

Из табл. 6.6 видно, что рассмотренные варианты метода Хука — Дживса приводят к заметному уменьшению количества шагов поиска, практически не изменяя точности нахождения точки минимума и значения функции в этой точке.

Отметим еще одну модификацию метода Хука — Дживса, позволяющую определить точку минимума целевой функции двух переменных за один шаг поиска. Графическая иллюстрация этого метода дана на рис. 6.20. Обратим внимание на то, что процесс поиска точки \mathbf{x}^1 в этом варианте совпадает с процессом ее поиска в варианте, представленном на рис. 6.19, б. Проводя исследующий поиск в точке \mathbf{x}^1 с помощью метода циклического покоординатного спуска и выбирая на этапе спуска ускоряющий множитель по условию минимума целевой функции в установленном направлении спуска, получим точку \mathbf{x}^2 , совпадающую с точкой минимума рассматриваемой функции.

6.6. Методы Розенброка и Пауэлла

Рассмотрим еще одну стратегию поиска точки минимума ограниченной снизу целевой функции $f(\mathbf{x})$, $\mathbf{x} \in \mathbb{R}^n$. Метод, реализующий эту стратегию поиска, также предусматривает проведение исследующего поиска на каждом k -м шаге. Целью исследующего поиска является выбор текущего направления спуска с учетом информации о поведении целевой функции в окрестности точки \mathbf{x}^{k-1} , найденной на предыдущем шаге.

Отличие этого метода от метода Хука — Дживса состоит в способе выбора направлений исследующего поиска. Если в методе Хука — Дживса они фиксированы и коллинеарны направлениям векторов стандартного базиса в \mathbb{R}^n , то в рассматриваемом методе выбор этих направлений проводят в процессе минимизации целевой функции путем построения на каждом k -м шаге поиска нового ортонормированного базиса в \mathbb{R}^n . Итогом выполнения этого этапа является нахождение точки $\tilde{\mathbf{x}}^k$, для которой $f(\tilde{\mathbf{x}}^k) < f(\mathbf{x}^{k-1})$. Тогда вектор $\tilde{\mathbf{x}}^k - \mathbf{x}^{k-1}$ определит направление спуска на k -м шаге.

Такая стратегия поиска впервые была реализована в 1960 году и получила название метода Розенброка*. В первоначальном варианте метода Розенброка процедура нахождения точки $\tilde{\mathbf{x}}^k$ (как и в методе Хука — Дживса) была основана на дискретных шагах исследующего поиска по выбранным направлениям. Опишем модификацию этого метода, в которой на каждом k -м шаге поиска выбор координат точки $\tilde{\mathbf{x}}^k$ проводят методом модифицированного *циклического покоординатного спуска*.

Пусть выбраны начальная точка $\mathbf{x}^0 \in \mathbb{R}^n$ и параметр $\varepsilon > 0$ точности поиска. Возьмем в качестве векторов \mathbf{p}_j^1 , $j = \overline{1, n}$, определяющих направления исследующего поиска на первом шаге, векторы \mathbf{e}_j стандартного базиса в \mathbb{R}^n . Полагаем $k = j = 1$, $\tilde{\mathbf{x}}_1^k = \mathbf{x}^0$ и переходим к основной части алгоритма.

1. Минимизируя функцию $\psi_j^{(k)}(\boldsymbol{\alpha}) = f(\tilde{\mathbf{x}}_j^k + \boldsymbol{\alpha}\mathbf{p}_j^k)$, находим значение $\boldsymbol{\alpha}_j^{(k)} \in \mathbb{R}$, вычисляем $\tilde{\mathbf{x}}_{j+1}^k = \tilde{\mathbf{x}}_j^k + \boldsymbol{\alpha}_j^{(k)}\mathbf{p}_j^k$ и переходим к п. 2.

2. Если $j < n$, то принимаем $j := j + 1$ и возвращаемся к п. 1. В противном случае полагаем $\mathbf{x}^k = \tilde{\mathbf{x}}_{n+1}^k$ и переходим к п. 3.

