax}_{\Pi} q(x)$ может быть весьма сложным. В частности, $\text{Argmin}_{\Pi} p(x)$ несвязно при

$$\left(m - \sum_{j=1}^n [a_j; b_j] \cdot u_j \right) = - \left(m - \sum_{j=1}^n [a_j; b_j] \cdot u_j \right),$$

а $\text{Argmax}_{\Pi} q(x)$ несвязно при $a_i = -b_i \neq 0$ для некоторого i .
В любом случае отдельные компоненты связности множеств

$\text{Argmin}_{\Pi} p(x)$ и $\text{Argmax}_{\Pi} q(x)$ представимы в виде прямых произведений $\Pi_1 \times \Pi_2 \times \dots \times \Pi_n$, где Π_i - либо вершина прямоугольника Π ,

либо целиком его ребро.

Договоримся, что впредь под $\text{Argmin}_n p(x)$ (или $\text{Argmax}_n q(x)$) мы будем понимать какую-либо одну компоненту связности множества всех точек, доставляющих $\min_n p(x)$ (соответственно, $\max_n q(x)$), а какую именно - безразлично. Численные эксперименты показывают, что эта неопределенность в конечном счете никак не влияет на сходимость организуемого нами алгоритма. Кроме того, множества

$$M = \mu_1 \times \mu_2 \times \dots \times \mu_n, \mu_i \in \mathcal{R} \text{ и } N = \nu_1 \times \nu_2 \times \dots \times \nu_n, \nu_i \in \mathcal{R},$$

со структурой прямого произведения обладают свойством

$$(M \cap N = \emptyset) \Leftrightarrow (\mu_i \cap \nu_i = \emptyset \text{ хотя бы для одного } i),$$

которое будет существенно использовано в дальнейшем.

После этих предварительных рассмотрений приступим к вычислению

$$z = \min_{\substack{x_j \in [a_j; b_j] \\ j=1, 2, \dots, n}} \left\{ \frac{\omega - |m - \sum_{j=1}^n x_j u_j|}{\sum_{j=1}^n |x_j|} \right\} = \min_n \frac{p(x)}{q(x)}, \quad (9)$$

где ω , m , u_1 , u_2 , ..., u_n - некоторые известные константы. Как уже отмечалось, простейший способ получения оценки снизу для z заключается в построении естественного интервального расширения для дроби в фигурных скобках (9). Его нижний конец, равный $\frac{\min_n p(x)}{\max_n q(x)}$ (в дальнейшем мы будем обозначать это выражение через $\text{Low}(\Gamma)$), и даст нужную оценку z .

Найдем множества $\text{Argmax}_n q(x)$ и $\text{Argmin}_n p(x)$. При этом может оказаться, что

либо 1) $\text{Argmin}_n p \cap \text{Argmax}_n q \neq \emptyset$,

либо 2) $\text{Argmin}_n p \cap \text{Argmax}_n q = \emptyset$.

В первом случае любая точка из пересечения $\text{Argmin } \rho \cap \text{Argmax } q$ доставляет глобальный минимум дроби $\frac{\rho(x)}{q(x)}$ на Π , так что задачу минимизации (9) можно считать успешно решенной. Если же $(\text{Argmin } \rho \cap \text{Argmax } q) = \emptyset$, то

$$\text{Low}(\Pi) = \frac{\min_{\Pi} \rho(x)}{\max_{\Pi} q(x)} \neq \min_{\Pi} \left\{ \frac{\rho(x)}{q(x)} \right\}.$$

Тогда существует номер $k \in \{1, 2, \dots, n\}$ такой, что в множествах $\text{Argmin}_{\Pi} \rho(x)$ и $\text{Argmax}_{\Pi} q(x)$ k -тые компоненты не имеют общих значений:

$$(\text{Argmin}_{\Pi} \rho(x))_k \cap (\text{Argmax}_{\Pi} q(x))_k = \emptyset. \quad (10)$$

Рассечем по k -ой компоненте область Π на подобласти Π' и Π'' так, что

$$\Pi' = [a_1; b_1] \times \dots \times [a_k; \frac{a_k + b_k}{2}] \times \dots \times [a_n; b_n],$$

$$\Pi'' = [a_1; b_1] \times \dots \times [\frac{a_k + b_k}{2}; b_k] \times \dots \times [a_n; b_n].$$

Далее будем называть Π' и Π'' областями - потомками для Π , а те компоненты области Π , для которых справедливо (10), - инцидентными для Π . Если у области Π более одной инцидентной компоненты, то k - номер любой из них. Далее мы подробнее обсудим эту ситуацию.

Из монотонности интервальной арифметики по включению следует, что

$$\text{Low}(\Pi') \geq \text{Low}(\Pi) \quad \text{и} \quad \text{Low}(\Pi'') \geq \text{Low}(\Pi),$$

а значит

$$\min \{ \text{Low}(\Pi'), \text{Low}(\Pi'') \} \geq \text{Low}(\Pi). \quad (11)$$

В действительности, при сделанных нами предположениях, в (II) имеет место строгое неравенство, поскольку при разбиении Π множества $\text{Argmin}_{\Pi} p(x)$ и $\text{Argmax}_{\Pi} q(x)$ попадают в разные области - потомки (если представимы в виде прямых произведений).

Пусть для определенности

$$\text{Argmin}_{\Pi} p \subseteq \Pi' \quad \text{и} \quad \text{Argmax}_{\Pi} q \subseteq \Pi'',$$

тогда $\min_{\Pi} p(x) = \min_{\Pi'} p(x)$, $\text{Argmin}_{\Pi} p(x) = \text{Argmin}_{\Pi'} p(x)$

и $\max_{\Pi} q(x) = \max_{\Pi''} q(x)$, $\text{Argmax}_{\Pi} q(x) = \text{Argmax}_{\Pi''} q(x)$.

В то же время

$$\max_{\Pi'} q(x) < \max_{\Pi} q(x) \quad , \quad \min_{\Pi''} p(x) > \min_{\Pi} p(x) \quad ,$$

так что

$$\frac{\min_{\Pi} p(x)}{\max_{\Pi} q(x)} < \frac{\min_{\Pi'} p(x)}{\max_{\Pi'} q(x)} \quad , \quad \frac{\min_{\Pi} p(x)}{\max_{\Pi} q(x)} < \frac{\min_{\Pi''} p(x)}{\max_{\Pi''} q(x)} \quad ,$$

а потому

$$\min_{\Pi} \left\{ \frac{p(x)}{q(x)} \right\} \geq \min \{ \text{Low}(\Pi'), \text{Low}(\Pi'') \} > \text{Low}(\Pi). \quad (I2)$$

Таким образом, изложенные рассуждения являются практически рецептом улучшения оценки снизу для (9). Процедуру разбиения можно повторить по отношению к областям - потомкам Π' и Π'' , получив дальнейшее улучшение оценки $\min \{ p(x)/q(x) \}$, затем

снова разбить потомков от Π' и Π'' и снова улучшить оценку для (9) и т.д. Целесообразно оформить такой процесс последовательного уточнения, используя общую схему известного метода ветвей и границ [9]. Действительно, рассечение какой-либо области на потомки - это не что иное, как разбиение задач на подзадачи, т.е. порождение "ветвей", а вычисление оценок Low - это построение "границ" целевой функции на "ветвях". Множество активных подзадач оформим в виде списка областей $S = \{ \Upsilon_1, \Upsilon_2, \dots \}$, $\Upsilon_m \subseteq \Pi$, Υ_m - интервальный вектор. Перед началом

работы алгоритма список S состоит из одной исходной области Π , а затем изменяется на каждом шаге, который заключается в следующем:

выбираем из списка S область Ω с наименьшей оценкой Low (Будем называть такую область и ее оценку Low текущими на данном шаге. Если в списке S содержатся несколько областей с наименьшими и равными оценками Low , то в качестве текущей берем любую из них.);

если $Argmin_{\Omega} p \cap Argmax_{\Omega} q \neq \emptyset$, то прекращаем алгоритм,

иначе рассекаем Ω по инцидентной координате, вычисляем параметры областей-потомков (т.е. значения $min p$, $max q$, $Argmin p$, $Argmax q$ и оценки Low), заносим области-потомки в список S вместо Ω .

