# Объединение стохастических и интервальных подходов для решения задач глобальной оптимизации функций

Н. В. Панов
*Учреждение Российской академии наук*
*Институт вычислительных технологий СО РАН, Новосибирск, Россия*
e-mail: rupo@ngs.ru

Рассматриваются интервальные методы глобальной оптимизации. Выявляются задачи, в которых существующие методы недостаточно эффективны, и анализируются причины этой неэффективности. Для преодоления слабых сторон традиционных детерминистских интервальных методов глобальной оптимизации предлагаются новые, вычислительно эффективные методы, сочетающие в себе доказательность интервальных и гибкость стохастических подходов.

*Ключевые слова*: интервал, интервальный анализ, глобальная оптимизация, генетический алгоритм, метод симулированного отжига, минимум.

## Введение

Предмет настоящей работы — поиск глобального оптимума вещественнозначной целевой функции $f : \mathbb{R}^n \to \mathbb{R}$ на некотором прямоугольном брусе $\boldsymbol{X}$ со сторонами, параллельными координатным осям, который является подмножеством области определения данной функции:

$$\text{найти} \quad \min_{x \in \boldsymbol{X}} f(x) \ . \tag{1}$$

Термин "глобальный" в применении к оптимуму означает, что мы должны найти точку, наилучшую в смысле целевой функции $f(x)$ *для всего* рассматриваемого бруса $\boldsymbol{X}$, а не только локально, в пределах некоторой окрестности начального приближения. При отсутствии "хороших" глобальных условий на целевую функцию эта особенность постановки задачи глобальной оптимизации делает ее весьма трудной для решения, так что соответствующая ветвь оптимизации выделяется обычно в самостоятельное научное направление в рамках общей теории оптимизации.

К настоящему моменту наработан богатый инструментарий поиска глобального оптимума [1 – 9]. Существующие методы можно условно разделить, с одной стороны, на детерминистские и стохастические, а с другой, по применяемой в них технике, — на классические (точечные) и интервальные. Точнее это разделение можно представить в виде дерева на рис. 1. В представленной схеме правая нижняя ветвь, отвечающая интервальным стохастическим методам, является весьма неразвитой и до сих пор представлена лишь публикациями [10 – 12]. Настоящая работа призвана в какой-то мере

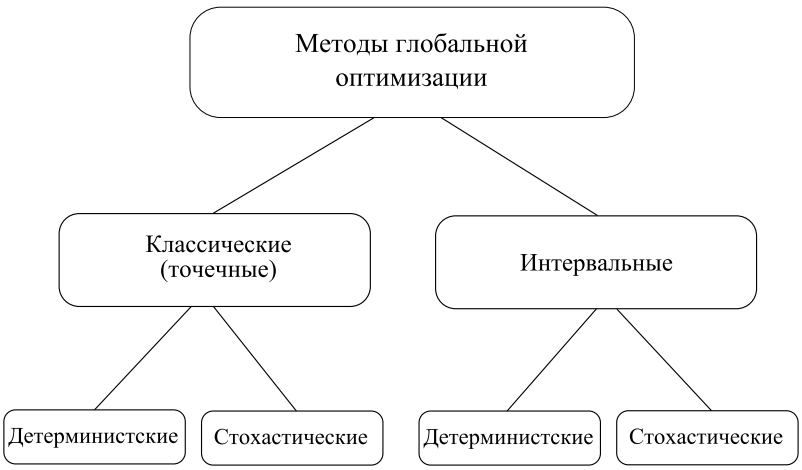


Рис. 1. Классификация алгоритмов глобальной оптимизации

восполнить этот дисбаланс и соответственно посвящена разработке интервальных стохастических алгоритмов глобальной оптимизации функций.

Традиционные детерминистские методы для решения задачи глобальной оптимизации — это методы, которые используют однозначно определенную вычислительную схему, и, по сути, они должны каким-то образом осуществить перебор "всех" значений функции на заданной области. Знание такой дополнительной информации о функции, как ее производная, константы Липшица и т. п., позволяет ограничить количество пробных точек, при этом гарантируя, что найденный оптимум будет глобальным.

Стохастические методы не дают такой гарантии, они выдают ответ, который справедлив лишь с некоторой достаточно высокой вероятностью. Методы такого типа весьма гибки при реализации, позволяют достаточно быстро и с приемлемой точностью находить глобальный оптимум, а потому пользуются большой популярностью на практике.

Прокомментируем традиционные (детерминистские) интервальные методы глобальной оптимизации, основанные на достижениях интервального анализа [6, 7, 9]. Их отличительная особенность — *доказательность*, весьма ценное свойство, заключающееся в гарантии того, что найденный результат в самом деле является оптимумом и действительно глобальным. По большому счету, никакой дополнительной информации о функции они не требуют и не накладывают ограничений на целевую функцию. Достаточно лишь ее записи в аналитическом виде либо в виде вычислительного алгоритма. Но за это приходится платить необходимостью дополнительной обработки этих выражений, и основным недостатоком интервальных методов является то, что зачастую их производительность меньше производительности классических (точечных) алгоритмов.

Под руководством С.П. Шарого автором разработаны и реализованы интервальные стохастические алгоритмы (методы) глобальной оптимизации функций. Проведены серии вычислительных экспериментов, результаты которых показывают перспективность такого подхода к решению задачи глобальной оптимизации функций.

## 1. Анализ существующих методов

В качестве базового в наших экспериментах выступает классический метод адаптивного интервального дробления, имеющий в своей основе стратегию "ветвей и границ", заимствованную из дискретной оптимизации.

На каждой итерации алгоритм находит подбрус исходного бруса $\boldsymbol{x}$, дающий наихудшую оценку оптимума, и делит его на две (или более) части, таким образом последовательно уточняя оценку глобального экстремума (см. псевдокод в табл. 1). Подробнее метод описывается в работах [9, 11].

На графике (рис. 2) сравнивается работа этого алгоритма на двух целевых функциях. По оси абсцисс отложено время работы метода в логорифмическом масштабе с учетом погрешности измерения, по оси ординат — точность интервальной оценки оптимума, достигаемая за это время.

Функция "Растригин10" (R10) [13],

$$f_{\mathrm{R10}}(x, y) = x^2 + y^2 - \cos(18x) - \cos(18y) \tag{2}$$

с ее множеством локальных минимумов (общая форма функции представлена на рис. 3, *а*, а ее вид вблизи начала координат показан на рис. 3, *б*) неожиданно оказалась "легкой" для алгоритма — минимум локализован за доли секунды, в то время