3. Если $|\mathbf{x}^k - \mathbf{x}^{k-1}| < \varepsilon$, где $\varepsilon > 0$ — достаточно малое число, характеризующее точность выполнения этапа исследующего поиска, то дальнейший поиск точки минимума прекращаем, принимая $\mathbf{x}^* \approx \mathbf{x}^k$ и $f(\mathbf{x}^*) \approx f(\mathbf{x}^k)$. В противном случае переходим к п. 4.

4. На k -м шаге поиска строим новый ортонормированный базис, векторы \mathbf{p}_j^{k+1} , $j = \overline{1, n}$, которого задают направления исследующего поиска на $(k+1)$ -м шаге. При построении этого базиса используем процесс ортогонализации Грама — Шмидта и проводим его следующим образом. Полагаем

$$\mathbf{a}_j^{k+1} = \begin{cases} \mathbf{p}_j^k, & \boldsymbol{\alpha}_j^{(k)} = 0; \\ \sum_{i=j}^n \boldsymbol{\alpha}_i^{(k)} \mathbf{p}_i^k, & \boldsymbol{\alpha}_j^{(k)} \neq 0. \end{cases} \quad j = \overline{1, n}.$$

* См.: Базара М., Шетти К.

Далее вычисляем

$$\mathbf{b}_j^{k+1} = \begin{cases} \mathbf{a}_j^{k+1}, & j = 1; \\ \mathbf{a}_j^{k+1} - \sum_{i=1}^{j-1} (\mathbf{a}_j^{k+1}, \mathbf{p}_i^{k+1}) \mathbf{p}_i^{k+1}, & j = \overline{2, n}, \end{cases}$$

и

$$\mathbf{p}_j^{k+1} = \frac{\mathbf{b}_j^{k+1}}{|\mathbf{b}_j^{k+1}|}, \quad j = \overline{1, n}.$$

После вычисления векторов \mathbf{p}_j^{k+1} , $j = \overline{1, n}$, переходим к п. 1, полагая $j := 1$, $\tilde{\mathbf{x}}_j^{k+1} = \mathbf{x}^k$ и затем $k := k + 1$.

На рис. 6.21 иллюстрируются этапы одного шага поиска точки минимума целевой функции двух переменных из начальной точки \mathbf{x}^0 . Отметим, что рассмотренный алгоритм можно модифицировать, вводя дополнительные правила выбора точки \mathbf{x}^k на каждом k -м шаге при проведении этапа исследующего поиска (см. 6.4).

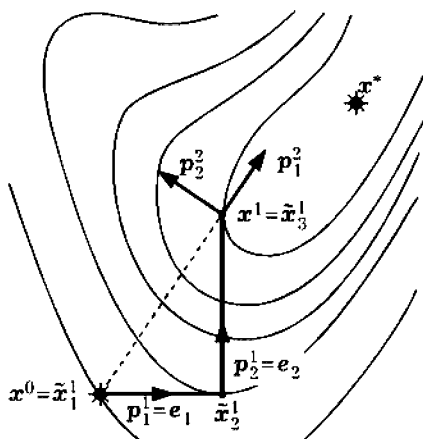


Рис. 6.21

Пример 6.6. Используем метод Розенброка для минимизации функции из примеров 6.4 и 6.5. Выберем начальную точку

$\mathbf{x}^0 = (-2, 1)$ и точность поиска $\varepsilon = 0,01$. Графическая иллюстрация первых трех шагов метода дана на рис. 6.22. Результаты поиска точки минимума по методу Розенброка приведены в табл. 6.7.

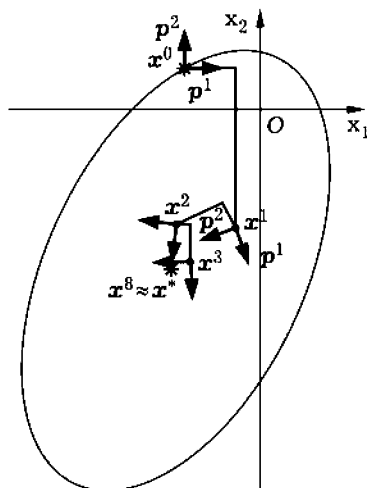


Рис. 6.22

Таблица 6.7

k	\mathbf{x}^k	$f(\mathbf{x}^k)$	k	\mathbf{x}^k	$f(\mathbf{x}^k)$
1	(-0,412, -3,259)	-12,482	5	(-2,168, -4,444)	-27,977
2	(-2,020, -3,178)	-23,817	6	(-2,219, -4,428)	-27,996
3	(-1,834, -4,234)	-27,245	7	(-2,233, -4,472)	-28,000
4	(-2,159, -4,177)	-27,794			

Из табл. 6.7 видно, что для рассматриваемой функции использование метода Розенброка не приводит к повышению эффективности поиска по сравнению, например, с модифицированным методом циклического покоординатного спуска (см. пример 6.4), поскольку при той же точности поиска необходимое число N шагов поиска не уменьшается.

