

образованное всеми решениями точечных систем $Ax = b$ с $A \in \mathbf{A}$ и $b \in \mathbf{b}$. Нас интересует вычисление по возможности более точных внешних покоординатных оценок этого множества решений:

$$\begin{aligned} &\text{Для данной интервальной линейной системы уравнений } \mathbf{Ax} = \mathbf{b} \\ &\text{найти оценки для величин } \min\{x_\nu \mid x \in \Xi_{uni}(\mathbf{A}, \mathbf{b})\} \text{ снизу и} \\ &\text{для величин } \max\{x_\nu \mid x \in \Xi_{uni}(\mathbf{A}, \mathbf{b})\} \text{ сверху, } \nu = 1, 2, \dots, n. \end{aligned} \quad (3)$$

В постановке (3) можно ограничиться требованием вычисления только минимума $\min\{x_\nu \mid x \in \Xi_{uni}(\mathbf{A}, \mathbf{b})\}$, поскольку

$$\max\{x_\nu \mid x \in \Xi_{uni}(\mathbf{A}, \mathbf{b})\} = -\min\{x_\nu \mid x \in \Xi_{uni}(\mathbf{A}, -\mathbf{b})\}$$

для всех $\nu = 1, 2, \dots, n$. Ниже мы для краткости иногда будем называть задачу (3) *внешней задачей* для интервальной линейной системы $\mathbf{Ax} = \mathbf{b}$.

Одним из основных итогов нашей предшествующей статьи [1] является идея алгоритмов, использующих для уточнения оценок множеств решений интервальных систем уравнений адаптивное дробление интервалов их параметров. От красивой идеи путь к эффективным и практичным алгоритмам идет, как правило, через кропотливую работу по модификации исходной схемы и ее “скрещиванию” с более мелкими усовершенствованиями, обеспечивающими успех на реальных вычислительных задачах. Цель этой публикации — представить работу по практическому усовершенствованию методов дробления параметров в применении к интервальным линейным системам уравнений, приведшую к созданию лучших в своем классе алгоритмов для оптимального внешнего оценивания множеств решений ИСЛАУ.

1. Модификации методов дробления параметров

Приступая к построению более совершенных методов дробления параметров для задачи внешнего оценивания объединенного множества решений ИСЛАУ, мы возьмем алгоритм из табл. 2 работы [1] как основу, которая будет развиваться и дополняться рядом усовершенствований, уже стандартных для интервальных методов глобальной оптимизации, базирующихся на стратегии “ветвей и границ”. Как правило, их перечень (не претендующий на полноту) включает в себя следующее (см., например, [2–7]):

1. Строят более качественную внешнюю оценивающую функцию (интервальное расширение) для целевой функции покоординатной оценки.
2. Выявлением монотонности целевой функции по тем или иным переменным на интервалах из рабочего списка \mathcal{L} добиваются уменьшения размерности этих интервалов.
3. На основе специфических локальных свойств целевой функции в соответствующих интервалах применяют более эффективные, чем бисекция, процедуры минимизации (например, методы градиентного спуска там, где целевая функция гладкая и выпуклая).
4. Наряду с оцениванием целевой функции по целым интервалам вычисляют ее значения в каких-то точках этих интервалов, — эти значения доставляют верхнюю

границу искомого глобального минимума, и ее знание позволяет чистить рабочий список \mathfrak{L} от тех записей, которые заведомо не могут быть ведущими.

Необходимость решительной модификации простейшего метода дробления параметров диктуется, в частности, его большой информационной сложностью: фактически каждый шаг алгоритма сопровождается ростом рабочего списка \mathfrak{L} , так что начиная с некоторого момента быстрая оперативная память ЭВМ может исчерпаться, а следующий за этим обмен с внешними носителями может стать фактором, который замедляет на порядки производительность процессора (см. § 2).

1.1. Тест на монотонность

Пусть дана интервальная система линейных алгебраических уравнений $\mathbf{Q}x = \mathbf{r}$ и нам известны

$$\frac{\partial x_\nu(\mathbf{Q}, \mathbf{r})}{\partial q_{ij}} \quad \text{и} \quad \frac{\partial x_\nu(\mathbf{Q}, \mathbf{r})}{\partial r_i}$$

— интервальные расширения соответствующих производных

$$\frac{\partial x_\nu(Q, r)}{\partial q_{ij}} \quad \text{и} \quad \frac{\partial x_\nu(Q, r)}{\partial r_i}$$

от ν -й компоненты вектора решения системы уравнений $Qx = r$ по ij -му элементу матрицы Q и i -му элементу вектора r . Если интервальные $n \times n$ -матрица $\tilde{\mathbf{Q}} = (\tilde{q}_{ij})$ и n -вектор $\tilde{\mathbf{r}} = (\tilde{r}_i)$ образованы из элементов

$$\tilde{q}_{ij} = \begin{cases} [\underline{q}_{ij}, \overline{q}_{ij}] & \text{при} \quad \frac{\partial x_\nu(\mathbf{Q}, \mathbf{r})}{\partial q_{ij}} \geq 0, \\ [\overline{q}_{ij}, \underline{q}_{ij}] & \text{при} \quad \frac{\partial x_\nu(\mathbf{Q}, \mathbf{r})}{\partial q_{ij}} \leq 0, \\ \mathbf{q}_{ij} & \text{при} \quad \text{int} \frac{\partial x_\nu(\mathbf{Q}, \mathbf{r})}{\partial q_{ij}} \ni 0, \end{cases} \quad (4)$$

$$\tilde{r}_i = \begin{cases} [\underline{r}_i, \overline{r}_i] & \text{при} \quad \frac{\partial x_\nu(\mathbf{Q}, \mathbf{r})}{\partial r_i} \geq 0, \\ [\overline{r}_i, \underline{r}_i] & \text{при} \quad \frac{\partial x_\nu(\mathbf{Q}, \mathbf{r})}{\partial r_i} \leq 0, \\ \mathbf{r}_i & \text{при} \quad \text{int} \frac{\partial x_\nu(\mathbf{Q}, \mathbf{r})}{\partial r_i} \ni 0, \end{cases} \quad (5)$$

где “int” обозначает внутренность интервала, то, очевидно,

$$\min\{x_\nu \mid x \in \Xi_{uni}(\tilde{\mathbf{Q}}, \tilde{\mathbf{r}})\} = \min\{x_\nu \mid x \in \Xi_{uni}(\mathbf{Q}, \mathbf{r})\}.$$

А поскольку количество существенно интервальных (с ненулевой шириной) элементов в $\tilde{\mathbf{Q}}$ и $\tilde{\mathbf{r}}$ может быть, вообще говоря, значительно меньшим, чем в \mathbf{Q} и \mathbf{r} , то, переходя от

исходной ИСЛАУ $\mathbf{Q}x = \mathbf{r}$ к системе $\tilde{\mathbf{Q}}x = \tilde{\mathbf{r}}$, мы тем самым добьемся упрощения задачи вычисления $\min\{x_\nu \mid x \in \Xi_{uni}(\mathbf{Q}, \mathbf{r})\}$.

Как найти фигурирующие в (4), (5) интервальные расширения производных? Это можно сделать, например, следующим образом. Из курса математического анализа известно, что $Y = (y_{ij})$ — обратная матрица для $Q = (q_{ij})$, поэтому производные решения вещественной линейной системы $Qx = r$ по ее коэффициентам даются формулами

$$\frac{\partial x_\nu}{\partial q_{ij}} = -y_{\nu i}x_j, \quad \frac{\partial x_\nu}{\partial r_i} = y_{\nu i}$$

(см., к примеру, [8, 9]). Следовательно, в случае, когда $\mathbf{Y} = (\mathbf{y}_{ij})$ — “обратная интервальная матрица” для \mathbf{Q} , т.е. внешняя интервальная оценка для множества обратных матриц

$$\mathbf{Y} \supseteq \{Q^{-1} \mid Q \in \mathbf{Q}\},$$

а \mathbf{x}_j — j -я компонента некоторого интервального вектора $\mathbf{x} \supseteq \Xi_{uni}(\mathbf{Q}, \mathbf{r})$, то мы можем принять следующие интервальные оценки производных:

$$\frac{\partial x_\nu(\mathbf{Q}, \mathbf{r})}{\partial q_{ij}} = -\mathbf{y}_{\nu i}\mathbf{x}_j, \quad \frac{\partial x_\nu(\mathbf{Q}, \mathbf{r})}{\partial r_i} = \mathbf{y}_{\nu i}. \quad (6)$$

Вычисление “обратной интервальной матрицы” можно выполнить как решение задачи внешнего оценивания объединенного множества решений интервального матричного уравнения

$$\mathbf{A}Y = I,$$

где I — единичная матрица, применив n раз тот же самый метод внешнего оценивания *Encl*, который выбран базовым для всего алгоритма. Иногда для вычисления \mathbf{Y} и \mathbf{x} уместно использовать какие-нибудь дешевые приближенные алгоритмы (вроде метода Хансена [10] для локализации “обратной интервальной матрицы”).

1.2. Стратегия дробления

Традиционно в интервальных методах глобальной оптимизации, которые основаны на схеме “ветвей и границ” и являются ближайшими родственниками методов дробления параметров, ведущие интервалы на каждом шаге дробятся по своей самой длинной грани. Этот способ дробления гарантирует, как известно (см. работы [7, 11]), сходимость использующих его алгоритмов, и мы тоже применили его в простейшем методе дробления параметров, хотя для него подобные проблемы не возникают из-за его конечности.

После того как собственно сходимость достигнута, на первый план выходит желание оптимизации метода, т.е. обеспечения наибольшей быстроты сходимости. В свою очередь, эту задачу далее обычно редуцируют к упрощенной постановке — обеспечить наиболее значительное улучшение целевой функции на каждом шаге алгоритма. Строгая и точная оптимизация методов дробления параметров даже в таком виде трудна и вряд ли целесообразна в полном объеме. Мы будем решать эту задачу, руководствуясь разумными эвристическими соображениями, на основе знания оценок производных целевой функции.

Если матрицы $\tilde{Q} = (\tilde{q}_{ij})$ и $\hat{Q} = (\hat{q}_{ij})$ отличаются друг от друга лишь в единственном (k, l) -м элементе, $\tilde{q}_{kl} < \hat{q}_{kl}$, то в силу теоремы Лагранжа о среднем имеем

$$(\hat{Q}^{-1}r)_\nu - (\tilde{Q}^{-1}r)_\nu = \frac{\partial x_\nu(\tilde{Q}, r)}{\partial q_{kl}} \text{wid} [\tilde{q}_{kl}, \hat{q}_{kl}]$$

для некоторой матрицы $\tilde{Q} \in [\check{Q}, \hat{Q}]$. Аналогично, если векторы $\check{r} = (\check{r}_i)$ и $\hat{r} = (\hat{r}_i)$ отличаются только в единственной k -й компоненте и $\check{r}_k < \hat{r}_k$, то

$$(Q^{-1}\hat{r})_\nu - (Q^{-1}\check{r})_\nu = \frac{\partial x_\nu(Q, \tilde{r})}{\partial r_k} \text{wid} [\check{r}_k, \hat{r}_k]$$

для некоторого вектора $\tilde{r} \in [\check{r}, \hat{r}]$.