Результатом работы этого алгоритма является неубывающая (в силу (12)) последовательность текущих оценок Low , приближающих снизу точное значение (9).

Предположим, что число шагов алгоритма конечно, т.е. для последней текущей области Ω справедливо

$$Argmin_{\Omega} p \cap Argmax_{\Omega} q \neq \emptyset \text{ и } Low(\Omega) = \min_{\Omega} \left\{ \frac{p(x)}{q(x)} \right\}.$$

В этом случае $Low(\Omega)$ равно точному значению (9), поскольку для любой точки $y \in \Pi$, такой что $y \in \mathbb{T}_y \in S$, имеет место

$$p(y)/q(y) \geq Low(\mathbb{T}_y) \geq Low(\Omega),$$

и оценка $Low(\Omega)$ достигается целевой функцией $\left\{ \frac{p(x)}{q(x)} \right\}$ на исходной области Π .

Если же число шагов алгоритма неограничено, то последовательность текущих оценок Low , будучи ограниченной точным значением (9) сверху, сходится к некоторому пределу. К какому именно, зависит от способа рассеечения текущей области на каждом шаге алгоритма.

Все предыдущие рассуждения и выводы остаются справедливыми при любом выборе грани, рассекаемой в текущей области, но последовательность текущих оценок Low не всегда сходится при этом к минимуму (9). Такое происходит, например, если на каждом шаге алгоритма в текущей области рассекается первая (по порядку) инцидентная компонента. Численные эксперименты показывают, что тогда, начиная с некоторого момента, рассекаемая координата может оставаться все время одной и той же, а текущая область списка неограниченно измельчается по ней, тогда как другие инцидентные компоненты алгоритмом не затрагиваются вообще. Разумней поэтому придерживаться следующей стратегии (похожей на предложенную в [15]): текущая область рассекается каждый раз по самой длинной из инцидентных компонент.

Методы типа ветвей и границ замечательны возможностью отсеивания бесперспективных ветвей на основе анализа уже достигнутых оценок целевой функции. В нашем алгоритме это можно осуществить следующим образом. Назовем область T , для которой $Low(T) = \min_T \left\{ \frac{p(x)}{q(x)} \right\}$, точно оцениваемой областью. Другими словами, точно оцениваемыми являются области T , у которых

$Argmin_T p(x) \cap Argmax q(x) \neq \emptyset$, т.е. все компоненты — неинцидентные. Если в списке областей S присутствует точно оцениваемая область T , то без ущерба для конечного результата можно выбросить из S все такие области T , что

$$Low(T) \geq Low(Y).$$

Действительно, для T возможны две ситуации:

1) $Low(T) > Low(Y)$, или 2) $Low(T) = Low(Y)$.

Если имеет место 1), то T может сделаться текущей лишь после того, как Y побывает текущей. Но к тому времени алгоритм уже завершит свою работу. Поэтому T никогда не будет текущей и ее удаление из списка областей никак не повлияет на сходимость алгоритма. Если 2), то пусть T стала текущей областью на шаге с номером S . Это означает, что после выполнения $(S-1)$ -ого шага алгоритма ВЕГРА наименьшая из оценок Low по всем областям

списка S равна $Low(T)$ и достигается она на областях T , T и, быть может, еще на некотором конечном множестве областей $T_1, T_2, \dots, T_t$. Поэтому при любом выборе текущих областей на $(s+1)$ -м и последующих шагах алгоритм БЕГРА остановится не позднее, чем на $(s+t)$ -ом шаге, когда текущей областью сделается T , а результатом его работы будет число $Low(T)$. Легко видеть, что выбрасывание из S областей $T_1, T_2, \dots, T_t$ никак не изменило бы такую концовку.

Следовательно, введя в рассматриваемый алгоритм процесс чистки списка областей, описанный выше, мы по окончании каждого шага будем получать список областей S , содержащий не более одной точно оцениваемой области, оценка Low - которой - наибольшая в S . В самом деле:

если при расщеплении текущей области появился точно оцениваемый потомок T , то при занесении его в список можно удалить из S все области с оценками Low , не меньшими $Low(T)$;

если при расщеплении текущей области появился потомок Ω такой, что $Low(\Omega) \geq \max_{T \in S} Low(T)$, а $\max_{T \in S} Low(T)$ достигается на точно оцениваемой области, то можно отбросить Ω без ущерба для результата.

Итак краткое описание алгоритма БЕГРА выглядит следующим образом. Перед началом работы БЕГРА список областей S состоит из единственной области - исходного прямоугольника

$\Pi = [a_1; b_1] \times [a_2; b_2] \times \dots \times [a_n; b_n]$, - для которой найдены $Low(\Pi)$, $Arg \min p$, $Arg \max q$. Один шаг алгоритма

заключается в выполнении нескольких этапов:

1. Находим текущую область Ω полученного с предыдущего шага списка областей S ;
2. Если область Ω - точно оцениваемая, прекращаем работу;
3. Рассекаем Ω по самой длинной из инцидентных компонент на потомки Ω' и Ω'' , вычисляем для них оценки Low , значения $Arg \min p(x)$, $Arg \max q(x)$;
4. Если $Low(\Omega') \geq \max_{T \in S} Low(T)$ и $\max_{T \in S} Low(T)$ достигается на точно оцениваемой области, то уничтожаем

Ω' . Иначе заносим Ω' в список областей S .

Аналогичную процедуру проделываем и с Ω'' .

б. Если Ω' является точно оцениваемой, то удаляем из S все области с оценкой *low*, не меньшей, чем *low*(Ω').

Аналогичное проделываем и в отношении Ω'' .

Конечно, идея построения подобных алгоритмов не нова, похожие алгоритмы описаны в [14-17], однако лишь в [16] явно упоминается общий принцип, лежащий в их основе – метод ветвей и границ. На наш взгляд, оригинальным в алгоритме ВЕГРА является то, что рассечение (измельчение) текущих областей осуществляется не по всем компонентам, а лишь по инцидентным, т.е. по таким, уменьшение которых только и улучшает оценки целевой функции.

Вскоре после того, как ВЕГРА был реализован на ЭВМ, выявились и некоторые нехорошие эффекты в его работе. Главная неприятность заключалась в том, что надежды, возлагавшиеся на точно оцениваемые области, почти не оправдывались: в задачах даже малой размерности они появлялись крайне редко. Поэтому так же редки были и чистки списка областей, а объем занимаемой программой оперативной памяти в процессе итерирования рос в арифметической прогрессии.