Т а б л и ц а 1. Упрощенная схема интервального адаптивного алгоритма глобальной оптимизации функций

| Вход |
|---|
| Брус $\boldsymbol{X} \subset \mathbb{R}^n$. Заданная точность $\epsilon > 0$.<br>Интервальное расширение $\boldsymbol{f} : \mathbb{I}\boldsymbol{X} \to \mathbb{IR}$ целевой функции $f$ |
| **Выход** |
| Оценка снизу глобального минимума $f^*$ функции $f$ на $\boldsymbol{X}$ |
| **Алгоритм** |
| $\boldsymbol{Y} \leftarrow \boldsymbol{X}$;<br>вычисляем $\boldsymbol{f}(\boldsymbol{Y})$ и инициализируем список $\mathcal{L}$ записью $\{ (\boldsymbol{Y}, \underline{\boldsymbol{f}(\boldsymbol{Y})}) \}$;<br>`DO WHILE` $\left( \mathrm{wid}(\boldsymbol{f}(\boldsymbol{Y})) \geq \epsilon \right)$<br>выбираем компоненту $l$, по которой брус $\boldsymbol{Y}$ имеет наибольшую длину, т. е. $\mathrm{wid}\boldsymbol{Y}_l = \max\limits_i \mathrm{wid}\boldsymbol{Y}_i$;<br>рассекаем $\boldsymbol{Y}$ по $l$-й координате на брусы $\boldsymbol{Y}'$ и $\boldsymbol{Y}''$ такие, что<br>$\boldsymbol{Y}' = (\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_{l-1}, [\underline{\boldsymbol{Y}}_l, \mathrm{mid}\boldsymbol{Y}_l], \boldsymbol{Y}_{l+1}, \ldots, \boldsymbol{Y}_n)$,<br>$\boldsymbol{Y}'' = (\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_{l-1}, [\mathrm{mid}\boldsymbol{Y}_l, \overline{\boldsymbol{Y}}_l], \boldsymbol{Y}_{l+1}, \ldots, \boldsymbol{Y}_n)$;<br>вычисляем интервальные оценки $\boldsymbol{f}(\boldsymbol{Y}')$ и $\boldsymbol{f}(\boldsymbol{Y}'')$;<br>удаляем запись $(\boldsymbol{Y}, \underline{\boldsymbol{f}(\boldsymbol{Y})})$ из рабочего списка $\mathcal{L}$;<br>помещаем записи $(\boldsymbol{Y}', \underline{\boldsymbol{f}(\boldsymbol{Y}')})$ и $(\boldsymbol{Y}'', \underline{\boldsymbol{f}(\boldsymbol{Y}'')})$ в список $\mathcal{L}$ в порядке возрастания второго поля;<br>обозначаем ведущую запись списка $\mathcal{L}$ через $(\boldsymbol{Y}, \underline{\boldsymbol{f}(\boldsymbol{Y})})$;<br>`END DO`<br>$F^* \leftarrow \underline{\boldsymbol{f}(\boldsymbol{Y})}$; |

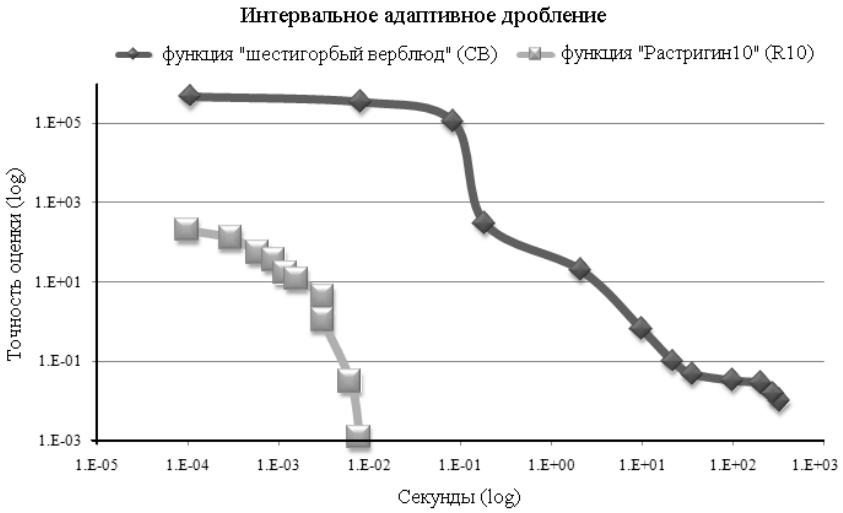


Рис. 2. График точности оценки глобального минимума двух целевых функций от времени для интервального алгоритма ветвей и границ

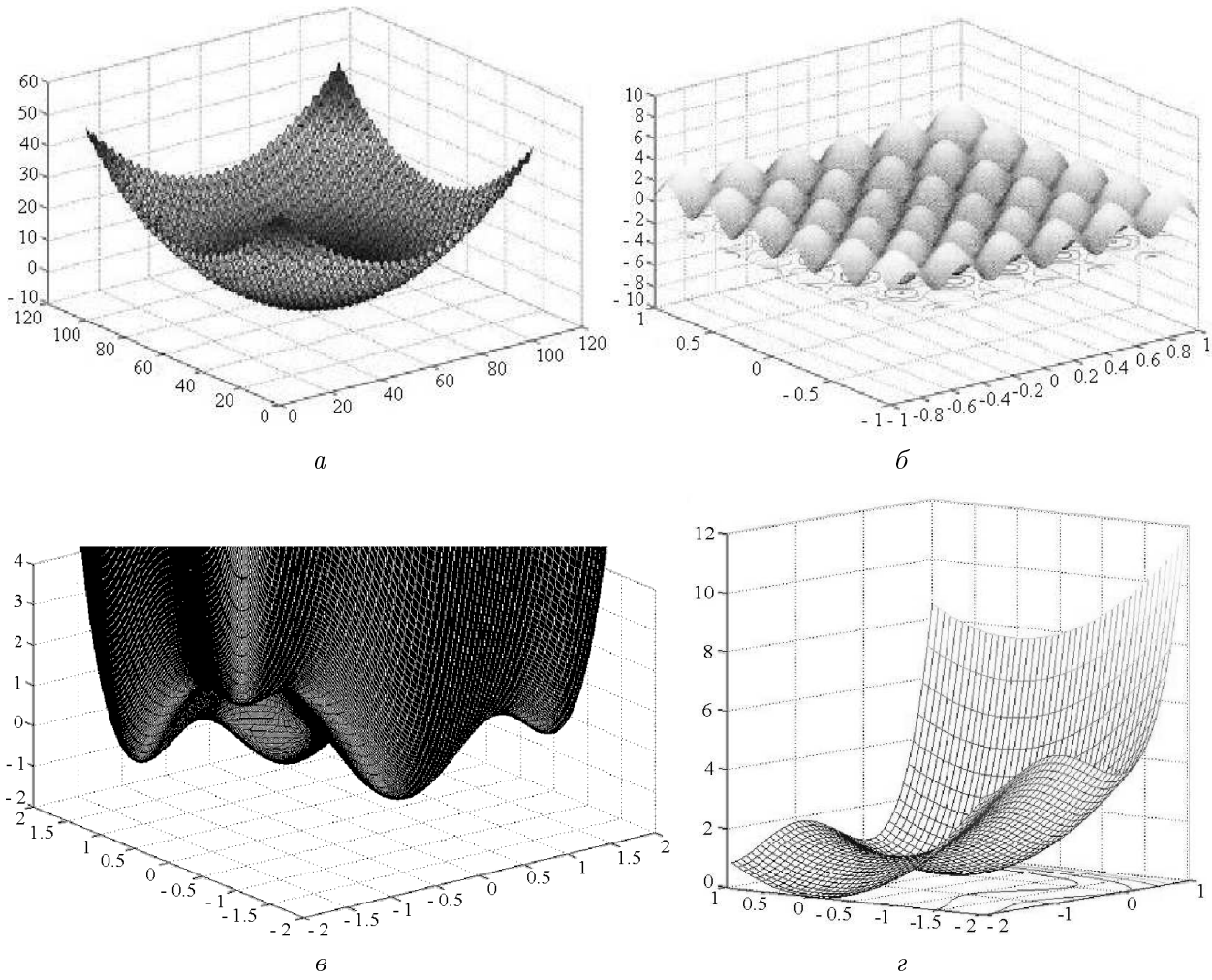


Рис. 3. Общая форма функции Растригина 10 (R10) (*а*) и ее вид вблизи начала координат (*б*); вид функции "шестигорбый верблюд" (СВ) вблизи начала координат (*в*); вид функции Трекани вблизи начала координат (*г*)