Предположим теперь, что интервальные матрицы \check{Q} и \hat{Q} получены из интервальной матрицы Q заменой элемента q_{kl} на его левую \underline{q}_{kl} и правую \overline{q}_{kl} границы:

$$\check{q}_{kl} = \underline{q}_{kl}, \quad \hat{q}_{kl} = \overline{q}_{kl}.$$

Допустим также, что

$$\min\{x_\nu \mid x \in \Xi_{uni}(\check{Q}, \mathbf{r})\} \quad \text{и} \quad \min\{x_\nu \mid x \in \Xi_{uni}(\hat{Q}, \mathbf{r})\}$$

достигаются на одном и том же семействе концов интервальных элементов матрицы и вектора правой части, что почти всегда имеет место для “достаточно узких” интервальных систем. Тогда

$$\min\{x_\nu \mid x \in \Xi_{uni}(\hat{Q}, \mathbf{r})\} - \min\{x_\nu \mid x \in \Xi_{uni}(\check{Q}, \mathbf{r})\} = \frac{\partial x_\nu(\hat{Q}, \hat{r})}{\partial q_{kl}} \text{wid} \mathbf{q}_{kl}$$

для некоторой матрицы $\hat{Q} \in Q$ и вектора $\hat{r} \in \mathbf{r}$. Сходным образом предположим, что \check{r} и \hat{r} — это интервальные векторы, полученные из интервального вектора \mathbf{r} заменой его k -й компоненты на левую \underline{r}_k и правую \overline{r}_k границы:

$$\check{r}_k = \underline{r}_k, \quad \hat{r}_k = \overline{r}_k.$$

При прежнем условии, что

$$\min\{x_\nu \mid x \in \Xi_{uni}(Q, \check{r})\}, \quad \min\{x_\nu \mid x \in \Xi_{uni}(Q, \hat{r})\},$$

достигаются на одном и том же множестве концов интервальных элементов матрицы и вектора правой части ИСЛАУ, снова получаем

$$\min\{x_\nu \mid x \in \Xi_{uni}(Q, \hat{r})\} - \min\{x_\nu \mid x \in \Xi_{uni}(Q, \check{r})\} = \frac{\partial x_\nu(Q, \hat{r})}{\partial r_k} \text{wid} \mathbf{r}_k$$

для некоторых матрицы $\hat{Q} \in Q$ и вектора $\hat{r} \in \mathbf{r}$. Следовательно, величина произведения ширины (или радиуса) интервального элемента на модуль интервального расширения соответствующей производной решения может служить в некотором смысле локальной мерой того, как бисекция элементов из Q либо \mathbf{r} влияет на $\min\{x_\nu \mid x \in \Xi_{uni}(Q, \mathbf{r})\}$ и размеры объединенного множества решений ИСЛАУ.

Далее, оценки объединенного множества решений ИСЛАУ, получаемые с помощью большинства существующих методов, являются тем более точными, чем меньше размеры этого множества решений. Д. Гей, например, в [12] доказал для своих методов оценки погрешности, пропорциональную ширине интервальной оболочки множества решений. Аналогичная оценка доказана А. Ноймайером для метода Кравчика [13]. С подобными базовыми методами уменьшение размеров множества решений $\Xi_{uni}(Q, \mathbf{r})$ должно приводить к аналогичному и сравнимому по величине изменению в оценке $\Upsilon(Q, \mathbf{r})$. При этом,

следовательно, требование наиболее быстрого улучшения целевой функции за один шаг метода дробления параметров становится по существу эквивалентным условию наиболее быстрого уменьшения размеров множества решений при дроблении ведущей ИСЛАУ.

Учитывая сделанные выше (частично эвристические) заключения, мы рекомендуем поэтому рассекать ведущие ИСЛАУ по элементам, на которых достигается максимальная из величин

$$\left| \frac{\partial x_\nu(\mathbf{Q}, \mathbf{r})}{\partial q_{ij}} \right| \text{wid } \mathbf{q}_{ij}, \quad \left| \frac{\partial x_\nu(\mathbf{Q}, \mathbf{r})}{\partial r_i} \right| \text{wid } \mathbf{r}_i, \quad (7)$$

$i, j \in \{1, 2, \dots, n\}$, т.е. по элементам, на которых достигается максимум произведения оценки производной решения на ширину интервала, а не просто по самым широким элементам ИСЛАУ.¹

1.3. Модификация Рона

Результат следствия 2 из теоремы 2.3 предшествующей работы [1], который мы использовали с целью вывода метода дробления параметров, для объединенных множеств решений ИСЛАУ был значительно усилен в 80-е годы И. Роном, уточнившим множество вершин матрицы \mathbf{A} и вектора правой части \mathbf{b} , на которых достигаются минимальное и максимальное значения компонент точек из $\Xi_{uni}(\mathbf{A}, \mathbf{b})$ [17].

Чтобы дать математически строгую формулировку результата И. Рона, нам потребуются некоторые дополнительные обозначения. Пусть

$$\mathcal{E} := \{x \in \mathbb{R}^n \mid |x_i| = 1 \text{ для всех } i = 1, 2, \dots, n\}$$

— множество n -векторов с компонентами ± 1 . Для данной интервальной матрицы \mathbf{A} и фиксированных $\sigma, \tau \in \mathcal{E}$ обозначим через $A^{\sigma\tau} = (a_{ij}^{\sigma\tau})$ матрицу размера $n \times n$, образованную элементами

$$a_{ij}^{\sigma\tau} := \begin{cases} \bar{\mathbf{a}}_{ij}, & \text{если } \sigma_i \tau_j = -1, \\ \underline{\mathbf{a}}_{ij}, & \text{если } \sigma_i \tau_j = 1. \end{cases}$$

Кроме того, через $b^\sigma = (b_i^\sigma)$ мы будем обозначать n -вектор, составленный из элементов

$$b_i^\sigma := \begin{cases} \bar{\mathbf{b}}_i, & \text{если } \sigma_i = 1, \\ \underline{\mathbf{b}}_i, & \text{если } \sigma_i = -1. \end{cases}$$

Матрица $A^{\sigma\tau}$ и вектор b^σ образованы, таким образом, наборами концов элементов матрицы \mathbf{A} и вектора \mathbf{b} соответственно, и всего имеются $2^n \cdot 2^n = 4^n$ различных матрично-векторных пар вида $(A^{\sigma\tau}, b^\sigma)$ для σ и τ , независимо пробегающих множество \mathcal{E} .

Оказывается [17], что для невырожденной матрицы \mathbf{A} как минимальное, так и максимальное значения компонент точек множества решений $\Xi_{uni}(\mathbf{A}, \mathbf{b})$ могут достигаться лишь на множестве 4^n матриц $A^{\sigma\tau}$ и ассоциированных с ними векторов b^σ , т.е.

$$\min\{x_\nu \mid x \in \Xi_{uni}(\mathbf{A}, \mathbf{b})\} = \min_{\sigma, \tau \in \mathcal{E}} \left((A^{\sigma\tau})^{-1} b^\sigma \right)_\nu, \quad (8)$$

$$\max\{x_\nu \mid x \in \Xi_{uni}(\mathbf{A}, \mathbf{b})\} = \max_{\sigma, \tau \in \mathcal{E}} \left((A^{\sigma\tau})^{-1} b^\sigma \right)_\nu \quad (9)$$

¹За рубежом имеется тенденция связывать такой способ выбора рассеканной компоненты с именем Д. Ратца, рассматривавшего его в своих работах [14, 15]. Мы не следуем этому, поскольку независимо от Д. Ратца и даже раньше такая стратегия дробления, требующая максимизации величин (7), была предложена автором в статье [16].

для каждого индекса $\nu = 1, 2, \dots, n$. Рассмотрим, как можно использовать этот факт в наших методах дробления параметров.

Важно осознавать, что приведенный выше результат не накладывает никаких ограничений на концы отдельно взятого интервального элемента матрицы \mathbf{A} либо вектора правой части \mathbf{b} , если в рассмотрение не привлечена информация о концах других элементов. Ограничение на ту или иную комбинацию концов интервалов, следующее из (8), (9), является существенно коллективным, и, чтобы принять его во внимание, мы должны отслеживать концы задействованных интервальных элементов по всей матрице \mathbf{A} и всему вектору \mathbf{b} . Для практической реализации этих идей с каждой интервальной линейной системой $\mathbf{Q}x = \mathbf{r}$, $\mathbf{Q} = (\mathbf{q}_{ij})$, $\mathbf{r} = (\mathbf{r}_i)$, порожденной в процессе дробления исходной ИСЛАУ (1), (2), мы связываем:

— вспомогательную $n \times n$ -матрицу $W = (w_{ij})$, элементы которой равны ± 1 или 0, такую что

$$w_{ij} := \begin{cases} -1, & \text{если } \mathbf{q}_{ij} = \bar{\mathbf{a}}_{ij}, \\ 0, & \text{если } \mathbf{q}_{ij} = \mathbf{a}_{ij}, \\ 1, & \text{если } \mathbf{q}_{ij} = \underline{\mathbf{a}}_{ij}; \end{cases}$$

— вспомогательные n -векторы $s = (s_i)$ и $t = (t_j)$ с компонентами ± 1 или 0, такие что

$$w_{ij} = s_i t_j \quad (10)$$

для всех $i, j = 1, 2, \dots, n$, и

$$s_i := \begin{cases} -1, & \text{если } \mathbf{r}_i = \underline{\mathbf{b}}_i, \\ 0, & \text{если } \mathbf{r}_i = \mathbf{b}_i, \\ 1, & \text{если } \mathbf{r}_i = \bar{\mathbf{b}}_i. \end{cases}$$

Кроме того, рабочий список \mathcal{L} будет состоять теперь из расширенных записей

$$\left(\mathbf{Q}, \mathbf{r}, \Upsilon(\mathbf{Q}, \mathbf{r}), W, s, t \right), \quad (11)$$

имеющих *шесть* полей за счет добавления трех полей для хранения W , s и t , которые получаются с предыдущих шагов алгоритма.

Векторам s и t предстоит быть “приближениями” соответственно к σ и τ , входящим в равенства (8), (9), в то время как $W = st^\top$ — это “приближение” к матрице $(\sigma \tau^\top)$. В начале работы алгоритма мы инициализируем все элементы W , s и t нулями, а затем перевычисляем их с целью заменить нулевые элементы (соответствующие неопределенности) на ненулевые (соответствующие тому или иному концу интервала). Матрица W и векторы s, t , взаимно влияя друг на друга и перевычисляясь в процессе работы алгоритма, предназначены, следовательно, для наблюдения над процессом дробления исходной интервальной линейной системы и его корректировки таким образом, чтобы порождать лишь варианты, допускаемые равенствами (8), (9). Это реализуется следующим образом.