Другим неясным моментом реализации алгоритма являлся критерий остановки, а также тесно связанный с этим вопрос о точности получающегося результата. Ясно, что при решении на реальной ЭВМ задач больших размерностей поводом для остановки может служить исчерпание ресурсов вычислительной системы, – машинного времени, памяти и т.п., тем более, что алгоритм ВЕГРА осуществляет последовательное уточнение оценки снизу для (9). Однако, в любом случае желательно иметь гарантированные оценки точности результата.

§ 4. ДАЛЬНЕЙШАЯ МОДИФИКАЦИЯ АЛГОРИТМА.

Плодотворной является идея [15-16] использовать для выявления бесперспективных областей списка S значения целевой функции в каких-либо точках этих областей. Пусть

$$Val(\gamma) = \frac{\rho(\tilde{x})}{q(\tilde{x})}, \text{ где } \tilde{x} - \text{какая-нибудь точка}$$

из γ .

Если, например, области $\mathcal{V} \in \Pi$ и $\mathcal{T} \in \Pi$ таковы, что

$$Val(\mathcal{V}) < Low(\mathcal{T}), \quad (I3)$$

то выбрасывание $\mathcal{T}$ из списка областей никак не повлияет на результаты работы алгоритма. Действительно,

$$\min_{\Pi} \left\{ \frac{p(x)}{q(x)} \right\} \leq Val(\mathcal{V}), \quad (I4)$$

поэтому из (I3) следует, что $\mathcal{T}$ никогда не делается текущей. Кроме того, то обстоятельство, что (I3) выполняется для любой области $\mathcal{V}$, когда-либо присутствовавшей в списке $\mathcal{S}$, может использоваться для контроля точности результатов вычислений.

Конечно, было бы идеальным выбирать $\tilde{x} \in \underset{\mathcal{T}}{Argmin} \{ p(x)/q(x) \}$.

Мы так и будем поступать в случае точно оцениваемых областей $\mathcal{T}$, полагая $Val(\mathcal{T}) = Low(\mathcal{T})$, но в общей ситуации подобный "удачный" подбор $\tilde{x}$ не менее труден, чем решение исходной задачи.

Поэтому в целях минимизации возможных отклонений $\tilde{x}$ от множества $\underset{\mathcal{T}}{Argmin} \{ p(x)/q(x) \}$ целесообразней всего брать в качестве $\tilde{x}$ центр области $\mathcal{T}$ (или близкую к нему внутреннюю точку $\mathcal{T}$ при $q(med \mathcal{T}) = 0$), если $\mathcal{T}$ не является точно оцениваемой. В [I5] процедура проверки условия, аналогичного (I3), так и называется - *midpoint test*.

Но на пути реализации изложенной выше идеи стоят значительные трудности. Дело в том, что в алгоритме ВЕПРА диаметры текущих областей не обязательно стремятся к нулю, т.к. их рассечение осуществляется лишь по инцидентным координатам. Эта особенность алгоритма ВЕПРА является, несомненно, положительной, поскольку именно благодаря ей обеспечивается экономия и целенаправленность вычислительных усилий: мы не скатываемся на беспорядочное измельчение прямоугольника Π , а рассекаем области только так, чтобы обеспечить наибольшее улучшение целевой функции. Но эта же особенность служит причиной того, что даже для текущих областей

разность ($Val(\Omega) - Low(\Omega)$) может оставаться всегда больше некоторого положительного числа. Вычислительные эксперименты свидетельствуют о том, что при этом неравенство (13), как правило, никогда не выполняется и ни одну из областей списка нельзя признать бесперспективной.

Ниже излагается специальный прием, позволяющий значительно упростить оптимизацию некоторых рациональных выражений. Он основан на вводимом нами далее понятии "отщепления" подмножества переменных (у функции). Вкратце его сущность можно охарактеризовать следующим образом: оптимизируя на прямоугольниках выражения вида $f(x) \cdot g(x)$ или $f(x)/g(x)$, мы, при выполнении некоторых специальных условий на f и g , имеем возможность *a priori*, не решая задачу целиком, узнать часть значений переменных, доставляющих искомый экстремум. Упомянутые "специальные условия" зависят как от вида функций f и g , так и от прямоугольника, на котором выполняется оптимизация, и в нужном нам случае легко проверяются. Все это позволит на основании полученной об экстремуме информации снижать размерность исходной оптимизационной задачи.

Основанная на этих соображениях техника будет использована при построении модифицированного алгоритма ВЕГРАМ вычисления (9). ориентированного на решение задач средней и не слишком большой размерности. Его отличительными особенностями являются меньший, в сравнении с ВЕГРА, объем требуемой оперативной памяти ЭВМ и гарантированная оценка точности получающегося результата.

Следующим определением выделяются функции, у которых некоторые переменные как бы независимы от остальных в том смысле, что конкретные значения этих переменных никак не влияют на достижение функцией экстремума по другим переменным.

Определение. Будем говорить, что в функции $f(x_1, x_2, \dots, x_n)$ на области $P = [a_1; b_1] \times [a_2; b_2] \times \dots$ совокупность (подмножество)

переменных $\{x_1, x_2, \dots, x_k\}$, $k < n$ - отщепляющаяся (отщепляется), если существуют такие фиксированные значения аргументов

$\tilde{x}_{k+1}, \tilde{x}_{k+2}, \dots, \tilde{x}_n$ из $[a_{k+1}; b_{k+1}], \dots, [a_n; b_n]$, соответственно, что они доставляют экстремум функции от $(n-k)$ переменных

$$\varphi_{\tilde{x}_1, \dots, \tilde{x}_k}(x_{k+1}, \dots, x_n) = f(\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_k, x_{k+1}, \dots, x_n)$$

на $[a_{k+1}; b_{k+1}] \times \dots \times [a_n; b_n]$ при любых $\tilde{x}_i \in [a_i; b_i], i=1, 2, \dots, k$.

В соответствии с типом рассматриваемого экстремума мы будем отмечать иногда, что совокупность $\{x_1, x_2, \dots, x_k\}$ отщепляется относительно максимума или же относительно минимума, а когда отщепляющаяся совокупность состоит из одной переменной, то саму эту переменную назовем отщепляющейся.

Равносильное определение:

подмножество $\{x_1, x_2, \dots, x_k\}$ отщепляется в $f(x_1, x_2, \dots, x_n)$ на $\Pi = [a_1; b_1] \times \dots \times [a_n; b_n]$ тогда и только тогда, когда

$$\bigcap_{\substack{\tilde{x}_i \in [a_i; b_i] \\ i=1, 2, \dots, k}} \left\{ \underset{\substack{x_i \in [a_i; b_i] \\ i=k+1, \dots, n}}{\text{Arg ext}} f(\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_k, x_{k+1}, \dots, x_n) \right\} \neq \emptyset. \quad (15)$$

Ввиду важности множества (15) в дальнейших рассмотрениях назовем

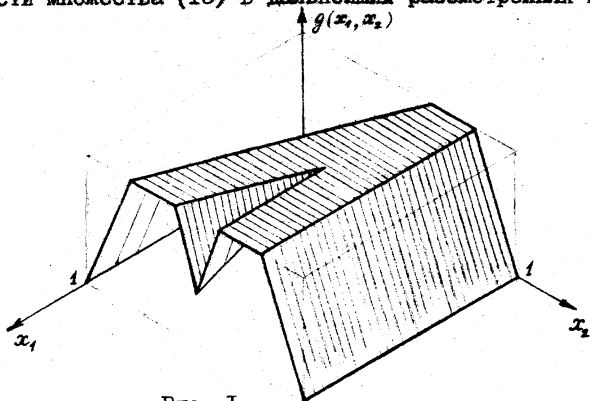


Рис. I

его гребнем функции f на Π и иногда будем обозначать

$\underset{\Pi}{cre} f$ (от английского *crest*), если нет двусмысленностей с типом экстремума и отщепляющимся подмножеством переменных. Отметим, что сами множества $\underset{x_i \in [a_i; b_i], i=k+1, \dots, n}{Argext_{\Pi} f(\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_k, x_{k+1}, \dots, x_n)}$

могут и не совпадать друг с другом, а их строение может быть достаточно сложным, как, например, у функции двух переменных $g(x_1, x_2)$, кусок графика которой изображен на рисунке I и у которой переменная x_1 отщепляется на $[0; 1] \times [0; 1]$ относительно максимума.

