ОБЪЕДИНЕННЫЙ ИНСТИТУТ ЯДЕРНЫХ ИССЛЕДОВАНИЙ

ВЫЧИСЛИТЕЛЬНЫЙ ЦЕНТР

И.Н. Силин

Р-1228

МИНИМИЗАЦИЯ ФУНКЦИЙ МНОГИХ ПЕРЕМЕННЫХ МЕТОДОМ СОПРЯЖЕННЫХ ПРЯМЫХ

Дубна 1963 год

И.Н. Силин

Р-1228

МИНИМИЗАЦИЯ ФУНКЦИЙ МНОГИХ ПЕРЕМЕННЫХ МЕТОДОМ СОПРЯЖЕННЫХ ПРЯМЫХ

Направлено в "Журнал вычислительной математики и математической физики"

Дубна 1963 год

После опубликования работы [1] ряд товарищей обращался к авторам с вопросом: нельзя ли построить метод минимизации произвольных гладких функций многих переменных, приближающийся по эффективности к методу линеаризации [1] для функционалов.

Подобный метод, по-видимому, должен за конечное число приближений находить минимум квадратичной функции и разумно обобщаться на другие случаи. Желательно также, чтобы он позволял устанавливать поведение функции в квазиквадратичной окрестности ее минимума.[x] По сравнению с методом линеаризации он может потребовать во многих случаях большей вычислительной работы, так как будучи более общим методом, не дает возможности использовать структурные особенности строения минимизируемой функции. Ниже описывается такой метод, предлагаемый автором.

§ 1. Построение алгоритма минимизации квадратичной функции

Докажем сначала следующую лемму.

Лемма. Пусть дана квадратичная функция m переменных $f(a_1 \dots a_m) \equiv f(\vec{a})$, которая вдоль заданного $\vec{r}$ на каждой из параллельных прямых семейства $\vec{a}_0 + t\vec{r}$ имеет строгий минимум, т.е. $\frac{\partial^2 f(\vec{a}_0 + t\vec{r})}{\partial t^2} > 0$. Тогда семейство точек, на которых на этих прямых осуществляется минимум $f(\vec{a})$, образует диаметральную гиперплоскость (гиперплоскость, проходящую через точку $\vec{M}$, в которой $f(\vec{a})$ имеет экстремум, конечно, если он вообще есть).

Доказательство. Минимум по t на прямых $\vec{a} = \vec{a}_0 + t\vec{r}$ достигается в точках:

$$\vec{a} = \vec{a}_0 - \left(\frac{\partial f}{\partial t}\Big|_{t=0} \Big/ \frac{\partial^2 f}{\partial t^2}\right)\vec{r}, \qquad (1)$$

где $\frac{\partial^2 f}{\partial t^2} = \text{const}(\vec{a}_0)$, а $\frac{\partial f}{\partial t}\Big|_{t=0}$ – линейная функция $\vec{a}_0$. Функция (1) отображает любую прямую в пространстве $\vec{a}_0$, не параллельную $\vec{r}$, в прямую же, но уже целиком принадлежащую интересующему нас $m-1$ -мерному семейству минимумов на прямых
Следовательно, любая прямая, соединяющая две произвольные точки семейства $a = a + tr$ $\vec{a} = \vec{a}_0 - \left(\frac{\partial f}{\partial t}\Big|_{t=0} \Big/ \frac{\partial^2 f}{\partial t^2}\right)\vec{r}$, принадлежит этому же семейству. Такое семейство есть гиперплоскость. Экстремальная точка $\vec{M}$ также принадлежит этой гиперплоскости, так как $\frac{\partial f(\vec{M} + t\vec{r})}{\partial t}\Big|_{t=0} = 0$. Лемма доказана.

Пользуясь результатом леммы, сразу можем построить алгоритм минимизации квадратичной функции.

Алгоритм 1. Пусть на пространстве m переменных $a_1 \dots a_m$ задана квадратичная

---

x) Мы будем понимать под квазиквадратичной окрестностью такую окрестность, в в которой функция вместе с ее первыми производными может быть с заданной точностью аппроксимирована квадратичной функцией с ее производными.