На каждом шаге алгоритма при разбиении интервального элемента \mathbf{h} ведущей ИСЛАУ $\mathbf{Q}x = \mathbf{r}$ мы смотрим на соответствующее значение:

— матрицы $W = (w_{ij})$, если элемент \mathbf{h} есть \mathbf{q}_{kl} из матрицы \mathbf{Q} ;

— вектора $s = (s_i)$, если элемент \mathbf{h} есть \mathbf{r}_k из вектора правой части \mathbf{r} .

В случае $w_{kl} = 0$ ($s_k = 0$ соответственно) мы порождаем, согласно обычной процедуре бисекции из простейшего метода дробления параметров табл. 2 работы [1], две интервальных системы-потомка — $\mathbf{Q}'x = \mathbf{r}'$ и $\mathbf{Q}''x = \mathbf{r}''$, которые отвечают обоим концам рассекаемого интервала \mathbf{h} . Но если $w_{kl} \neq 0$ ($s_k \neq 0$ соответственно), то после бисекции ведущей ИСЛАУ мы можем оставить в рабочем списке только *одного* потомка от $\mathbf{Q}x = \mathbf{r}$. Какого именно, зависит от знака элемента w_{kl} (s_k соответственно). Говоря более точно, мы выполняем процедуру, представленную в табл. 1.

Почему это в принципе возможно? Другими словами, не может ли отбрасывание второй интервальной системы-потомка в вышеописанной процедуре нарушить то свойство ведущих оценок $\Upsilon(\mathbf{Q}, \mathbf{r})$, что они приближают искомый $\min\{x_\nu \mid x \in \Xi_{uni}(\mathbf{A}, \mathbf{b})\}$ снизу?

Для ответа на этот вопрос заметим, что в новой процедуре дробления из табл. 1 мы отбрасываем лишь те системы, которые:

- не принадлежат множеству точечных систем $\{A^{\sigma\tau}x = b^\sigma \mid \sigma, \tau \in \mathcal{E}\}$;
- не содержат систем из этого множества.

Т а б л и ц а 1

Порождение систем-потомков

<pre> IF (рассекается \mathbf{q}_{kl}) THEN IF ($w_{kl} = 0$) THEN порождаем системы $\mathbf{Q}'x = \mathbf{r}'$ и $\mathbf{Q}''x = \mathbf{r}''$ так, что $\mathbf{q}'_{ij} := \mathbf{q}''_{ij} := \mathbf{q}_{ij}$ для $(i, j) \neq (k, l)$, $\mathbf{q}'_{kl} := \underline{\mathbf{q}}_{kl}$, $\mathbf{q}''_{kl} := \overline{\mathbf{q}}_{kl}$, $\mathbf{r}' := \mathbf{r}'' := \mathbf{r}$; ELSE порождаем систему $\mathbf{Q}'x = \mathbf{r}'$ так, что $\mathbf{r}' := \mathbf{r}$, $\mathbf{q}'_{ij} := \mathbf{q}_{ij}$ для $(i, j) \neq (k, l)$, $\mathbf{q}'_{kl} := \begin{cases} \underline{\mathbf{q}}_{kl} & \text{для } w_{kl} = 1, \\ \overline{\mathbf{q}}_{kl} & \text{для } w_{kl} = -1; \end{cases}$ END IF END IF IF (рассекается \mathbf{r}_k) THEN IF ($s_k = 0$) THEN порождаем системы $\mathbf{Q}'x = \mathbf{r}'$ и $\mathbf{Q}''x = \mathbf{r}''$ так, что $\mathbf{Q}' := \mathbf{Q}'' := \mathbf{Q}$, $\mathbf{r}'_i := \mathbf{r}_i$ для $i \neq k$, $\mathbf{r}'_k := \underline{\mathbf{r}}_k$, $\mathbf{r}''_k := \overline{\mathbf{r}}_k$, ELSE порождаем систему $\mathbf{Q}'x = \mathbf{r}'$ так, что $\mathbf{Q}' := \mathbf{Q}$, $\mathbf{r}'_i := \mathbf{r}_i$ для $i \neq k$, $\mathbf{r}'_k := \begin{cases} \underline{\mathbf{r}}_k & \text{для } s_k = -1, \\ \overline{\mathbf{r}}_k & \text{для } s_k = 1; \end{cases}$ END IF END IF </pre>

Следовательно, в силу свойства (C1) базисного оценивающего метода и, принимая во внимание равенство (8), мы получаем

$$\begin{aligned}
\min\{x_\nu \mid x \in \Xi_{uni}(\mathbf{A}, \mathbf{b})\} &= \min_{\sigma, \tau \in \mathcal{E}} \left((A^{\sigma\tau})^{-1} b^\sigma \right)_\nu \geq \min_{\sigma, \tau \in \mathcal{E}} \Upsilon(A^{\sigma\tau}, b^\sigma) \geq \\
&\geq \min\{\Upsilon(\mathbf{Q}, \mathbf{r}) \mid \mathbf{Q} \ni A^{\sigma\tau} \text{ и } \mathbf{r} \ni b^\sigma \text{ для некоторых } \sigma, \tau \in \mathcal{E}\} \geq \\
&\geq \min\{\Upsilon(\mathbf{Q}, \mathbf{r}) \mid \text{система } \mathbf{Q}x = \mathbf{r} \text{ присутствует в списке } \mathcal{L}\} = \\
&= \text{ведущая оценка } \Upsilon(\mathbf{Q}, \mathbf{r}).
\end{aligned}$$

Таким образом, с новой процедурой дробления ведущая оценка действительно приближает $\min\{x_\nu \mid x \in \Xi_{uni}(\mathbf{A}, \mathbf{b})\}$ снизу.

Обращение к описанию формальной вычислительной схемы “модификации Рона” мы начнем с установления правил перевычисления W , s и t в процессе работы алгоритма. Существует взаимно-однозначное соответствие между вектором s и правой частью интервальной системы $\mathbf{Q}x = \mathbf{r}$, в то время как дробление интервальной матрицы \mathbf{Q} рассматриваемой ИСЛАУ влияет на векторы s и t лишь неявно — посредством матрицы W и условий (10). Последнее тем не менее позволяет организовать перевычисление W , s и t на каждом таком шаге алгоритма, который завершается дроблением интервального элемента ведущей системы *на два конца*. В противном случае, если ведущая интервальная система порождает только одного потомка $\mathbf{Q}'x = \mathbf{r}'$ в соответствии с табл. 1, векторы s , t и матрица W остаются неизменными, так что

$$s' := s, \quad t' := t, \quad W' := W.$$

Итак, пусть ведущая интервальная система $\mathbf{Q}x = \mathbf{r}$ рассечена на два ИСЛАУ-потомка — $\mathbf{Q}'x = \mathbf{r}'$ и $\mathbf{Q}''x = \mathbf{r}''$, определенные в табл. 1. Каким должен быть закон, в соответствии с которым формируются матрицы W' , W'' и векторы s' , s'' , t' , t'' , соответствующие системам-потомкам? Сначала мы можем положить

$$W'' := W' := W, \quad s'' := s' := s, \quad t'' := t' := t,$$

а затем выполнить следующую двухэтапную процедуру перевычисления:

Во-первых, модифицируем W' , W'' и s' , s'' на основе информации о только что выполненной бисекции ведущей ИСЛАУ. Именно:

- (I) если рассеченным элементом был \mathbf{q}_{kl} из матрицы \mathbf{Q} , то в матрицах $W' = (w'_{ij})$ и $W'' = (w''_{ij})$ мы полагаем

$$w'_{kl} := 1 \quad \text{и} \quad w''_{kl} := -1;$$

- (II) если рассеченным элементом был \mathbf{r}_k из вектора правой части \mathbf{r} , то в векторах $s' = (s'_i)$ и $s'' = (s''_i)$ мы полагаем

$$s'_k := -1 \quad \text{и} \quad s''_k := 1.$$

Во-вторых, перевычисляем каждое из двух семейств взаимосвязанных объектов, — W' , s' , t' и W'' , s'' , t'' соответственно, — используя соотношения (10), а именно:

- (I) если s' или t' изменился, пытаемся перевычислить матрицу W' ;
- (II) если W' или t' изменился, пытаемся перевычислить вектор s' ;
- (III) если W' или s' изменился, пытаемся перевычислить вектор t' .

Инструкции (I)–(III) повторяются последовательно одна за другой в цикле до тех пор, пока изменения в W' , s' и t' не прекратятся. Та же самая процедура применяется затем к W'' , s'' , t'' .

Полная алгоритмическая схема приведенной выше процедуры оказывается довольно сложной, так что имеет смысл снабдить читателя более строгим и детальным ее описанием. Соответствующий псевдокод представляет табл. 2. Ниже мы даем объяснения к нему. Булевские переменные

$$W'Change, \quad s'Change, \quad t'Change, \quad W'C, \quad s'C, \quad t'C$$

и

$$W''Change, \quad s''Change, \quad t''Change, \quad W''C, \quad s''C, \quad t''C$$

— это “флаги”, введенные с целью отображения текущего статуса изменений в W' , s' , t' и W'' , s'' , t'' соответственно. Значение **true** показывает, что объект, с которым связан флаг, был изменен на текущей итерации процесса перевычисления, тогда как значение **false** — “без изменений”.

Для того чтобы завершить формальное описание модификации Рона, нам нужно лишь определить, что имеется в виду под “пытаемся перевычислить матрицу W' ”, “пытаемся перевычислить вектор s' ” и тому подобными инструкциями из табл. 2.

Обозначим прописными греческими буквами K' , Λ' и Ω' подмножества индексов элементов вектора s' , вектора t' и матрицы W' соответственно, которые изменили свои значения (с 0 на ± 1) на текущем шаге процедуры перевычисления табл. 2. K' и Λ' — это, следовательно, подмножества множества первых n натуральных чисел

$$\{1, 2, \dots, n\},$$

тогда как Ω есть подмножество множества всех пар, составленных из первых n натуральных чисел, т. е. подмножество множества

$$\{(1, 1), (1, 2), \dots, (1, n), (2, 1), \dots, (n, n)\}.$$

Каждое из множеств K' , Λ' , Ω' может быть пустым, но может содержать и более чем один элемент. Тогда наша “попытка перевычислить вектор s' ” может быть организована следующим образом (табл. 3).

“Попытка перевычислить вектор t' ” может быть выполнена аналогично тому, как это произведено выше с тем единственным отличием, что цикл “DO FOR $l \in \Lambda'$ ” во втором IF-операторе должен быть заменен на “DO FOR $k \in K'$ ”. Далее приводится процедура перевычисления W' (табл. 4).

