

УДК 519.642

МИНИМИЗАЦИЯ НЕЛИНЕЙНОГО ФУНКЦИОНАЛА НЕВЯЗКИ В ЗАДАЧАХ ПОТОКОВОЙ ОБРАБОТКИ ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ

Б. Г. Шпынёв¹, А. Л. Воронов¹

Рассматривается задача минимизации нелинейного функционала невязки, возникающая при обработке длинных рядов экспериментальных данных, когда измеряемый процесс описывается нелинейным интегро-дифференциальным или интегральным уравнением. Для задач подобного типа рассматриваются три взаимосвязанных метода решения: быстрый алгоритм спуска, метод регуляризации решения и поиск оптимального решения среди множества субоптимальных. Центральной частью метода является алгоритм спуска, работающий на сетке, образованной вершинами многомерных политопов. Регуляризация решения задачи проводится путем перехода для минимизации в функциональное пространство Соболева. Для устранения неоднозначности, связанной с наличием субоптимальных решений, используется специальная схема адаптации, работающая по принципу генетических алгоритмов и использующая результаты ранее проведенной обработки. Работа выполнена при поддержке Министерства образования и науки РФ (ГК № 14.518.11.7065) и гранта РФФИ № 11-05-00822-а.

Ключевые слова: нелинейные интегро-дифференциальные уравнения, метод спуска, регуляризация, генетические алгоритмы.

1. Введение. Обратные задачи часто возникают в экспериментальных научных приложениях, связанных с определением параметров процесса, результат действия которого измеряется в виде функций времени или частоты. В общем случае такие задачи выражаются в виде интегро-дифференциальных уравнений, которые описывают теоретическую модель физического процесса. Такие теоретические модели обычно имеют семейство решений, поэтому возникающие обратные задачи являются некорректными и требуют задания граничных и начальных условий. В настоящей статье рассматривается класс обратных задач, возникающих при решении интегро-дифференциальных уравнений второго порядка, которые часто встречаются на практике и для которых граничные условия, как правило, известны.

Наиболее детально исследованы обратные задачи, описываемые дифференциальными уравнениями в частных производных [18]. Для большей части таких задач возможен переход к системам обыкновенных дифференциальных уравнений или построение асимптотических решений. Среди интегральных уравнений наиболее полно исследован класс уравнений с линейным оператором (уравнения Фредгольма 1-го и 2-го рода и уравнения Вольтерра 1-го и 2-го рода) [1, 4, 6, 16, 22]. Для более общего случая, когда оператор интегрального уравнения сам зависит от искомой функции или ее производных, нелинейные обратные задачи, за единичными исключениями [7, 15], не упрощаются. Их решения ищутся в рамках задачи о наименьших квадратах, прямыми численными методами, с помощью разнообразных алгоритмов спуска в функциональном пространстве модели со специальными методами регуляризации [1, 22]. Кроме методов спуска, для решения нелинейных обратных задач используются эволюционные методы [20], в частности методы Монте-Карло [9], и разнообразные генетические алгоритмы [8, 5].

Особо отметим принципиальную трудность обратных задач с нелинейным оператором, которые рассматриваются в настоящей работе. Во-первых, задачи с нелинейным оператором не могут быть сведены к матричным задачам, как это возможно для уравнений Фредгольма и Вольтерра, т.е. для поиска решения всегда необходимы многократные прямые вычисления функционала невязки на каждой итерации. Во-вторых, нелинейные функционалы, как правило, имеют множественные локальные минимумы, поэтому задача минимизации должна включать в себя дополнительный алгоритм выбора глобального минимума среди множества субоптимальных. Это особенно критично, если ядро оператора обратной задачи зависит от периодических функций. По указанным причинам мы рассматриваем задачу минимизации с нелинейным оператором как три взаимосвязанных метода: метод спуска, метод регуляризации решения и метод поиска глобального минимума среди множества субоптимальных.

¹ Институт солнечно-земной физики Сибирского отделения РАН (ИСЗФ СО РАН); ул. Лермонтова, 126-а, а/я 291, 664033, г. Иркутск; Б. Г. Шпынёв, вед. науч. сотр., email: shpynev@iszf.irk.ru; А. Л. Воронов, вед. инженер-программист, e-mail: raven@iszf.irk.ru

Поскольку нелинейный оператор обратной задачи может зависеть от большого числа параметров, то основной проблемой методов спуска является необходимость вычисления большого числа (от $2n$ до 2^n) значений функционала на каждой итерации, чтобы обеспечить возможность выбора оптимального направления поиска решения. Для статистических методов требуется генерация статистически значимой выборки значений функционала, что требует еще большего числа вычислений. По этой причине исследователи, решающие обратные задачи, стремятся минимизировать число параметров задачи, вводя аналитические приближения для производных в методах градиентного спуска [14] или используя аппроксимацию сложных функций более простыми аналогами, которые допускают аналитическое упрощение задачи [22].

Методы градиентного спуска обладают хорошим быстродействием в задачах с малым числом параметров, однако становятся громоздкими и неустойчивыми при увеличении их числа. Основные сложности в этих методах — это корректное определение частных производных в шумах и неоднозначность интерпретации нуля производной, который может указывать как на минимум, так и на точку перегиба функции по параметру. В этой связи возникают проблемы с выработкой критерия останова алгоритма. Проблема наличия субоптимальных решений в большинстве методов градиентного спуска, как правило, не рассматривается в общем виде, т.е. эти методы не могут использоваться в их исходном виде для глобальной минимизации нелинейного функционала невязки.

Методы Монте-Карло и генетические алгоритмы, напротив, хорошо адаптированы к глобальной минимизации нелинейных функционалов, однако требуют расчета очень большого, статистически значимого числа случайных реализаций модели. Как следствие, для их работы требуются специальные алгоритмы генерации статистически независимых выборок, они нуждаются в больших вычислительных ресурсах и практически реализуются только на суперкомпьютерах [9].

Во многих экспериментальных приложениях часто возникает особый тип нелинейных обратных задач, позволяющий расширить возможности методов спуска, не прибегая к расчету статистически значимого числа значений функционала невязки. В этих задачах минимизация проводится для однородного (по структуре) потока экспериментальных данных, состоящих из отдельных реализаций измеряемого физического процесса. В качестве примера можно привести задачу радиоинтерферометрии Солнца или звездных источников, когда необходимо исключить из получаемого стандартного изображения влияние диаграммы направленности антенной системы, которая, в свою очередь, определена с погрешностью и нестационарна. Такого же типа задачи возникают в экспериментах по дистанционному зондированию атмосферы, океана и земной поверхности в акустическом и радио диапазонах, в ядерной физике при обработке данных ускорителей. Характерной особенностью данного типа задач является то, что в каждый момент времени известны решения обратной задачи, полученные в предыдущих сеансах обработки.

Очевидно, что в задачах потоковой обработки данных минимизацию можно проводить известными методами типа Монте-Карло для каждой реализации процесса. Однако на практике это часто невозможно по причине недостаточной скорости вычислительных устройств. Производительность компьютеров в настоящее время уже приближается к своему физическому пределу. Поэтому в условиях, когда точность научных измерений возрастает и поток данных многократно увеличивается, проблема потоковой обработки в реальном масштабе времени часто становится неразрешимой задачей в рамках традиционных математических подходов.