функция $f(\vec{a})$, имеющая строгий минимум. Зададим в пространстве $\vec{a}$ $m+1$ точки $\vec{M}_1^1 \dots \vec{M}_m^1, \vec{M}_{m+1}^1$, через которые нельзя провести гиперплоскость $m-1$ порядка. Через точки $\vec{M}_1^1$ и $\vec{M}_2^1$ проведем прямую с направляющим вектором $\vec{r}_1 = \lambda_1(\vec{M}_2^1 - \vec{M}_1^1)$ и найдем на ней точку $\vec{M}_1^2$, в которой достигается минимум $f(\vec{M}_1^1 + t\vec{r}_1)$. Через оставшиеся $m-1$ точки проведем прямые, параллельные первой, и на них также найдем минимумы в точках $\vec{M}_2^2 \dots \vec{M}_m^2$. Полученные точки $\vec{M}_1^2 \dots \vec{M}_m^2$ уже определяют диаметральную гиперплоскость, в которой, согласно лемме, лежит точка экстремума $f(\vec{a})$. Повторим процедуру, оставаясь в этой гиперплоскости. Через точки $\vec{M}_1^2$ и $\vec{M}_2^2$ проведем прямую с направляющим вектором $\vec{r}_2 = \lambda_2(\vec{M}_2^2 - \vec{M}_1^2)$, а через точки $\vec{M}_3^2 \dots \vec{M}_m^2$ параллельные ей прямые и на них найдем минимумы в точках $\vec{M}_1^3 \dots \vec{M}_{m-1}^3$, которые определяют диаметральную гиперплоскость $m-2$ порядка. Продолжаем процесс до тех пор, пока не получим прямую с направляющим вектором $\vec{r}_m = \lambda_m(\vec{M}_2^m - \vec{M}_1^m)$ и точку $\vec{M}_1^{m+1}$, в которой и достигается минимум $f(\vec{a})$.

В дальнейшем для удобства будем предполагать нормирующие множители λ_k такими, что

$$\frac{\partial^2 f(\vec{a}_0 + t\vec{r}_k)}{\partial t^2} = 2. \qquad (2)$$

§ 2. Некоторые свойства алгоритма 1

Докажем ряд теорем относительно свойств алгоритма 1.

Теорема 1. Нахождение минимума квадратичной функции согласно алгоритму 1 требует вычисления ее в минимальном числе точек, если не прибегать к аналитическому дифференцированию.

Доказательство. Запишем уравнение произвольной квадратичной функции в виде:

$$f(\vec{a}) = (c_{00} + c_{01}a_1 + \dots + c_{0m}a_m) + (c_{11}a_1^2 + c_{12}a_1a_2 + \dots + c_{1m}a_1a_m) + \dots + c_{mm}a_m^2 .$$

Подсчитаем число коэффициентов этого уравнения. Оно равно $(m+1) + m + (m-1) + \dots + 1 = \frac{(m+2)(m+1)}{2}$. Таким образом, чтобы однозначно задать квадратичную функцию, нужно задать ее не менее, чем в $\frac{(m+2)(m+1)}{2}$ точках. Минимизация по алгоритму 1 требует ее вычисления именно в таком числе точек. А именно, нам нужно вычислить ее значения в $m+1$ начальной точке $\vec{M}_1^1 \dots \vec{M}_{m+1}^1$. Далее, чтобы найти минимум вдоль каждой из параллельных прямых $\vec{a} = \vec{M}_1^1 + t\vec{r}_1$, нужно вычислить ее значения еще в m точках (на первой из прямых уже есть две точки, а на остальных по одной, но $\frac{\partial^2 t}{\partial t^2} = const$ $(\vec{M}_1^1)$). На последующих этапах нужно вычислить значение функции в $m-1, m-2, \dots, 1$ точках. Суммарное число точек равно

$$(m+1) + m + (m-1) + \dots + 1 = \frac{(m+2)(m+1)}{2} .$$

Теорема доказана.

Теорема 2. Точка минимума $\vec{M} \equiv \vec{M}_1^{m+1}$ и векторы $\vec{r}_1 \dots \vec{r}_m$, полученные при осуществлении алгоритма 1 и нормированные согласно (2), однозначно определяют квадратичную функцию $f(\vec{a})$.