То же следует произвести для перевычисления s'' , t'' и W'' . При этом нам необходимо организовать индексные подмножества K'' , Λ'' и Ω'' для представления индексов компонент векторов s'' , t'' и матрицы W'' соответственно, которые изменились на текущем шаге перевычислительного процесса.

Т а б л и ц а 2

Перевычисление W' , W'' , s' , s'' , t' , t'' .

```

W'Change := false;   s'Change := false;   t'Change := false;
W''Change := false;  s''Change := false;  t''Change := false;
IF ( ( пересекается  $q_{kl}$  из  $Q$  ) AND (  $q_{kl}$  пересекается на два конца ) ) THEN
     $m'_{kl} := 1$ ;  $m''_{kl} := -1$ ;  $W'Change := true$ ;  $W''Change := true$ ;
END IF
IF ( ( пересекается  $r_k$  из  $r$  ) AND (  $r_k$  пересекается на два конца ) ) THEN
     $s'_k := -1$ ;  $s''_k := 1$ ;  $s'Change := true$ ;  $s''Change := true$ ;
END IF
DO WHILE ( W'Change OR s'Change OR t'Change )
    IF ( s'Change OR t'Change ) THEN
        пытаемся перевычислить матрицу  $W'$ ;
        IF (  $W'$  изменилась ) THEN  $W'C := true$ 
        ELSE  $W'C := false$  END IF
    END IF
    IF ( W'Change OR t'Change ) THEN
        пытаемся перевычислить вектор  $s'$ ;
        IF (  $s'$  изменился ) THEN  $s'C := true$ 
        ELSE  $s'C := false$  END IF
    END IF
    IF ( W'Change OR s'Change ) THEN
        пытаемся перевычислить вектор  $t'$ ;
        IF (  $t'$  изменился ) THEN  $t'C := true$ 
        ELSE  $t'C := false$  END IF
    END IF
    W'Change := W'C; s'Change := s'C; t'Change := t'C;
END DO
DO WHILE ( W''Change OR s''Change OR t''Change )
    IF ( s''Change OR t''Change ) THEN
        пытаемся перевычислить матрицу  $W''$ ;
        IF (  $W''$  изменилась ) THEN  $W''C := true$ 
        ELSE  $W''C := false$  END IF
    END IF
    IF ( W''Change OR t''Change ) THEN
        пытаемся перевычислить вектор  $s''$ ;
        IF (  $s''$  изменился ) THEN  $s''C := true$ 
        ELSE  $s''C := false$  END IF
    END IF
    IF ( W''Change OR s''Change ) THEN
        пытаемся перевычислить вектор  $t''$ ;
        IF (  $t''$  изменился ) THEN  $t''C := true$ 
        ELSE  $t''C := false$  END IF
    END IF
    W''Change := W''C; s''Change := s''C; t''Change := t''C;
END DO

```

Т а б л и ц а 3

Перевычисление s'

```

IF (  $W'Change$  ) THEN
  DO FOR  $(k, l) \in \Omega'$ 
    IF (  $t'_l \neq 0$  )  $s'_k := w'_{kl}/t'_l$ 
  END DO
END IF
IF (  $t'Change$  ) THEN
  DO FOR  $l \in \Lambda'$ 
    DO FOR  $k = 1$  TO  $n$ 
      IF (  $s_k = 0$  AND  $w'_{kl} \neq 0$  )  $s'_k := w'_{kl}/t'_l$ 
    END DO
  END DO
END IF

```

Т а б л и ц а 4

Перевычисление W'

```

IF (  $s'Change$  ) THEN
  DO FOR  $k \in K'$ 
    DO FOR  $l = 1$  TO  $n$ 
      IF (  $t'_l \neq 0$  )  $w'_{kl} := s'_k t'_l$ 
    END DO
  END DO
END IF
IF (  $t'Change$  ) THEN
  DO FOR  $l \in \Lambda'$ 
    DO FOR  $k = 1$  TO  $n$ 
      IF (  $s'_k \neq 0$  )  $w'_{kl} := s'_k t'_l$ 
    END DO
  END DO
END IF

```

Наконец, мы должны упомянуть следующее замечательное свойство матрицы W : в каждой ее 2×2 -подматрице любой элемент равен произведению остальных трех элементов. Чтобы убедиться в этом, обозначим через i_1, i_2 номера строк и через j_1, j_2 — номера столбцов, образующих рассматриваемую подматрицу. Тогда в соответствии с определением (10)

$$w_{i_1 j_1} = \sigma_{i_1} \tau_{j_1}, \quad w_{i_1 j_2} = \sigma_{i_1} \tau_{j_2}, \quad w_{i_2 j_1} = \sigma_{i_2} \tau_{j_1}, \quad w_{i_2 j_2} = \sigma_{i_2} \tau_{j_2}.$$

Перемножая любые три из вышеприведенных равенств, к примеру, первое, второе и четвертое, получим

$$w_{i_1 j_1} w_{i_1 j_2} w_{i_2 j_2} = \sigma_{i_1} \tau_{j_1} \sigma_{i_1} \tau_{j_2} \sigma_{i_2} \tau_{j_2}.$$

Квадрат любой компоненты векторов σ и τ есть 1, так что

$$w_{i_1 j_1} w_{i_1 j_2} w_{i_2 j_2} = \tau_{j_1} \sigma_{i_2} = w_{i_2 j_1}. \quad (12)$$

Т а б л и ц а 5

Уточнение W посредством поиска 2×2 -подматриц

```

DO FOR  $i = 1$  TO  $n$ 
  DO FOR  $j = 1$  TO  $n$ 
    IF (  $i \neq k$  AND  $j \neq l$  ) THEN
      IF (  $w_{ij} \neq 0$  AND  $w_{il} \neq 0$  AND  $w_{kj} \neq 0$  ) THEN
         $w_{kl} := w_{ij}w_{il}w_{kj}$ ;
        EXIT
      END IF
    END IF
  END DO
END DO

```

То же самое с остальными элементами подматрицы:

$$w_{i_1 j_1} w_{i_1 j_2} w_{i_2 j_1} = w_{i_2 j_2}, \quad (13)$$

$$w_{i_1 j_2} w_{i_2 j_1} w_{i_2 j_2} = w_{i_1 j_1}, \quad (14)$$

$$w_{i_1 j_1} w_{i_2 j_1} w_{i_2 j_2} = w_{i_1 j_2}. \quad (15)$$

Иногда соотношения (12)–(15) могут оказаться полезными для дальнейшего уточнения матрицы W . Например, пусть мы намереваемся рассечь ведущую интервальную систему $\mathbf{Q}x = \mathbf{r}$ по элементу \mathbf{q}_{kl} , в то время как соответствующий элемент w_{kl} матрицы W равен нулю. При этом согласно обычному правилу дробления мы должны были бы породить две системы-потомка вместо ИСЛАУ $\mathbf{Q}x = \mathbf{r}$. Но разумно сделать попытку доопределить w_{kl} , поискав 2×2 -подматрицы в W , имеющие ненулевыми все элементы, за исключением w_{kl} . Если такая подматрица в W найдется, то мы присваиваем элементу w_{kl} значение произведения остальных трех элементов. Сказанное может быть реализовано в виде следующего алгоритма из табл. 5, где оператор EXIT означает выход из всех блоков и циклов выписанного псевдокода.

1.4. Отсев бесперспективных записей

Рассмотрим, наконец, модификацию простейшего метода дробления параметров, связанную с вычислением оценок $\Upsilon(\text{mid } \mathbf{Q}, \text{mid } \mathbf{r})$ для “середин” ведущих систем. Она позволит контролировать точность вычисления значения $\min\{x_\nu \mid x \in \Xi_{uni}(\mathbf{A}, \mathbf{b})\}$, а также удалять из списка \mathcal{L} бесперспективные записи, т. е. такие, которые никогда не сделаются ведущими. Благодаря последнему обстоятельству несколько ограничивается рост длины рабочего списка \mathcal{L} .

Действительно, пусть наряду с оценкой $\Upsilon(\mathbf{Q}, \mathbf{r})$ для интервальных систем $\mathbf{Q}x = \mathbf{r}$, порождаемых алгоритмом, вычисляются еще и величины $\Upsilon(\square\mathbf{Q}, \square\mathbf{r})$, где символом “ \square ” обозначена операция взятия какой-то фиксированной точки из интервала. Очевидно, что

$$\Upsilon(\square\mathbf{Q}, \square\mathbf{r}) \geq \Upsilon(\mathbf{Q}, \mathbf{r}),$$

и значения $\Upsilon(\square\mathbf{Q}, \square\mathbf{r})$ приближают $\min\{x \mid x \in \Xi_{uni}(\mathbf{A}, \mathbf{b})\}$ сверху. Если для каждого шага алгоритма определять величину

$$\omega = \min \Upsilon(\square\mathbf{Q}, \square\mathbf{r}) \quad (16)$$

по всем интервальным линейным системам $\mathbf{Q}x = \mathbf{r}$, когда-либо побывавшим в списке \mathcal{L} до этого шага, то

$$\min\{x_\nu \mid x \in \Xi_{uni}(\mathbf{A}, \mathbf{b})\} \leq \omega.$$

С другой стороны, если $\mathbf{Q}x = \mathbf{r}$ — ведущая ИСЛАУ, то

$$\Upsilon(\mathbf{Q}, \mathbf{r}) \leq \min\{x_\nu \mid x \in \Xi_{uni}(\mathbf{A}, \mathbf{b})\},$$

и потому еще одним критерием естественной остановки алгоритма может служить достижение требуемой малости разности $(\omega - \Upsilon(\mathbf{Q}, \mathbf{r}))$ для ведущей оценки $\Upsilon(\mathbf{Q}, \mathbf{r})$.

Далее, интервальная система $\mathbf{Q}x = \mathbf{r}$, удовлетворяющая на некотором шаге условию

$$\Upsilon(\mathbf{Q}, \mathbf{r}) > \omega, \tag{17}$$

при дальнейшем выполнении алгоритма уже никогда не станет ведущей, и удаление соответствующей записи из рабочего списка \mathcal{L} не окажет никакого влияния на результат работы алгоритма. Вообще говоря, тестироваться неравенством (12) должны на каждом шаге алгоритма все вновь заносимые в список \mathcal{L} записи, тогда как “чистки” списка — его просмотр и удаление всех удовлетворяющих (12) записей — имеет смысл проводить лишь после изменений (т. е. уменьшений) величины ω .

