

Х/О

Л.А. РАСТРИГИН
К.К. РИПА

АВТОМАТНАЯ ТЕОРИЯ СЛУЧАЙНОГО ПОИСКА

АКАДЕМИЯ НАУК ЛАТВИЙСКОЙ ССР
ИНСТИТУТ ЭЛЕКТРОНИКИ
И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

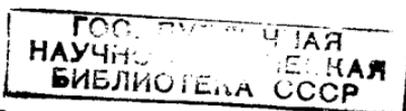
АКАДЕМИЯ НАУК ЛАТВИЙСКОЙ ССР
ИНСТИТУТ ЭЛЕКТРОНИКИ
И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

Л. А. РАСТРИГИН,
К. К. РИПА

АВТОМАТНАЯ
ТЕОРИЯ
СЛУЧАЙНОГО
ПОИСКА

ИЗДАТЕЛЬСТВО «ЗИНАТНЕ»
РИГА 1973

6S0.1.
P245
УДК 51:62-507



24
15925

УЧ-13965

АВТОМАТНАЯ ТЕОРИЯ СЛУЧАЙНОГО ПОИСКА.

Л. А. Растрин, К. К. Рипа. Рига, «Зинатне», 1973. 344 стр.

В монографии разработаны основы автоматной теории случайного поиска, которая дает возможность исследовать алгоритмы поиска с единой точки зрения. Алгоритмы поиска интерпретируются как конечные автоматы, а объект оптимизации — как случайная среда, в которой находится автомат. Исследование процесса оптимизации сводится к исследованию некоторой цепи Маркова, соответствующей данному алгоритму поиска и данному объекту оптимизации. Исследованы некоторые классы случайных и детерминированных алгоритмов поиска, рассмотрены вопросы их эквивалентности. Построены оптимальные алгоритмы в отдельных классах стохастических автоматов для различных объектов оптимизации.

Илл. 58, табл. 8, библи. 58 назв.

Печатается по решению Редакционно-издательского совета Академии наук Латвийской ССР от 17 февраля 1972 года.

© Издательство «Зинатне», 1973

ПРЕДИСЛОВИЕ

Эта книга является обобщением работ авторов за последние несколько лет. Идея представления случайного поиска в виде вероятностного автомата довольно естественна, хотя и была сформулирована только в 1967 г. [1]. В этом же году Ю. И. Неймарком с соавт. [2] была высказана мысль о возможности использования вероятностных автоматов для решения задачи многопараметрической оптимизации. Оба направления сначала развивались независимо, пока не было обнаружено, что они представляют собой разные ветви одного и того же дерева — автоматной стохастической оптимизации. Первую из них, которую представляют авторы, можно условно назвать «Случайный поиск как вероятностный автомат», вторую — «Вероятностный автомат как случайный поиск».

В данной книге нашли отражение работы и идеология первого из указанных направлений автоматной оптимизации, хотя в ней затронуты работы и второго направления.

Подобно тому как детерминированные автоматы являются частным случаем вероятностных, так регулярные процедуры поиска могут рассматриваться как частный случай процедур случайного поиска. В соответствии с этим в книге в виде частных случаев рассмотрены известные регулярные методы поисковой оптимизации как детерминированные автоматы и исследована работа этих автоматов в случайной среде, т. е. при оптимизации объекта с помехами.

Монография является пятой в серии книг по случайному поиску. В первой — «Случайный поиск в задачах оптимизации многопараметрических систем» (Рига, «Зинатне», 1965) — проблема случайного поиска была

только поставлена. Во второй — «Статистические методы поиска» (М., «Наука», 1968) — были исследованы преимущественно локальные алгоритмы случайного поиска. В третьей — «Случайный поиск с линейной тактикой» (Рига, «Зинатне», 1971) — рассмотрены алгоритмы случайного спуска, и в четвертой — «Случайный поиск в задачах адаптации» (Рига, «Зинатне», 1973) — исследованы процессы сходимости случайного поиска в точку или зону экстремума.

В настоящей книге работа алгоритмов случайного поиска анализируется в терминах вероятностных автоматов. Теория вероятностных автоматов дала эффективный язык, удобный для анализа и синтеза алгоритмов случайного поиска. Этот же язык дал возможность убедительно показать, что случайный поиск эффективно обобщает регулярные алгоритмы. Открылась возможность оптимального синтеза алгоритмов поиска по свойствам объекта.

При попытке автоматного представления алгоритмов случайного поиска с непрерывным распределением случайного шага в пространстве параметров обнаружилось любопытное обстоятельство: эти алгоритмы породили континуальные вероятностные автоматы.

Автоматное представление наряду со всеми своими достоинствами обладает одним существенным недостатком — трудностью анализа при большом числе переменных. А если вспомнить, что случайный поиск наиболее эффективен при оптимизации объектов с большим числом параметров [3], то очевидна необходимость дальнейших изысканий подходящего математического аппарата для решения таких задач. Следовательно, методы, рассмотренные в данной книге, нельзя считать универсальными. Их можно рекомендовать для анализа и синтеза алгоритмов поиска при оптимизации объектов не слишком высокой размерности.

В заключение авторы пользуются приятной возможностью высказать свою глубокую признательность лицам, которые просмотрели рукопись и высказали свое мнение по поводу ее удачных и неудачных мест, в результате чего последних стало значительно меньше.

Л. Растринг
К. Рина

ВВЕДЕНИЕ

§ 0.1. ЗАДАЧА ПОИСКОВОЙ ОПТИМИЗАЦИИ

Объектом поисковой оптимизации является $(n+1)$ -полюсник с n входами и одним выходом (см. рис. 0.1.1). Входы объекта x_1, \dots, x_n управляемы и определяются вектором

$$\mathbf{X} = (x_1, \dots, x_n). \quad (0.1.1)$$

Эти входы обычно называются параметрами или координатами объекта (в литературе по планированию экспериментов их называют факторами). На рис. 0.1.1 показано схематическое изображение объекта в скалярной (а) и векторной (б) форме.

Выход объекта Q является всегда скалярным и обычно называется показателем качества (или откликом) объекта, а его зависимость от входных параметров

$$Q = Q(x_1, \dots, x_n) = Q(\mathbf{X}) \quad (0.1.2)$$

— функцией качества или функцией отклика (последний термин обычно употребляют в планировании экспериментов).

Зависимость (0.1.2) может иметь недетерминированный характер. В этом случае используется запись

$$Q' = Q(\mathbf{X}, E), \quad (0.1.3)$$

где E — стационарный случайный фактор. Пусть

$$M_E[Q(\mathbf{X}, E)] = Q(\mathbf{X}), \quad (0.1.4)$$



Рис. 0.1.1. Представление объекта оптимизации.

где M — знак математического ожидания, а индекс E обозначает, что осреднение производится по случайному фактору E при $\mathbf{X} = \text{const}$.

Теперь можно сформулировать задачу оптимизации. Следует определить такое допустимое состояние \mathbf{X} , при котором $Q(\mathbf{X})$ достигает экстремума, т. е. решается задача

$$Q(\mathbf{X}) \rightarrow \underset{\mathbf{X} \in S}{\text{extr}}, \quad (0.1.5)$$

где S — заданное множество допустимых состояний объекта.

Как видно, постановка задачи оптимизации очень напоминает постановку задачи нелинейного программирования с той лишь разницей, что экстремизируемая функция $Q(\mathbf{X})$ неизвестна и задается своими реализациями, которые могут быть случайными (0.1.3). В последнем случае процесс образования показателя Q' очень удобно описывается аддитивной моделью

$$Q'(\mathbf{X}) = Q(\mathbf{X}) + \varepsilon(\sigma), \quad (0.1.6)$$

где $\varepsilon(\sigma)$ — случайное нормально распределенное число с нулевым математическим ожиданием и дисперсией σ^2 . Отдельные реализации предполагаются некоррелированными.

Ниже всюду, если это не оговорено особо, рассматривается задача минимизации

$$Q(\mathbf{X}) \rightarrow \underset{\mathbf{X} \in S}{\min}, \quad (0.1.7)$$

где множество допустимых входов S определено заранее.

Решением поставленной задачи является состояние \mathbf{X}^* , удовлетворяющее условию

$$Q(\mathbf{X}^*) \leq \underset{(\mathbf{X}^*, \mathbf{X} \in S)}{Q(\mathbf{X})} \quad (0.1.8)$$

или

$$\mathbf{X}^* = \arg \min_{\mathbf{X} \in S} Q(\mathbf{X}). \quad (0.1.9)$$

Последнее выражение означает, что искомое состояние \mathbf{X}^* является аргументом, принадлежащим S и соответствующим при этом минимуму функции качества.

Очевидно, что при отсутствии аналитического выражения для $Q(\mathbf{X})$ решение поставленной задачи (0.1.7) может производиться одним из трех способов:

1) путем восстановления (идентификации) функции качества $Q(\mathbf{X})$ с последующим решением соответствующей задачи нелинейного программирования;

2) путем поиска, т. е. соответствующим образом последовательно «запрашивая» значения показателя в состояниях $\mathbf{X}_1, \mathbf{X}_2, \dots$ с тем, чтобы постепенно приближаться к решению задачи \mathbf{X}^* ;

3) комбинированным путем, т. е. последовательно перемежая процесс идентификации и поиска. Этот путь, называемый иногда экстраполяционным поиском, опирается на простые модели функции качества $Q(\mathbf{X})$, которые используются для получения оценок положения точки минимума, уточняемых на следующем этапе.

Четкого различия между этими тремя способами указать нельзя, т. к. понятия модели объекта и ее идентификации используются в разной мере, но во всех трех случаях.

В данной книге будем рассматривать лишь поиск, смысл которого сводится к следующему [1, 3--5]. Для сбора информации на N -м этапе (или шаге) поиска делаются пробные шаги, т. е. определяется показатель качества в точках

$$\mathbf{X}_N^{(1)}, \mathbf{X}_N^{(2)}, \dots, \mathbf{X}_N^{(m_N)}, \quad (0.1.10)$$

выбранных некоторым случайным образом вокруг точки \mathbf{X}_{N-1} .

Выбор пробных шагов осуществляется в соответствии с заданным n -мерным законом распределения

$$p(\mathbf{X}_N^{(i)}/\mathbf{W}_{N-1}, \mathbf{X}_N^{(1)}, \dots, \mathbf{X}_N^{(i-1)}), \quad (0.1.11)$$

который формируется в процессе поиска. (Для детерминированных методов поиска это распределение вырождается в δ -функцию.)

Полученная информация о поведении функции качества в районе исходной точки \mathbf{X}_{N-1} позволяет принять решение о переводе объекта в другую точку

$$\mathbf{X}_N = \mathbf{X}_{N-1} + \Delta\mathbf{X}_N, \quad (0.1.12)$$

причем смещение $\Delta\mathbf{X}_N$ определяется в соответствии с выбранной функцией решения Φ :

$$\Delta\mathbf{X}_N = \Phi(\mathbf{W}_{N-1}, \mathbf{X}_N^{(1)}, \dots, \mathbf{X}_N^{(m_N)}). \quad (0.1.13)$$

Здесь вектор \mathbf{W}_{N-1} называется вектором памяти. Он

характеризует предысторию поиска и определяет влияние предыдущих шагов поиска на выбор N -го шага.

Вектор памяти \mathbf{W}_{N-1} с точки зрения информации, полученной на $(N-1)$ -м этапе поиска, корректируется в соответствии с алгоритмом самообучения

$$\mathbf{W}_N = \Psi(\mathbf{W}_{N-1}, \mathbf{X}_{N-1}, \mathbf{X}_N^{(1)}, \dots, \mathbf{X}_N^{(m_N)}). \quad (0.1.14)$$

При $\mathbf{W}_N = 0$ имеет место независимый поиск. Примером такого независимого поиска является поиск по методу градиента.

Для исследования и оценки эффективности поисковых процедур, для сопоставления различных алгоритмов поиска и для определения сходимости процесса поиска необходимо задать ситуацию, в которой эти алгоритмы действуют, т. е. иметь модели.

Приведем некоторые модели функции качества, на которых в дальнейшем будут исследованы и сопоставлены алгоритмы поиска.

1. *Линейная модель* объекта является наиболее простой из всех возможных. Она характеризуется линейной зависимостью показателя качества объекта от его управляемых параметров:

$$Q(x_1, \dots, x_n) = Q_0 + \sum_{i=1}^n a_i(x_i - x_{i0}), \quad (0.1.15)$$

или, в векторной форме,

$$Q(\mathbf{A}, \mathbf{X}) = Q_0 + [(\mathbf{X} - \mathbf{X}_0), \mathbf{A}], \quad (0.1.16)$$

где квадратными скобками выделено скалярное произведение. Вектор градиента функции качества равен

$$\text{grad } Q(\mathbf{X}_0) = (a_1, \dots, a_n) = \mathbf{A}, \quad (0.1.17)$$

т. е. он предполагается постоянным в зоне поиска. Это означает, что функция качества является линейной в пределах, определяемых смещением $\Delta \mathbf{X} = \mathbf{X} - \mathbf{X}_0$.

Эта модель хорошо отражает локальные свойства объекта вдали от экстремума или при достаточно малых шагах поиска $\Delta \mathbf{X}$. Поэтому она применима для локального анализа работы алгоритмов поиска.

2. *Стохастическая модель* объекта характеризуется тем, что в ней вектор ситуации объекта $\mathbf{A} = (a_1, \dots, a_n)$,

т. е. градиент функции качества $Q(\mathbf{X})$, меняется от одного состояния к другому по некоторому вероятностному закону. В общем виде такая модель выражается формулой

$$\mathbf{A}_{N+1} = \mathbf{F}(\mathbf{A}_N, \Delta \mathbf{X}_N), \quad (0.1.18)$$

где \mathbf{F} — случайный вектор, плотность распределения которого зависит от двух векторных аргументов — вектора ситуации \mathbf{A} в предыдущей исходной точке и вектора смещения $\Delta \mathbf{X}$.