Доказательство. Пользуясь информацией, указанной в условии теоремы, напишем уравнение функции $f(\vec{a})$ в явном виде. Для этого разложим вектор $\vec{a} - \vec{M}$ по векторам $\vec{r}_1 \ldots \vec{r}_m$ с коэффициентами разложения $a_1 \ldots a_m$. Для определения a_i получим систему уравнений

$$\sum_{i=1}^{m} a_i \vec{r}_i = \vec{a} - \vec{M},$$

откуда

$$a_i = \sum_{k=1}^{m} \bar{R}^{-1}_{ik} (a_k - M_k) , \qquad (3)$$

где R – матрица, строками которой являются векторы $\vec{r}_1 \ldots \vec{r}_m$, $\bar{R}$ – транспонированная матрица R, $\bar{R}^{-1}_{ik}$ – элементы матрицы, обратной к $\bar{R}$. Вычислим $f(\vec{a}) = f(\vec{M} + a_1 \vec{r}_1 + \ldots + a_m \vec{r}_m)$. $f(\vec{M} + a_m \vec{r}_m) = f(\vec{M}) + a_m^2$ (см. нормировку (2)). Замечая, что согласно алгоритму 1 и лемме на прямой $\vec{a} = \vec{M} + t_m \vec{r}_m$ лежат минимумы $f(\vec{a})$ по t_{m-1} на прямых $\vec{a} = (\vec{M} + t_m \vec{r}_m) + t_{m-1} \vec{r}_{m-1}$, получаем $f(\vec{M} + a_m \vec{r}_m + a_{m-1} \vec{r}_{m-1}) = f(\vec{M}) + a_m^2 + a_{m-1}^2$. Опять-таки, согласно алгоритму 1 и лемме, в плоскости $\vec{a} = \vec{M} + t_m \vec{r}_m + t_{m-1} \vec{r}_{m-1}$ лежат минимумы по t_{m-2} вдоль прямых $\vec{a} = (\vec{M} + t_m \vec{r}_m + t_{m-1} \vec{r}_{m-1}) + t_{m-2} \vec{r}_{m-2}$ и, следовательно, $f(M + a_m r_m + a_{m-1} r_{m-1} + a_{m-2} r_{m-2}) = f(\vec{M}) + a_m^2 + a_{m-1}^2 + a_{m-2}^2$. Продолжая рассуждения, получим:

$$f(\vec{M} + a_1 \vec{r}_1 + \ldots + a_m \vec{r}_m) = f(\vec{M}) + a_1^2 + \ldots + a_m^2 . \qquad (4)$$

Теорема доказана.

Следствия из теоремы 2. Подставляя (3) в (4), имеем

$$f(\vec{a}) = f(\vec{M}) + \sum_j [\sum_k \bar{R}^{-1}_{jk} (a_k - M_k)]^2 = f(\vec{M}) + \sum_{ik} [\sum_j \bar{R}^{-1}_{ji} (a_i - M_i) \bar{R}^{-1}_{jk} (a_k - M_k)] = \qquad (5)$$

$$= f(\vec{M}) + \sum_{i,k} [(\sum_j R^{-1}_{ij} \bar{R}^{-1}_{jk})(a_i - M_i)(a_k - M_k)] ,$$

откуда

$$\tfrac{1}{2} \frac{\partial^2 f}{\partial a_i \partial a_k} \equiv G_{ik} = \sum_j R^{-1}_{ij} \bar{R}^{-1}_{jk} \qquad (6)$$

$$G^{-1}_{ik} = \sum_j \bar{R}_{ij} R_{jk} . \qquad (7)$$

Теорема 3. Если квадратичная функция $f(\vec{a})$ не имеет строгого минимума (имеет седло, протяженный экстремум, или вообще не имеет экстремума), то при осуществлении алгоритма 1 вдоль одного из направлений минимизации $\vec{r}_1 \ldots \vec{r}_m$ обязательно не будет строгого минимума $/ \frac{\partial^2 f(\vec{a}_0 + t\vec{r}_k)}{\partial t^2} \leq 0 /$.

Доказательство. Действительно, либо строгого минимума не будет вдоль $\vec{r}_1$ на прямых $\vec{a}_0 + t\vec{r}_1$, либо его не будет на гиперплоскости минимумов по $t - A_{m-1}$, так как на прямых $\vec{a}_0 + t\vec{r}_1$, проходящих через область, где $f(\vec{a}) = f_{min}$, минимум будет достигаться при значениях $f = f_{min}$, а если функция не ограничена снизу, то она не будет ограничена снизу и на A_{m-1}. Продолжая рассуждения, придем к выводу, что если минимум будет на всех направлениях $\vec{r}_1 \ldots \vec{r}_{m-1}$, то его не будет вдоль $\vec{r}_m$. Теорема доказана.