как функция "шестигорбый верблюд" (Six Hump Camel Back или, сокращенно, CB) [13] (рис. 3, *в*)

$$f_{\text{CB}}(x,y) = 4x^2 - 2.1x^4 + \frac{1}{3}x^6 + xy - 4y^2 + 4y^4 \tag{3}$$

оказалась для данного алгоритма чрезвычайно сложной. Для нахождения минимума с приемлемой точностью пришлось затратить более 300 с, что характеризует эффективность метода как весьма низкую. Необходимо заметить, что во всех примерах в этой статье для простоты объяснения и большей наглядности результатов нигде не использовались так называемые *процедуры отбраковки* — вычислительные приемы, позволяющие достоверно определять и как результат отсеивать области, гарантированно не содержащие оптимум. Это чрезвычайно полезная процедура, и она реализуется на основе сугубо интервальной техники, позволяя существенно улучшить сходимость метода. Пожалуй, наиболее популярными на данный момент являются методы отбраковки по значению, тест на монотонность, проверка выпуклости, метод Ньютона (см., например, в [6]).

Возникает вопрос, почему на одной целевой функции алгоритм работает хорошо, в то время как на другой функции его производительность чрезвычайно низка?

Из графика функции "шестигорбый верблюд" (CB) (см. рис. 2) хорошо видны проявления так называемого эффекта застаивания интервальной оценки [10], когда за большой промежуток времени (на графике приведены логарифмические шкалы по обеим координатам) и соответственно за большое число итераций точность интервальной оценки улучшилась лишь незначительно.

Но это лишь одна из причин. Другая становится очевидной, если рассмотреть работу алгоритма на этой целевой функции шаг за шагом.

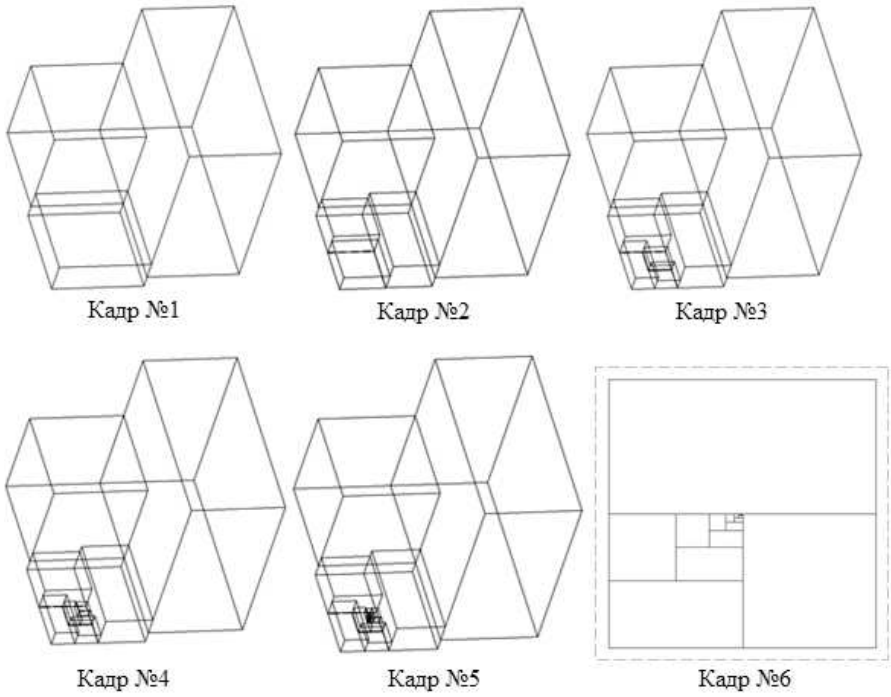


Рис. 4. Визуализация работы алгоритма интервального адаптивного дробления на целевой функции R10: кадры 1–5 — процесс дробления, кадр 6 — "вид сверху" на область определения

На рис. 4 приведены несколько кадров (шаги с номерами 1–5), полученных специальной программой-визуализатором (подробнее о реализации этой программы и методах исследования см. [14]) и показывающих, как менялась конфигурация рабочего списка при поиске глобального оптимума функции Растригина 10 (R10). Хорошо видно, что алгоритм ни разу "не ошибся" — каждое дробление действительно уточняло оценку глобального оптимума. Это отчетливо видно на кадре № 6 (рис. 4). На нем дан "вид сверху" и показано, как дробилась исходная область определения.

На рис. 5 приведены двумерные проекции, иллюстрирующие, как дробилась исходная область определения при поиске глобального оптимума функции "шестигорбый верблюд" (СВ).

Сделаем краткие выводы. В отличие от процесса на рис. 4 "ведущий брус" постоянно меняется, алгоритм последовательно мельчит соседние брусы. При этом точность оценивания улучшается лишь незначительно — имеет место эффект "застаивания интервальной оценки", т. е. ситуация, когда достаточно большое число последовательных дроблений не привело к значимому улучшению точности интервальной оценки (рис. 6). Дополнительно ухудшают поиск глобального минимума "ложные оптимумы", т. е. брусы, признаваемые ведущими, хотя реально и не содержащими глобального оптимума.

В заключение отметим основные причины неуспешности алгоритма:

— интервальные оценки областей значений функции, получаемые с помощью интервальных расширений, могут являться весьма грубыми;

— зачастую границы интервальных оценок целевой функции на подбрусах не коррелируют с действительной областью значений. Иными словами, если $\boldsymbol{a}$ и $\boldsymbol{b}$ — два каких-то подбруса, то, несмотря на неравенство $\min\limits_{x\in\boldsymbol{a}} f(x) > \min\limits_{x\in\boldsymbol{b}} f(x)$ для точных мини-

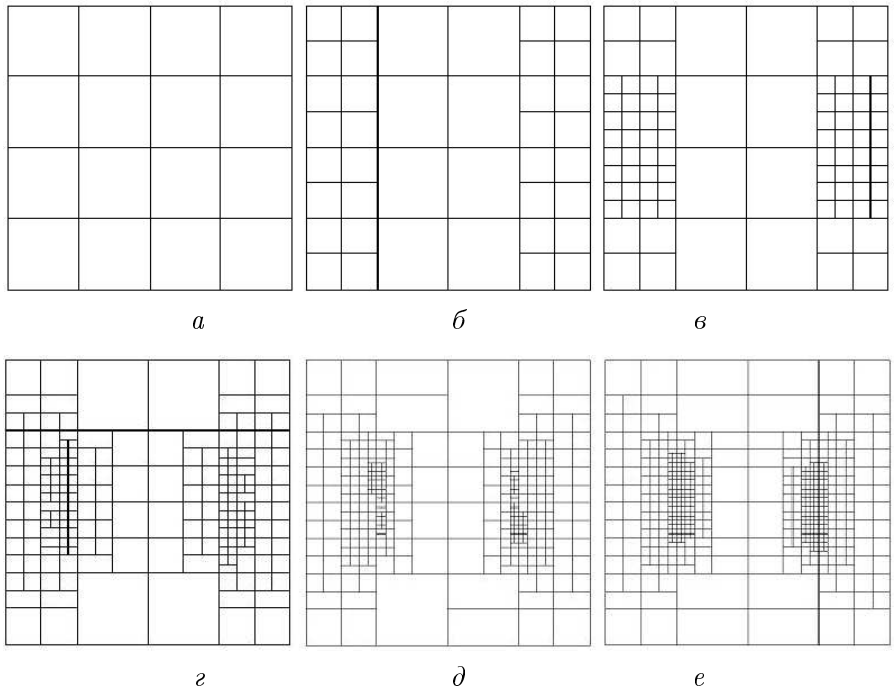


Рис. 5. Дробление исходной области определения алгоритмом интервального адаптивного дробления при поиске глобального минимума целевой функции "шестигорбый верблюд" (СВ): *а*—*е* соответствуют 16-, 40-, 88-, 202-, 308- и 402-й итерациям (дроблениям) соответственно