В настоящей работе для класса задач потоковой обработки данных мы предлагаем общий подход для глобальной минимизации нелинейного функционала невязки, основанный на использовании информации о ранее обработанных реализациях исследуемого физического процесса. Предлагаемый нами гибридный метод является быстрым алгоритмом спуска, в котором существенным образом используется информация о характеристиках процесса, полученная на основе предшествующих результатов обработки.

Указанные выше три составные части процесса минимизации разработаны в настоящем подходе на основе следующих принципов. Метод спуска разработан как многомерное обобщение использованного нами ранее [12] быстродействующего алгоритма фитирования. В этом алгоритме направление спуска в задаче с тремя параметрами определялось по значениям невязки в вершинах тетраэдра, центром которого является текущая точка проекционного пространства. Поскольку существуют многоугольники (политопы), которые можно обобщить на произвольную конечную размерность, в данной работе указанный метод спуска был применен к задачам с произвольным числом параметров. Требуемая в таких задачах регуляризация решения осуществляется путем перехода в функциональное пространство Соболева. Для обеспечения глобальной сходимости метода рассматривается специальная схема генерации начальных условий, использующая элементы генетических алгоритмов.

При решении каждой конкретной обратной задачи нами предполагается интерактивный метод конструирования алгоритма. Иными словами, выбор размерности проекционного пространства, выбор регу-

ляризирующих коэффициентов задачи, учет специфики исследуемых функций и граничных условий проводится путем последовательных приближений. Несмотря на то что для многих параметров задачи можно применять аналитические оценки, окончательное решение о характеристиках алгоритма принимается на основе визуального анализа результатов с учетом требований к гладкости решений и их физического смысла. То же касается и вопроса глобальной оптимизации, т.е. для окончательной адаптации алгоритма к потоковой обработке требуется ручная настройка в зависимости от специфики задачи.

2. Математическая формулировка задачи. Рассмотрим нелинейную обратную задачу для дважды непрерывно дифференцируемой функции f , определенной на $[a, b]$ и ограниченной вместе со своими производными f' и f'' на $[a, b]$. Такие задачи возникают в физических приложениях, когда процессы описываются дифференциальными уравнениями в частных производных второго порядка. Естественной областью определения этих уравнений является пространство Соболева $W_2^{(2)}$ [21], которое обладает свойствами, способствующими регуляризации решения обратной задачи. Из теоремы о вложении Соболева также следует, что $f \in C_2$.

Рассматривается операторное уравнение, описывающее некоторый физический процесс $Af = u$, $A : W_2^{(2)} \rightarrow L_2$. Результатом действия нелинейного оператора A на функцию f является функция $u \in L_2$, которую можно измерить в эксперименте. Обратная задача состоит в нахождении f . Функция $u(x)$ измеряется в присутствии случайных шумов, которые вместе с методическими погрешностями вычислений оператора задают ошибку $\varepsilon > 0$. Для большинства задач эта ошибка совпадает по порядку величины с дисперсией σ . По условию задачи известны N решений задачи $f_{-1}, f_{-2}, \dots, f_{-N}$, полученных в предыдущих сеансах обработки. Иными словами, фактически это класс нелинейных обратных задач, когда имеется несколько сотен (тысяч) уравнений с одинаковым ядром и разными правыми частями.

Задачи такого типа решаются методом регуляризации А. Н. Тихонова [22]. Вопросы сходимости, единственности решения и приближений для оператора с анализом точности подробно изложены в работе В. А. Морозова [19], поэтому мы на них останавливаться не будем и перейдем непосредственно к анализу функционала Тихонова для нелинейной обратной задачи в виде

$$\Phi = \|Af - u\|_{L_2}^2 + \alpha \|\Omega(A, f, u, f_{-1, -2, \dots, -N}, \varepsilon)\|_{L_2}^2. \tag{1}$$

Здесь $\alpha > 0$ — параметр регуляризации и Ω — оператор регуляризации, определяющий свойства функционального пространства модели и зависящий от априорной информации, которая включает в себя граничные условия, информацию о гладкости функции f , а также полученные ранее решения задачи. Поскольку естественным пространством для рассматриваемых обратных задач является пространство Соболева $W_2^{(2)}$, мы будем использовать оператор регуляризации в виде

$$\Omega = \sum_{k=0}^2 \int_a^b p_k(A, f, u, f_{-1, -2, \dots, -N}, \varepsilon) |f^{(k)}|^2 dx.$$

Весовые функции p_k определяют зависимость регуляризующей добавки от граничных условий и другой априорной информации. Параметр регуляризации α формально может быть включен множителем в весовые функции, поскольку они предполагаются независимыми друг от друга и могут быть произвольными. Однако чтобы сохранить традиционный смысл параметра регуляризации, мы его оставим в формулах.

Для функций f и u , не равных тождественно нулю, функционал (1) всегда положителен. Переход в функциональное пространство $W_2^{(2)}$ приводит к тому, что в минимумах функционала Φ находятся самые гладкие решения f . Для реальных физических систем это соответствует требованию вариационного принципа, когда функция, описывающая истинное движение, имеет минимальную вариацию (интегральную производную). Поэтому поиск минимума функционала Φ автоматически означает поиск наиболее гладкого и физически обоснованного решения. То, что интегральные производные эффективны при обработке экспериментальных данных, доказывает их широкое распространение в обработке цифровых изображений [10, 13, 2].

Функция p_0 позволяет ограничивать класс решений по норме. Однако в большинстве задач минимизации равенство норм модели и эксперимента является обязательным условием для расчета невязки, так как сравнивать функции с разной нормой бессмысленно. Поэтому в рассматриваемых задачах мы также потребуем обязательную нормировку:

$$\|Af\|_{L_2} = \|u\|_{L_2} = 1. \tag{2}$$

Условие (2) дает $p_0 = 0$. С учетом введенных обозначений и условий обратная задача ставится как задача минимизации функционала \mathfrak{F} методом наименьших квадратов в L_2 :

$$\mathfrak{F} \equiv \min_{L_2} \left(\int_a^b |Af - u|^2 dx + \alpha \int_a^b p_1 |f'|^2 dx + \alpha \int_a^b p_2 |f''|^2 dx \right). \quad (3)$$

3. Выбор параметров регуляризации. Выбор функций регуляризации является сложной проблемой, которая критически зависит от априорной информации. В наиболее общем виде эта проблема сформулирована в виде принципа обобщенной невязки в работах [19, 22]. Поскольку в рассматриваемом классе обратных задач функция f принадлежит компакту и имеет непрерывные и ограниченные производные, а минимизация интегральных производных исключает сингулярности решения, мы можем использовать несколько эвристических соображений для построения функций регуляризации для таких задач.

В этом разделе мы рассмотрим один из наиболее очевидных способов построения функций регуляризации, который может быть практически реализован для многих обратных задач. Он основан на классическом подходе Тихонова, в котором предполагаются два основных свойства функционала невязки. Первое — вклад регуляризующих добавок в функционал невязки должен быть мал относительно основной невязки. Второе — решение должно быть устойчивым к малым изменениям исходной информации, т.е. функции $u(x)$.