Конечно, было бы идеальным выбирать $(\square\mathbf{Q}, \square\mathbf{r}) \in \text{Arg min}\{\Upsilon(Q, r) \mid Q \in \mathbf{Q}, r \in \mathbf{r}\}$. Фактически мы так и будем поступать, если оценка $\Upsilon(\mathbf{Q}, \mathbf{r})$ — точная, полагая

$$\Upsilon(\square\mathbf{Q}, \square\mathbf{r}) = \Upsilon(\mathbf{Q}, \mathbf{r}).$$

Но в общем случае подобный “удачный” выбор не менее труден, чем решение исходной задачи, и потому в целях минимизации возможных отклонений $(\square\mathbf{Q}, \square\mathbf{r})$ от множества $\text{Arg min}\{\Upsilon(Q, r) \mid Q \in \mathbf{Q}, r \in \mathbf{r}\}$ целесообразней всего брать тогда в качестве $\square\mathbf{Q}$ и $\square\mathbf{r}$ середины матрицы \mathbf{Q} и вектора \mathbf{r} , т. е. $\text{mid } \mathbf{Q}$ и $\text{mid } \mathbf{r}$ соответственно.

1.5. Влияние базового алгоритма

Обсудим теперь очень важный вопрос о влиянии базового метода *Encli*, следовательно, способа получения оценки $\Upsilon(\mathbf{Q}, \mathbf{r})$ на вычислительную схему конкретных реализаций методов дробления параметров.

Для работы многих методов внешнего оценивания объединенного множества решений ИСЛАУ требуются некоторые начальные приближения, содержащие оцениваемое множество решений. Таковы, например, интервальный метод Гаусса — Зейделя, методы Кравчика, Гея и др. Нетрудно понять, что интервальный вектор внешней оценки объединенного множества решений ИСЛАУ $\mathbf{Q}x = \mathbf{r}$ может служить начальным приближением для процедуры внешнего оценивания объединенных множеств решений систем-потомков $\mathbf{Q}'x = \mathbf{r}'$ с $\mathbf{Q}' \subseteq \mathbf{Q}$ и $\mathbf{r}' \subseteq \mathbf{r}$. То же самое относится и к вычислению “обратной интервальной матрицы”, которая нужна в тесте на монотонность из § 1.1 и при выборе рассекаемого интервального элемента из § 1.2.

Имеет смысл поэтому при выборе соответствующих базовых методов хранить интервальные векторы внешней оценки множеств решений и “обратную интервальную матрицу” с предыдущих шагов алгоритма. Для этого потребуется расширить образующие рабочий список \mathcal{L} записи еще на два поля, так что теперь в методе с модификацией Рона мы будем оперировать не тройками, как в простейшем методе дробления параметров, или шестерками (11), а восьмерками

$$(\mathbf{Q}, \mathbf{r}, \Upsilon(\mathbf{Q}, \mathbf{r}), W, s, t, \mathbf{Y}, \mathbf{x}), \tag{18}$$

т. е. записями, состоящими из восьми членов, где

\mathbf{Q} — интервальная $n \times n$ -матрица, $\mathbf{Q} \subseteq \mathbf{A}$;

\mathbf{r} — интервальный n -вектор, $\mathbf{r} \subseteq \mathbf{b}$;

$\Upsilon(\mathbf{Q}, \mathbf{r})$ — нижний конец ν -й компоненты (для заданного фиксированного номера ν) внешней интервальной оценки множества решений $\Xi_{uni}(\mathbf{Q}, \mathbf{r})$;

W, s, t — вспомогательные величины, необходимые для реализации модификации Рона и определенные в § 1.3;

\mathbf{Y} — интервальная $n \times n$ -матрица, такая что $\mathbf{Y} \supseteq \{Q^{-1} \mid Q \in \mathbf{Q}\}$;

\mathbf{x} — интервальный n -вектор, такой что $\mathbf{x} \supseteq \Xi_{uni}(\mathbf{Q}, \mathbf{r})$.

Другое соображение. Как правило, любая из методик решения внешней задачи для ИСЛАУ, удовлетворяющая условию (С2) из [1], дает точную оценку $\Upsilon(\mathbf{Q}, \mathbf{r})$ не только тогда, когда \mathbf{Q} и \mathbf{r} — точечные (неинтервальные), но и на некотором более широком множестве начальных данных, когда часть элементов в \mathbf{Q} и \mathbf{r} могут быть интервалами. Например, интервальный метод Гаусса — Зейделя [6, 13], методы Гея [12] и другие итерационные методы, основанные на теореме о сжимающем отображении [8], обеспечивают точную оценку $\Upsilon(\mathbf{Q}, \mathbf{r})$ в случае вещественной матрицы \mathbf{Q} , а вектор правых частей \mathbf{r} может быть при этом любым. Получаемая по интервальному методу Гаусса оценка $\Upsilon(\mathbf{Q}, \mathbf{r})$, очевидно, точна для верхних треугольных матриц \mathbf{Q} и т. д. Возможны и более хитроумные условия на взаимное расположение элементов интервальной матрицы \mathbf{Q} , их знак, ширину и т. п. Как правило, выявление всех практически наиболее значимых подобных ситуаций нетрудно алгоритмизовать.

Таким образом, для естественной остановки метода дробления параметров совсем не обязательно дожидаться полного “овеществления” ведущей ИСЛАУ $\mathbf{Q}x = \mathbf{r}$ (условие завершения цикла “DO WHILE” в табл. 2 работы [1]). Достаточно того, чтобы ведущая оценка $\Upsilon(\mathbf{Q}, \mathbf{r})$ являлась точной.

Можно пойти еще дальше и предусмотреть возможность динамической смены в алгоритме базового метода *Encl*. Первоначально это может быть какой-нибудь метод с широкой сферой применимости, но и с большой погрешностью внешнего оценивания. Затем, по мере достижения в процессе дробления заданной узости интервальных систем-потомков, он заменяется на более точный специализированный алгоритм.

Вывод, вытекающий из вышеизложенного, прежний: для достижения наибольшей эффективности методов дробления параметров все составляющие реального алгоритма, применяемого к той или иной конкретной задаче, как-то: структура данных (т. е. записей в рабочем списке \mathfrak{L}); способ их обработки; способ дробления ведущих ИСЛАУ; другие характеристики алгоритма — должны быть тщательно увязаны с особенностями решаемой задачи. Общая схема методов дробления параметров предоставляет большую свободу разработчику, но нужно умело ей пользоваться.

1.6. Итоговая схема алгоритма

Приводимые ниже псевдокоды в табл. 6 и 7 суммируют развитые в предшествующих пунктах работы модификации методов дробления параметров (PPS-методов) для внешнего оценивания объединенного множества решений ИСЛАУ.

Теоретически модификация Рона позволяет уменьшить экспоненциальный множитель в верхней оценке трудоемкости PPS-методов с 2^{n^2+n} до 4^n , но необходимо отметить следующее:

— это сделано ценой существенного утяжеления алгоритма, так что его программная сложность становится весьма высокой;

Т а б л и ц а 6

Алгоритм LinPPS1

DO WHILE ((ведущая оценка $\Upsilon(\mathbf{Q}, \mathbf{r})$ неточна) ИЛИ ($\omega - \Upsilon(\mathbf{Q}, \mathbf{r}) > \epsilon$))

по формулам (6) вычисляем интервальные расширения для производных

$$\frac{\partial x_\nu(\mathbf{Q}, \mathbf{r})}{\partial q_{ij}} \quad \text{и} \quad \frac{\partial x_\nu(\mathbf{Q}, \mathbf{r})}{\partial r_i},$$

соответствующие элементам \mathbf{q}_{ij} и \mathbf{r}_i с ненулевой шириной;

“сжимаем” в соответствии с (4), (5) интервальные элементы в \mathbf{Q} и \mathbf{r} , на которых была выявлена монотонная зависимость x_ν от q_{ij} и r_i , и полученные в результате этой процедуры интервальную матрицу и интервальный вектор также обозначаем через \mathbf{Q} и \mathbf{r} ;

ищем среди элементов ведущей ИСЛАУ $\mathbf{Q}\mathbf{x} = \mathbf{r}$ интервальный элемент \mathbf{s} , которому соответствует наибольшее из произведений

$$\left| \frac{\partial x_\nu(\mathbf{Q}, \mathbf{r})}{\partial q_{ij}} \right| \cdot \text{wid } \mathbf{q}_{ij}, \quad \left| \frac{\partial x_\nu(\mathbf{Q}, \mathbf{r})}{\partial r_i} \right| \cdot \text{wid } \mathbf{r}_i, \quad i, j \in \{1, 2, \dots, n\};$$

порождаем интервальные системы-потомки $\mathbf{Q}'\mathbf{x} = \mathbf{r}'$ и $\mathbf{Q}''\mathbf{x} = \mathbf{r}''$:

если $\mathbf{s} = \mathbf{q}_{kl}$ для некоторых $k, l \in \{1, 2, \dots, n\}$, то полагаем

$$\mathbf{q}'_{ij} := \mathbf{q}''_{ij} := \mathbf{q}_{ij} \text{ для } (i, j) \neq (k, l), \quad \mathbf{q}'_{kl} := \underline{\mathbf{q}}_{kl}, \quad \mathbf{q}''_{kl} := \overline{\mathbf{q}}_{kl},$$

$$\mathbf{r}' := \mathbf{r}'' := \mathbf{r};$$

если $\mathbf{s} = \mathbf{r}_k$ для некоторого $k \in \{1, 2, \dots, n\}$, то полагаем

$$\mathbf{r}'_i := \mathbf{r}''_i := \mathbf{r}_i \text{ для } i \neq k, \quad \mathbf{r}'_k := \underline{\mathbf{r}}_k, \quad \mathbf{r}''_k := \overline{\mathbf{r}}_k,$$

$$\mathbf{Q}' := \mathbf{Q}'' := \mathbf{Q};$$

вычисляем интервальные векторы $\mathbf{x}' = \text{Encl}(\mathbf{Q}', \mathbf{r}')$ и $\mathbf{x}'' = \text{Encl}(\mathbf{Q}'', \mathbf{r}'')$;

вычисляем оценки $\Upsilon(\mathbf{Q}', \mathbf{r}')$ и $\Upsilon(\mathbf{Q}'', \mathbf{r}'')$;

вычисляем внешние оценки для “обратных интервальных матриц”

$$\mathbf{Y}' \supseteq (\mathbf{Q}')^{-1} \text{ и } \mathbf{Y}'' \supseteq (\mathbf{Q}'')^{-1};$$

вычисляем оценки $\Upsilon(\text{mid } \mathbf{Q}', \text{mid } \mathbf{r}')$ и $\Upsilon(\text{mid } \mathbf{Q}'', \text{mid } \mathbf{r}'')$ и полагаем

$$\mu := \min\{ \Upsilon(\text{mid } \mathbf{Q}', \text{mid } \mathbf{r}'), \Upsilon(\text{mid } \mathbf{Q}'', \text{mid } \mathbf{r}'') \};$$

удаляем из \mathcal{L} бывшую ведущую запись $(\mathbf{Q}, \mathbf{r}, \Upsilon(\mathbf{Q}, \mathbf{r}), p)$;

если $\Upsilon(\mathbf{Q}', \mathbf{r}') \leq \omega$, то заносим запись $(\mathbf{Q}', \mathbf{r}', \Upsilon(\mathbf{Q}', \mathbf{r}'), \mathbf{Y}', \mathbf{x}')$ в список \mathcal{L} в порядке возрастания значений третьего поля;

если $\Upsilon(\mathbf{Q}'', \mathbf{r}'') \leq \omega$, то заносим запись $(\mathbf{Q}'', \mathbf{r}'', \Upsilon(\mathbf{Q}'', \mathbf{r}''), \mathbf{Y}'', \mathbf{x}'')$ в список \mathcal{L} в порядке возрастания значений третьего поля;

если $\omega > \mu$, то полагаем $\omega := \mu$ и чистим список \mathcal{L} :

удаляем из него все такие записи $(\mathbf{Q}, \mathbf{r}, \Upsilon(\mathbf{Q}, \mathbf{r}), \mathbf{Y}, \mathbf{x})$, что $\Upsilon(\mathbf{Q}, \mathbf{r}) > \omega$;

END DO

— при решении больших практических задач размерности, превосходящей несколько десятков, как 2^{n^2+n} , так и 4^n делаются недостижимыми величинами, а методы дробления параметров следует рассматривать скорее как итерационную уточняющую процедуру, которая никогда не выполняется до своего естественного завершения.