Таким образом, свойство подмножества переменных "быть отщепляемым" зависит как от самой функции f , так и от прямоугольной области определения f .

Допустим, что в $f(x_1, x_2, \dots, x_n)$ на Π отщепляется подмножество $\{x_1, x_2, \dots, x_k\}$, причем известна некоторая априорная информация о множествах $\underset{x_i \in [a_i; b_i], i=k+1, \dots, n}{Argext_{\Pi} f(\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_k, x_{k+1}, \dots, x_n)}$

при различных $\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_k$. Например, что они не зависят от конкретных значений $\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_k$, или что все

$\underset{x_i \in [a_i; b_i], i=k+1, \dots, n}{Argext_{\Pi} f}$ являются одноточечными множествами. В подобных ситуациях соответствующий гребень может быть легко описан и это существенно упрощает оптимизационные задачи. Действительно, сначала можно зафиксировать какие-нибудь произвольные, выгодные по тем или иным причинам значения переменных $\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_k$ и сконцентрировать усилия на оптимизации функции от меньшего числа переменных

$$\varphi(x_{k+1}, x_{k+2}, \dots, x_n) = f(\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_k, x_{k+1}, \dots, x_n)$$

на прямоугольнике $[a_{k+1}; b_{k+1}] \times \dots \times [a_n; b_n]$. Отыскав доставляющие экстремум φ значения переменных $\tilde{x}_{k+1}, \dots, \tilde{x}_n$, мы найдем тем самым точку гребня $\underset{\Pi}{cre} f$. Для вычисления $\underset{\Pi}{ext} f$ достаточно теперь решить задачу

$$f(x_1, x_2, \dots, x_k, \tilde{x}_{k+1}, \dots, \tilde{x}_n) \rightarrow \underset{x_i \in [a_i; b_i], i=1, 2, \dots, k,}{ext},$$

с k переменными. Если же у такой функции $f(x_1, x_2, \dots, x_n)$ отщепляемой на Π является любая переменная, то оптимизация f на Π сводится просто к n одномерным задачам.

Примеры.

1). В линейном функционале

$$f(x_1, x_2, \dots, x_n) = \alpha_1 x_1 + \alpha_2 x_2 + \dots + \alpha_n x_n$$

любое подмножество переменных отщепляется на любой прямоугольной области в $\mathcal{R}^n$.

2). Если $f(x_1, x_2, \dots, x_n) = g(x_1, x_2, \dots, x_k) + h(x_{k+1}, \dots, x_n)$,

то множества переменных $\{x_1, x_2, \dots, x_k\}$ и $\{x_{k+1}, \dots, x_n\}$ отщепляются в f на любой прямоугольной области. В частности, у функции

$$g(x) = \sum_{j=1}^n |x_j|$$

отщепляемым является любое подмно-

жество переменных.

3). В функции $f(x_1, x_2) = x_1 x_2$ переменные отщепляются на любой прямоугольной области из $\mathcal{R}^2$, у которой соответствующая координата принимает только неположительные или только неотрицательные значения. Иначе отщепления может не быть, что видно на примере области $[-1; 1] \times [1; 2] \subseteq \mathcal{R}^2$. Действительно,

$$\min_{x_2} f(t, x_2) = t \quad \text{при } t \geq 0 \quad \text{и достигается при } x_2 = 1,$$

$$\min_{x_2} f(t, x_2) = -2t \quad \text{при } t \leq 0 \quad \text{и достигается при } x_2 = 2.$$

4). Если $f(x_1, x_2, \dots, x_n) = g(x_1, x_2, \dots, x_k) \cdot h(x_{k+1}, \dots, x_n)$, то подмножества переменных $\{x_1, x_2, \dots, x_k\}$ и $\{x_{k+1}, \dots, x_n\}$ отщепляются в f на любой прямоугольной области $\Pi = [a_1; b_1] \times \dots \times [a_k; b_k] \times [a_{k+1}; b_{k+1}] \times \dots \times [a_n; b_n]$ такой, что функции g и h сохраняют свой знак на $[a_1; b_1] \times \dots \times [a_k; b_k]$ и $[a_{k+1}; b_{k+1}] \times \dots \times [a_n; b_n]$, соответственно.

Легко видеть, что пример 4) сводится к 2) логарифмированием.

5). У функции $h: \mathcal{R}^2 \supseteq [0; 2] \times [-1; 1] \rightarrow \mathcal{R}$, график которой приведен на рисунке 2а), переменная x_1 отщепляется относительно максимума, соответствующим гребнем является отрезок $[0; 2] \times 0$. Относительно минимума x_1 не отщепляется, а переменная x_2 не отщепляется, как легко видеть, ни относительно максимума, ни относительно минимума.

График функции $h(x_1, x_2)$ получается довольно просто. Нужно

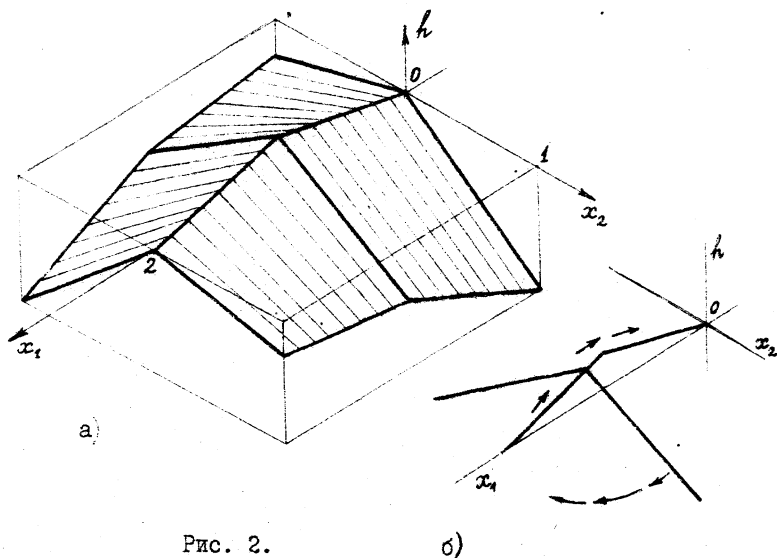


Рис. 2.

б)

плоский угол двигать вдоль "горбика" в плоскости Ox_1h так, как это показано на рисунке 2б) - чтобы вершина угла скользила по "горбику", сам угол в процессе движения поворачивался, а его плоскость оставалась параллельной координатной плоскости Ox_2h . Исходя из геометрического описания нетрудно выписать и аналитическое выражение для $h(x_1, x_2)$.