С точки зрения минимизации в пространстве $W_2^{(2)}$, можно выделить несколько основных принципов конструирования функций регуляризации.

1. Влияние регуляризации должно максимально сократить число локальных минимумов и сделать долину оптимального решения в $W_2^{(2)}$ достаточно глубокой, чтобы уменьшить влияние шумов. Иными словами, вклад регуляризующей добавки должен быть больше или хотя бы сопоставимым с вкладом шумов.

2. Вклад производных в функционал (1) не должен превышать некоторого порога λ_{\min} , при котором минимизация интегральных производных начинает отфильтровывать высокочастотную часть спектра полезного сигнала.

3. Вклад производных в невязку должен быть тем больше, чем меньше мощность измеряемого процесса $u(x)$. Это связано с тем, что при низком уровне сигнала решение становится неустойчивым вследствие случайных шумов. Если в этом случае увеличить вклад производных в функционал невязки, то в рассматриваемой области определения функции будут использоваться максимально гладкие решения.

4. По условию задачи нам известны значения функций f_{-1} , f'_{-1} и f''_{-1} , полученные в предыдущем сеансе обработки. Эта информация также может быть использована при конструировании функций регуляризации.

Для построения функций p_1 и p_2 воспользуемся следующими оценками.

Если функция f увеличивается на величину σ всюду на компакте, то основная невязка в (3) увеличивается на величину $\Delta_\sigma = \sigma^2(b-a)$.

Величина Δ_σ является оценкой, характеризующей информационный порог обратной задачи. Она определяет относительную долю шумов в общей невязке и может быть использована для задания абсолютной величины параметра регуляризации α .

Функции p_1 , p_2 должны быть обратно пропорциональны $u(x)$, поэтому, учитывая нормировку (2), можно записать: $p_1, p_2 \sim \frac{b-a}{|u| + \sigma}$. Добавка σ в знаменателе не позволяет ему обратиться в нуль. Вместо σ может быть использовано другое положительное число, ограничивающее величину функций регуляризации при малых значениях $|u|$.

Для контроля абсолютной величины регуляризующих членов уравнения (3) необходимо задать нормы интегральных производных. Для рассматриваемого класса задач можно применить два способа нормировки.

Первый способ удобен при отсутствии априорной информации о свойствах функций f , f' , f'' , например при первом запуске алгоритма. Если искомая функция f (или составляющие ее функции f_k) имеет единственный экстремум, то ее интегральная производная равна $2\|f\|_1$. Эта величина может быть использована в качестве нормы интегральной производной. Тогда функция с двумя экстремумами будет иметь интегральную производную в интервале 1–2, для функции с тремя экстремумами получим значение 1–3 и т.д. Как минимум, данная нормировка не даст регуляризующей добавке меняться на порядок величины.

Второй способ эффективен в условиях потоковой обработки, когда для нормировки можно использо-

вать норму производной $\|f'_{-1}\|$, полученную в предыдущем сеансе обработки. В этом случае не требуется вычислять величину $2\|f\|_1$ на каждой итерации. Аналогичную нормировку можно ввести для интеграла от второй производной: $2\|f'\|_1$ или $\|f''_{-1}\|$.

Как уже говорилось, регуляризирующая добавка должна быть мала по сравнению с основной невязкой. Однако она также должна быть больше информационного порога Δ_σ , иначе ее влияние на функционал может оказаться недостаточным для регуляризации. Из статистических соображений трехкратное превышение регуляризирующей добавки над Δ_σ должно обеспечить гарантированный вклад интегральных производных в общую невязку. Тогда, учитывая нормировки для f' , f'' и u , можно положить $\alpha = 3\Delta_\sigma$ и использовать функции регуляризации в виде

$$\alpha p_1 = 3\Delta_\sigma \frac{b-a}{|u|+\sigma} \frac{1}{\|f'_{-1}\|^2} = \frac{3\sigma^2(b-a)^2}{\|f'_{-1}\|^2} \frac{1}{|u|+\sigma}, \tag{4}$$

$$\alpha p_2 = 3\Delta_\sigma \frac{b-a}{|u|+\sigma} \frac{1}{\|f''_{-1}\|^2} = \frac{3\sigma^2(b-a)^2}{\|f''_{-1}\|^2} \frac{1}{|u|+\sigma}. \tag{5}$$

В общем случае вместо коэффициента 3 в выражениях (4) и (5) могут использоваться другие коэффициенты, которые подбираются для каждой задачи исходя из априорной информации. При использовании функций регуляризации в потоковой обработке под знаками интеграла в (3) остаются только вторые дроби из (4) и (5). Первые (нормировочные) члены будут постоянными в процессе минимизации. Использование функций регуляризации в виде (4) и (5) позволяет, с одной стороны, достаточно жестко контролировать вклад регуляризации в общую невязку, т.е. условие его малости. С другой стороны, обеспечивается адекватный вклад регуляризирующей добавки по сравнению с шумами.

Если f является комбинацией более простых функций f_k , то интегральные производные считаются для каждой из них в отдельности, а вклад их в общую невязку регулируется исходя из степени гладкости этих функций, которая обычно известна априори. Желательно, чтобы вклад производных каждой функции в невязку был больше Δ_σ , но общий вклад должен быть мал по сравнению с основной невязкой. Если дисперсия исходных данных слишком велика (сигнал в шумах), вклад регуляризирующих членов в (3) нужно дополнительно регулировать, уменьшая параметр α .

Рассмотренный метод регуляризации не исключает применения штрафных функций, однако их применение должно быть согласовано с общим принципом построения функций регуляризации. Так, если в задаче требуется ограничить минимальный порог интегральной производной λ_{\min} для предотвращения фильтрации полезного сигнала в процессе минимизации, то при уменьшении интегральной производной ниже данного порога можно использовать штрафную функцию в виде

$$\Delta_\sigma \int_a^b \left| f' - \frac{\lambda_{\min}}{b-a} \right|^2 \|f'_{-1}\|^{-1} dx. \tag{6}$$

Однако, по нашему опыту, следует избегать, по возможности, применения штрафных функций, поскольку их применение нарушает гладкость функционала Φ и иногда приводит к ошибочной остановке алгоритма спуска вблизи такой границы. Как будет показано далее, гораздо более эффективным является правильный выбор начальных условий для алгоритма спуска.

Еще раз отметим, что выбор параметров регуляризации по формулам (4) и (5) носит чисто эвристический характер. Поэтому в случае, когда данные оценки не дают желаемого результата при решении конкретной задачи, мы рекомендуем заменять их более приемлемыми аналогами, разработанными на основе принципов обобщенной невязки [19, 22].

4. Выбор проекционного пространства. Практическая реализация любого алгоритма решения обратной задачи предполагает ограничение на объем вычислений, что автоматически ограничивает число параметров функционала (1) и шаг изменения параметров в проекционном пространстве. В реальных обратных задачах используется конечномерный проектор $P_n: W_2^{(2)} \rightarrow R^n$, сводящий задачу (3) к минимизации в конечномерном пространстве с параметрами y_n . При этом функционал (1) проецируется в конечномерный функционал $\Phi_n = P_n \Phi$.