Рассмотрим еще один алгоритм прямого поиска, предложенный в 1964 году и называемый обычно методом Пауэлла* Предварительный этап и первые два пункта основной части этого алгоритма практически совпадают с алгоритмом метода Розенброка. Отличие состоит в построении системы векторов, определяющих направления спуска на каждом последующем шаге поиска и проявляется начиная с п. 3 описанного выше алгоритма метода Розенброка. В методе Пауэлла вычисления в п. 3 носят итерационный характер. Поэтому на предварительном этапе для номера i итерации принимают $i = 1$.

1. Минимизируя функцию $\psi_j^{(k)}(\boldsymbol{\kappa}) = f(\tilde{\boldsymbol{x}}_j^k + \boldsymbol{\kappa}\boldsymbol{p}_j^k)$, находим значение $\boldsymbol{\kappa}_j^{(k)} \in \mathbb{R}$, вычисляем $\tilde{\boldsymbol{x}}_{j+1}^k = \tilde{\boldsymbol{x}}_j^k + \boldsymbol{\kappa}_j^{(k)}\boldsymbol{p}_j^k$ и переходим к п. 2.

2. Если $j < n$, то полагаем $j := j + 1$ и возвращаемся к п. 1. В противном случае переходим к п. 3.

3. Полагаем $\boldsymbol{p} = \tilde{\boldsymbol{x}}_{n+1}^k - \tilde{\boldsymbol{x}}_1^k$ и находим значение $\hat{\boldsymbol{x}}$, минимизируя функцию $\psi(\boldsymbol{\kappa}) = f(\tilde{\boldsymbol{x}}_{n+1}^k + \boldsymbol{\kappa}\boldsymbol{p})$, а затем вычисляем $\boldsymbol{z}_i = \tilde{\boldsymbol{x}}_{n+1}^k + \hat{\boldsymbol{x}}\boldsymbol{p}$. Если $i < n$, то для всех $j = \overline{1, n-1}$ заменяем \boldsymbol{p}_j^k на \boldsymbol{p}_{j+1}^k , полагаем $\boldsymbol{p}_n^k = \boldsymbol{p}$, $\tilde{\boldsymbol{x}}_1^k = \boldsymbol{z}_i$, $j = 1$, $i := i + 1$ и переходим к п. 1 рассматриваемого алгоритма. В противном случае, т.е. при $i = n$, переходим к п. 4.

4. Принимаем $\boldsymbol{x}^k = \boldsymbol{z}_n$. Если $|\boldsymbol{x}^k - \boldsymbol{x}^{k-1}| < \varepsilon$, то вычисления прекращаем и полагаем $\boldsymbol{x}^* \approx \boldsymbol{x}^k$ и $f(\boldsymbol{x}^*) \approx f(\boldsymbol{x}^k)$. В противном случае считаем, что $\tilde{\boldsymbol{x}}_1^{k-1} = \boldsymbol{x}^k$, $i = 1$, $j = 1$, $k := k + 1$, и возвращаемся к п. 1 рассматриваемого алгоритма.

Рис. 6.23 иллюстрирует один шаг поиска точки минимума целевой функции двух переменных из начальной точки \boldsymbol{x}^0 . Можно доказать, что в случае квадратичной целевой функции $f(\boldsymbol{x}) = (Q\boldsymbol{x}, \boldsymbol{x})/2 + (\boldsymbol{c}, \boldsymbol{x})$ с положительно определенной матрицей Q с помощью алгоритма метода Пауэлла за один шаг поиска строится система векторов \boldsymbol{p}_j , $j = \overline{1, n}$, сопряженных относительно этой матрицы. При этом точка \boldsymbol{x}^1 , полученная в конце

* См.: Базара М., Шетти К.