Теорема 4. Параллелепипед с центром в точке $\vec{M} \equiv \vec{M}_1^{m+1}$, построенный на векторах $2\epsilon\vec{r}_1 \dots 2\epsilon\vec{r}_m$, где $\vec{r}_1 \dots \vec{r}_m$; $\vec{M}_1^{m+1}$ получены при осуществлении алгоритма 1, а ϵ – произвольное число, описан вокруг элипсоида $f(\vec{a}) = f(\vec{M}) + \epsilon^2$ и имеет наименьший объем из всех описанных вокруг этого элипсоида параллелепипедов.

Доказательство. Действительно, по формуле (4) в любой из точек $\vec{a} = \vec{M} \pm \epsilon\vec{r}_k$ $f(\vec{a}) = f(\vec{M}) + \epsilon^2$ и $\frac{\partial f}{\partial a_{i \neq k}} = 0$, то есть стенки параллелепипеда касаются элипсоида в точках $\vec{M} \pm \epsilon\vec{r}_k$. Объем параллелепипеда равен $(2\epsilon)^m |\,Det\,R\,| = (2\epsilon)^m \sqrt{Det\,G^{-1}}$ (см. (7)). Отсюда видно, что объем параллелепипеда не зависит от конкретных векторов $r_1 \dots r_m$, а определяется свойствами квадратичной функции. Можно убедиться, задавая в алгоритме 1 точку $\vec{M}_1^1$ в центре элипсоида, а точки $\vec{M}_2^1 \dots \vec{M}_{m+1}^1$ на концах его главных осей, что главные оси элипсоида также обладают свойствами векторов $\vec{r}_1 \dots \vec{r}_m$, и, следовательно, объем нашего параллелепипеда равен объему описанного параллелепипеда со стенками, перпендикулярными главным осям. Если мы теперь отобразим элипсоид $f(\vec{a}) = f(\vec{M}) + \epsilon^2$ на шар, линейно сжимая его вдоль главных осей, то прямоугольный параллелепипед отобразится в куб, а отношения объемов параллелепипеда к элипсоиду и куба к шару будут одинаковыми. Из всех же параллелепипедов, описанных вокруг шара, куб имеет наименьший объем (у косоугольного параллелепипеда та же высота, но больше площадь любого из оснований). Теорема доказана.

Следствие из теоремы 4. При линейном отображении элипсоида $f(\vec{a}) = f(\vec{M}) + \epsilon^2$ в шар параллелепипед, указанный в условии теоремы 4 (вообще говоря, косоугольный) отобразится в куб, так как у него наименьший объем, а система векторов $\vec{r}_1 \dots \vec{r}_m$ – в систему ортогональных векторов.

§ 3. Обобщение алгоритма на случай неквадратичных функций.

Из теорем 1–4 видно, что алгоритм 1 при обобщении его на случай неквадратичных функций, кроме предположительно хорошей сходимости, дал бы весьма большую информацию о квазиквадратичной окрестности минимума. Например, если $f(\vec{a}) = -2\ln L$, где L – функция правдоподобия[2,3], применяемая в статистике, $G^{-1} = \bar{R}R$ дает при выполнении некоторых условий[x] матрицу ошибок σ_{ik}^2 найденных оптимальных значений параметров a_k. Параллелепипед, указанный в условии теоремы 4, может хорошо аппроксимировать область низких значений $f(\vec{a})$. Из теоремы 3 видно, что практически устраняется возможность сойтись к седловой точке.

Однако полезно изменить порядок вычислений в алгоритме 1, чтобы иметь возможность лучше использовать получаемую в процессе счета информацию для последующих итераций.