Поскольку для стандартного вычисления частных производных по параметрам задачи при размерности n требуется от $2n$ до 2^n вычислений функционала (1) на каждом шаге, то возникает необходимость минимизировать число параметров. Для этого модель $f(x)$ обычно стремятся свести к функции небольшого числа параметров, входящих в некоторое аналитическое выражение.

Однако $f(x)$ может также быть произвольной функцией, не имеющей конкретного аналитического или параметрического представления. Так как проекционное пространство должно быть конечномерным, то в данном случае функция $f(x)$ должна быть представлена аппроксимацией полиномами или сплайнами с конечным числом коэффициентов, которые будут играть роль параметров минимизации. Если $f(x)$ имеет достаточно широкий спектр, то число параметров детальной аппроксимации может оказаться довольно большим (например, 30–100), и тогда процесс минимизации может стать серьезной проблемой из-за недостаточной скорости вычислительных средств. Тем не менее, мы осознанно рассматриваем возможность выбора проекционного пространства с большим числом параметров, поскольку такие возможности дает рассматриваемый далее метод спуска. Цель данного раздела — выбрать параметры проекционного пространства, оптимальные для конкретной задачи.

Введенная выше погрешность $\varepsilon > 0$ позволяет использовать ее в качестве нижней оценки для задания ε -сети покрытия, которое задается в функциональном пространстве и используется для определения ε -энтропии и ε -емкости компакта. Эта теория подробно изложена в работах А. Н. Колмогорова [17]. Минимальное число N_ε точек ε -сети определяется из выражения:

$$\log N_\varepsilon \approx \frac{|b - a|L}{\varepsilon}, \tag{7}$$

где константа L является оценкой сверху производной $f'(x)$, которая выражается условием Липшица $|f(x) - f(x')| \leq L|x - x'|$. В пространстве D_K конечномерных ограниченных действительных функций обычно определена норма $\|f\| = \sup_{x \in K} |f(x)|$.

Для функций из D_K вводится понятие гладкости, т.е. функции имеют гладкость $q > 0$ ($q = p + \alpha$, p — целое, $0 < \alpha \leq 1$), если для любых векторов $x \in K$ и $x + h \in K$ имеем

$$f(x + h) = \sum_{k=0}^p \frac{1}{k!} B_k(h, x) + R(h, x), \tag{8}$$

где $B_k(h, x)$ по h — это однородная форма степени k

$$\|R(h, x)\| \leq C \|h\|_0^q, \tag{9}$$

C — некоторая константа. При выполнении (8) и (9) для проекционного пространства с размерностью n оценка N_ε будет иметь вид $\log N_\varepsilon \approx \left(\frac{1}{\varepsilon}\right)^{n/q}$.

Наличие ε -сети и соответствующего ε -коридора в проекционном пространстве расширяет область допустимых решений задачи (3), так как позволяет использовать в качестве решения разные способы аппроксимации функции $f(x)$ полиномами или сплайнами с конечным рядом коэффициентов. Эта технология широко применяется на практике и строго обоснована для используемого класса обратных задач [19]. Если решение может быть представлено в виде комбинации более простых функций, то аппроксимация может быть применена к каждой из таких функций. Критерием применимости приближенного решения будет условие $\|\Phi_n(x, f(x)) - \Phi_n(x, f_1(x), f_2(x), \dots, f_k(x))\| < \varepsilon$. Тогда обратная задача сводится к задаче минимизации функции $n = k \times m$ параметров, которые составляют образ процесса:

$$I_m^k = \begin{cases} f_1^1, f_2^1, \dots, f_m^1 \\ f_1^2, f_2^2, \dots, f_m^2 \\ \dots\dots\dots \\ f_1^k, f_2^k, \dots, f_m^k \end{cases} \tag{10}$$

Способ аппроксимации зависит от специфики задачи. Предпочтительным будет представление $f(x)$ в виде комбинации медленно меняющихся функций $f_1(x), f_2(x), \dots, f_k(x)$ и отдельная аппроксимация каждой из таких функций. В этом случае интегралы от производных в (2) можно представить в виде комбинации интегралов от производных $f_1'(x), f_2'(x), \dots, f_k'(x)$ и задавать необходимую степень гладкости для каждой из функций разложения, учитывая априорную информацию, которая часто имеется для таких функций. Для аппроксимации медленно меняющихся функций удобно использовать разложение через локальные сплайны [3], которые определены на конечных интервалах изменения аргументов и делают процедуру спуска более устойчивой. Адекватность способа аппроксимации в каждой конкретной

задаче оценивается разработчиком алгоритма в тестовых испытаниях с использованием дополнительной априорной информации путем визуального или численного контроля.

На практике каждый из параметров модели может изменяться со своим собственным шагом, который можно определить на основе (7) в виде $\delta_i \approx \exp\left(\frac{\varepsilon}{(b_i - a_i)L_i}\right)$. Здесь индексами i обозначены области определения Φ_n по каждой функции разложения и соответствующие верхние оценки производных. Для удобства описания алгоритма спуска далее мы будем использовать нормированные величины параметров $y_i^j = \frac{f_i^j}{\delta_j}$, т.е. покрытием будет ε -сеть с единичным шагом.

5. Метод спуска. В предлагаемом подходе мы используем многомерный аналог алгоритма спуска, описанного нами в работе [12], который дополнен приведенным выше методом регуляризации и генетическим алгоритмом, обеспечивающим глобальную сходимость.

В рассматриваемом методе спуска мы отходим от традиционного расчета частных производных по параметрам. На каждой итерации невязка вычисляется в вершинах правильного политопа, центром которого является текущая точка проекционного пространства. Поскольку существуют только три правильных политопа с размерностью $n \geq 5$ (симплекс, гиперкуб и ортоплекс), мы рассмотрим применение данного способа перемещения в проекционном пространстве только для этих трех объектов. В табл. 1 приведены соответствующие схемы узлов политопов и число вычислений z невязки на каждом шаге. В центре политопов находится текущая точка проекционного пространства, из которой происходит движение алгоритма в вершину с наименьшей невязкой. В двумерном случае вершины этих многоугольников образуют равномерные сетки.

Таблица 1

Правильный политоп	Симплекс	Гиперкуб	Ортоплекс
Схема вершин для $n=2$			
Число вычислений Φ_n	$Z = n + 1$	$Z = 2^n$	$Z = 2n$
Геометрический коэффициент пути	$\frac{4}{3} - \frac{\sqrt{3}}{18} \cong 0.9484$	$\frac{\sqrt{2}}{2}$	1

Очевидно, что минимальное число вычислений требуется для симплекса; если число параметров n велико, то количество вычислений Φ_n в симплицальном методе почти вдвое меньше, чем в методе, основанном на ортоплексе. Число вычислений в методе, основанном на сетке вершин гиперкубов, растет по экспоненциальному закону, поэтому в многомерном случае ее использование крайне неэффективно и дальше эта сетка рассматриваться не будет.