На наш взгляд, в каждом конкретном случае пользователь, основываясь на информации о размере задачи, ее структуре, наличных вычислительных ресурсах, самостоятельно должен решать вопрос о целесообразности включения модификации Рона в ту или иную реализацию метода дробления параметров. По этой причине мы приводим ниже две итоговые вычислительные схемы методов дробления параметров: с модификацией Рона и без нее.

В табл. 6 и 7 предполагается, что в качестве базового метода внешнего оценивания *Encl* взят интервальный метод Гаусса — Зейделя, но это сделано лишь для определенности. Подчеркнем, что методы дробления параметров — это общая схема, а табл. 6 и 7 представляют лишь два ее возможных варианта. Развитие нами выше конструкции содержат ряд “свободных переменных”, которые должны быть настраиваемы под конкретную задачу, так что мы можем, следовательно, говорить о целом *классе методов*, основанных на общей идее дробления интервальных параметров системы.

Алгоритм, приведенный в табл. 6, описывает метод дробления параметров без модификации Рона. Его рабочий список \mathcal{L} образуется пятичленными записями вида

$$(\mathbf{Q}, \mathbf{r}, \Upsilon(\mathbf{Q}, \mathbf{r}), \mathbf{Y}, \mathbf{x}).$$

Для начала работы этого алгоритма, который мы называем *LinPPS1*, нам нужно:

- найти грубые внешние оценки для объединенного множества решений исходной интервальной системы и “обратной интервальной матрицы”, т. е. вычислить $\mathbf{x} \supseteq \Xi_{uni}(\mathbf{A}, \mathbf{b})$ и $\mathbf{Y} \supseteq \mathbf{A}^{-1}$;
- назначить точность $\epsilon > 0$;
- положить $\Upsilon(\mathbf{A}, \mathbf{b}) := \underline{\mathbf{x}}$ и $\omega := +\infty$;
- инициализировать рабочий список \mathcal{L} записью $(\mathbf{A}, \mathbf{b}, \underline{\mathbf{x}}, \mathbf{Y}, \mathbf{x})$.

Алгоритм, приведенный в табл. 7, названный нами *LinPPS2*, представляет метод дробления параметров с модификацией Рона, который оперирует с восьмичленными записями вида (18). Для начала его работы необходимо выполнить первые три пункта, как для алгоритма *LinPPS1*, далее положить

$$W := 0, \quad s := 0, \quad t := 0$$

для матрицы W и векторов s и t , введенных в § 2.3, и инициализировать рабочий список \mathcal{L} записью $(\mathbf{A}, \mathbf{b}, \underline{\mathbf{x}}, W, s, t, \mathbf{Y}, \mathbf{x})$.

В заключение полезно проследить, в каких отношениях построенные нами алгоритмы находятся к другим методам решения ИСЛАУ. Прежде всего, отметим, что имеется глубокая связь (фактически являющаяся двойственностью) между развиваемыми в настоящей работе методами дробления параметров и предложенным автором ранее классом методов дробления решений [11, 18]. Но к идее построения методов дробления параметров для интервальных линейных систем можно прийти и несколько иным путем, доводя до логического завершения некоторые из давно и широко известных в интервальном анализе подходов к решению ИСЛАУ.

Одной из прародительниц алгоритма *LinPPS1* можно, по-видимому, считать процедуру Купермана — Хансена (см. [8]), в которой внешние интервальные оценки для объединенного множества решений ИСЛАУ ищутся на основе пассивного однократного использования информации о производных компонент решения по элементам матрицы и правой части системы, подобно тому как это делается в § 1.1. Но в реальных интервальных системах не всегда удается выявить определенный знак этих производных, а потому какие-то интервальные элементы все равно останутся интервальными и после применения теста на монотонность. И Куперман, и Хансен в этом месте останавливались и завершали процесс решения “внешней задачи”. Дальнейшее уточнение

Т а б л и ц а 7

Алгоритм LinPPS2

DO WHILE ((ведущая оценка $\Upsilon(\mathbf{Q}, \mathbf{r})$ неточна) OR ($\omega - \Upsilon(\mathbf{Q}, \mathbf{r}) > \epsilon$))

используя формулы (6), вычисляем интервальные расширения производных

$$\frac{\partial x_\nu(\mathbf{Q}, \mathbf{r})}{\partial q_{ij}} \quad \text{и} \quad \frac{\partial x_\nu(\mathbf{Q}, \mathbf{r})}{\partial r_i},$$

которые соответствуют интервальным элементам \mathbf{q}_{ij} и \mathbf{r}_i с ненулевой шириной;

“сжимаем” в соответствии с (4), (5) в \mathbf{Q} и \mathbf{r} интервальные элементы, для которых выделена монотонность зависимости x_ν от q_{ij} и r_i , и обозначаем полученные при этом интервальные матрицу и вектор снова через \mathbf{Q} и \mathbf{r} ;

находим среди элементов системы $\mathbf{Q}\mathbf{x} = \mathbf{r}$ интервал \mathbf{h} , который соответствует наибольшему из произведений

$$\left| \frac{\partial x_\nu(\mathbf{Q}, \mathbf{r})}{\partial q_{ij}} \right| \cdot \text{wid } \mathbf{q}_{ij}, \quad \left| \frac{\partial x_\nu(\mathbf{Q}, \mathbf{r})}{\partial r_i} \right| \cdot \text{wid } \mathbf{r}_i, \quad i, j \in \{1, 2, \dots, n\};$$

пытаемся уточнить матрицу W в соответствии с процедурой табл. 5;

“дробим” элемент \mathbf{h} и порождаем одну или две интервальные системы-потомки $\mathbf{Q}'\mathbf{x} = \mathbf{r}'$ и $\mathbf{Q}''\mathbf{x} = \mathbf{r}''$ в соответствии с процедурой табл. 1;

если были порождены две системы-потомка, вычисляем новые матрицы W' , W'' и векторы s' , s'' , t' , t'' в соответствии с процедурами табл. 2–4; иначе полагаем $W' := W$, $s' := s$, $t' := t$;

вычисляем интервальные вектор $\mathbf{x}' = \text{Encl}(\mathbf{Q}', \mathbf{r}')$ и, возможно, $\mathbf{x}'' = \text{Encl}(\mathbf{Q}'', \mathbf{r}'')$, беря \mathbf{x} в качестве начального приближения;

присваиваем оценку $v' := \Upsilon(\mathbf{Q}', \mathbf{r}')$ и, возможно, $v'' := \Upsilon(\mathbf{Q}'', \mathbf{r}'')$;

уточняем внешнюю оценку для “обратной интервальной матрицы” $\mathbf{Y}' \supseteq (\mathbf{Q}')^{-1}$ и, возможно, для $\mathbf{Y}'' \supseteq (\mathbf{Q}'')^{-1}$, беря \mathbf{Y} в качестве начального приближения;

вычисляем оценку $\Upsilon(\text{mid } \mathbf{Q}', \text{mid } \mathbf{r}')$ и, возможно, $\Upsilon(\text{mid } \mathbf{Q}'', \text{mid } \mathbf{r}'')$ и полагаем $\mu := \min\{\Upsilon(\text{mid } \mathbf{Q}', \text{mid } \mathbf{r}'), \Upsilon(\text{mid } \mathbf{Q}'', \text{mid } \mathbf{r}'')\}$;

удаляем бывшую ведущую запись $(\mathbf{Q}, \mathbf{r}, v, \mathbf{Y}, W, s, t, \mathbf{x})$ из списка \mathcal{L} ;

если $v' \leq \omega$, то помещаем запись $(\mathbf{Q}', \mathbf{r}', v', W', s', t', \mathbf{Y}', \mathbf{x}')$ в список \mathcal{L} в порядке возрастания третьего поля;

если из ведущей ИСЛАУ были порождены две системы-потомка и $v'' \leq \omega$, то помещаем запись $(\mathbf{Q}'', \mathbf{r}'', v'', W'', s'', t'', \mathbf{Y}'', \mathbf{x}'')$ в список \mathcal{L} в порядке возрастания третьего поля;

если $\omega > \mu$, то полагаем $\omega := \mu$ и чистим список \mathcal{L} , т. е.

удаляем из него все такие записи $(\mathbf{Q}, \mathbf{r}, v, W, s, t, \mathbf{Y}, \mathbf{x})$, что $v > \omega$;

END DO

искомой оценки на этом пути возможно лишь при добавлении активной процедуры измельчения элементов матрицы ИСЛАУ, хотя рассматривать будет нужно все получающиеся при таком измельчении-дроблении интервальные системы.

Фактически это и реализуется в алгоритме LinPPS1. Список \mathcal{L} хранит все возникшие в процессе работы варианты (за исключением тех заведомо бесперспективных, которые выявляются посредством теста из § 1.4), а обрабатывается на каждом шаге алгоритма “самый перспективный” вариант, если мерой этой перспективности считать значение оценки $\Upsilon(\mathbf{Q}, \mathbf{r})$.

2. Сравнения и численные эксперименты

Как мы уже упоминали выше, задача внешнего оценивания объединенного множества решений ИСЛАУ является на сегодня одной из наиболее популярных и наиболее разработанных интервальных постановок задач. За сорок лет, прошедших с момента ее формулировки, были предложены четыре принципиально различных подхода к вычислению оптимальных (точных) оценок объединенного множества решений для общих интервальных линейных систем. Один из них восходит к работе У. Оеттли [19], который первым обнаружил, что пересечение объединенного множества решений с ортантами пространства \mathbb{R}^n является выпуклым полиэдром. Таким образом, точное значение $\min\{x_\nu \mid x \in \Xi_{uni}\}$ может быть найдено путем решения в каждом из ортантов некоторой задачи линейного программирования и последующим взятием минимального из результатов. Этот подход, развитие которого рассмотрено, например, Х. Янссоном в [20], и который мы распространили в [1] на задачи оценивания обобщенных множеств решений, основывается на пассивной переборной стратегии, а трудоемкость его экспоненциально растет в зависимости от размерности n . Поэтому практическая значимость его очень ограничена.