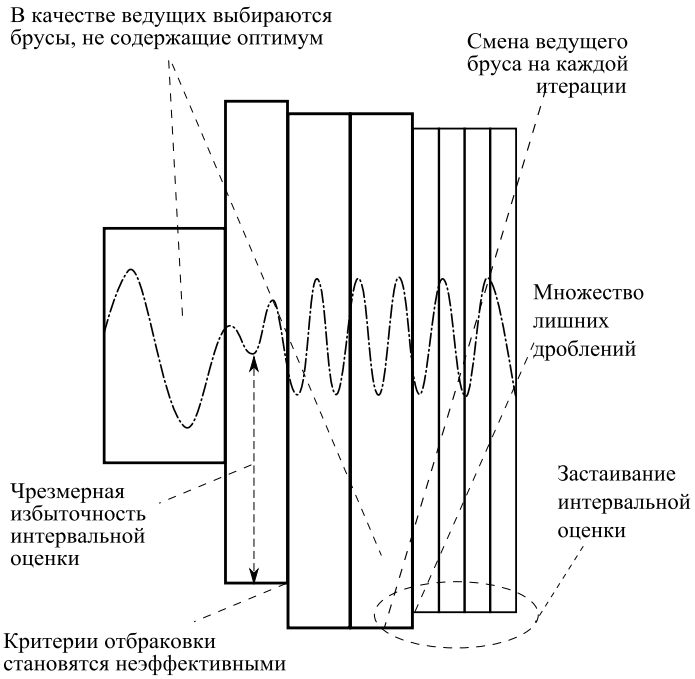


Рис. 6. Процесс работы детерминистского алгоритма: последовательные дробления области определения на все более мелкие подбрусы, практически не уточняющие границы глобального оптимума

мумов на этих подбрусах, нижние границы интервальных оценок часто оказываются в противоположном отношении, т. е. $\underline{\boldsymbol{f}(\boldsymbol{a})} < \underline{\boldsymbol{f}(\boldsymbol{b})}$ (рис. 6);

— эффект застаивания интервальной оценки вместе с двумя указанными выше явлениями заставляет алгоритм делать очень много дроблений впустую, фактически не приближаясь к глобальному оптимуму.

## 2. Интервальные стохастические алгоритмы

Из приведенного анализа понятно, что традиционные интервальные алгоритмы глобальной оптимизации скорее всего будут демонстрировать неблагоприятное поведение в задачах со многими локальными экстремумами и сложной целевой функцией. В соответствии со своей внутренней логикой эти алгоритмы будут последовательно мельчить множество брусов, не содержащих на самом деле оптимум, при этом лишь незначительно улучшая точность интервальной оценки.

Очевидно, что надо как-то изменить процедуру выбора бруса для дробления. Естественным, хотя и несколько необычным шагом является введение случайности (рандомизации) в этот процесс. Впервые подобная идея — отказаться от жесткого детерминизма классических методов интервальной оптимизации с целью получения принципиально новых алгоритмов — была высказана С.П. Шарым в публикациях [11, 12]. В этих работах был предложен, в частности, алгоритм “случайного интервального дробления”, блок-схема которого изображена на рис. 7.

Мы испытали алгоритм случайного интервального дробления для функции Трекани (Trec) [13]:

$$f_{\mathrm{Trec}}(x, y) = x^4 + 4x^3 + 4x^2 + y^2, \tag{4}$$

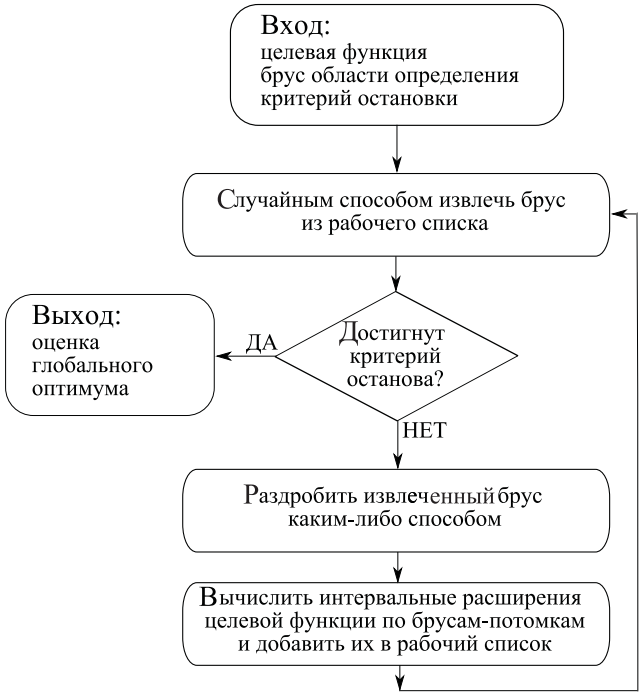


Рис. 7. Блок-схема случайного интервального дробления

на начальном брусе $[-1,1]\times[-1,1]$, (вид функции вблизи начала координат показан на рис. 3, *г*). Через 0.01 с алгоритм был принудительно остановлен. За это время минимум был локализован как находящийся на брусе $[-1,0]\times[-0.125,0]$ и значение целевой функции — в интервале $[-1.4,3.3]$. При этом список брусов имел вид, показанный на рис. 8.

Практически в соответствии с этим распределялась и точность интервальной оценки функции. Буквой D на диаграммах обозначена ширина брусов. Видно, что основную долю составляют достаточно мелкие брусы, но есть и несколько крупных. Несмотря на весьма хорошее распределение, именно эти несколько нераздробленных брусов и не позволяют алгоритму отыскать глобальный оптимум. Действительно, с увеличением числа итераций какого-то значимого улучшения значения оптимума не происходит — ведь с каждым дроблением брусов становится больше, а вероятность выбрать тот самый широкий — меньше. Так, в этом эксперименте оценка не была существенно улучшена даже за 1000 с (напомним, что процедуры отбраковки не использовались).

Итак, подведем промежуточные итоги.

Используя метод случайного интервального дробления, мы получили алгоритм, не тратящий времени на такие второстепенные задачи, как упорядочивание рабочего списка или поиск наиболее перспективного бруса, а выполняющий практически только дробления. Но в силу отсутствия какой-либо другой управляющей логики, кроме случайного дробления брусов по равновероятному закону, он быстро замедляется, не справляясь с порожденным самим собой множеством подбрусов.