Перепишем определение отщепления совокупности переменных в более формализованном виде:

$\{x_1, x_2, \dots, x_n\}$ отщепляется в $f(x_1, x_2, \dots, x_n)$ относительно

минимума на $\Pi = [a_1; b_1] \times \dots \times [a_n; b_n]$ тогда и только тогда, когда $\exists \check{x}_i \in [a_i; b_i], i = k+1, \dots, n$, такие, что $\forall x_i \in [a_i; b_i], i = 1, 2, \dots, n$ имеет место неравенство $f(x_1, x_2, \dots, x_n) \geq f(x_1, x_2, \dots, x_k, \check{x}_{k+1}, \dots, \check{x}_n)$. (I6)

Аналогично и с отщеплением относительно максимума, но тогда неравенство (I6) заменяется на противоположное:

$$f(x_1, x_2, \dots, x_n) \leq f(x_1, x_2, \dots, x_k, \hat{x}_{k+1}, \dots, \hat{x}_n) \quad (I7)$$

для некоторых $\hat{x}_i \in [a_i; b_i], i = k+1, \dots, n$.

Пусть требуется отыскать минимум дроби $f(x)/g(x)$ на прямоугольнике $\Pi \subseteq R^n$, причем первые k переменных отщепляются на Π у функции f относительно минимума, а у функции g — относительно максимума. Тогда существуют $\check{x}_i, \hat{x}_i \in [a_i; b_i], i = k+1, \dots, n$, такие, что в силу (I6) и (I7) справедливо

$$\frac{f(x_1, x_2, \dots, x_n)}{g(x_1, x_2, \dots, x_n)} \geq \frac{f(x_1, x_2, \dots, x_k, \check{x}_{k+1}, \dots, \check{x}_n)}{g(x_1, x_2, \dots, x_k, \hat{x}_{k+1}, \dots, \hat{x}_n)} \quad \text{для всех } x \in \Pi.$$

Если к тому же $\check{x}_i = \hat{x}_i, i = k+1, \dots, n$, то

$$\min_{x \in \Pi} \left\{ \frac{f(x)}{g(x)} \right\} = \min_{\substack{x_i \in [a_i; b_i] \\ i=1, 2, \dots, k}} \left\{ \frac{f(x_1, x_2, \dots, x_k, \check{x}_{k+1}, \dots, \check{x}_n)}{g(x_1, x_2, \dots, x_k, \hat{x}_{k+1}, \dots, \hat{x}_n)} \right\} \quad (I8)$$

количество переменных задачи уменьшится до k . Похожее рассуждения можно провести и при максимизации дроби $f(x)/g(x)$, а также при минимизации и максимизации произведения $f(x) \cdot g(x)$ на прямоугольнике $\Pi \subseteq R^n$.

В связи с вышеизложенным возникают по крайней мере два вопроса: как выявить отщепление того или иного подмножества переменных? если известно, что подмножество переменных отщепляется, то как найти хотя бы одну точку из гребня? Мы не собираемся разви-

вать здесь большую и общую теорию, наша цель существенно более скромна - это вычисление минимума (9), и усовершенствование, по возможности, алгоритма ВЕГРА.

В знаменателе минимизируемой дроби стоит "хорошее" выражение

$$q(x) = \sum_{j=1}^n |x_j| ,$$

у которого любое подмножество переменных отщепляется относительно максимума на любой прямоугольной области Π , и гребень

в $q(x)$ находится очень просто. С числителем посложнее: наличие операции взятия модуля приводит к тому, что нужное нам отщепление не всегда имеет место. Однако, если на прямоугольнике

$\Pi \subseteq \mathcal{R}^n$ выражение под знаком модуля сохраняет один определенный знак, то

$$p(x) = \omega - |m - \sum_{j=1}^n x_j u_j| = \begin{cases} \omega - m + \sum_{j=1}^n x_j u_j, & \text{или же} \\ \omega + m - \sum_{j=1}^n x_j u_j, \end{cases}$$

и отщепляющимся в $p(x)$ является любое подмножество переменных. Но подобное "полное расщепление", как правило, никогда и не требуется. Чаще нужно отщепление вполне определенного подмножества переменных, а в таких ситуациях условие знаковостояния выражения под знаком модуля является чересчур грубым и избыточным.

Как же выявить отщепление переменных? Применять для этой цели непосредственно определение практически невозможно, т.к. для функции $f(x_1, x_2, \dots, x_n)$ проверка отщепления даже одной переменной x_i требует решения бесконечной совокупности оптимизационных задач

$$(f(t, x_2, \dots, x_n) \rightarrow \text{extr} , x_i \in [a_i; b_i] , i=2, 3, \dots, n)$$

$$\text{для } t \in [a_1; b_1]$$

и сравнения получающихся результатов между собой. К счастью, для $p(x)$ удастся получить сравнительно простой критерий отщепления подмножества переменных, работающий за $O(n)$ арифметических операций.

Принимая во внимание наши допущения о структуре множества $\text{Argmin } p$, и поскольку при любых $t_j \in [a_j; b_j]$, $j=1, 2, \dots, k$, имеет место

$$\begin{aligned} \min_{\substack{x_j \in [a_j; b_j] \\ j=k+1, \dots, n}} p(t_1, t_2, \dots, t_k, x_{k+1}, \dots, x_n) = \\ = \min_{x_j \in [a_j; b_j]} \left\{ \omega - \left| m - \sum_{j=1}^k t_j u_j - \sum_{j=k+1}^n x_j u_j \right| \right\} \\ = \omega - \max \left\{ \left| m - \sum_{j=1}^k t_j u_j - \sum_{j=1}^n [a_j; b_j] \cdot u_j \right|, \right. \\ \left. \left| m - \sum_{j=1}^k t_j u_j - \sum_{j=1}^n [a_j; b_j] \cdot u_j \right| \right\}, \end{aligned}$$

мы можем заключить, что при любых $t \in \mathbb{R}^k$

либо

$$\text{Argmin}_{x_j \in [a_j; b_j]} p(t_1, \dots, t_k, x_{k+1}, \dots) = \Lambda = \text{Argmin}_{x_j \in [a_j; b_j]} \left(\sum_{j=k+1}^n x_j u_j \right) \in \mathbb{R}^{n-k},$$

либо

$$\text{Argmin}_{x_j \in [a_j; b_j]} p(t_1, t_2, \dots, t_k, x_{k+1}, \dots) = P = \text{Argmax}_{x_j \in [a_j; b_j]} \left(\sum_{j=k+1}^n x_j u_j \right) \in \mathbb{R}^{n-k}$$

Пересекаться множества Λ и P могут лишь когда

$$(a_i - b_i) u_i = 0 \quad \text{для } i=k+1, \dots, n \quad \text{и при этом } \Lambda = P.$$

Следовательно, отщепление у функции $p(x)$ подмножества переменных $\{x_1, x_2, \dots, x_k\}$ автоматически влечет то, что соответствующий гребень сегр совпадает с $\text{Argmin}_{x_j \in [a_j; b_j]} p(t_1, \dots, t_k, x_{k+1}, \dots)$

при любых $t_i \in [a_i; b_i]$, $i=1, \dots, k$. Аналогично обстоит дело и с другими отщепляющимися подмножествами.