В общем случае плотность распределения вектора ситуации \mathbf{A} на $(N+1)$ -м шаге зависит от вектора ситуаций на N предыдущих шагах $\mathbf{A}_N, \dots, \mathbf{A}_1$ и от смещения $\Delta \mathbf{X}_N$ на N -м шаге, т. е.

$$p(\mathbf{A}_{N+1}/\mathbf{A}_N, \dots, \mathbf{A}_1, \Delta \mathbf{X}_N). \quad (0.1.19)$$

Такая модель объекта может быть использована для моделирования весьма широкого класса объектов оптимизации. Объект в этом случае задается при помощи функции $\mathbf{F}(\mathbf{A}_N, \Delta \mathbf{X}_N)$ или плотности распределения (0.1.19).

Далее будем исследовать работу алгоритмов поиска на стохастической модели, ситуации которой принимают значения из конечного множества векторов

$$\mathbf{A} = \{\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_v\}. \quad (0.1.20)$$

Плотность распределения (0.1.19) в простейшем случае можно задавать в виде однородной цепи Маркова

$$p(\mathbf{A}_{N+1}/\mathbf{A}_N, \Delta \mathbf{X}_N^{(i)}) = \left\| \begin{array}{ccc} \delta_{11}^{(i)} & \dots & \delta_{1v}^{(i)} \\ \dots & \dots & \dots \\ \delta_{v1}^{(i)} & \dots & \delta_{vv}^{(i)} \end{array} \right\| \quad (0.1.21)$$

где i — номер реализации вектора $\Delta \mathbf{X}_N$ на N -м шаге поиска ($i=1, \dots, m$);

v — число направлений, которые может принимать вектор градиента.

§ 0.2. АЛГОРИТМЫ СЛУЧАЙНОГО ПОИСКА

Алгоритмы случайного поиска характеризуются тем, что выбор пробных шагов $\mathbf{X}_N^{(1)}, \dots, \mathbf{X}_N^{(m_N)}$, а также рабочего смещения $\Delta \mathbf{X}_N$ производится случайно

в соответствии с вероятностным распределением (0.1.11) и функцией решения (см. формулу (0.1.13)). Различают алгоритмы случайного поиска без самообучения и с самообучением.

АЛГОРИТМ ПОИСКА БЕЗ САМООБУЧЕНИЯ

При работе алгоритмов поиска без самообучения не учитывается предыстория процесса поиска, т. е. $\mathbf{W}=0$, и не используется выражение (0.1.14).

Среди алгоритмов поиска без самообучения можно условно выделить два класса алгоритмов: 1) алгоритмы поиска с линейной тактикой и 2) алгоритмы поиска с нелинейной тактикой.

Алгоритм с линейной тактикой [4] характеризуется тем, что при неудачном шаге ($\Delta Q'_N \geq 0$) делаются пробные шаги в соответствии с распределением (0.1.11), а потом передается управление на выбор $\Delta \mathbf{X}_N$ в соответствии с функцией решения (0.1.13). При удачном шаге ($\Delta Q'_N < 0$) пробные шаги не делаются, а блок пробных шагов сразу передает управление на блок функции решения, который определяет последующий как равный предыдущему, т. е.

$$\Delta \mathbf{X}_N = \Delta \mathbf{X}_{N-1}. \quad (0.2.1)$$

Примером алгоритма поиска с линейной тактикой может быть, например, следующий алгоритм:

$$\Delta \mathbf{X}_N = \begin{cases} \Delta \mathbf{X}_{N-1} & \text{при } \Delta Q_{N-1} = Q(\mathbf{X}_{N-1}) - Q(\mathbf{X}_{N-2}) < 0; \\ a \Xi & \text{при } \Delta Q_{N-1} \geq 0. \end{cases} \quad (0.2.2)$$

Алгоритм с нелинейной тактикой. При удачном шаге делаются пробные шаги в соответствии с распределением (0.1.11), а потом передается управление на выбор $\Delta \mathbf{X}_N$ в соответствии с решающим правилом (0.1.13). При неудачном шаге пробные шаги не делаются, а сразу управление передается на блок функции решения.

Для некоторых нелинейных алгоритмов поиска этот шаг может быть равным предыдущему с противоположным знаком, т. е.

$$\Delta \mathbf{X}_N = -\Delta \mathbf{X}_{N-1}. \quad (0.2.3)$$

Ниже в основном будут рассмотрены алгоритмы поиска с одним пробным шагом, совпадающим со смещением $\Delta \mathbf{X}_N$.

Пример алгоритма поиска с нелинейной тактикой:

$$\Delta \mathbf{X}_N = \begin{cases} a\Xi & \text{при } \Delta Q_{N-1} < 0; \\ -\Delta \mathbf{X}_{N-1} & \text{при } \Delta Q_{N-1} \geq 0, \end{cases} \quad (0.2.4)$$

где

$$\Delta Q_{N-1} = Q(\mathbf{X}_{N-1}) - Q(\mathbf{X}_{N-2}); \quad (0.2.5)$$

$$\mathbf{X}_{N-1} = \mathbf{X}_{N-2} + \Delta \mathbf{X}_{N-1}; \quad (0.2.6)$$

Ξ — реализация случайного вектора, равномерно распределенного на единичной гиперсфере; $\Xi = (\xi_1, \dots, \xi_n)$;

a — длина смещения $\Delta \mathbf{X}_N$:

$$a = |\Delta \mathbf{X}_N|. \quad (0.2.7)$$

АЛГОРИТМ ПОИСКА С САМООБУЧЕНИЕМ

В алгоритмах поиска с самообучением [6—9] вектор \mathbf{W}_N изменяется по формуле (0.1.14) в зависимости от сделанного шага $\Delta \mathbf{X}_{N-1}$ и полученной информации (опыта) ΔQ_{N-1} . Ниже будут рассмотрены алгоритмы поиска с самообучением с одним пробным шагом, совпадающим с рабочим шагом $\Delta \mathbf{X}_{N-1}$. В этом случае алгоритм самообучения можно задать, например, при помощи следующей рекуррентной формулы:

$$\mathbf{W}_N = \mathbf{W}_{N-1} - \delta \operatorname{sign} \Delta Q_{N-1}(\mathbf{X}) \operatorname{sign} \Delta \mathbf{X}_{N-1}, \quad (0.2.8)$$

где δ — параметр, характеризующий скорость обучения ($\delta > 0$); $\Delta Q_{N-1}(\mathbf{X})$ определяется формулой (0.2.5);

$$\operatorname{sign} \Delta \mathbf{X}_{N-1} = (\operatorname{sign} \Delta x_1^{(N-1)}, \dots, \operatorname{sign} \Delta x_n^{(N-1)}) \quad (0.2.9)$$

($\Delta x_i^{(N-1)}$ — i -я координата вектора $\Delta \mathbf{X}_{N-1}$).

Чтобы устранить нежелательное детерминирование поиска, на изменение вектора накладывается ограничение

$$|\mathbf{W}| < d\sqrt{n} \quad (0.2.10)$$

($0 < d \leq 1$).

В настоящей книге везде будут рассмотрены алгоритмы *дискретно распределенного поиска* [10]. Исследование таких алгоритмов поиска обосновано хотя бы тем, что они широко применяются в технических устройствах — автоматических оптимизаторах [11], работающих по алгоритмам случайного поиска.

Дискретно распределенный поиск характеризуется тем, что множество реализаций случайного вектора ΔX_N , определяемое формулой (0.1.13), содержит конечное число векторов.

Дискретно распределенный поиск будет определяться при помощи задания конечного множества из m элементов, которые могут быть значениями вектора $\Delta X^{(i)}$:

$$\Delta X^{(i)} = (\Delta x_1^{(i)}, \dots, \Delta x_n^{(i)}) \quad (0.2.11)$$

$$(i = 1, 2, \dots, m).$$

Поиск по вершинам гиперкуба. Пусть, например, координаты вектора ΔX_N равны одному из двух чисел: $+1$ или -1 , т. е.

$$x_i^{(N)} \in \{+1, -1\}. \quad (0.2.12)$$

При таком поиске векторы смещения ΔX_N направлены к вершинам n -мерного гиперкуба. Число возможных реализаций вектора смещения ΔX для такого поиска равно числу вершин n -мерного гиперкуба, т. е.

$$m = 2^n. \quad (0.2.13)$$

Векторы возможных смещений ΔX занумеруем в следующем порядке:

$$\Delta X^{(i+m/2)} = -\Delta X^{(i)} \quad (0.2.14)$$

$$\left(i = 1, 2, \dots, \frac{m}{2} \right).$$

Запись алгоритмов (0.2.2) и (0.2.4) для дискретно распределенного поиска остается такой, как для непрерывно распределяемого поиска, только Ξ здесь представляет собой случайный вектор, образующийся путем равновероятного выбора одного вектора смещения ΔX из 2^n возможных.

К дискретно распределенному поиску применима идея самообучения [1, 3, 5, 6, 9]. При алгоритме покоординат-

ного самообучения для каждой координаты на N -м шаге задаются вероятности положительного и отрицательного смещения:

$$\Delta x_i^{(N)} = \begin{cases} +1 & \text{с вероятностью } p_i^{(N)}; \\ -1 & \text{с вероятностью } 1 - p_i^{(N)} \end{cases} \quad (0.2.15)$$

$$(i = 1, \dots, n).$$

Вероятности $p_i^{(N)}$ зависят от i -й координаты вектора памяти $\mathbf{W}_N = (\omega_1^{(N)}, \dots, \omega_n^{(N)})$ на N -м шаге, т. е. от $\omega_i^{(N)}$. Эта зависимость, например, может иметь следующий вид:

$$p_i^{(N)} = \begin{cases} 0 & \text{при } \omega_i^{(N)} < -1; \\ \frac{1}{2} (1 + \omega_i^{(N)}) & \text{при } |\omega_i^{(N)}| < 1; \\ 1 & \text{при } \omega_i^{(N)}, \end{cases} \quad (0.2.16)$$

где $\omega_i^{(N)}$ — i -я координата вектора памяти \mathbf{W}_N на N -м шаге поиска. Следовательно, решающую функцию поиска задают формулы (0.2.15) и (0.2.16). Алгоритм самообучения задается для каждой координаты в виде

$$\omega_i^{(N+1)} = \omega_i^{(N)} - \delta \operatorname{sign} (\Delta Q_N \Delta x_i^{(N)}) \quad (0.2.17)$$

$$(i = 1, \dots, n).$$

На изменение $\omega_i^{(N)}$ накладывается ограничение

$$|\omega_i^{(N)}| \leq d_i \quad (0.2.18)$$

$$(0 < d_i).$$

Другой алгоритм самообучения образует систему поиска с применением оптимизирующих автоматов [2, 12—16]. Здесь параметр памяти $\omega_i^{(N)}$ соответствует состоянию i -го автомата на N -м шаге. Оно принимает одно из дискретных значений, число которых равно числу состояний автомата. Вероятность $p_i^{(N)}$ может принимать значения 1 или 0 в зависимости от значения параметра $\omega_i^{(N)}$. Переход вектора памяти $\omega_i^{(N)}$ из одного состояния в другое от N -го шага поиска к $(N+1)$ -му шагу является случайным. Следовательно, решающая функция поиска задается формулой (0.2.15) и выражением

$$p_i^{(N)} = \begin{cases} \operatorname{sign} \omega_i^{(N)}, & \text{если } \omega_i^{(N)} \geq 0; \\ 1 + \operatorname{sign} \omega_i^{(N)}, & \text{если } \omega_i^{(N)} < 0. \end{cases} \quad (0.2.19)$$

Алгоритм самообучения можно также задать, например, при помощи рекуррентной формулы

$$\begin{aligned} \omega_i^{(N+1)} &= \\ &= \begin{cases} \omega_i^{(N)} - \delta \operatorname{sign}(\Delta Q_N \Delta x_i^{(N)}) & \text{с вероятностью } q_i; \\ \omega_i^{(N)} + \delta \operatorname{sign}(\Delta Q_N \Delta x_i^{(N)}) & \text{с вероятностью } 1 - q_i. \end{cases} \end{aligned} \quad (0.2.20)$$

Если решающую функцию поиска задать формулами (0.2.15) и (0.2.16), а алгоритм самообучения — формулой (0.2.20), то получаем более общий алгоритм, обобщающий как алгоритм покоординатного самообучения, так и систему автоматного поиска.

Покоординатный поиск [17—19] также является дискретно распределенным поиском. При этом поиске одна из координат вектора $\Delta \mathbf{X}_N$ принимает значение $+1$ или -1 , а все другие координаты этого вектора равны нулю, т. е.

$$\begin{aligned} \Delta x_i^{(N)} &\in \{1, 0, -1\} \\ &(i = 1, 2, \dots, n) \end{aligned} \quad (0.2.21)$$

при выполнении условия

$$\sum_{i=1}^n |\Delta x_i^{(N)}| = 1. \quad (0.2.22)$$

Число возможных направлений смещения $\Delta \mathbf{X}$ для покоординатного поиска в этом случае равно

$$m = 2n. \quad (0.2.23)$$

Троичный поиск [20] характеризуется тем, что все координаты вектора $\Delta \mathbf{X}_N$ принимают одно из значений $+1, 0$ или -1 , т. е. имеет место (0.2.21) без ограничения (0.2.22). Число возможных направлений смещения $\Delta \mathbf{X}$ в пространстве \mathbf{X} для троичного поиска равно

$$m = 3^n. \quad (0.2.24)$$

§ 0.3. НЕКОТОРЫЕ СВОЙСТВА ВЕРоятностных АВТОМАТОВ

Под вероятностным автоматом понимается объект [21—24]

$$A = \langle X, Y, U; p(u', y|u, x) \rangle, \quad (0.3.1)$$

где X , Y , U — конечные или счетные множества соответственно входных символов, выходных символов (букв) и состояний автомата;

$$p(u', y/u, x) \quad (0.3.2)$$

— условная вероятность того, что автомат, будучи в состоянии u ($u \in U$), при подаче на вход буквы x ($x \in X$) выдает на выход букву y ($y \in Y$), переходя одновременно в состояние u' ($u' \in U$). При этом функция (0.3.2) должна удовлетворять очевидным условиям

$$p(u', y/u, x) \geq 0 \text{ при всех } u' \in U \text{ и } y \in Y; \quad (0.3.3)$$

$$\sum_{\substack{u' \in U \\ y \in Y}} p(u', y/u, x) = 1 \text{ при всех } u \in U \text{ и } x \in X. \quad (0.3.4)$$

Начальные условия вероятностного автомата определяются распределением вероятностей начального состояния $p^{(0)}(u)$ на множестве состояний $u \in U$, причем очевидно, что

$$p^{(0)}(u) \geq 0; \quad \sum_{u \in U} p^{(0)}(u) = 1. \quad (0.3.5)$$

Содержательное функционирование вероятностного автомата состоит в следующем. В начальный момент времени $t=0$ известно распределение вероятностей начальных состояний автомата $p^{(0)}(u)$.