Дополнение простейшего стохастического алгоритма интервальной глобальной оптимизации процедурами отбраковки позволило решить эту проблему и существенно улучшить производительность метода за счет отсеивания более половины брусов. И как следствие локализовать минимум с абсолютной точностью $10^{-4}$ за 0.38 с. Правда, чтобы повысить точность до $10^{-5}$, пришлось потратить еще 2 с.

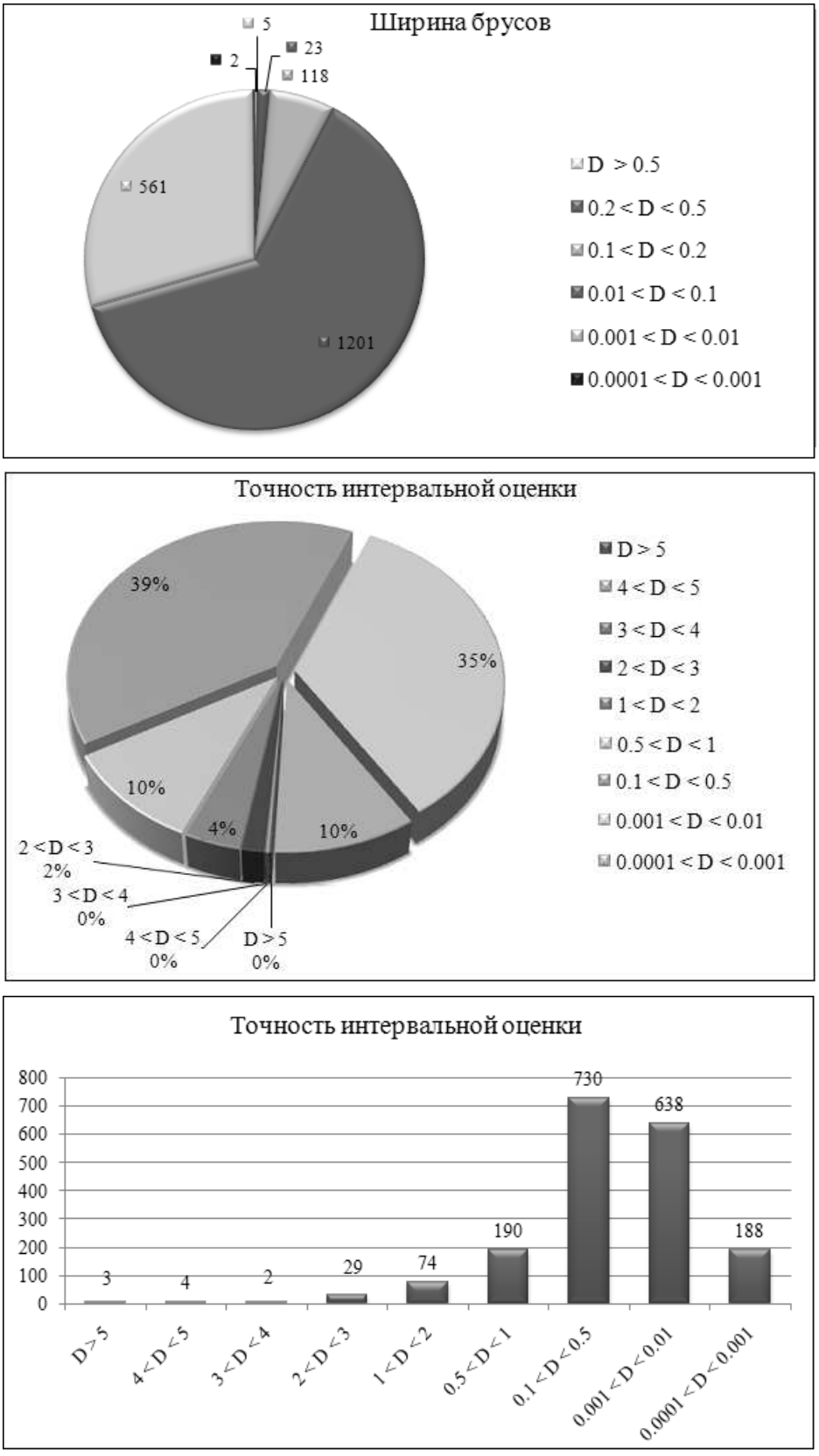


Рис. 8. Распределение ширины брусов и ширина интервальной оценки функции Трекани

Таким образом, с одной стороны имеем жестко детерминированный метод классического алгоритма "ветвей и границ", а с другой — "случайное интервальное дробление". Проблемы обоих подходов очевидны. Для классического адаптивного интервального дробления — это застаивание интервальной оценки и ложные оптимумы; для "случайного интервального дробления" — отсутствие адаптации к задаче.

Естественным шагом представляется модификация процедуры выбора ведущего бруса таким образом, чтобы более широкие брусы выбирались с большей вероятностью. То есть мы лишь слегка модифицировали процедуру выбора очередного кандидата на дробление. Выбор по-прежнему происходит случайно, но более широкие брусы имеют бóльшую вероятность быть раздробленными. Опуская технические подробности, скажем, что это уже вполне конкурентоспособный алгоритм, на "простых" задачах несколько уступающий адаптивному интервальному дроблению, зато многократно превосходящий его на целевых функциях, для которых характерны неточные интервальные оценки и проявление эффекта застаивания интервальной оценки.

Рассмотрим теперь следующий стохастический метод интервальной глобальной оптимизации из предложенных в [11, 12] — интервальный метод имитации отжига. За его основу был взят популярный классический (точечный) метод (известный так же как "алгоритм Метрополиса"), моделирующий физические процессы отжига или кристаллизации и хорошо зарекомендовавший себя для многоэкстремальных проблем с большим количеством возможных решений [3, 15]. Упрощенный псевдокод этого алгоритма приведен в табл. 2.

Алгоритм оперирует таким понятием, как температура $T$. Чем она выше, тем больше вероятность, что будет выбран брус, не доставляющий рекордную на данный момент оценку оптимума, т. е. тем больше "рысканье" по области определения. По мере работы алгоритма и уточнения оценки глобального оптимума температура понижается и описанные колебания постепенно уменьшаются, в пределе прекращаясь вовсе. Более подробно вычислительная схема и детали этого метода описаны в [10].

Нами была осуществлена реализация этого метода на ЭВМ и проведен ряд вычислительных экспериментов [10, 14]. Тестовые задачи брались в основном из работы [13]. Эксперименты показали преимущество нового метода в скорости нахождения глобального оптимума над классическим (детерминированным) алгоритмом интервальной глобальной оптимизации, описанным в начале нашей статьи, в случае, если целевая функция $f$ имела много локальных экстремумов и сложный рельеф. Если же $f$ имела не очень сложную структуру, то разработанный метод заметно проигрывал детерминистскому.

Так, на рис. 9 продемонстрировано сравнение работы интервального алгоритма имитации отжига с работой классического алгоритма интервального адаптивного дробления на целевой функции (3) "шестигорбый верблюд".

С другой стороны, применение метода интервального симулированного отжига с теми же настройками, что и в предыдущем случае, для поиска глобального минимума функции Растригин10 (2) дало чрезвычайно плохие результаты (рис. 10, *б*).