Общая эффективность рассматриваемых политопов применительно к задаче минимизации также зависит от геометрических свойств образуемых ими n -мерных пространств. Рассмотрим количество шагов (итераций) s , необходимых для достижения точки минимума функционала в симплицальном методе и в ортоплексном методе. Для традиционного способа координатного спуска, в котором применяется именно ортоплекс (городская метрика), путь от точки старта до точки оптимального решения всегда равен сумме путей по каждому из параметров. В симплицальном комплексе практически всегда реализуется одно-временное (диагональное) движение по нескольким переменным. За счет этого путь в симплицальном комплексе всегда короче, чем путь в городской метрике. Геометрические коэффициенты пути в разных метриках также приведены в табл. 1.

В двумерном случае путь по сотовой структуре (сетке из вершин двумерных симплексов) может отличаться от пути вдоль прямоугольной сетки в соотношении $K_{\text{way}}^2 = \frac{s_{\text{sim}}}{s_{\text{ort}}} = \frac{4}{3} - \frac{\sqrt{3}}{18} \cong 0.9484$. На первый взгляд это отличие выглядит не столь существенным, однако при использовании многомерной сетки коэффициент сокращения пути может составить $K_{\text{way}}^n = (0.9484)^{n-1}$. Иными словами, получаемый эффект “смешивания переменных” коренным образом изменяет вычислительные характеристики алго-

ритма спуска. На рис. 1 приведены значения произведения zK_{way}^n в зависимости от размерности задачи. Так, для задачи с 50 параметрами общее число вычислений в симплицеальном комплексе и в городской метрике может отличаться до 25 раз. При размерности задачи больше двадцати эффект перемешивания переменных преобладает над эффектом увеличения размерности и среднее число вычислений на каждую итерацию даже уменьшается.

Симплициальный метод имеет подавляющее преимущество по скорости спуска при большом числе параметров функционала невязки. Поэтому оптимальным методом минимизации из заданной точки P_n будет алгоритм, производящий $n + 1$ вычислений функционала Φ_n в вершинах соответствующего n -мерного симплекса Δ^n и сдвигающий начало координат в вершину с наименьшим значением Φ_n . Критерием останковки алгоритма является минимум Φ_n в центре симплекса по сравнению со значениями во всех вершинах.

Программный код, реализующий данный алгоритм, будем называть *симплекс-процессором* (СП). Блок-схема алгоритма приведена на рис. 2. На вход СП поступает начальный набор параметров $I_0 = y_1, y_2, \dots, y_n$, на выходе СП будет набор $I_{\min} = y'_1, y'_2, \dots, y'_n$ параметров оптимального или ближайшего субоптимального решения задачи (3).

Алгоритм СП предельно прост и легко реализуется на языках высокого уровня.

Правильный $(n + 1)$ -угольник (симплекс) строится из симплекса размерности $n - 1$ по следующему алгоритму: для симплекса размерности $n = 2$, т.е. равностороннего треугольника с центром в начале координат, задается высота $h_2 = \sqrt{3}/2$ и координаты его вершин $(-1/2, -h_2/3)$, $(1/2, -h_2/3)$, $(0, 2h_2/3)$.

Далее в цикле выполняются следующие операции.

1. Для симплекса размерности $n + 1$ рассчитывается новая высота $h_n = \sqrt{1 - h_{n-1}^2}$.
2. Симплекс размерности $n - 1$ сдвигается на $-h_n/n$ вдоль новой переменной x_n .
3. Добавляется новая точка с координатами $(0, 0, 0, \dots, 1 - h_n/n)$.

Когда координаты вершин заданы, производится расчет значений Φ_n в этих вершинах и определяется вершина с наименьшей невязкой, куда переносится текущая точка проекционного пространства. Для описания n -мерного симплекса необходим двумерный массив с размерностью $(n + 1; n + 1)$. Для рассмотренного ниже примера программный код СП на языке Delphi составляет около 30 строк, в то время как алгоритм расчета целевой функции содержал порядка 600 строк текста.

Специфические свойства симплицеального комплекса делают симплекс-процессор удобным инструментом для решения многомерных обратных задач, поскольку в рассматриваемом подходе этот алгоритм снимает принципиальные ограничения на размерность проекционного пространства. Увеличение числа параметров минимизации существенно не уменьшает, а даже может увеличивать эффективную скорость работы алгоритма спуска. В этом подходе возможен широкий выбор методов аппроксимации решения задачи (3), что дает новые возможности в постановке и численном решении нелинейных задач математической физики. Переход для минимизации в $W_2^{(2)}$ в таких

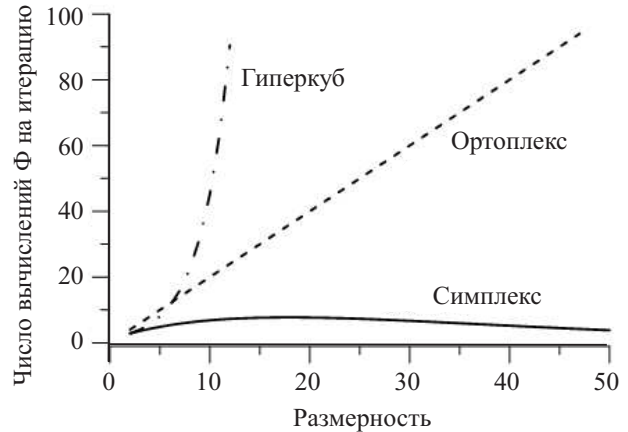


Рис. 1. Число вычислений функционала Φ_n на итерацию для разных политопов

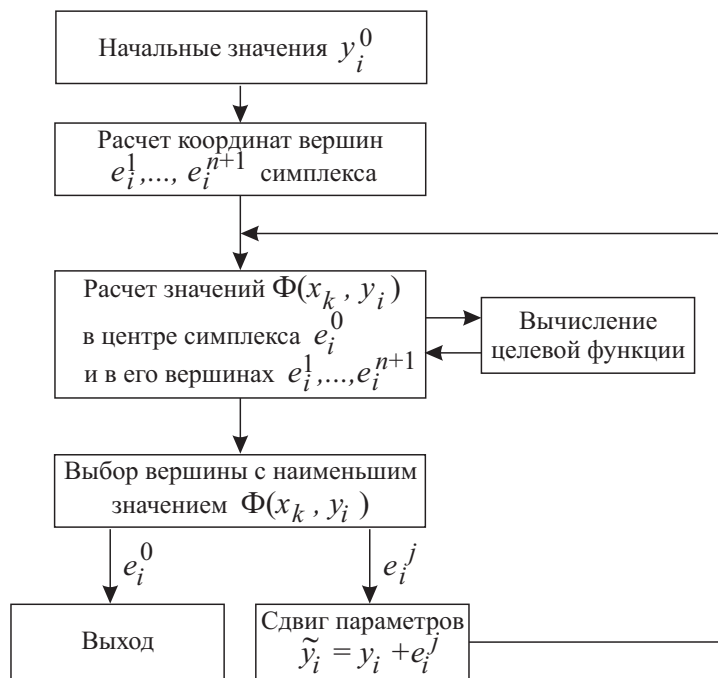


Рис. 2. Блок-схема симплекс-процессора

задачах будет выполнять функцию контроля за соблюдением вариационного принципа и исключит вероятность получения сингулярных решений. Если ядро оператора обратной задачи является линейным, то симплекс-процессор может применяться самостоятельно без дополнительных алгоритмов глобальной оптимизации.