На вход вероятностного автомата в дискретные моменты времени

$$t = 1, 2, 3, \dots, N, \quad (0.3.6)$$

подается некоторая последовательность

$$x_1 x_2 x_3 \dots x_N \dots \quad (0.3.7)$$

из множества входных символов X . Автомат работает потактно, т. е. в каждый дискретный момент времени выдает на выход некоторый символ y из множества выходных символов Y и переходит одновременно в новое состояние $u' \in U$ в соответствии с функцией условной вероятности (0.3.2). Таким образом, после прохождения N тактов на выходе автомата получается последовательность

$$y_1 y_2 y_3 \dots y_N \quad (0.3.8)$$

и автомат находится в состоянии u_N .

В дальнейшем алгоритмы поиска будут рассмотрены как вероятностные автоматы. При этом будет показано, что различные классы поисковых алгоритмов представляются как различные вероятностные автоматы.

НЕКОТОРЫЕ ТИПЫ ВЕРОЯТНОСТНЫХ АВТОМАТОВ

Вероятностный автомат типа Мили [21, 22, 24]. Вероятностный автомат является вероятностным автоматом типа Мили, если для условной плотности вероятности (0.3.2) выполнено условие

$$p(u', y/u, x) = p_1(u'/u, x) p_2(y/u, x), \quad (0.3.9)$$

т. е. последующее состояние, в которое переходит автомат $u' \in U$, не зависит от выхода автомата $y \in Y$, а выход y не зависит от состояния, в которое переходит автомат для любого значения входа $x \in X$ автомата и для любого состояния автомата $u \in U$. Это означает, что выход автомата и его состояние не зависят друг от друга. Они определяются лишь значениями входа и предыдущего состояния.

Вероятностный автомат типа Мура [21, 22, 24]. Вероятностный автомат является автоматом типа Мура, если всюду, где

$$p(u'/u, x) \neq 0, \quad (0.3.10)$$

выполнено соотношение

$$p(y/u, x, u') = p(y/u'), \quad (0.3.11)$$

следовательно, если выход автомата зависит только от его состояния и не зависит от входа. Таким образом, для вероятностного автомата типа Мура имеем

$$p(u', y/u, x) = p(u'/u, x) p(y/u'), \quad (0.3.12)$$

т. е. такой автомат является частным случаем автомата Мили.

Автомат со случайными реакциями [21, 25]. Вероятностный автомат называется автоматом со случайными реакциями (с детерминированной функцией переходов), если он является автоматом типа Мили и условная вероятность $p(u'/u, x)$ принимает только два значения: 0 и 1.

Отсюда следует, что существует функция $\delta(u, x)$ такая, что

$$p(u'/u, x) = \begin{cases} 1, & \text{если } u' = \delta(u, x); \\ 0 & \text{— в противном случае.} \end{cases} \quad (0.3.13)$$

Это означает, что состояние автомата вполне регулярно зависит от его входа и предыдущего состояния. Стохастичность автомата образуется за счет функции выхода.

Марковский автомат [21, 25]. Вероятностный автомат называется марковским (с детерминированной функцией выходов), если он является автоматом типа Мили и условная вероятность $p(y/u, x)$ принимает только два значения: 0 или 1. Следовательно, существует такая функция $\lambda(u, x)$, что

$$p(y/u, x) = \begin{cases} 1, & \text{если } y = \lambda(u, x); \\ 0 & \text{— в противном случае,} \end{cases} \quad (0.3.14)$$

т. е. вход и состояние детерминированно определяют выход автомата; случайно лишь состояние автомата.

Детерминированный автомат является частным случаем вероятностного автомата и получается из определения (0.3.1), где условная вероятность $p(u', y/u, x)$ принимает только два значения: 0 или 1.

ВИДЫ ЭКВИВАЛЕНТНОСТИ В ТЕОРИИ ВЕРОЯТНОСТНЫХ АВТОМАТОВ

Ниже будут рассмотрены вопросы эквивалентности различных поисковых алгоритмов. При этом эквивалентность алгоритмов поиска будет сформулирована как эквивалентность их автоматного представления. Поэтому приводим определение и основные свойства эквивалентности автоматов [21, 23, 26, 27].

Эквивалентность состояний вероятностного автомата. Предположим, что вначале автомат находится в состоянии u_0 и на его вход подается последовательность входных символов (0.3.7) длиной N . Тогда на выходе автомата получаем последовательность выходных символов

той же длины (0.3.8). Вероятность появления последовательности (0.3.8), согласно определению (0.3.2), равна

$$p_{u_0}(y_1 y_2 \dots y_N / x_1 x_2 \dots x_N) = \sum_{u_k \in U} \prod_{k=1}^N p(y_k, u_k / u_{k-1}, x_k). \quad (0.3.15)$$

Если для $u_{01} \in U$, $u_{02} \in U$ и для любого положительного N имеет место равенство

$$p_{u_{01}}(y_1 y_2 \dots y_N / x_1 x_2 \dots x_N) = p_{u_{02}}(y_1 y_2 \dots y_N / x_1 x_2 \dots x_N) \quad (0.3.16)$$

для всех

$$x_k \in X; \quad y_k \in Y \quad (0.3.17)$$

$$(k = 1, 2, \dots, N),$$

то состояния u_{01} и u_{02} называются эквивалентными. Физически это означает, что состояния u_{01} и u_{02} неразличимы как внутренние начальные условия, поскольку они приводят к одному и тому же внешне наблюдаемому поведению, или соотношению вход—выход.

Эквивалентность начальных распределений вероятностного автомата. Для каждой пары (x, y) входного и выходного символов данного автомата через $T(y/x)$ обозначим матрицу, ij -элементом которой является

$$t_{ij}(y/x) = p(y, u_j / u_i, x), \quad (0.3.18)$$

т. е. вероятность перехода из состояния i в состояние j и появления на выходе сигнала y при подаче на вход сигнала x . Тогда $\{T(y/x)\}$, где $x \in X$, $y \in Y$, является семейством $m \times m$ матриц с неотрицательными элементами, где m — количество внутренних состояний автомата. Из условия (0.3.4) следует, что матрицы $T(y, x)$ должны быть такими, чтобы для каждого входного сигнала $x \in X$ переходная матрица состояний

$$T(x) = \sum_y T(y/x) \quad (0.3.19)$$

являлась стохастической. Переходная матрица для последовательностей входных (0.3.7) и выходных (0.3.8) символов определяется равенством

$$T(y_1 y_2 \dots y_N / x_1 x_2 \dots x_N) = T(y_1 / x_1) T(y_2 / x_2) \dots T(y_N / x_N). \quad (0.3.20)$$

Если $\mathbf{P}^{(0)} = (p_1^{(0)}, \dots, p_m^{(0)})$ — любое начальное распределение внутренних состояний u_1, u_2, \dots, u_m автомата, то число

$$p_{\mathbf{P}^{(0)}}(y_1 y_2 \dots y_N / x_1 x_2 \dots x_N) = e T' (y_1 y_2 \dots y_N / x_1 x_2 \dots x_N) \mathbf{P}^{(0)} \quad (0.3.21)$$

обозначает вероятность появления последовательности $y_1 y_2 \dots y_N$ на выходе, когда на вход подается последовательность $x_1 x_2 \dots x_N$ при начальном распределении $\mathbf{P}^{(0)}$. Здесь e является m -мерным вектором, элементы которого равны единице; T' — транспонированная матрица T .

Два начальных распределения $\mathbf{P}_1^{(0)}$ и $\mathbf{P}_2^{(0)}$ называются N -эквивалентными, если для любых последовательностей (0.3.7) и (0.3.8) длины N имеем

$$p_{\mathbf{P}_1^{(0)}}(y_1 y_2 \dots y_N / x_1 x_2 \dots x_N) = p_{\mathbf{P}_2^{(0)}}(y_1 y_2 \dots y_N / x_1 x_2 \dots x_N). \quad (0.3.22)$$

Два начальных распределения $\mathbf{P}_1^{(0)}$ и $\mathbf{P}_2^{(0)}$ называются эквивалентными, если они N -эквивалентны для всех N .

Когда распределения $\mathbf{P}_1^{(0)}$ и $\mathbf{P}_2^{(0)}$ являются вырожденными, т. е. элементы векторов $\mathbf{P}_1^{(0)}$ и $\mathbf{P}_2^{(0)}$ равны нулю или единице, тогда эквивалентность начальных распределений совпадает с эквивалентностью состояний.

Достаточным условием [23] эквивалентности начальных распределений $\mathbf{P}_1^{(0)}$ и $\mathbf{P}_2^{(0)}$ является их $(m-1)$ -эквивалентность, где m — количество внутренних состояний автомата.

Эквивалентность двух вероятностных автоматов по заданным начальным распределениям. Пусть даны два вероятностных автомата A_1 и A_2 с матрицами переходов $T_1(y/x)$ и $T_2(y/x)$. Пусть автомат A_1 имеет m_1 внутренних состояний, а автомат A_2 m_2 внутренних состояний. Предположим, что $\mathbf{P}_1^{(0)}$ есть начальное распределение состояний для автомата A_1 , а $\mathbf{P}_2^{(0)}$ — то же для автомата A_2 . Обозначим вероятность появления на выходе автомата A_i последовательности $y_1 y_2 \dots y_N$ при условии,

что на вход была подана последовательность $x_1 x_2 \dots x_N$ и начальное распределение состояний автомата было равно $\mathbf{P}_i^{(0)}$, величиной $p_{\mathbf{P}_i^{(0)}}^{T_i}(y_1 y_2 \dots y_N / x_1 x_2 \dots x_N)$;

$$\begin{aligned} p_{\mathbf{P}_i^{(0)}}^{T_i}(y_1 y_2 \dots y_N / x_1 x_2 \dots x_N) &= \\ &= \mathbf{e}_i T'_i(y_1 y_2 \dots y_N / x_1 x_2 \dots x_N) \mathbf{P}_i^{(0)}, \end{aligned} \quad (0.3.23)$$

где \mathbf{e}_i — m_i -мерный вектор, элементы которого равны единице;

T'_i — транспонированная матрица T_i ($i=1, 2$).

Если для всех входных последовательностей $x_1 x_2 \dots x_N$ и выходных последовательностей $y_1 y_2 \dots y_N$, имеющих длину N ,

$$\begin{aligned} p_{\mathbf{P}_1^{(0)}}^{T_1}(y_1 y_2 \dots y_N / x_1 x_2 \dots x_N) &= \\ &= p_{\mathbf{P}_2^{(0)}}^{T_2}(y_1 y_2 \dots y_N / x_1 x_2 \dots x_N), \end{aligned} \quad (0.3.24)$$

то начальное распределение $\mathbf{P}_1^{(0)}$ для автомата A_1 называется N -эквивалентным начальному распределению $\mathbf{P}_2^{(0)}$ для автомата A_2 . Или, иначе, система $(T_1, \mathbf{P}_1^{(0)})$, т. е. система «переходная матрица T_1 и начальное распределение $\mathbf{P}_1^{(0)}$ », называется N -эквивалентной системе $(T_2, \mathbf{P}_2^{(0)})$, т. е. системе «переходная матрица T_2 и начальное распределение $\mathbf{P}_2^{(0)}$ ».

Системы $(T_1, \mathbf{P}_1^{(0)})$ и $(T_2, \mathbf{P}_2^{(0)})$ называются эквивалентными, если они N -эквивалентны для всех N .

Достаточным условием [23] эквивалентности систем $(T_1, \mathbf{P}_1^{(0)})$ и $(T_2, \mathbf{P}_2^{(0)})$ является их $(m_1 + m_2 - 1)$ -эквивалентность, где m_1 — количество внутренних состояний автомата A_1 , а m_2 — автомата A_2 .

Эквивалентность автоматов по состояниям. Автоматы A_1 и A_2 называются эквивалентными по состояниям [22], если для каждого $u_i \in \{U^{A_1}\}$ существует такое $u_j \in \{U^{A_2}\}$, что

$$\begin{aligned} p_{u_i}^{A_1}(y_1 y_2 \dots y_N \dots / x_1 x_2 \dots x_N \dots) &= \\ &= p_{u_j}^{A_2}(y_1 y_2 \dots y_N \dots / x_1 x_2 \dots x_N \dots), \end{aligned} \quad (0.3.25)$$

и наоборот, для каждого $u_k \in \{U^{A_2}\}$ существует такое $u_l \in \{U^{A_1}\}$, что

$$\begin{aligned}
 p_{u_{i_1}^{A_2}}(y_1 y_2 \dots y_N \dots / x_1 x_2 \dots x_N \dots) = \\
 = p_{u_{i_1}^{A_1}}(y_1 y_2 \dots y_N \dots / x_1 x_2 \dots x_N \dots)
 \end{aligned}
 \quad (0.3.26)$$

при всех парах входных и выходных последовательностей $x_1 x_2 \dots x_N \dots$ и $y_1 y_2 \dots y_N \dots$. Здесь $\{U^{A_1}\}$ и $\{U^{A_2}\}$ — множества внутренних состояний автоматов A_1 и A_2 соответственно; $p_{u_{i_1}^{A_1}}(y_1 y_2 \dots / x_1 x_2 \dots)$ — вероятность появления на выходе автомата A_i ($i=1, 2$) последовательности $y_1 y_2 \dots$, если на вход была подана последовательность $x_1 x_2 \dots$ и в начальный момент автомат A_i находился в состоянии u_j .