Объяснение дается на рис. 10, *а*, на котором проиллюстрирован непосредственно процесс дробления. Напомним, что при всех своих многоэкстремальности, неудобности и сложности для глобальной оптимизации функции (ее вид вблизи начала координат приводится на рис. 3), метод интервального адаптивного дробления успешно решает эту задачу. Вернемся к рис. 4, на котором показано, как, ни разу не ошибаясь, метод адаптивного интервального дробления движется к цели. В то же время, в метод имитации

Т а б л и ц а 2. Псевдокод простейшего интервального алгоритма имитации отжига

**Вход**

Брус области определения $\boldsymbol{x} \subset \mathbb{R}^n$,
Интервальное расширение $\boldsymbol{f} : \boldsymbol{x} \to \mathbb{IR}$ целевой функции $f$,
Начальное $T_0$ и конечное $T_{fin}$ значения "температуры"

**Выход**

Оценка $f^*$ глобального минимума функции $f$ на $\boldsymbol{X}$

**Алгоритм**

присваиваем $T \leftarrow T_0$ и $\boldsymbol{y} \leftarrow \boldsymbol{x}$;
назначаем целочисленную величину $N_T$
— количество испытаний на один температурный уровень;
вычисляем $\boldsymbol{f}(\boldsymbol{Y})$ и инициализируем список $\mathcal{L}$ записью $\{ (\boldsymbol{Y}, \underline{\boldsymbol{f}(\boldsymbol{Y})}) \}$;
`DO WHILE` $(T > T_{fin})$
`DO FOR` $j = 1$ `TO` $N_T$
по некотрому случайному правилу $\mathcal{S}(y)$ выбираем из рабочего списка $\mathcal{L}$ брус $\boldsymbol{z}$
`DO` (с вероятностью $\mathsf{P}_T(y, z)$ )
принимаем $z$, присваивая $y \leftarrow z$,
выбираем компоненту $l$, по которой брус $\boldsymbol{z}$ имеет наибольшую длину, т. е. $\mathrm{wid}\boldsymbol{z}_l = \max\limits_i \mathrm{wid}\boldsymbol{z}_i$;
рассекаем $\boldsymbol{z}$ по $l$-й координате на брусы $\boldsymbol{z}'$ и $\boldsymbol{z}''$ такие, что
$\boldsymbol{z}' = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{l-1}, [\underline{\boldsymbol{z}}_l, \mathrm{mid}\boldsymbol{z}_l], \boldsymbol{z}_{l+1}, \ldots, \boldsymbol{z}_n)$,
$\boldsymbol{z}'' = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{l-1}, [\mathrm{mid}\boldsymbol{z}_l, \overline{\boldsymbol{z}}_l], \boldsymbol{z}_{l+1}, \ldots, \boldsymbol{z}_n)$;
вычисляем интервальные оценки $\boldsymbol{f}(\boldsymbol{z}')$ и $\boldsymbol{f}(\boldsymbol{z}'')$;
удаляем запись $(\boldsymbol{z}, \underline{\boldsymbol{f}(\boldsymbol{z})})$ из рабочего списка $\mathcal{L}$;
помещаем записи $(\boldsymbol{z}', \underline{\boldsymbol{f}(\boldsymbol{z}')})$ и $(\boldsymbol{z}'', \underline{\boldsymbol{f}(\boldsymbol{z}'')})$ в список $\mathcal{L}$ в порядке возрастания второго поля;
обозначаем ведущую запись списка $\mathcal{L}$ через $(\boldsymbol{y}, \underline{\boldsymbol{f}(\boldsymbol{y})})$;
`END DO`
`END DO`
уменьшаем значение температуры $T \leftarrow \alpha T$;
`END DO`
$f^* \leftarrow f(y)$;

отжига заложено рысканье по области определения, которое с "плохими" настройками приводит к "метаниям" и множеству бессмысленных дроблений.

Мы привели этот пример, чтобы еще раз подчеркнуть, что поведение алгоритма существенным образом зависит от параметра $T$ (температуры) — как от начального его значения, так и от сценария его изменения со временем. В классическом (точечном) ва-

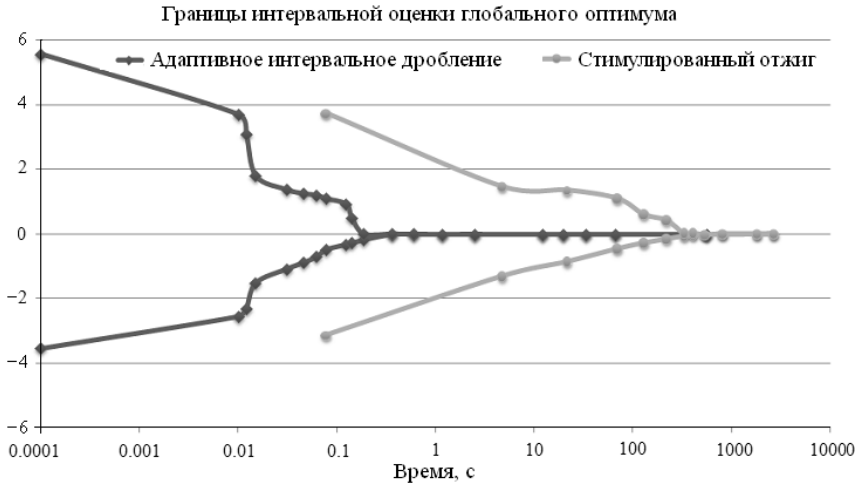


Рис. 9. Ширина оценки оптимума при интервальном методе имитации отжига и адаптивном интервальном дроблении в зависимости от времени (логарифмический масштаб по оси времени)

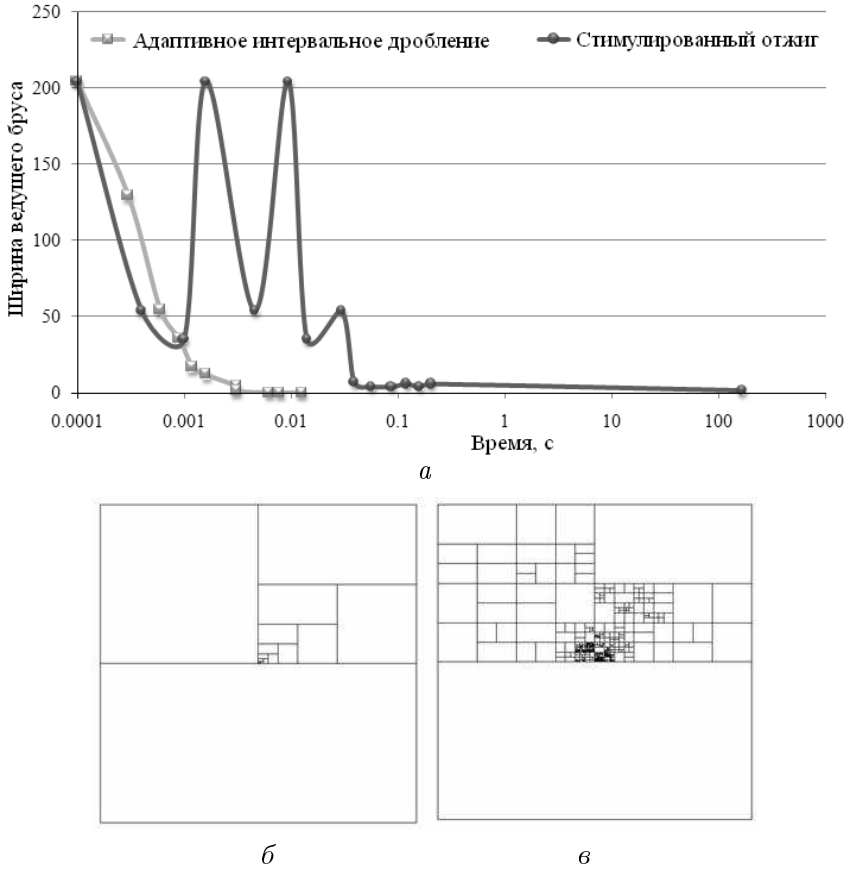


Рис. 10. Ширина интервальной оценки на ведущем брусе в зависимости от времени для алгоритма имитации отжига с “плохими” настройками и классическим алгоритмом интервального адаптивного дробления при поиске оптимума функции Растригин10 (*а*); процесс поиска минимума функции Растригин10, методом адаптивного интервального дробления, планомерное и безошибочное движение к глобальному минимуму (*б*); неудачные настройки метода отжига в процессе поиска минимума функции Растригин10, провоцирующие множество разрозненных дроблений (*в*)