Алгоритм 2. Зададим m линейно независимых "затравочных" векторов, например, координатные вектора, $\vec{l}_1 \dots \vec{l}_m$ и начальные шаги вдоль них $h_1 \dots h_m$. Из точки начального приближения $\vec{M}_1^1$ делается шаг $\vec{M}_2^1 = \vec{M}_1^1 + h_1\vec{l}_1$ и из точки $\vec{M}_2^1$ вдоль вектора $\vec{r}_1 = \lambda_1(\vec{M}_2^1 - \vec{M}_1^1)$ находится минимум $f(\vec{a})$ в точке $\vec{M}_1^2$ ($\vec{r}_1$ нормируется согласно (2), $\frac{\partial^2 f}{\partial t^2}$ берется в минимуме). Далее делается шаг $\vec{M}_3^1 = \vec{M}_1^2 + h_2\vec{l}_2$ и из этой точки снова ищется минимум в точке $\vec{M}_2^2$ вдоль $\vec{l}_1$. Вдоль вектора $\vec{r}_2 = \lambda_2(\vec{M}_2^2 - \vec{M}_1^2)$ на прямой, соединяющей точки $\vec{M}_1^2$ и $\vec{M}_2^2$, находится минимум в точке $\vec{M}_1^3$. Далее делается шаг

---

x) В частности, если минимум не вырожденный, как это понимается в работе[1], и поверхность $f(\vec{a}) = f(\vec{M}) + 1$ лежит внутри области квазиквадратичности.

$\vec{M}_4^1 = \vec{M}_1^3 + h_3 \vec{\ell}_3$ и из этой точки ищется минимум в точке $\vec{M}_3^2$ вдоль вектора $\vec{r}_1$, а из точки $\vec{M}_3^2$ – минимум в точке $\vec{M}_2^3$ вдоль $\vec{r}_2$. Вдоль $\vec{r}_3 = \lambda_3(\vec{M}_2^3 - \vec{M}_1^3)$ находим $\vec{M}_1^4$ и т.д.

Будем называть точки, найденные при минимизации вдоль уже известных векторов $\vec{r}_1 \dots \vec{r}_k$ вспомогательными приближениями, а точки $\vec{M}_1^{k+2}$, находимые при минимизации вдоль вновь получаемого вектора $\vec{r}_{k+1}$, – основными приближениями. На каждое следующее основное приближение приходится делать на одно вспомогательное больше. Процесс продолжается, пока не найдется вектор $\vec{r}_m$ и точка $\vec{M}_1^{m+1}$. Сравнивая с алгоритмом 1, и обращая внимание на совпадение нумерации точек $\vec{M}_1^j$ и одинаковый смысл этой нумерации, можно видеть, что геометрическое построение в обоих алгоритмах совпадает. Если бы функция $f(\vec{a})$ была квадратичной, точка $\vec{M}_1^{m+1}$ указала бы ее минимум. В общем случае описанная выше процедура является разгоном. Итерационный процесс может быть продолжен следующим образом. Сделаем некоторый шаг $\vec{M}_{m+1}^2 = \vec{M}_1^{m+1} + h_1^1 \vec{r}_1$, после чего понизим индексы j точек M_1^j и k векторов $\vec{r}_k$ на единицу, отбрасывая те из них, у которых появились нулевые индексы. В результате создается такая же ситуация, как и после нахождения предпоследнего основного приближения. Исходя из точки $\vec{M}_{m+1}^1$ (бывшей $\vec{M}_{m+1}^2$), снова можем последовательно найти вспомогательные приближения вдоль векторов $\vec{r}_1 \dots \vec{r}_{m-1}$, (бывших $\vec{r}_2 \dots \vec{r}_m$), находя точку $\vec{M}_2^m$, и вдоль $\vec{r}_m = \lambda_m(\vec{M}_2^m - \vec{M}_1^m)$ ($\vec{M}_1^m$ – бывшая точка M_1^{m+1}) находим новое основное приближение в точке $\vec{M}_1^{m+1}$. Геометрическое построение снова приобрело структуру построения из алгоритма 1, хотя мы вычислили только $m-1$ новое вспомогательное приближение.