Стохастическая эквивалентность автоматов. Вероятностные автоматы A_1 и A_2 называются стохастически эквивалентными, если для каждого начального распределения $P_i^{(0)} \in \{P_{A_1}^{(0)}\}$ найдется такое начальное распределение $P_j^{(0)} \in \{P_{A_2}^{(0)}\}$, что

$$\begin{aligned}
 p_{P_i^{(0)}}^{T_1}(y_1 y_2 \dots y_N / x_1 x_2 \dots x_N) = \\
 = p_{P_j^{(0)}}^{T_2}(y_1 y_2 \dots y_N / x_1 x_2 \dots x_N)
 \end{aligned}
 \quad (0.3.27)$$

и для каждого начального распределения $P_h^{(0)} \in \{P_{A_2}^{(0)}\}$ найдется такое начальное распределение $P_l^{(0)} \in \{P_{A_1}^{(0)}\}$, что

$$\begin{aligned}
 p_{P_h^{(0)}}^{T_2}(y_1 y_2 \dots y_N / x_1 x_2 \dots x_N) = \\
 = p_{P_l^{(0)}}^{T_1}(y_1 y_2 \dots y_N / x_1 x_2 \dots x_N)
 \end{aligned}
 \quad (0.3.28)$$

для всех пар входных и выходных последовательностей $x_1 x_2 \dots x_N$ и $y_1 y_2 \dots y_N$ и для всех N . Здесь $\{P_{A_1}^{(0)}\}$ и $\{P_{A_2}^{(0)}\}$ обозначают класс начальных распределений автоматов A_1 и A_2 соответственно.

В некоторых случаях для установления эквивалентности автоматов удобно пользоваться понятием гомоморфизма автоматов.

Гомоморфизм вероятностных автоматов [22]. Вероятностный автомат $A_1 = \langle X, Y, \{T_1(X/Y)\} \rangle$ гомоморфно отображается на вероятностном автомате $A_2 = \langle X, Y, \{T_2(X/Y)\} \rangle$, если существует такая матрица B , число строк которой равно числу внутренних состояний автомата A_1 , число столбцов — числу внутренних состояний

автомата A_2 , все строки матрицы B стохастические, а все ее столбцы линейно независимые, и для всех $x \in X$, $y \in Y$ справедливо равенство

$$T_1(y_1/x)B = BT_2(y/x). \quad (0.3.29)$$

Если вероятностный автомат A_1 гомоморфно отображается на вероятностном автомате A_2 , то они стохастически эквивалентны, причем начальному вектору $\mathbf{P}_1^{(0)}$ эквивалентен начальный вектор $\mathbf{P}_2^{(0)}$, т. е.

$$\mathbf{P}_2^{(0)} = \mathbf{P}_1^{(0)}B. \quad (0.3.30)$$

Если матрица гомоморфизма B состоит только из нулей и единиц, то автоматы A_1 и A_2 эквивалентны по состояниям.

Приведем еще некоторые положения, которые понадобятся в дальнейшем при исследовании эквивалентности алгоритмов поиска.

1. Для того чтобы два вероятностных автомата A_1 и A_2 со случайными реакциями были эквивалентны по состояниям, необходимо и достаточно, чтобы существовал автомат A_3 со случайными реакциями, на котором гомоморфно отображаются автоматы A_1 и A_2 [21].

2. Любой вероятностный автомат стохастически эквивалентен некоторому вероятностному автомату Мура. При этом всякий вероятностный автомат со случайными реакциями эквивалентен вероятностному автомату Мура со случайными реакциями [21].

3. Для всякого вероятностного автомата типа Мили, имеющего m внутренних состояний и l входных символов, можно построить эквивалентный ему по состояниям вероятностный автомат типа Мура, имеющий $l(m+1)$ внутренних состояний [21].

§ 0.4. АВТОМАТЫ В СЛУЧАЙНЫХ СРЕДАХ

Модель объекта в общем случае можно представить как некоторую случайную среду, с которой взаимодействует автомат — алгоритм поиска. Поэтому естественно рассмотреть основные положения и результаты, полученные в области исследования поведения ав-

томатов в случайных средах [28], учитывая, что имеется в виду автомат поиска.

Некоторые дополнительные определения. Рассмотрим автомат типа Мура. Задаем этот автомат его каноническими уравнениями

$$u(t+1) = \Phi(u(t), x(t+1)); \quad (0.4.1)$$

$$y(t) = F(u(t)). \quad (0.4.2)$$

Переменная t здесь обозначает время. Обычно она принимает целочисленные значения $t=1, 2, \dots, N, \dots$. Предполагается, что входная переменная автомата $x(t)$ может принимать лишь два значения: 0 и 1, т. е.

$$X = \{0, 1\}. \quad (0.4.3)$$

Значение $x=0$ соответствует единичному выигрышу автомата, а значение $x=1$ — его единичному проигрышу. Предполагается, что выходная переменная $y(t)$ автомата A может принимать κ различных значений $y_1, y_2, \dots, y_\kappa$.

Значения переменной $y(t)$ будем также называть действиями автомата. В момент времени t автомат A произвел действие y_α , если $y(t) = y_\alpha$ ($\alpha=1, 2, \dots, \kappa$). Автомат может находиться в некотором из m его состояний u_1, \dots, u_m . Число m называется памятью автомата. Очевидно, что для автомата с детерминированной функцией выхода $F(u(t))$ $m \geq \kappa$. Автомат A в момент времени t находится в j -м состоянии ($j=1, 2, \dots, m$), если $u(t) = u_j$. Действие y_α соответствует состоянию u_j , если $F(u_j) = y_\alpha$.

Уравнение (0.4.2) описывает зависимость действия автомата от его состояния, а уравнение (0.4.1) — изменения его состояния под воздействием входной переменной $x(t)$. В случае детерминированных автоматов $\Phi(u(t), x(t+1))$ и $F(u(t))$ являются детерминированными функциями, а в случае вероятностных автоматов — функциями плотности вероятностей.

Входная переменная $x(t)$ принимает лишь два значения: 0 и 1. Поэтому формула (0.4.1) задает пару «отображений в себе» множества состояний автомата. Эти отображения будем записывать в виде переходных матриц автомата:

$$\|a_{ij}(x)\| \quad (0.4.4)$$

$$(i, j=1, 2, \dots, m; x=0, 1).$$

Переходные матрицы детерминированного автомата просты: каждая ее строка при любом фиксированном значении x содержит только один элемент, равный единице, а остальные элементы равны нулю. Переходы детерминированного автомата из одного состояния в другое определяются следующим образом: если в момент t автомат находился в состоянии u_i , то в момент $t+1$ он перейдет в такое состояние u_j , для которого

$$a_{ij}(x(t+1)) = 1. \quad (0.4.5)$$

Для вероятностного (стохастического) автомата переходные матрицы являются стохастическими. Элементы переходных матриц $a_{ij}(x)$ обозначают вероятность перехода из i -го состояния в j -е при заданном значении входной переменной x , т. е.

$$a_{ij}(x) = \text{ВЕР}(u(t+1) = u_j / u(t) = u_i, x). \quad (0.4.6)$$

Детерминированный автомат является частным случаем вероятностного автомата.

Случайная среда. Обычно различают стационарную и составную случайные среды. Говорят, что автомат находится в стационарной случайной среде [28]

$$C = (a_1, a_2, \dots, a_n), \quad (0.4.7)$$

если действия автомата и значения его входной переменной связаны следующим образом: действие y_α ($\alpha = 1, 2, \dots, n$), произведенное автоматом в момент t , влечет за собой в момент $t+1$ значение $x=1$ (проигрыш, штраф) с вероятностью

$$s_\alpha = \frac{1 - a_\alpha}{2} \quad (0.4.8)$$

и $x=0$ (выигрыш) с вероятностью

$$1 - s_\alpha = \frac{1 + a_\alpha}{2}. \quad (0.4.9)$$

Здесь a_α — средний выигрыш автомата при действии y_α . При этом предполагается, что

$$|a_\alpha| \leq 1. \quad (0.4.10)$$

Пусть в момент t автомат находится в состоянии u_i ($i=1, 2, \dots, m$), которому соответствует действие $y_{\alpha_i} =$

$=F(u_i)$. Тогда вероятность p_{ij} перехода автомата из состояния u_i в состояние u_j определяется формулой

$$p_{ij} = s\alpha_i a_{ij}(1) + (1 - s\alpha_i) a_{ij}(0) \quad (0.4.11)$$

$$(i, j = 1, 2, \dots, m).$$

Нетрудно убедиться, что матрица

$$A(S) = \|p_{ij}\| \quad (0.4.12)$$

является стохастической. Следовательно, функционирование автомата в стационарной случайной среде описывается цепью Маркова [29, 30].

Введем диагональную матрицу размерности $m \times m$, элементами которой являются вероятности штрафов действия автомата [14, 31]:

$$S = \begin{vmatrix} s_1 & 0 & 0 & \dots & 0 & 0 \\ 0 & s_2 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 0 & s_m \end{vmatrix} \quad (0.4.13)$$

где s_j ($j=1, \dots, m$) определяется формулой (0.4.8), а m — память автомата.

Тогда при учете формулы (0.4.11) имеем цепь Маркова с матрицей переходных вероятностей

$$A(S) = SA_1 + (I - S)A_0, \quad (0.4.14)$$

где A_1 и A_0 — переходные матрицы автомата при проигрыше и выигрыше соответственно.

Обозначим через p_i ($i=1, \dots, m$) финальную вероятность состояния u_i автомата, находящегося в стационарной случайной среде C , а через σ_α ($\alpha=1, 2, \dots, \kappa$) — сумму финальных вероятностей таких состояний u_i , которым соответствует действие y_α . σ_α имеет смысл вероятности действия y_α автомата A в среде C . Тогда в случае эргодической цепи Маркова (0.4.14) математическое ожидание $W(A, C)$ выигрыша для автомата A в среде C выражается формулой

$$W(A, C) = \sum_{\alpha=1}^{\kappa} \sigma_\alpha a_\alpha. \quad (0.4.15)$$

Будем говорить, что автомат A обладает целесообразным поведением в среде C , если

$$W(A, C) > \frac{1}{\kappa} (a_1 + a_2 + \dots + a_\kappa). \quad (0.4.16)$$

Для автомата, совершающего свои действия равновероятно и независимо от реакции среды,

$$W = \frac{1}{\kappa} (a_1 + a_2 + \dots + a_\kappa). \quad (0.4.17)$$

Очевидно, что $W(A, C) \geq W$. В работе [28] рассмотрены множества автоматов с целесообразным поведением в стационарных случайных средах.

Рассмотрим поведение автомата в составной случайной среде [28], т. е. в среде, свойства которой изменяются случайным образом. Будем считать, что среда, с которой взаимодействует автомат, состоит из стационарных случайных сред $C^{(\alpha)}$, переключение которых осуществляется по марковской схеме.

Рассмотрим цепь Маркова

$$K(C^{(1)}, \dots, C^{(v)}, \Delta), \quad (0.4.18)$$

имеющую v состояний $C^{(1)}, \dots, C^{(v)}$, причем переходы из одного состояния в другое задаются матрицей переходных вероятностей

$$\Delta = \begin{vmatrix} \delta_{11} & \delta_{12} & \dots & \delta_{1v} \\ \dots & \dots & \dots & \dots \\ \delta_{v1} & \delta_{v2} & \dots & \delta_{vv} \end{vmatrix} \quad (0.4.19)$$

Состояние $C^{(\alpha)}$ цепи Маркова соответствует стационарной случайной среде

$$C^{(\alpha)} = C(a_1^{(\alpha)}, a_2^{(\alpha)}, \dots, a_\kappa^{(\alpha)}). \quad (0.4.20)$$

Будем говорить, что автомат находится в составной среде K , если в каждый момент времени он находится в одной из сред $C^{(\alpha)}$ ($\alpha = 1, 2, \dots, v$), т. е. если его действия y и значения входной переменной x связаны формулами (0.4.8) и (0.4.9); если при этом в момент t автомат находился в среде $C^{(\alpha)}$, то в момент $t+1$ он будет находиться в среде $C^{(\beta)}$ с вероятностью $\delta_{\alpha\beta}$.