рианте недостаточно высокая начальная температура или слишком быстрое охлаждение (т. е. уменьшение $T$) повышают вероятность того, что алгоритм сможет найти лишь локальный оптимум. В интервальном случае ситуация несколько лучше. Если система "охлаждается" очень быстро, то алгоритм становится классическим детерминистским методом адаптивного дробления. В противном случае, если температура высока и практически не понижается, получающийся алгоритм практически совпадает со случайным интервальным дроблением.

Наши вычислительные эксперименты показывают, что для различных типов целевых функций лучшие результаты достигаются на разных методах. Мы разработали стратегию управления температурой, достаточно успешно справляющуюся с различными целевыми функциями в отсутствие априорной информации об их характере. Этот способ хорошо зарекомендовал себя на различных типах задач и может использоваться как универсальный.

Суть его состоит в том, что вначале температура весьма высока, но быстро снижается до тех пор, пока доля отвергнутых неведущих брусов не станет примерно равной 20 %. Затем температура достаточно плавно понижается, пока не будет отвергаться более половины брусов, после чего скорость охлаждения опять возрастает. Если после понижения температуры до нуля ответ не был предоставлен за какое-то разумное время, мы рекомендуем снова увеличить температуру системы и заново повторить интервальную имитацию отжига.

## 3. Интервальные генетические методы

Далее разрабатывалась интервальная версия генетического алгоритма.

Использование идеи биологической эволюции для синтеза алгоритмов случайного поиска явилось, пожалуй, первой бионической идеей, которая была реализована немедленно и в многочисленных модификациях.

Для целей интервальной глобальной оптимизации нам показался наиболее удобным эволюционный подход, остановимся на нем подробнее.

Методы этого типа традиционно относятся к огромному пласту генетических алгоритмов. Строго говоря, генетическим алгоритмом следует называть лишь методы, оперирующие понятиями *ген* и *генетические операторы*. В данном контексте это означает, что вводится еще один уровень абстракции, позволяющий закодировать задачу таким образом, чтобы ее решение могло быть представлено в виде вектора ("хромосома"). Также отличительной особенностью классического генетического алгоритма является акцент на использование таких генетическиех операторов, как "скрещивание" (crossover) и "мутация" (mutation).

Выбранный же нами метод не использует понятия "ген" и не применяет генетических операторов, и является скорее алгоритмом эволюционного поиска, встречающимся в литературе также под названиями "эволюционное программирование" (Evolutionary Programming). Подробное описание его можно найти, например, в [16, 17]. Кратко, основные особенности метода состоят в следующем. В природе особи в популяции конкурируют друг с другом за такие ресурсы, как пища и вода. Кроме того, члены популяции часто конкурируют за привлечение брачного партнера. Те особи, которые наиболее приспособлены к окружающим условиям, будут иметь больше шансов выжить и произвести потомство. Слабо приспособленные особи не произведут потомства либо их потомство

будет очень немногочисленным. Эволюционные алгоритмы (как, впрочем, и все генетические алгоритмы с той или иной степенью детализации) используют прямую аналогию с таким механизмом. Они работают с популяцией — с совокупностью "особей", каждая из которых представляет возможное решение задачи. Каждая такая особь оценивается мерой ее "приспособленности" согласно тому, насколько "хорошо" соответствующее ей решение задачи. В природе это эквивалентно оценке того, насколько эффективен организм при конкуренции за ресурсы. Наиболее приспособленные особи получают возможность "воспроизводить" потомство. Эти идеи хорошо известны и успешно применяются. Мы распространили этот подход на интервальный случай.

Так, основными объектами — отдельными особями — в нашем случае выступают брусы, на которые дробится исходная область определения. А дробление — не что иное, как размножение, в терминах генетических алгоритмов. Причем "детей" всегда не менее двух (иначе интервальная оценка не улучшится, так как не произойдет уменьшения интересующей подобласти области определения целевой функции). Кроме того, можно заметить, что для размножения нужен лишь один родитель, который погибает в процессе (брусы-потомки полностью замещают родителя).

Для того чтобы наш алгоритм был по-прежнему доказательным (гарантирующим), т. е. сохранял отличительную черту интервальных методов, пришлось пересмотреть механизмы "гибели" особей. Так, они или рождаются нежизнеспособными (не выдерживают критериев отбраковки, применяемых сразу после дробления), или погибают в результате "эпидемий" — срабатывания уточненного критерия отбраковки.

Алгоритм во многом похож на уже описанные в настоящей статье — основным механизмом в нем является адаптивное дробление исходной области определения на все более мелкие подбрусы, означающее постепенное уточнение интервального расширения

*Основа интервального генетического алгоритма*

**Произвольным образом задается начальная популяция**
(Выбор начальной популяции практически никак не сказывается на дальнейшей работе алгоритма. Начальную популяцию можно создавать любым способом. Например, произвести несколько случайных дроблений исходной области определения. Этот шаг вообще можно пропустить, приняв брус исходной области определения за единственную существующую особь)

**Начальная популяция передается в основной цикл**

*Основной цикл*

{

1. **Вычислить значения функции приспособленности новорожденных особей** (этот шаг включает вычисление интервального расширения целевой функции по новым подбрусам, как необходимый для определения приспособленности подбруса).
2. $N$ **из наиболее приспособленных брусов с вероятностью** $P_n$ **оставляют от** $L_n$ **до** $U_n$ **потомков**.
3. $M$ **из неприспособленных брусов с вероятностью** $P_m$ **оставляют от** $L_m$ **до** $U_m$ **потомков**.
4. **Потомки проверяются на жизнеспособность** (применяются интервальные критерии отбраковки, описанные ранее).
5. **Если критерий отбраковки был улучшен, организуется "эпидемия"** (улучшенные критерии применяются ко всем особям).

}

функции и локализацию глобального оптимума. Предложенный метод принципиально отличается лишь стратегией выбора перспективного для уточнения оценки бруса.

В качестве меры приспособленности можно выбрать, например, нижнюю границу (для определенности рассматриваем ситуацию поиска минимума) интервальной оценки целевой функции на брусе. Чем она лучше (чем меньше нижняя граница), тем больше шансов брусу размножиться (быть раздробленным) и тем многочисленнее будет его потомство (брус будет раздроблен на бóльшее количество подбрусов).