Пусть

$$(\alpha_1, \alpha_2, \dots, \alpha_k) \in \underset{x_j \in [a_j; b_j]}{\operatorname{Argmin}} \left(\sum_{j=1}^k x_j u_j \right) \in \mathbb{R}^k,$$

$$(\beta_1, \beta_2, \dots, \beta_k) \in \underset{x_j \in [a_j; b_j]}{\operatorname{Argmax}} \left(\sum_{j=1}^k x_j u_j \right) \in \mathbb{R}^k.$$

Лемма 5. В функции $\rho(x)$ совокупность переменных $\{x_1, x_2, \dots, x_k\}$ отщепляется на области $\Pi = [a_1; b_1] \times \dots \times [a_n; b_n]$ относительно минимума тогда и только тогда, когда

$$\underset{x_j \in [a_j; b_j]}{\operatorname{Argmin}} \rho(\alpha_1, \dots, \alpha_k, x_{k+1}, \dots) \cap \underset{x_j \in [a_j; b_j]}{\operatorname{Argmin}} \rho(\beta_1, \dots, \beta_k, x_{k+1}, \dots) \neq \emptyset. \quad (I9)$$

Учитывая изложенные ранее замечания о строении множеств

$\underset{x_j \in [a_j; b_j]}{\operatorname{Argmin}} \rho(t_1, \dots, t_k, x_{k+1}, \dots)$ при произвольном $t \in \mathcal{K}^k$, мы можем условие (I9) заменить на эквивалентное:

$$\underset{x_j \in [a_j; b_j]}{\operatorname{Argmin}} \rho(\alpha_1, \dots, \alpha_k, x_{k+1}, \dots) = \underset{x_j \in [a_j; b_j]}{\operatorname{Argmin}} \rho(\beta_1, \dots, \beta_k, x_{k+1}, \dots).$$

Доказательство. Необходимость очевидна.

Достаточность докажем "от противного", предположив, что в условиях леммы подмножество $\{x_1, x_2, \dots, x_k\}$ не отщепляющееся. Тогда для некоторой точки $t = (t_1, \dots, t_k) \in [a_1; b_1] \times \dots \times [a_k; b_k]$ имеет место

$$\underset{x_j \in [a_j; b_j]}{\operatorname{Argmin}} \rho(t_1, \dots, t_k, x_{k+1}, \dots) \neq \left(\begin{array}{l} \text{общее значение } \underset{x_j \in [a_j; b_j]}{\operatorname{Argmin}} \rho(\alpha_1, \dots, \alpha_k, x_{k+1}, \dots) \\ \text{и } \underset{x_j \in [a_j; b_j]}{\operatorname{Argmin}} \rho(\beta_1, \dots, \beta_k, x_{k+1}, \dots) \end{array} \right)$$

Предположим для определенности, что

$$\underset{x_j \in [a_j; b_j]}{\operatorname{Argmin}} \rho(t_1, t_2, \dots, t_k, x_{k+1}, \dots, x_n) = \mathcal{A}, \quad (20)$$

$$\begin{aligned}
 a \quad & \underset{x_j \in [a_j; b_j]}{\operatorname{Argmin}} \rho(\alpha_1, \alpha_2, \dots, \alpha_k, x_{k+1}, \dots, x_n) = \\
 & = \underset{x_j \in [a_j; b_j]}{\operatorname{Argmin}} \rho(\beta_1, \beta_2, \dots, \beta_k, x_{k+1}, \dots, x_n) = P.
 \end{aligned} \tag{20}$$

(Противоположный случай может быть рассмотрен совершенно аналогично.)

Из определения векторов α и β следует, что

$$\sum_{j=1}^k \alpha_j u_j \leq \sum_{j=1}^k t_j u_j \leq \sum_{j=1}^k \beta_j u_j,$$

т.е.

$$\sum_{j=1}^k (t_j - \beta_j) u_j \leq 0 \leq \sum_{j=1}^k (t_j - \alpha_j) u_j. \tag{21}$$

Далее, если $\lambda = (\lambda_{k+1}, \lambda_{k+2}, \dots) \in L$, $\rho = (\rho_{k+1}, \dots, \rho_n) \in P$,

то

$$\left| m - \sum_{j=1}^k t_j u_j - \sum_{j=k+1}^n \lambda_j u_j \right| \geq \left| m - \sum_{j=1}^k t_j u_j - \sum_{j=k+1}^n \rho_j u_j \right|.$$

И снова для определенности предположим, что подмодульное выражение в левой части этого неравенства неотрицательно. Разбор случая

$$m - \sum_{j=1}^k t_j u_j - \sum_{j=k+1}^n \lambda_j u_j \leq 0$$

ничем не отличается от того, что будет изложено ниже.

Итак,

$$m - \sum_{j=1}^k t_j u_j - \sum_{j=k+1}^n \lambda_j u_j \geq \left| m - \sum_{j=1}^k t_j u_j - \sum_{j=k+1}^n \rho_j u_j \right|.$$

Добавим к обеим частям этого неравенства неотрицательную в силу (21) величину $\sum_{j=1}^k (t_j - \alpha_j) u_j$, получим

$$m - \sum_{j=1}^k \alpha_j u_j - \sum_{j=k+1}^n \lambda_j u_j \geq \left| m - \sum_{j=1}^k t_j u_j - \sum_{j=k+1}^n \rho_j u_j \right| + \left| \sum_{j=1}^k (t_j - \alpha_j) u_j \right|$$

или, поскольку левая часть остается неотрицательной,

$$\left| m - \sum_{j=1}^k \alpha_j u_j - \sum_{j=k+1}^n \lambda_j u_j \right| \geq \left| m - \sum_{j=1}^k \alpha_j u_j - \sum_{j=k+1}^n \rho_j u_j \right|,$$

что означает

$$\rho(\alpha_1, \alpha_2, \dots, \alpha_k, \lambda_{k+1}, \dots, \lambda_n) \leq \rho(\alpha_1, \alpha_2, \dots, \alpha_k, \rho_{k+1}, \dots, \rho_n).$$

Очевидное противоречие с допущением (20) завершает доказательство леммы.

Изложенный в лемме 5 критерий отщепления группы переменных в $\rho(x)$ можно переформулировать в виде, более удобном для алгоритмической реализации:

совокупность переменных $\{x_1, x_2, \dots, x_k\}$ отщепляется у функции $\rho(x)$ на прямоугольнике $\Pi = [a_1; b_1] \times \dots \times [a_n; b_n]$ относительно минимума тогда и только тогда, когда разности

$$\left| \frac{m - \sum_{j=1}^k [a_j; b_j] \cdot u_j - \sum_{j=k+1}^n [a_j; b_j] \cdot u_j}{m - \sum_{j=1}^k [a_j; b_j] \cdot u_j - \sum_{j=k+1}^n [a_j; b_j] \cdot u_j} \right| - \quad (22)$$

$$\begin{aligned}
 & \left| \frac{m - \sum_{j=1}^K [a_j; b_j] \cdot u_j - \sum_{j=K+1}^n [a_j; b_j] \cdot u_j}{m - \sum_{j=1}^K [a_j; b_j] \cdot u_j - \sum_{j=K+1}^n [a_j; b_j] \cdot u_j} \right| - \\
 & - \left| \frac{m - \sum_{j=1}^K [a_j; b_j] \cdot u_j - \sum_{j=K+1}^n [a_j; b_j] \cdot u_j}{m - \sum_{j=1}^K [a_j; b_j] \cdot u_j - \sum_{j=K+1}^n [a_j; b_j] \cdot u_j} \right| \quad (22)
 \end{aligned}$$

либо одновременно ≥ 0 , либо одновременно ≤ 0 .