Обозначим через $\psi_i^{(\beta)}$ ($\beta=1, \dots, v; i=1, 2, \dots, m$) такое состояние системы автомат — составная среда, при котором автомат находится в состоянии u_i , а составная среда — в состоянии $C^{(\beta)}$. Тогда вероятность $p_{ij}^{(\beta)(\gamma)}$ перехода системы из состояния $\psi_i^{(\beta)}$ в состояние $\psi_j^{(\gamma)}$ выражается формулой

$$p_{ij}^{(\beta)(\gamma)} = [p_{\alpha_i}^{(\beta)} a_{ij}(1) + (1 - p_{\alpha_i}^{(\beta)}) a_{ij}(0)] \delta_{\beta\gamma}, \quad (0.4.21)$$

где $\|a_{ij}(x)\|$ — переходные матрицы ($x=0, 1$) автомата A ;

$p_{\alpha_i}^{(\beta)}$ и $1 - p_{\alpha_i}^{(\beta)}$ — соответственно вероятности проигрыша и выигрыша автомата A в среде $C^{(\beta)}$ при действии $y_{\alpha_i} = F(u_i)$;

$$p_{\alpha_i}^{(\beta)} = \frac{1 - a_{\alpha_i}}{2} \quad (0.4.22)$$

и

$$1 - p_{\alpha_i}^{(\beta)} = \frac{1 + a_{\alpha_i}^{(\beta)}}{2}. \quad (0.4.23)$$

Матрица

$$A(x^{(\beta)}, \Delta) = \|p_{ij}^{(\beta, \gamma)}\| \quad (0.4.24)$$

$(\beta, \gamma = 1, \dots, v; i, j = 1, \dots, m)$

является стохастической. Следовательно, поведение системы автомат — составная среда описывается некоторой цепью Маркова. В случае эргодической цепи Маркова (0.4.24) математическое ожидание выигрыша $W(A, K)$ автомата A в среде K вычисляется по формуле

$$W(A, K) = \sum_{\alpha=1}^{\kappa} \sum_{\beta=1}^v \sigma_{\alpha}^{(\beta)} a_{\alpha}^{(\beta)}. \quad (0.4.25)$$

Здесь $\sigma_{\alpha}^{(\beta)}$ — суммарная вероятность таких состояний системы, в которых автомат производит действие y_{α} , а составная среда находится в состоянии $C^{(\beta)}$.

Простейшая составная случайная среда, когда $v=2$, задается матрицей переходных вероятностей

$$\Delta = \left\| \begin{array}{cc} 1-\delta & \delta \\ \delta & 1-\delta \end{array} \right\| \quad (0.4.26)$$

Здесь параметр δ представляет собой среднюю частоту переключения состояний составной среды.

Автоматы с формируемой структурой в случайных средах [28, 32, 33]. Структура автомата задается переходными матрицами $\|a_{ij}(x)\|$ ($x=0,1$), которые определяют переходы состояний $u(t) \in \{u_1, u_2, \dots, u_m\}$ автомата при том или ином значении входной переменной x , и функцией $y(t) = F(u(t))$, определяющей действия $y(t) \in \{y_1, y_2, \dots, y_n\}$ автомата, или, в случае автомата со случайными действиями, вероятности действий автомата в зависимости от его состояний.

Для автоматов с формируемой структурой матрицы переходов изменяются в зависимости от значений входной переменной x .

Изменения переходных матриц $\|a_{ij}(x)\|$ осуществляются следующим образом. Пусть в момент t под воздействием входной переменной $x(t)$ состояние u_i переходит в состояние u_j , а затем, в момент $t+1$, входная переменная принимает значение $x(t+1)=0$ (выигрыш) либо $x(t+1)=1$ (проигрыш). Величина переходной вероятности $a_{ij}(t+1)$ увеличивается в случае $x(t+1)=0$ и уменьшается при $x(t+1)=1$, а остальные элементы $a_{ik}(t, x(t))$ ($k \neq j$) i -й строки изменяются таким образом, чтобы сохранилась стохастичность матрицы, т. е. чтобы соблюдалось условие

$$\sum_{i=1}^m a_{ii}(t+1, x(t)) = 1. \quad (0.4.27)$$

Остальные строки матрицы $\|a_{ij}(x)\|$ не изменяются. В каждый момент времени t изменяется лишь одна из матриц $\|a(t, x(t))\|$ ($x=0,1$), и именно та, которая соответствует значению x , равному $x(t)$.

Предложено несколько алгоритмов изменения элементов переходных матриц [32—36]. Например, в работе [32] рассмотрены следующие, весьма общие, формулы изменения переходных вероятностей:

$$a_{ij}^{x(t)}(t+1) = \frac{a^{x(t+1)} a_{ij}^{x(t)}(t)}{a^{[1-x(t+1)]} + (-1)^{x(t+1)} a_{ij}^{x(t)}(1-a)}; \quad (0.4.28)$$

$$a_{ik}^{x(t)}(t+1) = \frac{a^{[1-x(t+1)]} a_{ij}^{x(t)}(t)}{a^{[1-x(t+1)]} + (-1)^{x(t+1)} a_{ij}^{x(t)}(1-a)} \quad (0.4.29)$$

$(k \neq j).$

Здесь $a = \text{const}$; $0 < a < 1$.

Некоторые основные сведения о цепях Маркова. Из изложенного видно, что функционирование автоматов в случайной среде описывается цепью Маркова. Поэтому исследование автомата в случайной среде сводится к исследованию соответствующей цепи Маркова. Приводим основные положения о цепях Маркова [29, 30].

Под цепью Маркова будем понимать некоторую систему U , которая в каждый момент времени может находиться в одном из m состояний u_1, u_2, \dots, u_m и меняет свое состояние только в дискретные моменты времени t_1, t_2, \dots, t_N . Вероятностный переход системы U из одного состояния в другое задается некоторой переходной вероятностью $p_{ij}(N)$. Это вероятность перехода из i -го состояния в j -е в момент времени $t=N$. В общем случае, когда вероятности переходов $p_{ij}(N)$ зависят от времени $t=N$, цепи называются неоднородными цепями Маркова. Если p_{ij} не зависят от времени t_N , т. е.

$$p_{ij}(N) = p_{ij}, \quad (0.4.30)$$

то цепь Маркова называется однородной. Такая цепь задается матрицей переходных вероятностей

$$\pi = \begin{vmatrix} p_{11} & \dots & p_{1m} \\ \dots & \dots & \dots \\ p_{m1} & \dots & p_{mm} \end{vmatrix} \quad (0.4.31)$$

где p_{ij} ($i, j=1, \dots, m$) — вероятность перехода из i -го состояния в j -е. При этом должно выполняться условие стохастичности матрицы π , т. е.

$$\sum_{j=1}^m p_{ij} = 1 \quad (0.4.32)$$

$$(i=1, \dots, m).$$

Для полного задания цепи Маркова необходимо иметь также вектор начальных вероятностей

$$\mathbf{P}^{(0)} = (p_1^{(0)}, \dots, p_m^{(0)}), \quad (0.4.33)$$

где $p_i^{(0)}$ ($i=1, \dots, m$) — вероятность того, что система U в момент времени $t=0$ находится в состоянии u_i .

Вероятности переходов системы из одного состояния в другое за N шагов определяются формулой

$$\mathbf{P}^{(N)} = (\pi')^{(N)} \mathbf{P}^{(0)}. \quad (0.4.34)$$

Здесь $(\pi')^{(N)}$ — N -я степень транспонированной матрицы π' ;
 $\mathbf{P}^{(0)}$ — вектор начальных вероятностей
 (0.4.33);

$$\mathbf{P}^{(N)} = (p_1^{(N)}, \dots, p_m^{(N)}), \quad (0.4.35)$$

где $p_i^{(N)}$ — вероятность того, что система находится в i -м состоянии после N шагов.

Элементы $p_{ij}^{(N)}$ матрицы $\pi^{(N)}$ называются переходными вероятностями за N шагов. Их можно определить по формуле Перрона [37]

$$p_{ij}^{(N)} = \sum_{\nu=1}^r \frac{1}{(m_{\nu}-1)!} D_{\lambda}^{m_{\nu}-1} \left[\frac{\lambda^{(N)} \pi_{ji}(\lambda)}{\psi_{\nu}(\lambda)} \right] \Big|_{\lambda=\lambda_{\nu}} \quad (\nu=1, \dots, r). \quad (0.4.36)$$

Обозначим через λ_{ν} характеристические числа матрицы π , т. е. корни характеристического полинома матрицы π , который задается определителем

$$\pi(\lambda) = \begin{vmatrix} \lambda - p_{11} & -p_{12} & \dots & -p_{1m} \\ -p_{21} & \lambda - p_{22} & \dots & -p_{2m} \\ \dots & \dots & \dots & \dots \\ -p_{m1} & -p_{m2} & \dots & \lambda - p_{mm} \end{vmatrix} \quad (0.4.37)$$

В формуле (0.4.36)

$\pi_{ji}(\lambda)$ — алгебраическое дополнение элемента определителя $\pi(\lambda)$, стоящего на пересечении его j -й строки и i -го столбца;

m_{ν} — кратность ν -го характеристического числа λ_{ν} ;

$D_{\lambda}^{m_{\nu}-1} [\]$ — производная по переменной λ порядка $m_{\nu}-1$; подстановка $\lambda = \lambda_{\nu}$ производится после дифференцирования;

$$\psi_{\nu}(\lambda) = \frac{\pi(\lambda)}{(\lambda - \lambda_{\nu})^{m_{\nu}}}. \quad (0.4.38)$$

Особенно простой вид формула Перрона (0.4.36) приобретает в случае, когда все характеристические числа матрицы π (0.4.37) имеют кратность, равную единице, т. е. когда

$$m_1 = m_2 = \dots = m_r = 1. \quad (0.4.39)$$

В этом случае $r = n$ и формула Перрона имеет вид

$$p_{ij}^{(N)} = \sum_{\nu=1}^m \frac{\lambda_{\nu}^{(N)} \pi_{ji}(\lambda_{\nu})}{\psi_{\nu}(\lambda_{\nu})} \quad (i, j = 1, 2, \dots, n). \quad (0.4.40)$$

Если существует предел

$$\lim_{N \rightarrow \infty} p_{ij}^{(N)} = p_{ij}^{(\infty)}, \quad (0.4.41)$$

то вероятности $p_{ij}^{(\infty)}$ называются предельными переходными вероятностями. Имея вектор начальных вероятностей $\mathbf{P}^{(0)} = (p_1^{(0)}, \dots, p_m^{(0)})$, можно получить предельные вероятности $p_i^{(\infty)}$ ($i = 1, \dots, m$) различных состояний цепи Маркова по формулам

$$p_i^{(\infty)} = \sum_{j=1}^m p_j^{(0)} p_{ji}^{(\infty)}. \quad (0.4.42)$$

Если предельные вероятности цепи Маркова $p_i^{(\infty)}$ ($i = 1, \dots, m$) не зависят от ее начальных вероятностей $p_i^{(0)}$ ($i = 1, \dots, m$), то такая цепь Маркова называется эргодической. Легко видеть, что цепь Маркова является эргодической, если вероятности $p_{ij}^{(\infty)}$ для любого индекса j не зависят от индекса i , т. е. если все строки матрицы $\pi^{(\infty)} = \|\pi_{ij}^{(\infty)}\|$ предельных переходных вероятностей одинаковы. Для того чтобы цепь Маркова была эргодической, необходимо и достаточно, чтобы одним из значений простого корня характеристического полинома ее матрицы переходных вероятностей была единица, а модули всех других корней этого полинома были строго меньше единицы.

Для случая эргодической конечной цепи Маркова предельные переходные вероятности определяются формулами

$$p_{ij}^{(\infty)} = \frac{\pi_{ji}(\lambda)}{\psi_1(\lambda)} \quad (0.4.43)$$

$$(i, j = 1, \dots, m).$$

Приведем еще другой признак, по которому определяется эргодичность цепи Маркова. Этот признак следующий. Если начиная с некоторого $N \geq 1$ все элементы N -й степени матрицы переходных вероятностей π положительны, то цепь Маркова является эргодической и ее предельные вероятности равны

$$p_j^{(\infty)} = \lim_{N \rightarrow \infty} p_{ij}^{(N)}. \quad (0.4.44)$$

На практике использование формул Перрона или возведение переходных матриц в N -ю степень для цепей Маркова со многими состояниями является сложной задачей. Поэтому более целесообразно предельные вероятности $p_i^{(\infty)}$ находить как решение следующей системы алгебраических уравнений [38]:

$$\mathbf{P} = \mathbf{P}\pi, \quad (0.4.45)$$

где π — матрица (0.4.31);

\mathbf{P} — вектор предельных вероятностей; $\mathbf{P} = (p_1, \dots, p_m)$.
Формула (0.4.45) имеет место для эргодических цепей Маркова.

ПОИСК БЕЗ САМООБУЧЕНИЯ

§ 1.1. СЛУЧАЙНЫЙ ПОИСК КАК ВЕРОЯТНОСТНЫЙ АВТОМАТ

Рассмотрим задачу минимизации скалярной функции многих переменных

$$Q = Q(x_1, x_2, \dots, x_n) \quad (1.1.1)$$

методами случайного поиска как задачу функционирования вероятностного автомата в некоторой среде [39]. Автоматом в данном случае является алгоритм случайного поиска, а среда, в которой действует этот автомат, представляет собой объект оптимизации. Среда в общем случае предполагается стохастической, т. е. на одно и то же воздействие X она каждый раз отзывается некоторым случайным образом.

Одним из примеров такого взаимодействия автомата и среды может служить гомеостат Эшби [40], представляющий собой динамическую систему

$$\frac{dU}{dt} = F(U, X, E), \quad (1.1.2)$$

Состояние системы (1.1.2) описывается вектором $U = (u_1, u_2, \dots, u_m)$ и определяется как вектором управляемых параметров гомеостата $X = (x_1, x_2, \dots, x_n)$, так и вектором неуправляемых параметров $E = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n)$, характеризующим стохастические свойства среды.

Управление состоянием U гомеостата осуществляется путем воздействия на его параметры x_1, x_2, \dots, x_n , причем целью управления является выведение гомеостата в заданное состояние U^* , т. е. минимизация показателя

$$Q = |U - U^*|. \quad (1.1.3)$$

Управление параметрами гомеостата производится методом проб и ошибок, который сводится, по сути дела, к случайному перебору элементов некоторого допустимого множества управлений $\{X\}$ с последующей проверкой их эффективности и реакции на каждое случайное управление. При этом четко разграничиваются два вида реакций. Отрицательная реакция R^- возникает в ответ на управление, которое не приводит к выполнению поставленных целей. Эта реакция в соответствии с алгоритмом гомеостата вызывает очередную случайную пробу управления. Положительная реакция R^+ следует за достижением цели управления. Она сохраняет в объекте то управление, которое привело к положительному результату. Алгоритм такого поведения гомеостата можно записать в виде

$$X_{i+1} = \begin{cases} E & \text{при } R^-; \\ X_i & \text{при } R^+, \end{cases} \quad (1.1.4)$$

где X_i — управление на i -м шаге работы гомеостата;
 E — оператор случайного управления из класса допустимых управлений, т. е. оператор случайного определения параметров гомеостата.