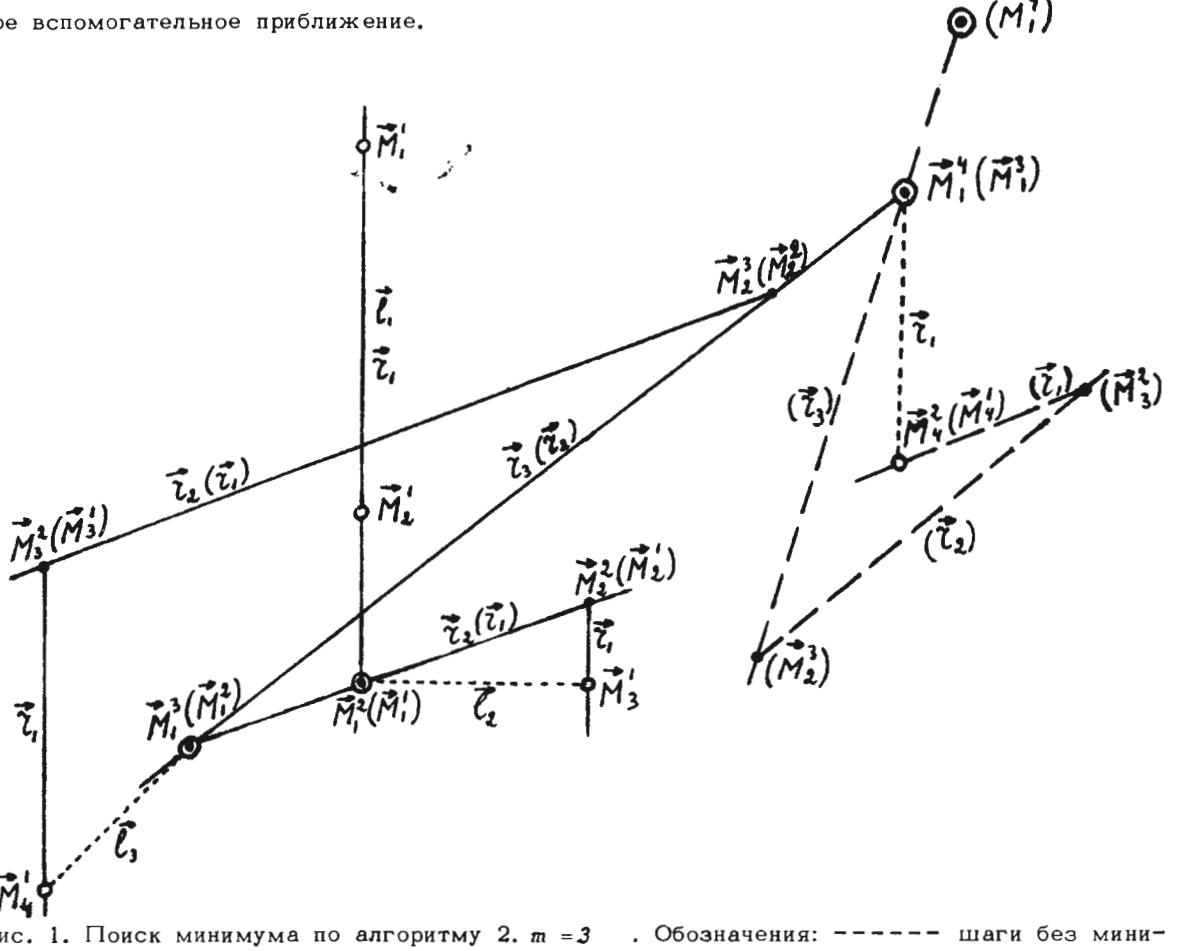

Рис. 1. Поиск минимума по алгоритму 2. $m = 3$. Обозначения: ------ шаги без минимизации, ———— разгон, — — — — — продолжение процесса для неквадратичной функции. ○ – точки не являющиеся минимумами, ● – вспомогательные приближения, ◉ – основные приближения. В скобках указаны обозначения точек и векторов после переиндексации.

Практически весьма важным является выбор шагов h_1^1, с помощью которых обеспечивается продолжение процесса. Следует иметь в виду, что слишком большие шаги могут приводить к ухудшению сходимости из-за нелинейности "оврага", вдоль которого происходит спуск, а слишком малые шаги - к потере точности при получении очередного $\vec{r}_m$ (так как минимумы вдоль прямых находятся приближенно) и как следствие - тоже к ухудшению сходимости. Кроме того, если нас интересует поведение функции в квадратичной окрестности минимума, шаги не должны выводить из этой окрестности, иначе векторы $\vec{r}_k$ и матрица G^{-1} будут сильно искажены.