Равносильность этих условий утверждению леммы 5 достаточно очевидна.

Теперь все готово для построения алгоритма БЕГРАМ. Его главное отличие от алгоритма БЕГРА заключается в наличии процедуры проверки условия (22) "отщепления" переменных, а также в действиях, предпринимаемых при наличии этого отщепления.

Пусть на некотором шаге алгоритма текущая область Ω разбивается на потомки Ω' и Ω'' . Если для инцидентных компонент в Ω' (Ω'' , соответственно) выполнено аналогичное (22) условие, то это означает их отщепление в $\rho(x)$ на Ω' (Ω''). Тогда осуществляем "сжатие" области Ω' (или Ω''), т.е. заменяем $\Omega' = [a'_1; b'_1] \times \dots \times [a'_n; b'_n]$ на такую область

$$[\bar{a}'_1; \bar{b}'_1] \times [\bar{a}'_2; \bar{b}'_2] \times \dots \times [\bar{a}'_n; \bar{b}'_n] \quad , \text{ что}$$

$$\left\{ \begin{array}{l} \bar{a}'_i = a'_i, \bar{b}'_i = b'_i, \text{ если } i\text{-ая компонента инцидентна в } \Omega'; \\ \bar{a}'_i = \bar{b}'_i = (\text{общему значению } (\text{Argmin}_{\Omega'} \rho)_i \text{ и } (\text{Argmax}_{\Omega'} q)_i), \text{ если } i\text{-ая компонента неинцидентна в } \Omega'. \end{array} \right.$$

В силу (18) такая замена никак не изменит величину $\min_{\Omega'} \left\{ \frac{\rho(x)}{q(x)} \right\}$.

Кроме того, как легко видеть, и значения $\text{Argmin}_{\Omega'} \rho(x)$,

$\text{Argmax}_{\Omega'} q(x)$, $\text{Low}(\Omega')$ останутся неизменными при "сжатии" области Ω' . Следовательно, процедура "сжатия" никак не повлияет

жет на сходимость алгоритма. В новой, сжатой области Ω' все компоненты с ненулевой длиной инцидентны. Поскольку инцидентные компоненты постоянно подвергаются рассечению алгоритмом, то это позволяет надеяться, что диаметры текущих областей могут быть сделаны в процессе работы алгоритма сколь угодно малыми, и, следовательно $\inf (Val(\Omega) - Low(\Omega)) = 0$ для текущих Ω . Все это

создаст более благоприятные условия для использования описанного в начале § 4 рецепта выявления бесперспективных областей.

Целесообразно оформить его следующим образом. С алгоритмом связывается вещественный параметр $Minval$, принимающий значение наименьшей из оценок Val по всем областям, когда-либо присутствовавшим в списке S . $Minval$ можно перевычислять на каждом шаге алгоритма или же периодически через некоторое количество шагов. Ясно, что последовательность параметров $Minval$

— невозрастающая. Область T признается бесперспективной и удаляется из списка S , если $Low(T) \geq Minval$.

Этот тест области на бесперспективность является более сильным, нежели используемый в алгоритме ВЕГРА. В самом деле, если в процессе работы алгоритма ВЕГРАМ в списке S появилась точно оцениваемая область γ , то значение параметра $Minval$, начиная с этого же шага, становится не большим $Low(\gamma)$. Далее, следовательно, из списка S всегда будут удаляться области с оценкой Low , не меньшей чем $Low(\gamma)$, что соответствует этапам 4 и 5 одного шага алгоритма ВЕГРА.

Докажем правомерность использования теста в случае, когда

$Minval = Val(\gamma)$ для области γ , не являющейся точно оцениваемой. В начале параграфа мы уже отмечали, что если некая область T такова, что $Low(T) > Val(\gamma)$, то она никогда не сделается текущей. Рассмотрим поэтому ситуацию

$$Low(T) = Val(\gamma)$$

Пусть T все-таки стала текущей. Тогда

$$\text{Low}(T) \leq \min_P \left\{ \frac{\rho(x)}{q(x)} \right\},$$

да еще

$$\min_P \left\{ \frac{\rho(x)}{q(x)} \right\} \leq \text{Val}(T).$$

Следовательно, глобальный минимум целевой функции $\left(\frac{\rho(x)}{q(x)} \right)$ на исходном прямоугольнике P (а, значит, и на T) равен оценке $\text{Val}(T)$. Поскольку мы предположили, что T — не точно оцениваемая, то $\text{Val}(T) = \rho(\tilde{x})/q(\tilde{x}) = \min_T \left\{ \frac{\rho(x)}{q(x)} \right\}, \tilde{x} \in \text{int } T$.

Насколько это возможно?

В [6] была показана квазивогнутость дроби $\rho(x)/q(x)$, из которой следовало, что ее минимум достигается в крайних точках выпуклой области определения. Справедливым оказывается и более сильное утверждение: минимум дроби $\rho(x)/q(x)$ на интервальном векторе не может достигаться в точках, не являющихся его вершинами. Докажем это.

Пусть $\tilde{x}$ — какая-нибудь точка, отличная от вершин области T . Таким образом, $\tilde{x}$ является внутренней для некоторого отрезка I , целиком лежащего в T , причем в нашем случае можно считать этот отрезок параллельным какой-нибудь координатной оси, например, k -ой (если длина k -ой компоненты T ненулевая). Ясно, что если целевая функция $(\rho(x)/q(x))$ определена в $\tilde{x}$, то $q(\tilde{x}) = \sum_{j=1}^k |\tilde{x}_j| \neq 0$ и это неравенство выполняется в некоторой достаточно малой окрестности O точки $\tilde{x}$ относительно I . Зафиксируем значения всех компонент аргумента кроме k -ой, тогда

$$\frac{\rho(x)}{q(x)} = \frac{\omega - |C_1 - x_k u_k|}{C_2 + |x_k|},$$

где $C_1 = m - \sum_{\substack{j=1 \\ j \neq k}}^n \tilde{x}_j u_j$, $C_2 = \sum_{\substack{j=1 \\ j \neq k}}^n |\tilde{x}_j|$ и

$$C_2 + |x_k| \neq 0 \quad \text{в} \quad O.$$

Далее могут представиться два случая.

I) $\tilde{x}_k = 0$ или $C_1 - \tilde{x}_k u_k = 0$. (23)

Положим

$$\mathcal{O}_1 = \{y \in \mathcal{R} \mid |C_1 - (\tilde{x}_k + y) u_k| \geq |C_1 - \tilde{x}_k u_k|\},$$

$$\mathcal{O}_2 = \{y \in \mathcal{R} \mid |\tilde{x}_k + y| > |\tilde{x}_k|\}.$$

Из свойств абсолютной величины числа следует, что

либо $\mathcal{O}_1 \supseteq \{y \in \mathcal{R} \mid y \geq 0\}$, либо $\mathcal{O}_1 \supseteq \{y \in \mathcal{R} \mid y \leq 0\}$,

либо вообще $\mathcal{O}_1 = \mathcal{R}$, когда $C_1 - \tilde{x}_k u_k = 0$ или $u_k = 0$.