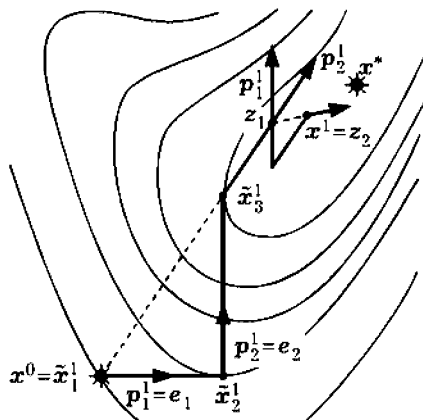


Рис. 6.23

первого шага, совпадает с искомой точкой \mathbf{x}^* минимума такой целевой функции. Покажем это на конкретном примере.

Пример 6.7. Используем метод Пауэлла для минимизации ранее рассмотренной функции (см. примеры 6.1, 6.4–6.6). Выберем начальную точку $\mathbf{x}^0 = (-2, 1)$ и точность поиска $\varepsilon = 0,01$. Графическая иллюстрация метода представлена на рис. 6.24. На рисунке видно, что точка \mathbf{x}^* минимума целевой функции достигается за один шаг поиска. В результате этого поиска получаем $\mathbf{x}^* \approx (-2,236, -4,472)$ и $f(\mathbf{x}^*) \approx -28,0$, что с точностью $5 \cdot 10^{-5}$ совпадает с точным решением $(-\sqrt{5}, -2\sqrt{5})$ задачи. #

Известно много модификаций метода Пауэлла. На рис. 6.25 представлена одна из таких модификаций в применении к функции из примеров 6.1, 6.4–6.7. В качестве начальной выбрана точка $\mathbf{x}^0 = (1, 1)$. Алгоритм, реализующий этот вариант метода Пауэлла, включает проведение на каждом k -м шаге поиска одной дополнительной итерации циклического покоординатного спуска в направлении вектора* \mathbf{p}^n . При минимизации

*См.: Реклейтис Г., Рейвиндран А., Рэгсдел К., а также: Химмельблау Д.

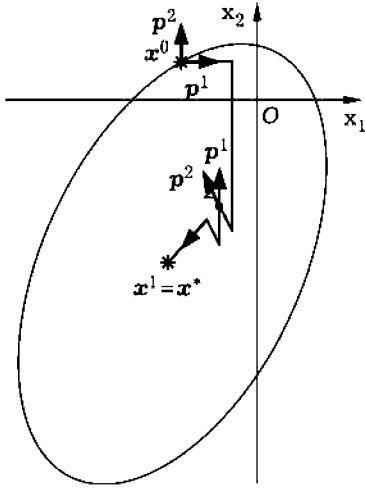


Рис. 6.24

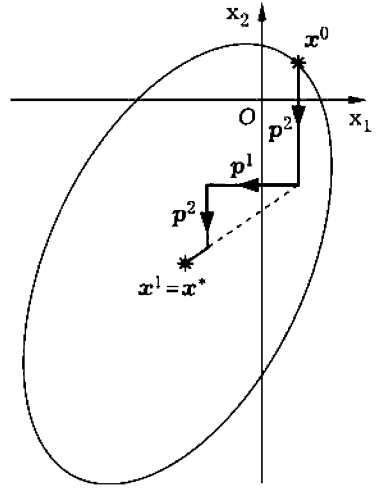


Рис. 6.25

функции, не являющейся квадратичной, дополнительная итерация гарантирует линейную независимость векторов, определяющих направление спуска на каждом k -м шаге поиска.

Вопросы и задачи

6.1. Перечислите методы прямого поиска, выделите те из них, в которых поиск точки минимума целевой функции проводится в соответствии с рекуррентным соотношением (6.8). Объясните, почему при построении алгоритмов таких методов направления поиска должны быть линейно независимыми. Графически проиллюстрируйте на примере решения двумерной задачи минимизации возможные подходы к построению таких направлений поиска.

6.2. В задаче

$$f(x_1, x_2) = 11x_1^2 + 6x_1x_2 + 3x_2^2 - 2\sqrt{10}(x_1 - 3x_2) - 22 \rightarrow \min,$$