Алгоритм 2 был осуществлен автором на электронной вычислительной машине для случая четырех переменных и показал достаточно высокую эффективность. Шаги h_1^1 выбирались следующим образом. В процессе минимизации вдоль каждого из $\vec{r}_k$ определялся радиус области квазиквадратичности h_k, внутри которой параболическая интерполяция по трем точкам предсказывает положение минимума t_{min} с заданной точностью ϵ, постоянной для всех $\vec{r}_k$ (с учетом нормировки (2)). При выполнении шага $h_1^1 \vec{r}_1$ бралось $h_1^1 = \min_k |h_k|$, чтобы не выходить за пределы эквипотенциальной поверхности $f = f_{min} + (\min_k h_k)^2$, внутри которой ожидается квадратичность. Такой выбор шага позволяет также экономить время при нахождении вспомогательных приближений.

На практике весьма част случай, когда параллельно с вычислением функции $f(\vec{a})$ без заметных дополнительных затрат времени может быть вычислен по аналитическим формулам и ее градиент $\vec{g}(\vec{a})$. Желательно построить алгоритм, требующий в этом случае существенно меньшего объема вычислений.

Можно было бы, используя (7) в качестве оценки обратной матрицы вторых производных, сразу предсказывать направление спуска. Однако, как можно увидеть на простых примерах, при этом последовательные шаги $\vec{r}_k$ не приобретают свойств $\vec{r}_k$ из алгоритма 1 даже при попадании в квадратичную область. Чтобы не накапливать погрешности, можно все время повторять разгон, беря в качестве "затравочных" векторов векторы, полученные в результате предыдущего разгона.

Алгоритм 3. Процесс разгона тот же, что и в алгоритме 2, однако вспомогательные приближения $\vec{M}_2^j$ не вычисляются, а предсказываются, исходя из (4) и (2) по формуле

$$\vec{M}_2^j = \vec{M}_1^j + h_j \vec{\ell}_j - \tfrac{1}{2} \sum_{k=1}^{j-1} (\vec{g}(\vec{M}_2^j + h_j \vec{\ell}_j), \vec{r}_k) \vec{r}_k ,$$

откуда

$$\vec{r}_j = \lambda_j \left[h_j \vec{\ell}_j - \tfrac{1}{2} \sum_{k=1}^{j-1} (\vec{g}(\vec{M}_1^j + h_j \vec{\ell}_j), \vec{r}_k) \vec{r}_k \right] . \qquad (8)$$

Вдоль вектора $\vec{r}_j$ сразу ищется очередное основное приближение. При этом уже по двум точкам можно делать кубическую интерполяцию, используя знание градиента. После того как найдено $\vec{r}_m$, разгон повторяется. В качестве векторов $\vec{\ell}_1 \ldots \vec{\ell}_m$ берутся векторы $\vec{r}_m, \vec{r}_1 \ldots \vec{r}_{m-1}$ (минимум вдоль $\vec{r}_m$ уже найден ранее).

Алгоритм 3 также был осуществлен автором на машине. Шаги h_k выбирались аналогично h_1^1 в алгоритме 2. В не слишком сложных случаях алгоритм 3 дает заметную экономию счета по сравнению с алгоритмом 2.

Заметим, что вместо алгоритма 3 может быть предложено несколько более экономич-

ных алгоритмов, требующих, однако, в несколько раз большей оперативной памяти. Алгоритмы же 2 и 3 требуют $m^2 + 4m$ рабочих ячеек.

В связи с тем, что в описываемом методе большая нагрузка падает на одномерную минимизацию вдоль прямых, следует применять для нее весьма надежные и экономичные алгоритмы. Автор применял алгоритмы, осуществляющие поиск интервала, внутри которого наверняка есть минимум, и затем стягивающие границы этого интервала с обеих сторон к минимуму с квадратической или кубической интерполяцией при нахождении минимума, что обеспечивает ньютоновскую сходимость для достаточно гладких функций.

Строгое рассмотрение сходимости алгоритмов 2 и 3 весьма затруднительно. Автор приводит результаты качественного исследования.

Следует сразу оговорить условия, при которых процедуры более или менее оправданы.

1) Имеется замкнутая эквипотенциальная поверхность $f(\vec{a}) = F > f(\vec{a}_0)$, охватывающая точку начального приближения $\vec{a}_0$.