Легко заметить, что такой алгоритм имеет целесообразное поведение, направленное на поиск и сохранение в системе состояния, которое обеспечивает положительную реакцию R^+ .

Итак, смысл случайного поиска по рассмотренному алгоритму (в данном случае — слепого поиска) заключается в том, чтобы случайно перебирать значения параметров системы до тех пор, пока не будут найдены такие их варианты, которые обеспечивают выполнение определенных заданных условий. В случае гомеостата — это наличие устойчивого состояния системы в заданных границах.

Такое поведение гомеостата, по-видимому, наиболее целесообразно в том случае, когда управляющее устройство не имеет никаких сведений о структуре объекта, т. е. последний представляет собой «черный ящик».

Естественно задать вопрос: всегда ли можно найти условия, при которых объект удовлетворяет целям управления, т. е. можно ли случайным перебором допустимых

управлений всегда и наверняка достигать цели? Разобьем множество возможных реализаций параметров объекта на два подмножества $\{X^{(0)}\}$ и $\{X^*\}$, в первое из которых объединены значения параметров, приводящие к отрицательной реакции, во второе — вызывающие положительную реакцию. Тогда решение задачи управления будет заключаться в случайном отыскании хотя бы одного элемента второго множества за конечное число шагов поиска (под шагом поиска здесь понимается однократный случайный выбор параметров объекта). Для этого необходимо, чтобы оба подмножества имели одинаковую мощность, т. е. чтобы при случайном переборе элементов множества всех возможных управлений (под управлением здесь и далее подразумевается определение параметров объекта x_1, x_2, \dots, x_n) представители подмножества $\{X^*\}$ встречались не слишком редко. Тогда вероятность выбора одного из элементов этого подмножества будет конечна. Следовательно, процесс завершится в конечное время и система обязательно придет в состояние, удовлетворяющее целям управления.

Представим работу гомеостата как функционирование некоторого вероятностного автомата, действующего в случайной среде [41]. Тогда гомеостат следует «рас-слоить» на среду и управляющее устройство УУ (см. рис. 1.1.1). Под средой подразумевается объект управления, реализующий зависимость (1.1.2), а управляющее устройство работает в соответствии с алгоритмом случайного поиска (1.1.4). В этом случае отрицательная реакция R^- , эквивалентная $Q > q = \text{const}$, соответствует штрафу, а положительная реакция $R^+ (Q \leq q)$ — нештрафу [39].

Алгоритм случайного поиска (1.1.4), реализуемый управляющим устройством, является вероятностным автоматом, выход которого X изменяется в соответствии со входом Q . Стохастическая матрица, характеризующая

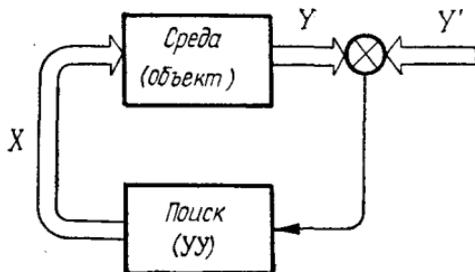


Рис. 1.1.1. Блок-схема гомеостатического управления.

функционирование этого автомата в случайной среде, имеет вид

$$\begin{array}{l} R^- \\ R^+ \end{array} \left\| \begin{array}{cc} 1-p & p \\ 0 & 1 \end{array} \right\| \quad (1.1.5)$$

где p — вероятность отыскания решения, удовлетворяющего поставленным целям управления ($Q \leq q$). Эффективность функционирования гомеостата существенно зависит от величины p . Чем больше p , тем быстрее находится решение и, следовательно, тем эффективнее работа гомеостата.

Стохастический характер данного автомата определяется двумя факторами — оператором случайного шага E и неопределенностью среды (стохастичность среды зависит от неизвестного вектора ситуации E).

Применение описанной схемы слепого поиска к оптимизации многопараметрических систем не приводит к успеху. Это связано с тем, что для более или менее сложных объектов управления вероятность p отыскания цели «сразу» очень мала, и процесс слепого поиска необходимого управления связан, как правило, с огромными затратами времени.

Рассмотренный алгоритм случайного поиска (1.1.3) пригоден для отыскания решения в принципе, т. е. гарантирует конечность времени отыскания условий, удовлетворяющих целям управления. Однако вопросы быстродействия не решаются этим алгоритмом, поскольку он предназначен для управления объектами самого широкого класса с единственным ограничением, связанным с конечностью вероятности отыскания решения.

В задачах экстремального управления часто содержатся дополнительные сведения об объекте, например о характере поведения показателя качества объекта при различных управлениях. Эта информация дает возможность применить новые алгоритмы управления, которые построены на базе случайного гомеостатического поиска, но решают вопросы быстродействия уже с учетом имеющихся дополнительных сведений об объекте.

Дополнительные сведения об объекте оптимизации несет функция качества $Q(\mathbf{X})$, которую нужно минимизировать. Значение $Q(\mathbf{X})$ очень часто может служить мерой близости к цели, т. е. мерой успеха поиска.

Сформулируем один из алгоритмов случайного поиска, который основывается на высказанных соображениях:

$$\Delta X_{i+1} = \begin{cases} \Xi & \text{при } R_i^-, \text{ т. е. если } \Delta Q_i \geq 0; \\ \Delta X_i & \text{при } R_i^+, \text{ т. е. если } \Delta Q_i < 0, \end{cases} \quad (1.1.6)$$

где ΔX_i — рабочее смещение на i -м шаге, а случайные пробы Ξ предполагаются достаточно малыми по модулю, чтобы обеспечить достаточно большую вероятность достижения подцели. Нетрудно заметить, что этот алгоритм является дифференциальным аналогом гомеостатического алгоритма и обычно называется алгоритмом случайного спуска или случайным поиском с линейной тактикой. Смысл его прост и естествен. Система делает случайные шаги в пространстве управляемых параметров, пока не будет найден такой шаг, который приведет к уменьшению функции качества. Положительная реакция алгоритма заключается в движении по выбранному направлению, т. е. в повторении этого шага до тех пор, пока показатель качества не начнет увеличиваться, что вызовет отрицательную реакцию — случайные пробы новых направлений и т. д.

Эффективность такого алгоритма поиска гарантируется условием небыстрого изменения функции $Q(X)$, согласно которому успех в X_1 может повториться в X_2 , если расстояние между X_1 и X_2 не очень велико. На этом основании в алгоритме (1.1.6) предусмотрены повторные шаги в выбранном направлении.

Рассмотрим работу этого алгоритма как стохастического автомата в случайной среде, которая в данном случае представлена объектом оптимизации. Направленный граф работы такого автомата показан на рис. 1.1.2, где Ξ — оператор случайного смещения, а R^+ — оператор положительной реакции. Стрелками на рисунке пред-

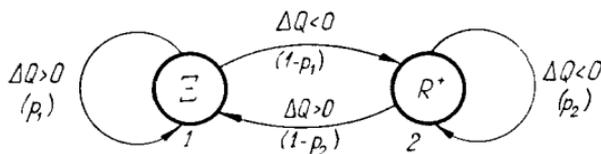


Рис. 1.1.2. Граф гомеостатического алгоритма оптимизации (линейная тактика).

ставлены переходы от одного оператора к другому, причем рядом указаны события, вызывающие эти переходы, а в скобках — вероятности этих событий. Автомат может пребывать в двух состояниях и задается матрицей переходов

$$\begin{matrix} E \\ R^+ \end{matrix} \left\| \begin{matrix} p_1 & 1-p_1 \\ 1-p_2 & p_2 \end{matrix} \right\| \quad (1.1.7)$$

где вероятности p_1 и p_2 зависят от свойств среды, т. е. от оптимизируемого объекта, и изменяются вместе с этим объектом.

Нетрудно заметить, что эффективность рассматриваемого алгоритма зависит от того, как часто применяется оператор R^+ .

Именно в этом цикле показатель качества уменьшается и происходит настройка системы. Поэтому значение вероятности p_2 в значительной мере определяет эффективность применения данного алгоритма. Алгоритм эффективен, если объект таков, что значение p_2 достаточно близко к единице.

Однако эффективность алгоритма зависит и от вероятности выхода автомата на указанный цикл, т. е. от величины $1-p_1$, которая соответствует вероятности случайного определения удачного направления ($\Delta Q < 0$). Достаточно большое значение этой величины обеспечивает работоспособность алгоритма.

Таким образом, для эффективной работы алгоритма случайного поиска (1.1.6) необходимо, чтобы объект оптимизации удовлетворял следующим требованиям: 1) значение p_1 не должно быть очень малым, 2) значение p_2 должно быть велико.

Рассмотренный алгоритм случайного поиска построен на принципе «наказания» случайностью, в соответствии с которым оператор случайного шага E вводится как отрицательная реакция на неудачу при отыскании подцели. В случае удачи поиск осуществляется тем же способом, который привел к удаче. Такая форма поведения, безусловно, разумна и целесообразна для линейных и близких к ним объектов, свойства которых с переходом из одного состояния в другое изменяются незначительно.

Именно поэтому автомат, реализующий подобный ал-

горитм, называется автоматом с линейной тактикой [39]. Если же условие линейности объекта не выполняется и вероятность p_2 мала, то рассмотренный линейный алгоритм поиска имеет ограниченное применение.

Объекты, для которых вероятность p_2 мала, часто бывает целесообразно оптимизировать, применяя алгоритм с нелинейной тактикой. В этом алгоритме отрицательная реакция R^- заключается в регулярном «исправлении» ошибок, т. е. каждый раз принимаются меры для того, чтобы устранить последствия неудачи. Этот алгоритм можно записать в следующем виде:

$$\Delta X_{i+1} = \begin{cases} \Xi & \text{при } R^+; \\ F(\Delta X_i) & \text{при } R^-. \end{cases} \quad (1.1.8)$$

Оператор случайного шага Ξ здесь вводится как положительная реакция R^+ на удачный шаг, который привел к уменьшению функции качества $\Delta Q < 0$. Отрицательная реакция R^- вызывает действие, направленное на преодоление полученного отрицательного эффекта ($\Delta Q > 0$), после чего снова следует случайный шаг Ξ . Таким образом, работоспособность этого алгоритма поиска достигается исправлением ошибок, допущенных в процессе случайных проб.

Граф алгоритма показан на рис. 1.1.3. Матрица переходов графа имеет вид

$$\begin{array}{c} \Xi \\ R^- \end{array} \left\| \begin{array}{cc} p & 1-p \\ 1 & 0 \end{array} \right\| \quad (1.1.9)$$

где p — вероятность случайного удачного шага ($p = p_1$). Очевидно, что для эффективной работы этого алгоритма объект вовсе не обязан обладать свойствами «гладкости». Единственное условие, предъявляемое алгоритмом к объекту, заключается в том, чтобы вероятность p была не очень мала.

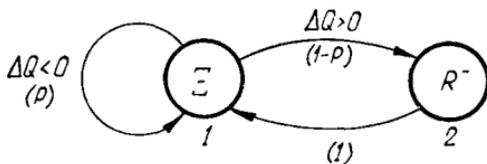


Рис. 1.1.3. Граф алгоритма случайного поиска с нелинейной тактикой.

Рассмотренные выше алгоритмы случайного поиска являются стохастическими автоматами, переходные матрицы которых определяют статистические свойства оператора случайного шага, которые в процессе поиска остаются неизменными. Естественно использовать возможность изменения вероятностных свойств оператора Ξ для целенаправленного воздействия на переходные вероятности, чтобы повысить эффективность указанных алгоритмов. Данную задачу решают алгоритмы самообучения, т. е. способы перестройки статистических свойств оператора Ξ , основанные на предыстории поиска. Такое самообучение вносит в поиск дополнительную обратную связь и является реакцией более высокого порядка на поведение объекта в процессе поиска, чем R^+ или R^- .

Таким образом, можно утверждать, что алгоритмы случайного поиска являются вероятностными автоматами, адаптация которых производится путем введения самообучения в процессе их функционирования.

§ 1.2. ПРЕДСТАВЛЕНИЕ СЛУЧАЙНОГО ПОИСКА В ВИДЕ ВЕРОЯТНОСТНОГО АВТОМАТА

В предыдущем параграфе показано, что алгоритмы случайного поиска могут рассматриваться как простейшие вероятностные автоматы, имеющие два состояния — случайный шаг Ξ и реакцию R на результат этого случайного шага. В данном параграфе эта точка зрения расширяется и обобщается, что позволяет не только привлечь теорию вероятностных автоматов к исследованию работы алгоритмов случайного поиска, но и построить новые алгоритмы.

Начнем с определения исходных понятий, точнее, с установления связи между понятиями теорий поисковой оптимизации и вероятностных автоматов. Эта связь отражена в таблице 1.2.1.

Рассмотрим алгоритм случайного поиска с ограниченным, т. е. конечным, числом m возможных направлений движения в пространстве оптимизируемых параметров.