6. Поиск оптимального решения. Как уже говорилось, если ядро оператора обратной задачи является нелинейным, то функционал $\Phi_n(x_k, y_i)$ имеет множественные локальные минимумы, в которые может попадать симплекс-процессор. Задача поиска глобального минимума $\Phi_n(x_k, y_i)$ в этом случае будет состоять в поиске оптимальных стартовых образов модели $I_0 = y_1, y_2, \dots, y_n$, относительно которых СП начинает спуск. Подобную задачу решают генетические алгоритмы (ГА), в которых формируется переменная популяция образов, постепенно приближающаяся к оптимальному решению. Данная задача может также решаться методами Монте-Карло путем генерации случайной популяции образов, в которой один из образов рано или поздно окажется вблизи оптимального решения.

Так как симплекс-процессор обладает высокой скоростью работы и хорошей локальной сходимостью, то задача глобальной оптимизации может быть сведена к задаче поиска долины глобального минимума. Под “долиной” здесь следует понимать множество точек проекционного пространства, до которых из глобального минимума существует ε -коридор [17] с неотрицательным градиентом. Так как число локальных минимумов функционала невязки обычно невелико, то для поиска оптимального решения достаточно, чтобы в каждой долине функционала $\Phi_n(x_k, y_i)$ находился хотя бы один стартовый образ. Это позволяет отказаться от генерации случайных образов, как в методах Монте-Карло, поскольку число долин конечно. В случае “гладких” функционалов $\Phi_n(x_k, y_i)$ для поиска глобального минимума будет достаточно всего лишь единиц или десятков стартовых образов. Наоборот, идеи селекции, заложенные в генетических алгоритмах, оказываются вполне плодотворными для формирования ограниченной популяции стартовых образов.

Предлагаемая в данном разделе методика поиска оптимального решения близка по идеологии принципам, реализуемым в генетических алгоритмах, с той лишь разницей, что случайной выборкой для построения популяции оптимальных стартовых образов здесь является обработанный ранее и достаточно большой массив реализаций случайного процесса. В генетических алгоритмах случайная популяция эволюционирует к конечному решению посредством механизмов скрещивания и мутации. В настоящем подходе популяция полученных ранее решений задачи минимизации эволюционирует к небольшому числу стартовых образов, достаточных для решения задачи минимизации в будущем.

Очевидно, что образы процесса (10) используются в процедуре селекции как структурированные записи, организованные в виде массива или базы данных, позволяющие проводить необходимые процедуры выбора или модификации образов. Для эффективной работы процедуры селекции будет рациональным разделить всю популяцию образов на блоки в соответствии с происхождением, качеством аппроксимации моделью случайной реализации процесса и частотой использования образов. Один из вариантов такого деления представлен в табл. 2.

Таблица 2

Тип памяти	Назначение
Базовая рефлекторная (БР)	Образы, вносимые в память до первого использования алгоритма (аналог — генетически заложенная информация)
Приобретенная рефлекторная (ПР)	Образы, которые часто встречаются в процессе работы (аналог — приобретенные рефлекссы, навыки)
Прошлый (удачный) Опыт (ПО)	Удачные образы, полученные в прошлом, но относительно редко встречающиеся

Предлагаемое деление памяти на разделы является условным, однако оно имеет очевидное функциональное назначение, позволяющее алгоритму минимизации приспосабливаться к изменяющимся условиям задачи. В процесс адаптации алгоритма входят четыре стадии.

1. *Базовое обучение.* Алгоритм использует для старта образы, созданные заранее из каких-либо теоретических соображений или прошлого опыта (по наследству). Удачные образы из БР и удачные образы, полученные в результате поиска, заносятся в раздел ПР если они имеют заметное отличие от других образов, находящихся в ПР. Если ни один из образов в ПР и БР не приводит к удачному решению, алгоритм использует процедуры мутации/скрещивания, применяемые в ГА.

2. *Приобретение опыта.* По мере заполнения раздела ПР образы в этом разделе периодически сортируются по убыванию коэффициента повторяемости $\omega(y_i)$, который увеличивается на единицу при каждом

удачном для данного образа результате:

- а) периодически, например при каждом сохранении раздела памяти на жесткий диск, $\omega(y_i)$ умножается на коэффициент забывания $\phi(t) < 1$, равный, например, $\phi = 0.8$;
- б) через некоторое время для образов с редкой повторяемостью будет выполняться условие $\omega(y_i) < \lambda$, где λ — порог; при выполнении этого условия образы переносятся в раздел ПО;
- с) среди образов с высокой повторяемостью $\omega(y_i)$ периодически проводится проверка на предмет их принадлежности к одной и той же долине функционала $\Phi_n(x_k, y_i)$; для этого подходит процедура скрещивания, применяемая в ГА, т.е. добавляется образ — средний по параметрам от пары; если результат будет удачным, то один из родительских образов уйдет в ПО; выживший стартовый образ может быть далеким от любого родительского, но быть лучшей точкой старта для большого процента реализаций процесса;
- д) после продолжительного времени работы алгоритма образы с высоким значением $\omega(y_i)$ могут быть сохранены в БР как геном модели;
- е) в случае неудачи алгоритм сначала использует данные раздела ПО, затем раздел БР и далее; переходит к процедуре мутации/скрещивания.

3. *Забывание редких образов.*

Данные в разделе прошлого опыта сортируются в порядке убывания повторяемости и ухудшения качества (результатирующей невязки). Для этого раздела также вводится свой коэффициент забывания, чтобы данные забывались через длительный срок, например после 100 циклов записи разделов на диск. Чтобы раздел не переполнялся, лишние образы в конце списка удаляются.

4. *Скрещивание/мутация.*

В случае, когда ни один из образов в памяти не попадает в долину оптимального решения, тогда запускается алгоритм скрещивания, описанный выше, или мутации по одной из схем, используемых в ГА. Мутация проводится путем добавления к базовым образам квазипериодических функций малой амплитуды (например, $\sin(\alpha y_i)$, $\cos(\alpha y_i)$) или случайного шума, которые смогут переместить стартовый образ в долину оптимального решения. Если ни одна из перечисленных процедур не достигает цели, то алгоритм выдает признак плохой реализации и записывает квазиоптимальный результат.