2) $f(\vec{a})$ непрерывна вместе со своими производными до второго порядка включительно на замкнутом множестве P, ограниченном указанной выше эквипотенциальной поверхностью. При этом условие 1) является существенным для любого метода минимизации $f(\vec{a})$ на неограниченном пространстве.

Если ряд основных приближений сойдется к некоторой точке $\vec{M}$ и не произойдет вырождения системы векторов $\vec{r}_1 \dots \vec{r}_m$ в систему линейно зависимых векторов, то в точке $\vec{M}$ будет по крайней мере экстремум (минимум или, что маловероятно, седло) так как

$$\frac{\partial f(\vec{M} + t\vec{r}_k)}{\partial t} = 0;, \quad k = 1 \dots m ; \qquad \vec{r}_k \text{ - линейны независимы.}$$

Алгоритмы 2 и 3 построены таким образом, что ни на одном из шагов с конечным номером система $\vec{r}_k$ не может стать вырожденной. Однако вырождение может происходить в пределе. Его нельзя полностью исключить без изменения алгоритмов, особенно принимая во внимание конечную точность, с которой может вестись счет. При попадании же в достаточно малую окрестность минимума, в котором $Det\, G_{ik} \neq 0$, где $G_{ik} = \frac{\partial^2 f}{\partial a_i \partial a_k}$ при достаточно малых принудительных шагах h_1' в алгоритме 2 и h_k в алгоритме 3 и, соответственно, достаточно большой точности вычислений опасность вырождения исчезает.

Возможность вырождения можно полностью устранить, если в алгоритме 2 отказаться от продолжения процесса после разгона и каждый раз повторять разгон с использованием исходных "затравочных" векторов $\vec{\ell}_1 \dots \vec{\ell}_m$, в алгоритме 3 также повторять разгон с исходных $\vec{\ell}_1 \dots \vec{\ell}_m$. Общность видоизмененных алгоритмов в смысле их применимости такая же, как и для метода скорейшего спуска или релаксационного метода, за исключением того, что имеется возможность при нормировке (2) получить вектор бесконечной длины. Но в принципе, можно ввести искусственное ограничение на длину вектора $\vec{r}_k$ с использованием, например, размеров области P и разности $F - f(\vec{a})$.

Тем не менее автору кажется, что не следует отказываться от алгоритмов 2 и 3 в их основной форме, так как в большинстве случаев такой отказ приведет к увеличению

времени счета. При подозрении, что произошло вырождение векторов $\vec{r}_1 \ldots \vec{r}_m$ эти алгоритмы могут быть возобновлены с исходными $\vec{\ell}_1 \ldots \vec{\ell}_m$, но уже с той точки, к которой перед этим с заданной точностью сошлись.

Хотя описанный выше метод и локален, он является в некотором смысле развитием метода "оврагов"[4]. Однако, автор предалагает назвать его методом сопряженных прямых - по аналогии с методом сопряженных градиентов[5] для решения систем линейных уравнений.

Следует обратить внимание, что алгоритм, аналогичный алгоритму 1, в принципе может быть применен к решению систем линейных уравнений, если вместо минимума на прямой $(a + tr)$ искать нуль величины $(\vec{r}, \vec{g})$, где $\vec{g}$ вектор невязок. Выше изложенные теоремы могут быть с некоторыми изменениями перефразированы на этот случай. Алгоритм может быть обобщен на некоторый класс нелинейных уравнений без приведения к задаче минимизации, хотя во многих случаях это не выгодно.

Автор благодарен С.Н.Соколову за полезные обсуждения и Е.П.Жидкову за замечания по работе.

Литература

1. С.Н.Соколов, И.Н.Силин. Нахождение минимумов функционалов методом линеаризации. Препринт ОИЯИ Д-810, Дубна, 1961.
2. Т.Крамер. Математические методы статистики. ИИЛ Москва, 1948.
3. Н.П.Клепиков, С.Н.Соколов. Анализ экспериментальных данных методом максимума правдоподобия. Препринт ОИЯИ Р-235, Дубна, 1958.
4. И.М.Гельфанд, М.Л.Цетлин. Принципы нелокального поиска в системах автоматической оптимизации. ДАН, т.137, №2, 1961.
5. Н.С.Березин, Н.П.Жидков. Методы вычислений, т.2, гл.6, § 5, физматгиз, Москва, 1959.

Рукопись поступила в издательский отдел
6 марта 1963 года.