Таблица 1.2.1

Теория поисковой оптимизации	Теория вероятностных автоматов
Объект	Среда
Оптимизатор, работающий по методу случайного поиска	Вероятностный автомат
Реакция объекта $\Delta Q' \leq 0$ $\Delta Q' > 0$	Входной алфавит автомата $C = (0, 1)$ нештраф ($c=0$) штраф ($c=1$)
$\Delta X_1, \Delta X_2, \dots, \Delta X_m$	Выходной алфавит автомата
Состояние объекта $\text{grad } Q(\mathbf{X}) = \left(\frac{\partial Q}{\partial x_1}, \dots, \frac{\partial Q}{\partial x_n} \right)$	Состояние среды $S = (s_1, \dots, s_m)$

Будем рассматривать сначала покоординатный поиск, когда вдоль координатных осей равновероятно делаются единичные шаги в обоих направлениях

$$\Delta \mathbf{X} = (\pm 1, \pm 1, \dots, \pm 1). \quad (1.2.1)$$

Очевидно, что в этом случае m четно. Определим индексы следующим образом:

$$\Delta \mathbf{X}^{(i)} = -\Delta \mathbf{X}^{(i+m/2)} \quad (1.2.2)$$

$$\left(i \leq \frac{m}{2} \right),$$

т. е. шаги с индексами, различающимися величиной $\frac{m}{2}$, равны по модулю и противоположно направлены.

Покажем теперь, как некоторые описанные ранее алгоритмы покоординатного поиска записываются в терминах теории автоматов.

Нелинейный алгоритм случайного поиска:

$$\Delta X_i = \begin{cases} E, & \text{если } \Delta Q'_{i-1} \leq 0; \\ R, & \text{если } \Delta Q'_{i-1} > 0, \end{cases} \quad (1.2.3)$$

где

$$\Delta Q'_{i-1} = Q'(X_{i-1}) - Q'(X_{i-2}) \quad \text{и} \quad X_{i-1} = X_{i-2} + \Delta X_{i-1}. \quad (1.2.4)$$

Здесь E — оператор случайного шага, который представляет собой равновероятный выбор из $m=2^n$ возможностей (1.2.1);

R — оператор реакции на $\Delta Q' > 0$ (напомним, что рассматривается случай минимизации). Пусть для конкретности R выражает операцию возврата в исходную точку, т. е. $R_i = -\Delta X_{i-1}$.

Блок-схема оптимизации по этому алгоритму показана на рис. 1.2.1, а. Здесь Π — блок памяти предыдущего значения показателя качества; ИМ — исполнительный механизм, выполняющий операцию (1.2.4); буквой E обозначен генератор случайных шагов.

Заметим, что данный алгоритм не тождествен описанному в работе [1]: в последнем после возврата необходимо следует случайный шаг, в алгоритме же (1.2.3) реакция R может повторяться, если повторяется неравенство $\Delta Q' > 0$.

Теперь сформулируем этот алгоритм в терминах функционирования вероятностных автоматов [39].

Вероятностный автомат, описывающий указанный алгоритм случайного поиска, оперирует двумя стохастическими матрицами, соответствующими двум реакциям среды (штраф и нештраф). Нештрафу ($c=0$) соответствует стохастическая матрица выбора случайного шага ΔX :

$$A_0 = \|p_{ij}\| \quad (1.2.5)$$

$$(i, j = 1, \dots, m),$$

где

$$p_{ij} = \frac{1}{m}. \quad (1.2.6)$$

Штрафу ($c=1$) соответствует нестохастическая матрица реакции

$$A_1 = \|a_{ij}\| \quad (1.2.7)$$

$$(i, j = 1, \dots, m),$$

где

$$a_{ij} = \begin{cases} 1 & \text{при} \begin{cases} j = i + \frac{m}{2} & (i = 1, \dots, \frac{m}{2}); \\ j = i - \frac{m}{2} & (i = \frac{m}{2} + 1, \dots, m); \end{cases} \\ 0 & \text{при остальных значениях } i, j. \end{cases} \quad (1.2.8)$$

Иначе говоря, матрица реакции состоит из четырех блоков — двух единичных и двух нулевых:

$$A_1 = \left\| \begin{array}{c|c} 0 & I \\ \hline I & 0 \end{array} \right\| \quad (1.2.9)$$

Нетрудно заметить, что эта матрица обеспечивает возврат системы в исходное состояние при $\Delta Q' > 0$.

Таким образом, вероятностный автомат, эквивалентный алгоритму случайного поиска с возвратом, представляется в виде следующей комбинации матриц:

$$A = (1 - c)A_0 + cA_1. \quad (1.2.10)$$

Блок-схема оптимизации с применением этого вероятностного автомата показана на рис. 1.2.1, б. Здесь блок c выполняет естественную операцию:

$$c = \begin{cases} 1, & \text{если } \Delta Q' > 0; \\ 0, & \text{если } \Delta Q' \leq 0. \end{cases} \quad (1.2.11)$$

Блок A_1 представляет собой автомат, срабатывающий при $c = 1$ в соответствии с матрицей A_1 , т. е. реализующий поведение при штрафе. Блок A_0 реализует вероятностный автомат A_0 , т. е. образует случайный шаг Ξ .

Заметим, что обе матрицы A_0 и A_1 можно объединить в одну следующего вида:

$$A(c) = \|a_{ij}^{\Xi}\|, \quad (1.2.12)$$

где

$$a_{ij}^{\Sigma} = \begin{cases} \frac{1+c(m-1)}{m} & \text{при } \left. \begin{array}{l} j=i+\frac{m}{2} \\ j=i-\frac{m}{2} \end{array} \right\} \quad \left(i=1, \dots, \frac{m}{2} \right); \\ \frac{1-c}{m} & \text{при остальных значениях } i, j. \end{cases} \quad (1.2.13)$$

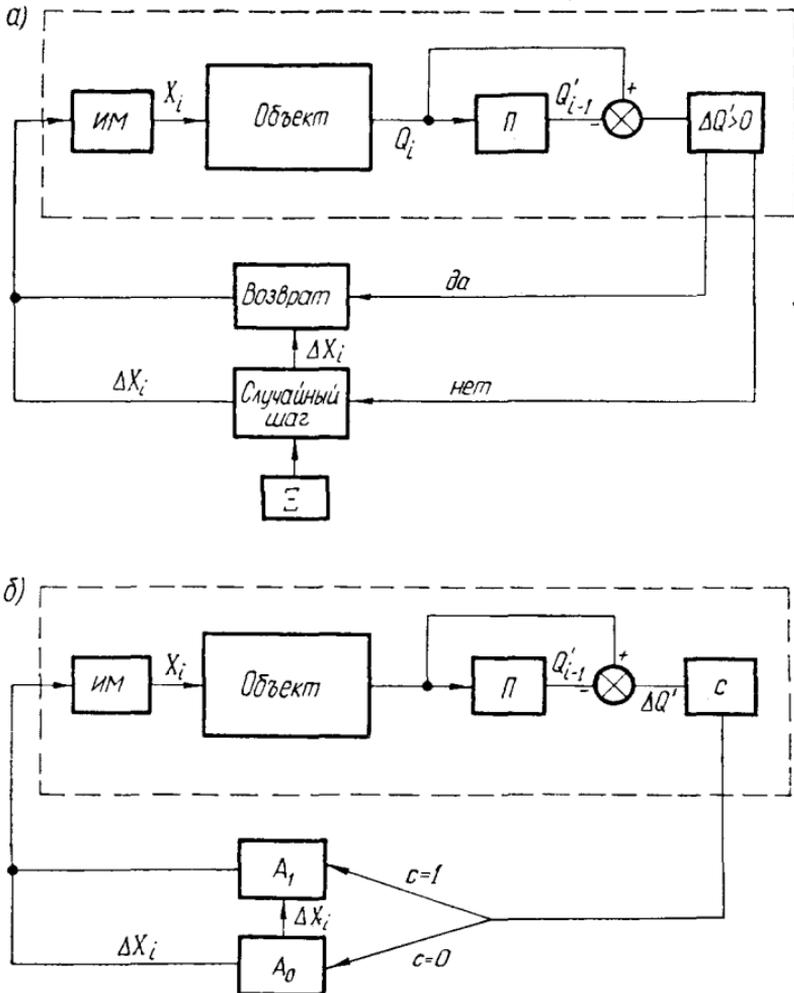


Рис. 1.2.1. Блок-схема оптимизации:
 а — с алгоритмом случайного поиска; б — с применением вероятностного автомата.

Обе схемы на рис. 1.2.1, *a* и *б* работают строго идентично. Поэтому дальнейшее изучение поведения системы оптимизации, работающей по алгоритму случайного поиска (1.2.3), можно производить с привлечением аппарата теорий вероятностных автоматов.

Среда в этой теории описывается вектором

$$\mathbf{S} = (s_1, s_2, \dots, s_m), \quad (1.2.14)$$

где s_i — вероятность штрафа ($c=1$) при $\Delta\mathbf{X}=\Delta\mathbf{X}^{(i)}$.

При отсутствии помех в определении приращения показателя качества получаем

$$\Delta Q^i = (\Delta\mathbf{X}_i, \text{grad } Q(\mathbf{X})), \quad (1.2.15)$$

и вероятность штрафа ($\Delta Q^{(i)} > 0$) равна

$$s_i = \text{sgn } \Delta Q^{(i)} \quad (1.2.16)$$

где

$$\text{sgn } z = \begin{cases} 1, & \text{если } z > 0; \\ 0, & \text{если } z \leq 0. \end{cases}$$

При наличии нормальных некоррелированных помех, накладывающихся на показатель качества, получаем более общую формулу:

$$s_i = \frac{1}{2} \left[1 + \Phi \left(\frac{\Delta Q^{(i)}}{2\sigma} \right) \right], \quad (1.2.17)$$

где $\Phi(z)$ — интеграл вероятности;
 σ^2 — дисперсия нормальной помехи;

$$\Phi(z) = \frac{2}{\pi} \int_0^z e^{-t^2} dt.$$

Очевидно, что с изменением градиента показателя качества вектор \mathbf{S} будет также изменяться.

Далее, для полного описания функционирования системы случайного поиска как вероятностного автомата необходимо установить связь между поведением внешней среды и действиями автомата, т. е. замкнуть цепь обратной связи. Для этого в формулу (1.2.13) вместо коэффициента c подставим вероятность штрафа при i -м действии автомата, т. е. s_i :

$$a_{ij}^{\Sigma} = \begin{cases} \frac{1+s_i(m-1)}{m} & \text{при } \begin{cases} j=i+\frac{m}{2} & (i=1, \dots, \frac{m}{2}); \\ j=i-\frac{m}{2} & (i=\frac{m}{2}+1, \dots, m); \end{cases} \\ \frac{1-s_i}{m} & \text{при остальных значениях } i, j. \end{cases} \quad (1.2.18)$$

Тогда процесс оптимизации случайным поиском описывается простой цепью Маркова со следующей матрицей переходных вероятностей (см. § 0,4):

$$A(S) = (I-S)A_0 + SA_1; \quad (1.2.19)$$

здесь A_0 и A_1 — матрицы (1.2.5) и (1.2.7) соответственно;

I — единичная матрица;
 S — диагональная матрица

$$S = \begin{vmatrix} s_1 & 0 & 0 & \dots & 0 & 0 \\ 0 & s_2 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & s_{m-1} & 0 \\ 0 & 0 & 0 & \dots & 0 & s_m \end{vmatrix} \quad (1.2.20)$$

где s_i — вероятность штрафа при i -м действии автомата.

Располагая этими данными и пользуясь аппаратом теории вероятностных автоматов, можно определить статистические свойства поиска.

Пусть

$$\mathbf{P} = (p_1, p_2, \dots, p_m) \quad (1.2.21)$$

— вектор предельных вероятностей шагов $\Delta\mathbf{X}^{(1)}, \dots, \Delta\mathbf{X}^{(m)}$. Тогда осредненный шаг, определяющий эффективность поиска, равен

$$M[\Delta\mathbf{X}] = \sum_{i=1}^m p_i \Delta\mathbf{X}^{(i)}, \quad (1.2.22)$$

а средняя вероятность штрафа

$$p_{\text{ср}} = \sum_{i=1}^m p_i s_i. \quad (1.2.23)$$

Линейный алгоритм случайного поиска:

$$\Delta X_i = \begin{cases} \Delta X_{i-1}, & \text{если } \Delta Q'_{i-1} \leq 0; \\ \Xi, & \text{если } \Delta Q'_{i-1} > 0, \end{cases} \quad (1.2.24)$$

где Ξ — случайный вектор из множества (1.2.1).

В случае штрафа ($\Delta Q' > 0$, т. е. $c=1$) матрица A_1 совпадает с (1.2.5) и соответствует случайному шагу, т. е. равновероятному выбору одного из m возможных направлений. При «нештрафе» ($\Delta Q' \leq 0$, т. е. $c=0$) матрица A_0 — единичная, что соответствует повторению предыдущего шага. Таким образом, матрица переходных вероятностей алгоритма случайного спуска в процессе оптимизации имеет вид (1.2.18), где

$$a_{ij}^{\Sigma} = \begin{cases} 1 - s_i + \frac{s_i}{m}, & \text{если } i=j; \\ \frac{s_i}{m}, & \text{если } i \neq j. \end{cases} \quad (1.2.25)$$

§ 1.3. ЛИНЕЙНЫЙ АЛГОРИТМ СЛУЧАЙНОГО ПОИСКА

Линейные алгоритмы случайного поиска характеризуются тем, что после удачного шага ($\Delta Q' \leq 0$) последующий шаг делается в том же направлении, а после неудачного шага ($\Delta Q' > 0$) выбирается новое случайное направление. В данном параграфе исследуем линейный алгоритм случайного поиска, определенный формулами (1.2.24). Для этого алгоритма матрица переходных вероятностей при нештрафе ($\Delta Q' \leq 0$) имеет вид [42]

$$A_0 = \begin{vmatrix} 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 1 \end{vmatrix} \quad (1.3.1)$$

а при штрафе ($\Delta Q' > 0$)

$$A_1 = \begin{vmatrix} \frac{1}{m} & \dots & \frac{1}{m} \\ \dots & \dots & \dots \\ \frac{1}{m} & \dots & \frac{1}{m} \end{vmatrix} \quad (1.3.2)$$

где $m = 2^n$.