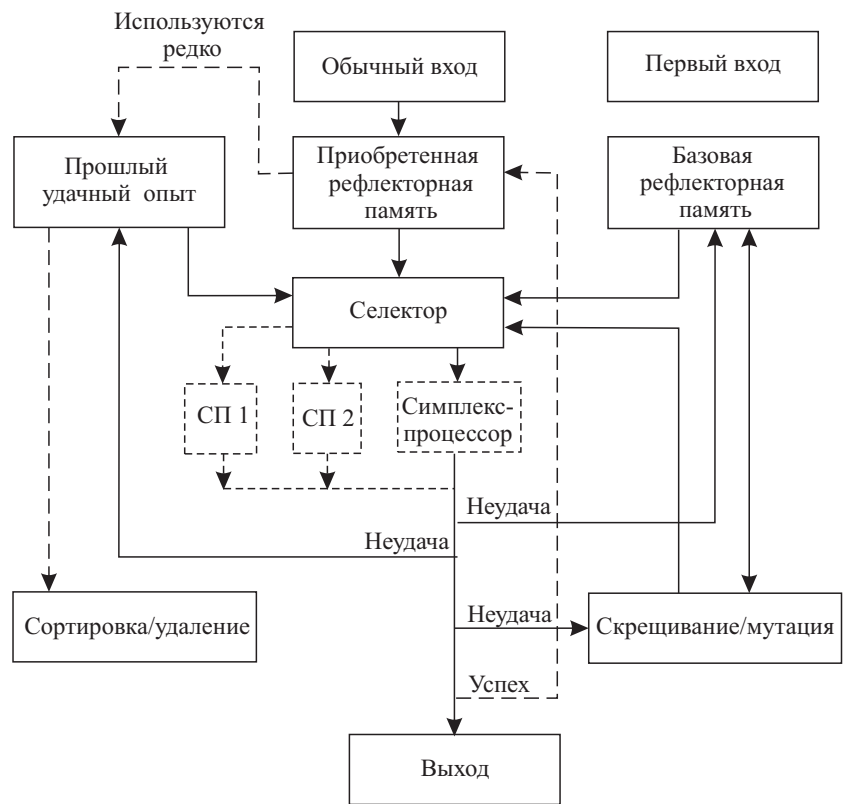


Рис. 3. Блок-схема алгоритма АСП

Описанный алгоритм и его возможные вариации можно назвать методом Адаптивных Симплекс-Процессоров (АСП). Блок-схема алгоритма представлена на рис. 3. Сплошными стрелками показаны направления передачи образов для обработки. Длинные пунктирные линии — запись/удаление образов. Тонким пунктиром показана возможность параллельной работы группы симплекс-процессоров, каждый из которых может запускаться как независимый процесс в современных операционных системах.

Следует отметить, что возможность параллельной обработки на многопроцессорных системах легко реализуется и для самого симплекс-процессора в виде n одинаковых процессов, вычисляющих невязку каждый в своей вершине симплекса, среди которых определяется направление спуска. Время работы такого алгоритма будет равно времени одного вычисления функционала на каждую итерацию. Если задача оптимизации жестко определена, то для ее решения можно использовать однотипные микропроцессоры, объединенные в кластер.

Предлагаемая схема адаптации симплекс-процессора к условиям задачи делает метод АСП гибким и максимально эффективным по количеству требуемых вычислительных ресурсов. В задачах потоковой обработки однотипных экспериментальных данных метод АСП оптимизирует процесс настолько, что даже для сильно нелинейной обратной задачи остается несколько десятков стартовых образов, а спуск к минимуму от них требует всего единиц или нескольких десятков итераций.

7. Пример использования метода АСП. Рассмотрим один пример использования метода АСП для решения нелинейного интегро-дифференциального уравнения, описывающего зондирование ионосферы методом некогерентного рассеяния [11]. Уравнение обратной задачи имеет вид

$$P(\tau) = B \int_0^\tau N_e(r) \cos^2 \left(\gamma \int_0^r N_e(\tilde{r}) d\tilde{r} \right) \left(r^2 \left(1 + \frac{T_e(r)}{T_i(r)} \right) \right)^{-1} \left| a \left(\tau - \frac{2r}{c} \right) \right|^2 dr + Q(\tau). \quad (11)$$

Здесь B, γ — константы, $a(t)$ — известная огибающая зондирующего импульса, $N_e(r)$ — электронная концентрация на высоте r , $\frac{T_e}{T_i}$ — отношение температуры электронов и ионов на заданной высоте, $Q(\tau)$ — шум эксперимента. В результате эксперимента измеряется вертикальный профиль мощности $P(\tau)$ в виде 650 дискретных отсчетов АЦП с эквивалентным шагом по высоте ~ 1.5 км. Обратная задача состоит в том, чтобы найти функции $N_e(r)$ и $\frac{T_e(r)}{T_i(r)}$.

Задача рассматривается в ограниченной области высот $D \in 100-1000$ км. Имеется следующая априорная информация относительно свойств функций и граничных условий для них:

- 1) функция $\frac{T_e(r)}{T_i(r)} \geq 1$ и строго равна единице на границах D ;
- 2) функция $N_e(r) > 0$ и равна нулю на нижней границе D ; в подавляющем числе экспериментов функция $N_e(r)$ имеет один максимум, и в этих случаях она хорошо аппроксимируется функцией Чепмена $N(z) = \exp \left(1 - \frac{z - z_0}{H} - \exp \left(-\frac{z - z_0}{H} \right) \right)$; функция $N_e(r)$ имеет большой динамический диапазон изменения $10^4 \div 10^6$;
- 3) предполагается, что шум эксперимента $Q(\tau)$ — это нормальный гауссов процесс со средним q и дисперсией σ .

Очевидно, что оператор обратной задачи является сильно нелинейным, так как содержит функцию $\cos^2(x)$, аргументом которой является интегральная концентрация электронов. Так как данный член уравнения (11) является квазипериодическим, функционал невязки в рассматриваемой задаче часто имеет субоптимальные минимумы, связанные с возможностью сдвига профиля $P(\tau)$ на целое число периодов. Функция $N_e(r)$ имеет большой динамический диапазон, и число квазипериодических вариаций колеблется от 2 до 20. При большом числе вариаций проблема сдвига на один-два периода становится весьма актуальной, поскольку свертка подынтегральной дроби в (11) с $a(t)$ сглаживает эти вариации и иногда даже делает обратную задачу несовместной. Поэтому в случае большого числа вариаций процедура селекции стартовых образов становится очень ответственным моментом, и попадание в долину глобального минимума может требовать значительного числа попыток.

Так как функция $N_e(r)$ имеет большой динамический диапазон изменения, прологарифмируем ее. Заменяем $\ln(N_e(r))$ и $\frac{T_e(r)}{T_i(r)}$ их аппроксимацией кубическими B-сплайнами [3] на равномерной сетке из 20 узлов по переменной r . Число узлов выбирается так, чтобы аппроксимация локальными сплайнами позволяла обеспечить необходимую вариативность функций и не приводила к фильтрации полезного сигнала. Этот эффект контролировался визуально, но он так же хорошо проявлялся в увеличении невязки.

Каждый образ процесса состоит из 20 коэффициентов $N_{j'}$, описывающих $\ln(N_e(r))$, и такого же числа коэффициентов T_j для $\frac{T_e(r)}{T_i(r)}$. Добавим к этому набору значение шума эксперимента q . В результате задача минимизации сводится к задаче (3) с количеством параметров 41.

По условию задачи минимизации $\|Af\|_{L_2} = \|u\|_{L_2}$, поэтому константа B исчезает при нормировке. Граничное условие $N_e(100) = 0$ позволяет исключить первый коэффициент из списка параметров $N_{j'}$. Кроме того, исключаются первый и последний коэффициенты аппроксимации для T_j , т.е. общее число параметров оптимизации равно 38.