Аналогично, либо $\mathcal{O}_2 \supseteq \{y \in \mathcal{R} \mid y > 0\}$, либо

$\mathcal{O}_2 \supseteq \{y \in \mathcal{R} \mid y < 0\}$, либо, при $\tilde{x}_k = 0$, $\mathcal{O}_2 = \mathcal{R}$.

Другими множества $\mathcal{O}_1$ и $\mathcal{O}_2$ быть не могут. В любом случае пересечение $\mathcal{O}_1 \cap \mathcal{O}_2$, при условии (23), включает в себя все отрицательные числа или все положительные числа.

Существует поэтому $\delta \in \mathcal{O}_1 \cap \mathcal{O}_2$, $\delta \neq 0$ такое, что

$\tilde{x} + \delta e_k \in O \subseteq I \subseteq \mathcal{U}$, где e_k - орт k -ой координатной оси. Легко видеть, что $\rho(\tilde{x} + \delta e_k) \leq \rho(\tilde{x})$, $q(\tilde{x} + \delta e_k) > q(\tilde{x})$,

а потому

$$\frac{\rho(\tilde{x} + \delta e_k)}{q(\tilde{x} + \delta e_k)} < \frac{\rho(\tilde{x})}{q(\tilde{x})}.$$

Итак, в O существует точка, в которой значение целевой функции меньше, чем в $\tilde{x}$.

2). $\tilde{x}_k \neq 0$ и $C_1 - \tilde{x}_k u_k \neq 0$.

Обозначим через O_1 ту окрестность $\tilde{x}$, в которой выполнены эти неравенства. В $O \cap O_1$ справедливо

$$\frac{\rho(x)}{q(x)} = \frac{\omega - (C_1 - x_k u_k) \cdot \operatorname{sgn}(C_1 - \tilde{x}_k u_k)}{C_2 + x_k \operatorname{sgn} \tilde{x}_k}. \quad (24)$$

Хорошо известно, что дробно-линейная функция, каковой является (24), строго монотонна на вещественной оси всюду, где она определена. Следовательно, и в этом случае в $O_1 \cap O$ обязательно найдутся точки, в которых значение дроби $\rho(x)/q(x)$ меньше, чем $\rho(\tilde{x})/q(\tilde{x})$.

Утверждение доказано. Вместе с тем нами полностью обоснован тест области на бесперспективность.

Условием остановки алгоритма может служить

$$(Minval - \text{текущая оценка } Low) \leq \varepsilon$$

для заранее заданного ε , или же

$$\frac{Minval - \text{текущая оценка } Low}{\text{текущая оценка } Low} \leq \varepsilon$$

в зависимости от того, требуется обеспечить абсолютную или относительную точность ϵ .

С учетом всего изложенного выше краткое описание алгоритма ВЕГРАМ - модификации алгоритма ВЕГРА - выглядит следующим образом. Перед началом работы ВЕГРАМ список областей S состоит из единственной области - исходного прямоугольника $\Pi =$

$= [a_1; b_1] \times [a_2; b_2] \times \dots \times [a_n; b_n]$. Кроме того, вычислены $Low(\Pi)$, $Val(\Pi)$, $Argmin p$, $Argmax q$, а начальное значение параметра $Minval$ полагается равным $Val(\Pi)$. Каждый шаг алгоритма ВЕГРАМ заключается в выполнении нескольких этапов:

1. Просматривая полученный с предыдущего шага список областей S , находим в нем текущую область Ω и выбрасываем из S все такие области Υ , что

$$Low(\Upsilon) \geq Minval$$

2. Если Ω - точно оцениваемая область или $Minval - Low(\Omega) \leq \epsilon$, останавливаем алгоритм.

3. Рассекаем Ω по самой длинной из инцидентных компонент на потомки Ω' и Ω'' , вычисляем для них значения $Argmin p$, $Argmax q$, оценки Low , Val .

4. Проверяем отщепление у функции $p(x)$ на Ω' подмножества переменных, соответствующего инцидентным компонентам Ω' . Если это отщепление имеет место, "сжимаем" область Ω' , перевычисляем оценку Val для новой области Ω' .

Аналогичную процедуру проделываем и для области Ω''

5. Заносим Ω' и Ω'' в список S вместо Ω .

6. Перевычисляем параметр $Minval$:

$$\text{если } Minval > \min \{ Val(\Omega'), Val(\Omega'') \} = v,$$

то присваиваем $Minval$ значение v .

Остается лишь сообщить, что многочисленные тестовые расчеты, проведенные с алгоритмами ВЕГРА и ВЕГРАМ, свидетельствуют об

их надежной сходимости к глобальному минимуму (9). При этом, в случае алгоритма ВЕГРАМ через некоторое количество шагов рост длины списка областей, как правило, замедлялся (в сравнении с ВЕГРА), а при дальнейшем итерировании происходила стабилизация длины списка.

Л И Т Е Р А Т У Р А

1. Алефельд Г., Херцбергер Ю. Введение в интервальные вычисления. - М.: Мир, 1987.
2. Калмыков С.А., Шокин Ю.И., Юлдашев З.Х. Методы интервального анализа. - Новосибирск: Наука, 1986.
3. Хлебалин Н.А. Аналитический метод синтеза регуляторов в условиях неопределенности параметров объекта // Аналитические методы синтеза регуляторов. - Саратов: Саратовск. политехн. ин-т, 1981. - С. 107-123.
4. Neumaier A. Tolerance analysis with interval arithmetic // Freiburger Intervall-Berichte. - 1986. - №9. - P. 5-19.
5. Захаров А.В., Шокин Ю.И. Алгебраическое интервальное решение систем линейных интервальных уравнений $Ax = b$ и $Ax + d = b$. - Красноярск, 1987. (Препринт / ВЦ СО АН СССР: №5).
6. Шайдуров В.В., Шарый С.П. Решение интервальной алгебраической задачи о допусках. - Красноярск, 1988. (Препринт / ВЦ СО АН СССР: №5).
7. Базара М., Шетти К. Нелинейное программирование. Теория и алгоритмы. - М.: Мир, 1982.
8. Келли Дж.Л. Общая топология. - М.: Наука, 1981.
9. Пападимитриу Х., Стайглиц К. Комбинаторная оптимизация. Алгоритмы и сложность. - М.: Мир, 1985.
10. Шор Н.З. Методы минимизации недифференцируемых функций и их приложения. - Киев: Наукова думка, 1979.
11. Поляк Б.Т. Введение в оптимизацию. - М.: Наука, 1963.
12. Демьянов В.Ф., Васильев Л.В. Недифференцируемая оптимизация. - М.: Наука, 1981.
13. Polak E. On the mathematical foundations of nondifferentiable optimization in engineering design // SIAM Rev. - 1987. - V.29,

№ 1.- P. 21-89.

14. Skelboe S. Computation of rational interval functions // BIT.- 1974.- Vol.14.- P.87-95.
15. Asathambi H.S., Shen Zuhe, Moore R.E. On computing the range of values // Computing.- 1982.- Vol.28, №3. - P.225-237.
16. Ratschek H., Rokne J. Computer methods for the range of functions.- Chichester: Horwood, 1984.
17. Волков Е.А. О поиске максимума функции и приближенном глобальном решении системы нелинейных уравнений // Труды МИАН СССР.- 1974.- Т.131.- С.64-80.

Типография Красноярского Научного Центра СО АН СССР
Заказ № 87. Тираж 180 экз. АЛ 02208 .
Подписано к печати 04.04.89 г.