Следующие два вычислительных подхода к оптимальному решению “внешней задачи” для интервальных линейных систем — это методы дробления решений и методы дробления параметров (называемые также PSS-методами и PPS-методами), предложенные автором (см., например, работы [11, 16, 18, 21]). Хотя для $n \times n$ -системы сложность выполнения методов дробления решений в худшем случае пропорциональна 2^n , а верхняя оценка сложности выполнения методов дробления параметров равна даже 2^{n^2+n} , эти методы, имея в основе стратегию “ветвей и границ”, являются *адаптивными*. При исполнении каждого последующего шага как в методах дробления решений, так и в методах дробления параметров мы полностью используем информацию о результатах предыдущих шагов.

Наконец, четвертый и в настоящий момент наиболее популярный в зарубежной литературе подход к оптимальному решению “внешней задачи” предложен И. Роном [17] (см. также [13, 22]). Отталкиваясь от характеристики Оеттли — Прагера для объединенного множества решений, И. Рон показывает, что для случая квадратной невырожденной матрицы \mathbf{A} искомые значения $\min\{x_\nu \mid x \in \Xi_{uni}(\mathbf{A}, \mathbf{b})\}$ и $\max\{x_\nu \mid x \in \Xi_{uni}(\mathbf{A}, \mathbf{b})\}$, $\nu = 1, 2, \dots, n$, достигаются на множестве не более чем 2^n решений так называемого уравнения Оеттли — Прагера

$$|\text{mid } \mathbf{A} \cdot x - \text{mid } \mathbf{b}| = \text{rad } \mathbf{A} \cdot |x| + \text{rad } \mathbf{b}. \quad (19)$$

Вычисляя все возможные решения этого уравнения и сравнивая их между собой, мы получим оптимальные оценки множества решений за конечное число шагов. Поскольку процесс определения каждого последующего решения для (19) никак не зависит от решений, уже найденных раньше, алгоритм Рона в целом не является адаптивным (т. е. подобен переборным методикам), в то время как его трудоемкость пропорциональна 4^n в худшем случае и 2^n в лучшем.

Таким образом, все подходы, разработанные к настоящему моменту для вычисления оптимальных оценок объединенного множества решений интервальных линейных систем, имеют экспоненциальную трудоемкость в наихудшем случае. Как уже отмечалось, этот факт не является

следствием “плохости” предложенных алгоритмов, а отражает глубокие свойства самого объединенного и других множеств решений интервальных линейных систем: задача вычисления его оптимальных покоординатных оценок оказалась NP-трудной [23–28]. Следовательно, экспоненциальная сложность всех перечисленных выше алгоритмов существенна и не может быть устранена [29].

Каковы же преимущества и недостатки методов дробления решений и методов дробления параметров в сравнении с другими подходами для нахождения оптимальных решений “внешней задачи”? Наиболее важная их особенность состоит в том, что они порождают последовательности приближенных оценок искомых величин “с нужной стороны”, т. е. для $\min\{x_\nu \mid x \in \Xi_{uni}\}$ снизу, а для $\max\{x_\nu \mid x \in \Xi_{uni}\}$ сверху. Именно такие оценки и требуются в соответствии со смыслом “внешней задачи”. Процесс выполнения метода дробления параметров, например, разбивается на ряд эффективно вычисляемых этапов-шагов, в результате каждого из которых мы получаем некоторое решение “внешней задачи”. После того как в таком алгоритме проработал хотя бы один из этих этапов, его прерывание в любой момент приведет к тому, что мы все равно будем иметь в своем распоряжении некоторое решение “внешней задачи”. Иными словами, если у нас имеются достаточные вычислительные мощности, то, реализуя методы дробления параметров (как и методы дробления решений), мы можем быть вполне уверены, что некоторый ответ к задаче будет наверняка получен, хотя, возможно, и не оптимальный. Следовательно, как методы дробления решений, так и методы дробления параметров являются *последовательно гарантирующими*. С учетом труднорешаемости “внешней задачи” для интервальных линейных систем это свойство радикальным образом выделяет методы дробления решений и методы дробления параметров из всех других подходов для вычисления оптимального решения.

Напротив, подходы Оеттли и Рона к нахождению оптимальных решений “внешней задачи”, имея экспоненциальную в худшем случае трудоемкость, дают желаемые “внешние” оценки объединенного множества решений лишь в финале, при естественном завершении своей работы, поскольку раньше нельзя гарантировать то, что вычисленная оценка в действительности не превосходит $\min\{x_\nu \mid x \in \Xi_{uni}\}$ (соответственно, не меньше $\max\{x_\nu \mid x \in \Xi_{uni}\}$). Следовательно, эти алгоритмы являются *финально гарантирующими*. Если размерность интервальной линейной системы достаточно велика (больше нескольких десятков), то в силу труднорешаемости “внешней задачи” количество арифметических и логических операций, необходимое для того, чтобы задача была наверняка решена, начинает превосходить количество операций, выполнимое на сколь угодно мощном компьютере за любое разумное время (час, день, год или даже столетие). В этих условиях нельзя быть уверенным, что финально гарантирующий алгоритм, будучи примененным к задаче, вообще завершит свою работу и, таким образом, будет получено решение поставленной задачи. Иначе говоря, применяя финально гарантирующий алгоритм, мы рискуем попусту растратить время и ресурсы без получения хоть какого-то ответа к нашей задаче.

Этот пессимистичный прогноз особенно справедлив для пассивных переборных алгоритмов. Если адаптивные (последовательные) алгоритмы достигают экспоненциальной трудоемкости лишь на наихудших вариантах, а в среднем для “не очень плохих задач” работают с приемлемыми трудозатратами, то, напротив, пассивные переборные алгоритмы, какими являются подходы Оеттли — Прагера и Рона, требуют экспоненциальных трудозатрат *всегда*, вне зависимости от характера решаемой задачи.

Отметим, что алгоритмы из [17, 19] для нахождения оптимальных решений “внешней задачи” — все-таки последовательно гарантирующие, но относительно так называемого “слабого внутреннего” оценивания. Фактически эти алгоритмы решают не “внешнюю”, а некоторую другую задачу для интервальной линейной системы, требующую оценивания $\min\{x_\nu \mid x \in \Xi_{uni}(\mathbf{A}, \mathbf{b})\}$ сверху и $\max\{x_\nu \mid x \in \Xi_{uni}(\mathbf{A}, \mathbf{b})\}$ снизу.

Сказанное мы проиллюстрируем результатами вычислительных экспериментов на рабочей станции Sun ULTRA-10 (тактовая частота процессора 300 МГц, шины — 100 МГц, емкость ОЗУ 128 Мбайт), которые были проведены с несколькими версиями методов дробления параметров,

реализованными на языке программирования Fortran фирмы Sun Microsystems.²

Для того чтобы продемонстрировать влияние различных факторов на поведение методов дробления параметров для внешнего оценивания объединенного множества решений ИСЛАУ, мы приводим в табл. 8 результаты тестовых расчетов с четырьмя методами:

- A* — простейший метод дробления параметров (т. е. алгоритм из табл. 2 работы [1]), но дополненный процедурой удаления из рабочего списка \mathcal{L} бесперспективных записей;
- B* — тот же алгоритм, что и *A*, но дополненный процедурой сжатия элементов матрицы ИСЛАУ на основе информации о монотонности оценки (см. § 1.1);
- C* — тот же алгоритм, что и *B*, но дополненный процедурой сжатия компонент правой части ИСЛАУ на основе информации о монотонности оценки (см. § 1.1);
- D* — тот же алгоритм, что и *C*, но в котором отсекаемая компонента выбирается из условия максимизации произведения оценки производной на ширину интервала (как это описано в § 1.2).

Т а б л и ц а 8

4 × 4-система Ноймайера с диагональю $t = 5.5$

Алгоритм	A	B	C	D
Количество итераций	51	51	39	15
Максимальная длина списка	45	45	21	9
Время счета	< 1 с	< 1 с	< 1 с	< 1 с

5 × 5 система Ноймайера с диагональю $t = 7$

Алгоритм	A	B	C	D
Количество итераций	506	506	421	59
Максимальная длина списка	503	503	285	48
Время счета	< 1 с	2 с	2 с	< 1 с

6 × 6-система Ноймайера с диагональю $t = 8.5$

Алгоритм	A	B	C	D
Количество итераций	15760	15760	10655	441
Максимальная длина списка	15744	15744	4644	302
Время счета	110 с	190 с	103 с	6 с

²Это фактически язык Fortran-90 со встроенными интервальными типами данных, операциями и отношениями между ними, см. <http://www.sun.com>.

Модификация Рона из § 1.3 нами в этих алгоритмах не использовалась. Фактически на основании приводимых ниже результатов можно сравнивать даже пять алгоритмов, включая самый простейший метод дробления параметров, поскольку он отличается от алгоритма A лишь длиной списка.

В качестве базового метода мы использовали интервальный метод Гаусса — Зейделя со стандартным предобуславливанием обратной средней матрицей, этот же метод использовался для внешнего оценивания “обратной интервальной матрицы” ИСЛАУ. Модельной задачей нам служила интервальная линейная система

$$\begin{pmatrix} t & [0, 2] & \cdots & [0, 2] \\ [0, 2] & t & \cdots & [0, 2] \\ \vdots & \vdots & \ddots & \vdots \\ [0, 2] & [0, 2] & \cdots & t \end{pmatrix} x = \begin{pmatrix} [-1, 1] \\ [-1, 1] \\ \vdots \\ [-1, 1] \end{pmatrix}, \quad (20)$$

которая была предложена А. Ноймайером в книге [13]. Оказывается, что матрицы системы (20) четного порядка n невырождены при $t > n$, а матрицы нечетного порядка n невырождены при $t > \sqrt{n^2 - 1}$.

Некоторую информацию о строении объединенного множества решений $\tilde{\Xi}$ интервальной линейной системы (20) можно получить из соображений симметрии. Из-за уравновешенности вектора правой части рассматриваемая ИСЛАУ инвариантна относительно изменения знаков всех компонент решения на противоположный, так что ее множество решений $\tilde{\Xi}$ центрально симметрично относительно начала координат, в частности,

$$\min\{x_i \mid x \in \tilde{\Xi}\} = -\max\{x_i \mid x \in \tilde{\Xi}\}, \quad i = 1, 2, \dots, n. \quad (21)$$

Так как для любых $i, j \in \{1, 2, \dots, n\}$ после замены x_i на x_j и наоборот интервальная система (20) остается неизменной, множество $\tilde{\Xi}$ симметрично относительно биссектрисы положительного и отрицательного ортантов пространства \mathbb{R}^n , так что

$$\begin{aligned} \min\{x_i \mid x \in \tilde{\Xi}\} &= \min\{x_j \mid x \in \tilde{\Xi}\}, \\ \max\{x_i \mid x \in \tilde{\Xi}\} &= \max\{x_j \mid x \in \tilde{\Xi}\} \end{aligned}$$

для любых $i, j \in \{1, 2, \dots, n\}$. Сопоставляя эти соотношения с (21), можно заключить, что интервальная оболочка множества решений $\tilde{\Xi}$ системы (20) является гиперкубом с центром в начале координат.