В нашей реализации этой модели есть несколько параметров:

— вероятности $P_n$ и $P_m$;

— максимальное количество потомков $L_n$ и $L_m$;

— минимальное количество потомков $U_n$ и $U_m$;

— величина $N$: сколько объектов, начиная с самого приспособленного, могут оставить потомство;

— "разновесные" дети: брусы дробятся на равные части или разбиение происходит в случайной пропорции;

— количество особей начальной популяции.

Необходимо заметить, что если изменить алгоритм таким образом, чтобы потомки были равноправны и их было обязательно двое, но оставить потомство мог лишь "вожак", т. е. самый приспособленный, то в более привычных терминах это будет звучать как "на каждой итерации дробится брус с наименьшей нижней оценкой". Фактически получается классический алгоритм адаптивного интервального дробления, предложенный еще Муром, дополненный Скелбоу, обогащенный Хансеном и уже обсуждавшийся выше.

Таким образом, становится очевидно, что поведение алгоритма, его успешность зависят от этих параметров весьма существенно. Соответственно, встает вопрос о выборе оптимальных настроек алгоритма (в частности, правил задания вероятности и правил их модификации). Над этим мы работаем в настоящий момент. Общее предположение, подтверждающееся результатами наших экспериментов, следующее: чем проще функция, тем меньше должно быть особей, способных оставить потомство. Чем сложнее для оптимизации целевая функция (т. е. чем избыточнее интервальная оценка для нее, чем сильнее выражен эффект застаивания, чем больше ложных оптимумов), тем эффективнее множественное неравноправное потомство от многих родителей. Для более эффективной борьбы с застаиванием мы можем рекомендовать разрешить еще какому-то количеству плохо приспособленных брусов размножаться с некоторой вероятностью.

В качестве меры приспособленности в описанном алгоритме мы использовали нижнюю границу интервальной оценки функции на подбрусе (т. е. оценку оптимума на подбрусе снизу).

Исследования показывают, что если строить функцию приспособленности, учитывая еще и ширину интервальной оценки, и размер бруса, то на некоторых целевых функциях время работы может уменьшиться на треть.

Таким образом, функция приспособленности бруса (при поиске минимума) приобретает следующий вид:

$$\boldsymbol{F}(\boldsymbol{b}) = \sum_{i}^{n} \left( \alpha_i \cdot \mathrm{wid}(b_i) \right) + \beta \cdot \underline{\boldsymbol{f}(\boldsymbol{b})} + \gamma \cdot \mathrm{wid}(\boldsymbol{f}(\boldsymbol{b})). \tag{5}$$

Здесь $\boldsymbol{f}(\boldsymbol{b})$ — интервальная оценка целевой функции $f$ на брусе $\boldsymbol{b}$; $\underline{\boldsymbol{f}(\boldsymbol{b})}$ — нижняя граница интервальной оценки (оценка минимума снизу); $\mathrm{wid}(\boldsymbol{f}(\boldsymbol{b}))$ — ширина (точность)

интервальной оценки; $\mathrm{wid}(\boldsymbol{b}_i)$ — размер $i$-й стороны бруса; $\alpha_i$ — соответствующий весовой коэффициент.

Необходимо заметить, что поведение алгоритма опять-таки очень сильно зависит от величин указанных перапметров.

Наконец финальным шагом стала разработка самоподстраивающегося алгоритма, т. е. алгоритма, способного изменять коэффициенты $\alpha, \beta, \gamma$ во время своей работы в соответствии со спецификой задачи. На данный момент достигнуты следующие показатели. Максимальный проигрыш нашего адаптивного генетического алгоритма составляет порядка 13 % (проигрыш адаптивному дроблению при поиске оптимума константной целевой функции), в то время как на более сложных функциях с множеством локальных оптимумов новый интервальный генетический алгоритм в основном существенно быстрее прочих, описанных в настоящей работе.

## Заключение

Объединение интервальных и стохастических (рандомизированных) подходов позволило создать принципиально новые, эффективные методы для решения задачи глобальной оптимизации — одной из востребованных проблем современной вычислительной и прикладной математики. Результаты наших вычислительных экспериментов демонстрируют перспективность дальнейших исследований.

Автор выражает глубокую признательность научному руководителю — д-ру физ.-мат. наук Сергею Петровичу Шарому за постоянную помощь и полезные обсуждения и благодарит рецензента за внимательную рецензию, позволившую устранить допущенные неточности в изложении материала.

## Список литературы

[1] Васильев О.В., Аргучинцев А.В. Методы оптимизации в задачах и упражнениях. М.: Физматлит, 1999.

[2] Евтушенко Ю.Г. Численный метод поиска глобального экстремума функций (перебор на неравномерной сетке) // Журн. вычисл. математики и мат. физики. 1971. Т. 11, № 6. С. 1390–1403.

[3] Жиглявский А.А., Жилинскас А.Г. Методы поиска глобального экстремума. М.: Наука, 1991.

[4] Жилинскас А.А., Шалтянис В.Р. Поиск оптимума: компьютер расширяет возможности. М.: Наука, 1989.

[5] Törn A., Ali M., Viitanen S. Stochastic global optimization: Problem classes and solution techniques // J. of Global Optimization. 1999. Vol. 14, N 4. P. 437–447.

[6] Hansen E., Walster G. Global optimization using interval analysis. N. Y.: Marcel Dekker, 2004.

[7] Moore R. Methods and Applications of Interval Analysis. Philadelrhia: SIAM, 1979.

[8] Ding-Zhu Du, Pardalos P., Weili Wu. Mathematical Theory of Optimization. Dordrecht: Kluwer, 2001.

[9] Kearfott R. Rigorous Global Search: Continuous Problems. Dordrecht: Kluwer, 1996.

[10] Панов Н.В., Шарый С.П. Стохастические подходы в интервальных методах глобальной оптимизации // Всероссийское (с международным участием) совещание по интервальному анализу и его приложениям (ИНТЕРВАЛ-06), 1–4 июля 2006 года, Петергоф, Россия. Расширенные тезисы докладов. СПб.: ВВМ, 2006. С. 101–105.

[11] Шарый С.П. Стохастические подходы в интервальной глобальной оптимизации // Труды XIII Байкальской международной школы-семинара "Методы оптимизации и их приложения", Иркутск—Северобайкальск, 2–8 июля 2005. Иркутск, ИСЭМ СО РАН, 2005. Т. 4. С. 85–105.

[12] Шарый С.П. Рандомизированные алгоритмы в интервальной глобальной оптимизации // Сиб. журн. вычисл. мат. 2008. Т. 11, № 4. С. 457–474.

[13] Janson C., Knuppel O. A global minimization method: The multi-dimensional case // Technical Report 92.1. Forschungsschwerpunkt Informations- und Kommunikationstechnik. TU Hamburg-Harburg, 1992.

[14] Панов Н.В., Колдаков В.В. Программный комплекс для графического представления процесса и результатов работы интервальных алгоритмов // Труды 5-й международной конференции памяти академика А.П. Ершова "Перспективы систем информатики". Новосибирск, 2003. Т. 1. Международное совещание по интервальной математике и методам распространения ограничений. С. 38–46.

[15] Metropolis N., Rosenbluth M. et al. Equations of state calculations by fast computing machines // J. Chem. Phys. 1953. Vol. 21. P. 1087.

[16] Koza J. Genetic Programming: on the Programming of Computers by Means of Natural Selection. MIT Press, 1992.

[17] Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы. М.: Физматлит, 2006.

*Поступила в редакцию 25 декабря 2008 г.,*
*в переработанном виде — 15 мая 2009 г.*