В качестве первичных стартовых образов для $N_{j'}$ рассчитывается набор функций простого слоя Че-

пмена $\ln(N(z)) = 1 - \frac{z - z_0}{H} - \exp\left(-\frac{z - z_0}{H}\right)$, в которых параметры z_0 и H изменяются на сетке 30×30 значений (всего 900 профилей), равномерно перекрывающих предполагаемый диапазон изменения функции $N_e(r)$. Начальные значения коэффициентов аппроксимации T_j полагаем равными единице. Шум q полагаем равным среднему значению $u(x_k)$ в конце отрезка аппроксимации. Теперь все граничные условия и стартовые параметры заданы и можно применять метод АСП.

Образы модели обрабатываются и хранятся в виде записей, состоящих из независимых параметров и двух рядов коэффициентов $N_{j'}$ и T_j . В число параметров включены: частота использования образа $\omega(y_i)$, нормы производных $\|f'\|$, $\|f''\|$ и нормировочный коэффициент модели $\|Af\|$, чтобы не считать его повторно. Для расчета целевой функции сначала по коэффициентам сплайнов рассчитываются значения $\ln(N_e(r))$ и $\frac{T_e(r)}{T_i(r)}$ в точках, соответствующих отсчетам АСП, вычисляется профиль мощности (11),

вычисляется вклад соответствующих интегральных производных (4) и (5) для $\ln(N_e(r))$ и $\frac{T_e(r)}{T_i(r)}$. Далее рассчитывается невязка (1), в которой регуляризующие добавки суммируются.

Экспериментальные данные за каждые сутки записаны в отдельных файлах, поэтому автоматически задается время запуска процедуры селекции, описанной выше. При коэффициенте забывания $\phi = 0.8$ после обработки пяти файлов в блоке основной памяти (ПР) остается 200–300 стартовых образов. После многократного проведения процедуры скрещивания число стартовых образов в ПР сокращается еще на треть. Число итераций симплекс-процессора от этих стартовых образов при выбранном шаге сетки составляет от 1 до 40.

Каждый файл экспериментальных данных содержит профили мощности (11) с временным разрешением 1.5 минуты, всего за сутки около 960 профилей. Обработка суточного эксперимента на Pentium (R) Dual Core 2.5 ГГц занимает около 30 минут (в среднем 3 секунды на профиль). Таким образом, задача оптимизации по 38 параметрам легко решается на стандартном персональном компьютере с минимальными вычислительными затратами и с минимальными требованиями к оперативной памяти.

8. Выводы. Рассмотренный в настоящей статье метод решения нелинейных обратных задач позволяет удачно сочетать скорость сеточного алгоритма спуска и адаптационные свойства генетических алгоритмов. Ядром метода является симплекс-процессор, который использует минимальное число вычислений невязки на каждой итерации, а геометрические свойства симплицеального комплекса позволяют многократно уменьшить общее число итераций при большом числе параметров задачи. Простота реализации симплекс-процессоров позволяет легко распараллеливать процесс вычислений на многопроцессорных вычислительных системах. Эти качества симплекс-процессора позволяют использовать его в обратных задачах со многими параметрами, которые не могут быть решены традиционными численными методами из-за ограниченной мощности вычислительных систем. В частности, решение обратной задачи может быть получено для функций, которые не имеют аналитического описания, путем их детальной конечномерной аппроксимации.

Использование пространства $W_2^{(2)}$ в качестве функционального пространства обратной задачи реализует базовое свойство реальных физических систем, заложенное в вариационном принципе, когда минимум функционала невязки ищется среди функций с минимальной вариацией производных. Это позволяет избавиться от применения сложных схем регуляризации типа штрафных функций, которые часто используются в подобных задачах.

Вычислительные возможности симплекс-процессоров позволяют проводить глобальную оптимизацию путем использования элементов эволюционных алгоритмов. Глобальная оптимизация фактически состоит только в выборе стартовых образов для работы СП и эволюционной методике их селекции.

СПИСОК ЛИТЕРАТУРЫ

1. Bard Y. Nonlinear parameter estimation. New York: Academic Press, 1970.
2. Chambolle A. An algorithm for total variation minimization and applications // J. of Mathematical Imaging and Vision. 2004. 20. 89–97.
3. de Boor C. A practical guide to splines. Berlin: Springer, 1978.
4. Деннис Дж., Шнабель Р. Численные методы безусловной оптимизации и решения нелинейных уравнений. М.: Мир, 1988.
5. Goldberg D.E. Genetic algorithms in search, optimization and machine learning. Boston: Addison-Wesley, 1989.
6. Хемминг Р.В. Численные методы для научных работников и инженеров. М.: Наука, 1979.
7. Лэм Дж. Введение в теорию солитонов. М.: Мир, 1983.
8. Fogel L.J. Intelligence through simulated evolution: forty years of evolutionary programming. New York: Wiley, 1999.

9. *Foulkes W.M.C., Mitas L., Needs R.J., Rajagopal G.* Quantum Monte Carlo simulations of solids // *Reviews of Modern Physics*. 2001. **73**. 33–83.
10. *Rudin L.I., Osher S., Fatemi E.* Nonlinear total variation based noise removal algorithms // *Physica D*. 1992. **60**. 259–268.
11. *Shpynev B.G.* Incoherent scatter Faraday rotation measurements on a radar with single linear polarization // *Radio Sci*. 2004. **39**, N 3. RS3001 (DOI: 10.1029/2001RS002523).
12. *Shpynev B.G., Voronov A.L.* Discrete direction method in nonlinear problems of incoherent scattering spectrum fitting // *Geomagnetism and Aeronomy*. 2010. **50**, N 7. 908–913.
13. *Strong D., Chan T.* Edge-preserving and scale-dependent properties of total variation regularization // *Inverse Problems*. 2003. **19**. S165–S187.
14. *Swartz W.E.* Analytic partial derivatives for least-squares fitting incoherent scatter data // *Radio Sci*. 1978. **13**. 581–589.
15. *Абловитц М., Сигур Х.* Солитоны и метод обратной задачи. М.: Мир, 1987.
16. *Верлань А.Ф., Сизиков В.С.* Интегральные уравнения: методы, алгоритмы, программы. Киев: Наукова Думка, 1986.
17. *Колмогоров А.Н.* Избранные труды. Т. 3. Теория информации и теория алгоритмов. М.: Наука, 2005.
18. *Курант Р.* Уравнения с частными производными. М.: Мир, 1964.
19. *Морозов В.А.* Регулярные методы решения некорректно поставленных задач. М.: Наука, 1987.
20. *Саттон Р.С., Барто Э.Г.* Обучение с подкреплением. М.: БИНОМ, 2011.
21. *Соболев С.Л.* Некоторые применения функционального анализа в математической физике. Л.: Изд-во ЛГУ, 1950.
22. *Тихонов А.Н., Гончарский А.В., Степанов В.В., Ягола А.Г.* Численные методы решения некорректных задач. М.: Наука, 1990.
23. *Шеннон К.* Работы по теории информации и кибернетике. М.: ИЛ, 1963.

Поступила в редакцию
17.06.2013