В случае приближения значения t к границам невырожденности (n для четных размерностей и $\sqrt{n^2 - 1}$ для нечетных) размеры объединенного множества решений ИСЛАУ (20) неограниченно возрастают. Варьируя t и n , легко получить из (20) набор тестов для проверки развитых нами алгоритмов оптимального решения “внешней задачи” для ИСЛАУ.

Результаты тестовых расчетов с интервальными линейными системами Ноймайера размера 4×4 , 5×5 и 6×6 представлены в табл. 10. С одной стороны, подобные задачи являются все-таки не слишком тяжелыми и просчитываются рассматриваемыми нами алгоритмами “до конца”, т. е. до того момента, когда ведущая оценка станет точной. А с другой стороны, комбинаторная сложность этих задач при решении “лобовым перебором” (количество крайних точечных систем уравнений, равное соответственно 2^{20} , 2^{30} и 2^{42}) достаточно велика и позволяет предметно судить о сравнительной эффективности вычислительных схем.

и матрицы

$$\begin{pmatrix} 1 & 2 & 3 & \cdots & n-1 & n \\ 2 & 2 & 3 & \cdots & n-1 & n \\ 3 & 3 & 3 & \cdots & n-1 & n \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ n-1 & n-1 & n-1 & \cdots & n-1 & n \\ n & n & n & \cdots & n & n \end{pmatrix}. \quad (24)$$

Интервальный вектор правой части для всех трех случаев брался одинаковым и равным

$$([0.999, 1.001], [0.999, 1.001], \dots, [0.999, 1.001])^\top.$$

Интересно, что еще в начале 90-х годов на персональном компьютере с процессором Intel 80486DX и математическим сопроцессором К. Мадсен и О. Тофт не смогли успешно просчитать до конца методом Рона интервальные линейные 30×30 -системы:

- 1) с матрицей (22), все элементы которой равномерно уширены на интервал $[-0.002, 0.002]$;
- 2) с матрицей (23), все элементы которой уширены на $[-0.0003, 0.0003]$;
- 3) с матрицей (24), все элементы которой уширены на $[-0.00001, 0.00001]$.

Вскоре задача 2 из этого списка все-таки была успешно решена, но для этого К. Мадсену и О. Тофту потребовались привлечение транспьютера Meiko Transputer System и распараллеливание вычислений на 32 процессорах. Общее последовательное время работы всех процессоров составило при этом 2471.27 с (более 40 мин). О решении задач 1 и 3 даже на транспьютере в работах [30, 31] ничего не сообщается.

Что касается методов дробления параметров, то алгоритм, приведенный в табл. 8 под литерой *D*, успешно справился со всеми тремя упомянутыми задачами на однопроцессорной рабочей станции SUN Ultra-10. При этом:

Задача 1 была решена всего за две итерации-бисекции и время порядка одной секунды по каждой отдельной оцениваемой компоненте множества решений.

Задача 2 была решена всего за две итерации-бисекции и время порядка одной секунды по каждой из отдельных оцениваемых компонент множества решений.

Задача 3 даже с более широкой матрицей радиуса 0.0001 успешно решалась для отдельной оцениваемой компоненты множества решений за время, не превосходящее 14 мин и не более чем за 1646 итераций-бисекций. Этот худший результат был достигнут для 10-й компоненты множества решений, а для других компонент трудоемкость исполнения варьировалась в широких пределах — от 50–60 итераций-бисекций до нескольких сотен и тысячи с лишним.

Наконец, с помощью методов дробления параметров автором был получен рекордный результат. Алгоритм *D* позволил успешно вычислять оптимальные по координатным оценки множества решений интервальной ИСЛАУ размера 300×300 с матрицей (22), уширенной на $[-0.002, 0.002]$ всего за две итерации-бисекции и время порядка 20 мин (по любой из компонент). Конечно, задача 1 с интервализацией матрицы (22) умеренно сложная, но тем не менее столь большие размеры решаемых систем в задаче оптимального внешнего оценивания множеств решений общих ИСЛАУ не рассматривал еще никто в мире!

В заключение можно резюмировать, что приведенные выше данные тестовых прогонов, а также свойство последовательной гарантии методов дробления параметров (см., например, [1, 33]) убедительно свидетельствуют в пользу их решительного преимущества перед методами Рона.

Список литературы

- [1] Шарый С.П. Оптимальное внешнее оценивание множеств решений интервальных систем уравнений. Ч. 1 // Вычисл. технологии. 2000. Т. 7, № 6. С. 90–113.

- [2] Панков П. С. Алгоритм доказательного поиска экстремума с использованием миноранты по области // Изв. АН Киргизской ССР. 1979. № 6. С. 12, 13.
- [3] Панков П. С. Алгоритмы доказательства устойчивых утверждений и глобальной оптимизации в ограниченной области. Фрунзе, 1984. 13 с. Деп. в ВИНТИ, № 5250-84.
- [4] ASAITHAMBI N. S., SHEN ZUHE, MOORE R. E. On computing the range of values // Computing. 1982. Vol. 28, No 3. P. 225–237.
- [5] HANSEN E. Global Optimization Using Interval Analysis. N. Y.: Marcel Dekker, 1992.
- [6] KEARFOTT R. B. Rigorous Global Search: Continuous Problems. Dordrecht: Kluwer, 1996.
- [7] RATSCHЕК H., ROKNE J. New Computer Methods for Global Optimization. Chichester, N. Y.: Ellis Horwood, Halsted Press, 1988.
- [8] Алефельд Г., Херцбергер Ю. Введение в интервальные вычисления. М.: Мир, 1987.
- [9] Зорич В. А. Математический анализ. Т. 1. М.: Наука, 1981. Т. 2; М.: Наука, 1984.
- [10] Калмыков С. А., Шокин Ю. И., Юлдашев З. Х. Методы интервального анализа. Новосибирск: Наука, 1986.
- [11] SHARY S. P. On optimal solution of interval linear equations // SIAM J. Numer. Analysis. 1995. Vol. 32, No 2. P. 610–630.
- [12] GAY D. M. Solving interval linear equations // SIAM J. Numer. Analysis. 1982. Vol. 19, No 4. P. 858–870.
- [13] NEUMAIER A. Interval Methods for Systems of Equations. Cambridge: Cambridge Univ. Press, 1990.
- [14] RATZ D. Automatische Ergebnisverifikation bei globalen Optimierungsproblemen. Ph.D. Dissertation. Karlsruhe: Univ. Karlsruhe, 1992.
- [15] RATZ D., CSENDES T. On the selection of subdivision directions in interval branch-and-bound methods for global optimization // J. Global Optimization. 1995. Vol. 7. P. 183–207.
- [16] SHARY S. P. A new class of algorithms for optimal solution of interval linear systems // Interval Computations. 1992. No 2(4). P. 18–29.
- [17] ROHN J. Systems of linear interval equations // Linear Algebra Appl. 1989. Vol. 126. P. 39–78.
- [18] SHARY S. P. Optimal solution of interval linear algebraic systems. I // Interval Computations. 1991. Vol. 1, No 2. P. 7–30.
- [19] OETTLI W. On the solution set of a linear system with inaccurate coefficients // SIAM J. Numer. Analysis. 1965. Vol. 2, No 1. P. 115–118.
- [20] JANSSON C. Calculation of exact bounds for the solution sets of linear interval systems // Linear Algebra Appl. 1997. Vol. 251. P. 321–340.
- [21] Шарый С. П. Новый класс алгоритмов для оптимального решения интервальных линейных систем // Актуальные проблемы прикладной математики: Мат. конф., Саратов, 20–22 мая 1991. Саратов, 1991. С. 113–119.
- [22] KOLEV L. V. Interval Methods for Circuit Analysis. Singapore: World Scientific, 1993.

- [23] Лакеев А. В., Носков С. И. Описание множества решений линейного уравнения с интервально заданными оператором и правой частью // Докл. РАН. 1993. Т. 330, № 4. С. 430–433.
- [24] Лакеев А. В., Носков С. И. О множестве решений линейного уравнения с интервально заданными оператором и правой частью // Сиб. мат. журнал. 1994. Т. 35, № 5. С. 1074–1084.
- [25] KREINOVICH V., LAKEYEV A. V., NOSKOV S. I. Optimal solution of interval linear systems is intractable (NP-hard) // Interval Computations. 1993. No 1. P. 6–14.
- [26] KREINOVICH V., LAKEYEV A. V., NOSKOV S. I. Approximate linear algebra is intractable // Linear Algebra Appl. 1996. Vol. 232. P. 45–54.
- [27] KREINOVICH V., LAKEYEV A., ROHN J., KANL P. Computational Complexity and Feasibility of Data Processing and Interval Computations. Dordrecht: Kluwer Acad. Publ., 1997.
- [28] ROHN J., KREINOVICH V. Computing exact componentwise bounds on solutions of linear system is NP-hard // SIAM J. Matrix Analysis Appl. 1995. Vol. 16. P. 415–420.
- [29] Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982.
- [30] MADSEN K., TOFT O. A parallel method for linear interval equations // Interval Computations. 1994. No 3. P. 81–105.
- [31] TOFT O. Sequential and parallel solution of linear interval equations // Eksamensproject: NI-E-92-04, Numerisk Institute, Danmarks Tekniske Hojskole. Lyngby, 1992. 98 p.
- [32] GREGORY R.T, KARNEY D.L. A Collection of Matrices for Testing Computational Algorithms. N. Y.: Wiley Interscience, John Wiley, 1969.
- [33] Шарый С.П. Интервальные алгебраические задачи и их численное решение: Дис. ... д-ра физ.-мат. наук. Новосибирск, 2000.
- [34] Пападимитриу Х., Стайглиц К. Комбинаторная оптимизация. Алгоритмы и сложность. М.: Мир, 1985.
- [35] Шарый С. П. Алгебраический подход во “внешней задаче” для интервальных линейных систем // Вычисл. технологии. 1998. Т. 3, № 2. С. 67–114.
- [36] HEINDL G., KREINOVICH V., LAKEYEV A. Solving linear interval systems is NP-hard even if we exclude overflow and underflow // Reliable Computing. 1998. Vol. 4. P. 383–388.
- [37] MOORE R. E. Methods and Applications of Interval Analysis. Philadelphia: SIAM, 1979.
- [38] SHARY S. P. Algebraic approach in the “outer problem” for interval linear equations // Reliable Computing. 1997. Vol. 3, No 2. P. 103–135.