Далее предположим, что в процессе поиска вероятности штрафов s_i ($i=1, \dots, m$) постоянные. Это имеет место, например, для линейной функции $Q(X)$. Тогда функционирование автомата описывается простой однородной цепью Маркова со следующей матрицей переходных вероятностей (см. формулу (1.2.25)):

$$A(S) = \begin{vmatrix} b_1 & a_1 & a_1 & \dots & a_1 & a_1 & a_1 \\ a_2 & b_2 & a_2 & \dots & a_2 & a_2 & a_2 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ a_{m-1} & a_{m-1} & a_{m-1} & \dots & a_{m-1} & b_{m-1} & a_{m-1} \\ a_m & a_m & a_m & \dots & a_m & a_m & b_m \end{vmatrix} \quad (1.3.3)$$

где

$$a_i = \frac{s_i}{m}; \quad b_i = 1 - s_i + \frac{s_i}{m}. \quad (1.3.4)$$

По этой матрице с учетом (1.3.4) и (1.2.17) видно, что для $\sigma \neq 0$ цепь Маркова является эргодической. Поэтому вектор предельных вероятностей $\mathbf{P} = (p_1, \dots, p_m)$ является решением системы уравнений

$$\mathbf{P} = A'(S) \mathbf{P}, \quad (1.3.5)$$

где $A'(S)$ — транспонированная матрица (1.3.3). Решая эту систему, находим

$$p_2 = \frac{b_1 - a_1 - 1}{b_2 - a_2 - 1} p_1 = \frac{s_1}{s_2} p_1; \\ \dots \dots \dots \quad (1.3.6)$$

$$p_m = \frac{b_1 - a_1 - 1}{b_m - a_m - 1} p_1 = \frac{s_1}{s_m} p_1.$$

После нормировки p_i ($i=1, \dots, m$) имеем

$$p_i = \frac{1}{s_i} S^* \quad (1.3.7)$$

$$(i=1, \dots, m),$$

где

$$S^* = \left(\sum_{j=1}^m \frac{1}{s_j} \right)^{-1}. \quad (1.3.8)$$

Предельный средний штраф равен

$$p_{\text{ср}} = m S^*. \quad (1.3.9)$$

Можно показать, что имеет место неравенство

$$0 < p_{\text{ср}} < \frac{1}{2}, \quad (1.3.10)$$

причем при $\sigma \rightarrow \infty$ $p_{\text{ср}} \rightarrow \frac{1}{2}$ и при $\sigma \rightarrow 0$ $p_{\text{ср}} \rightarrow 0$. Следовательно, рассмотренный автомат, соответствующий линейному алгоритму случайного поиска, обладает целесообразным поведением в процессе оптимизации при наличии помехи с ограниченной дисперсией.

Пронумеруем направления смещения $\Delta X^{(i)}$ в пространстве X следующим образом:

$$\begin{aligned} \Delta X^{(1)} &= (+1, \dots, +1); \quad \Delta X^{(2)} = (+1, \dots, +1, -1); \quad \Delta X^{(3)} = \\ &= (+1, \dots, +1, -1, +1); \quad \Delta X^{(4)} = (+1, \dots, +1, \\ &-1, -1); \quad \dots; \quad \Delta X^{(m/2-1)} = (+1, -1, \dots, -1, +1); \\ \Delta X^{(m/2)} &= (+1, -1, \dots, -1); \quad \Delta X^{(m/2+1)} = -\Delta X^{(i)} \end{aligned} \quad (1.3.11)$$

$$\left(i=1, \dots, \frac{m}{2} \right).$$

Тогда средний вектор смещения в пространстве X будет равен

$$\begin{aligned}
M[\Delta X] = & \left[\left(\frac{S_{1+m/2} - S_1}{S_1 S_{1+m/2}} + \frac{S_{2+m/2} - S_2}{S_2 S_{2+m/2}} + \dots \right. \right. \\
& \dots + \left. \frac{S_m - S_{m/2}}{S_m S_{m/2}} \right) S^*, \left(\frac{S_{1+m/2} - S_1}{S_1 S_{1+m/2}} + \dots \right. \\
& \dots + \frac{S_{m/4+2} - S_{m/4}}{S_{m/4} S_{m/4+m/2}} - \frac{S_{m/4+1+m/2} - S_{m/4+1}}{S_{m/4+1} S_{m/4+1+m/2}} - \dots \\
& \dots - \left. \frac{S_m - S_{m/2}}{S_m S_{m/2}} \right) S^*, \dots, \left(\frac{S_{1+m/2} - S_1}{S_1 S_{1+m/2}} - \right. \\
& - \frac{S_{2+m/2} - S_2}{S_2 S_{2+m/2}} + \frac{S_{3+m/2} - S_3}{S_3 S_{3+m/2}} - \\
& \left. - \frac{S_{4+m/2} - S_4}{S_4 S_{4+m/2}} + \dots - \frac{S_m - S_{m/2}}{S_{m/2} S_m} \right) S^* \left. \right], \quad (1.3.12)
\end{aligned}$$

где S^* определяется формулой (1.3.8). Среднее изменение показателя качества

$$\begin{aligned}
M[\Delta Q] = & S^* \left[\frac{S_{1+m/2} - S_1}{S_1 S_{1+m/2}} (\alpha_1 + \alpha_2 + \dots + \alpha_n) + \right. \\
& + \frac{S_{2+m/2} - S_2}{S_2 S_{2+m/2}} (\alpha_1 + \dots + \alpha_{n-1} - \alpha_n) + \\
& + \frac{S_{3+m/2} - S_3}{S_3 S_{3+m/2}} (\alpha_1 + \dots + \alpha_{n-2} - \alpha_{n-1} + \alpha_n) + \\
& + \frac{S_{4+m/4} - S_4}{S_4 S_{4+m/2}} (\alpha_1 + \dots + \alpha_{n-2} - \alpha_{n-1} - \alpha_n) + \dots + \\
& + \frac{S_{m-1} - S_{m/2-1}}{S_{m-1} S_{m/2-1}} (\alpha_1 - \dots - \alpha_{n-1} + \alpha_n) + \\
& \left. + \frac{S_m - S_{m/2}}{S_{m/2} S_m} (\alpha_1 - \dots - \alpha_n) \right], \quad (1.3.13)
\end{aligned}$$

где $\sum_{i=1}^n \alpha_i^2 = 1$ (напомним, что α_i — направляющие косинусы градиента).

Случай без воздействия помехи необходимо рассмотреть отдельно. Предположим, что направление градиента функции $Q(\mathbf{X})$ удовлетворяет условиям

$$\alpha_1 > \alpha_2 > \dots > \alpha_n > 0; \quad \alpha_1 > \alpha_2 + \dots + \alpha_n. \quad (1.3.14)$$

Тогда при $\sigma \rightarrow 0$ имеем

$$s_i = \begin{cases} 1 & \text{при } i=1, \dots, \frac{m}{2}; \\ 0 & \text{при } i=\frac{m}{2}+1, \dots, m. \end{cases} \quad (1.3.15)$$

Цепь Маркова является неэргодической (имеются $\frac{m}{2}$ поглощающих состояний) с матрицей переходных вероятностей

$$A(S) = \left\| \begin{array}{cccccc} \frac{1}{m} & \dots & \frac{1}{m} & \frac{1}{m} & \frac{1}{m} & \dots & \frac{1}{m} & \frac{1}{m} \\ \dots & \dots \\ \frac{1}{m} & \dots & \frac{1}{m} & \frac{1}{m} & \frac{1}{m} & \dots & \frac{1}{m} & \frac{1}{m} \\ 0 & \dots & 0 & 1 & 0 & \dots & 0 & 0 \\ \dots & \dots \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 & 1 \end{array} \right\| \begin{array}{l} \frac{m}{2} \\ \frac{m}{2} \end{array} \quad (1.3.16)$$

N -я степень этой матрицы будет иметь вид $A^N(S) =$

$$= \left\| \begin{array}{cccccc} \frac{1}{2^{N-1}m} & \dots & \frac{1}{2^{N-1}m} & \frac{2^N-1}{2^{N-1}m} & \frac{2^N-1}{2^{N-1}m} & \dots & \frac{2^N-1}{2^{N-1}m} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \frac{1}{2^{N-1}m} & \dots & \frac{1}{2^{N-1}m} & \frac{2^N-1}{2^{N-1}m} & \frac{2^N-1}{2^{N-1}m} & \dots & \frac{2^N-1}{2^{N-1}m} \\ 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & 0 & \dots & 1 \end{array} \right\| \begin{array}{l} \frac{m}{2} \\ \frac{m}{2} \end{array} \quad (1.3.17)$$

По матрице (1.3.17) находим вектор вероятностей $\mathbf{P}^{(N)} = (p_1^{(N)}, \dots, p_m^{(N)})$ пребывания автомата в состоянии i на N -м шаге:

$$p_i^{(N)} = \begin{cases} \frac{1}{2^{N-1}m} (p_1^{(0)} + \dots + p_{m/2}^{(0)}) & \text{при } i=1, \dots, \frac{m}{2}; \\ \frac{2^N - 1}{2^{N-1}m} (p_1^{(0)} + \dots + p_{m/2}^{(0)}) + p_i^{(0)} & \text{при } i = \\ & = \frac{m}{2} + 1, \dots, m, \end{cases} \quad (1.3.18)$$

где $\mathbf{P}^{(0)} = (p_1^{(0)}, \dots, p_m^{(0)})$ — вектор начальных вероятностей. При предельном переходе $N \rightarrow \infty$ получаем вектор предельных вероятностей:

$$p_i = \begin{cases} 0 & \text{при } i=1, \dots, \frac{m}{2}; \\ p_i^{(0)} + \frac{2}{m} (p_1^{(0)} + \dots + p_{m/2}^{(0)}) & \text{при } i = \frac{m}{2} + 1, \dots, m. \end{cases} \quad (1.3.19)$$

Средний штраф $p_{\text{ср}} = 0$ и средний вектор смещения в пространстве \mathbf{X}

$$M[\Delta \mathbf{X}] = \left[-1, - \left(\sum_{i=1+m/2}^{m/4+m/2} p_i^{(0)} - \sum_{i=m/4+1+m/2}^m p_i^{(0)} \right), \right. \\ - \left(\sum_{i=m/2+1}^{m/8+m/2} p_i^{(0)} - \sum_{i=m/8+1+m/2}^{m/4+m/8} p_i^{(0)} + \sum_{i=m/4+1+m/2}^{3/8m+m/2} p_i^{(0)} - \right. \\ \left. - \sum_{i=3/8m+1+m/2}^m p_i^{(0)} \right), \dots, - (p_{1+m/2}^{(0)} - p_{2+m/2}^{(0)} + \\ \left. + p_{3+m/2}^{(0)} - p_{4+m/2}^{(0)} + \dots - p_m^{(0)}) \right]. \quad (1.3.20)$$

Среднее изменение показателя качества

$$\begin{aligned}
 M[\Delta Q] = & -\alpha_1 \sum_{i=1}^{m/2} p_i^{(0)} - p_{1+m/2}^{(0)} (\alpha_1 + \alpha_2 + \dots \\
 & \dots + \alpha_n) - p_{2+m/2}^{(0)} (\alpha_1 + \dots + \alpha_{n-1} - \alpha_n) - \\
 & - p_{3+m/2}^{(0)} (\alpha_1 + \dots + \alpha_{n-2} - \alpha_{n-1} + \alpha_n) - \\
 & - p_{4+m/2}^{(0)} (\alpha_1 + \dots + \alpha_{n-2} - \alpha_{n-1} - \alpha_n) - \dots \\
 & \dots - p_{m-1}^{(0)} (\alpha_1 - \dots - \alpha_{n-1} + \alpha_n) - \\
 & - p_m^{(0)} (\alpha_1 - \dots - \alpha_n).
 \end{aligned} \tag{1.3.21}$$

Рассмотрим двумерный случай при наличии помехи ($n=2$; $m=4$). Векторы $\Delta \mathbf{X}^{(i)}$ пронумеруем в следующем порядке:

$$\begin{aligned}
 \Delta \mathbf{X}^{(1)} = (1, 1); & \quad \Delta \mathbf{X}^{(2)} = (-1, 1); \\
 \Delta \mathbf{X}^{(3)} = (-1, -1); & \quad \Delta \mathbf{X}^{(4)} = (1, -1).
 \end{aligned} \tag{1.3.22}$$

Матрица $A(\mathbf{S})$ имеет вид

$$A(\mathbf{S}) = \begin{vmatrix} b_1 & a_1 & a_1 & a_1 \\ a_2 & b_2 & a_2 & a_2 \\ a_3 & a_3 & b_3 & a_3 \\ a_4 & a_4 & a_4 & b_4 \end{vmatrix} \tag{1.3.23}$$

где

$$a_i = \frac{s_i}{4}; \quad b_i = \frac{4-3s_i}{4}. \tag{1.3.24}$$

$$\begin{aligned}
 s_1 = \frac{1}{2} \left[1 + \Phi \left(\frac{\alpha_1 + \alpha_2}{2\sigma} \right) \right]; & \quad s_2 = \frac{1}{2} \left[1 - \Phi \left(\frac{\alpha_1 - \alpha_2}{2\sigma} \right) \right]; \\
 s_3 = \frac{1}{2} \left[1 - \Phi \left(\frac{\alpha_1 + \alpha_2}{2\sigma} \right) \right]; & \quad s_4 = \frac{1}{2} \left[1 + \Phi \left(\frac{\alpha_1 - \alpha_2}{2\sigma} \right) \right]
 \end{aligned} \tag{1.3.25}$$

(α_1, α_2 — направляющие косинусы вектора градиента; $\alpha_1^2 + \alpha_2^2 = 1$).

Из матрицы (1.3.23) следует, что цепь Маркова при постоянных s_i однородная и при $\sigma \neq 0$ эргодическая.

Вектор предельных вероятностей $\mathbf{P} = (p_1, p_2, p_3, p_4)$ находим, решая систему уравнений

$$\mathbf{P} = A'(\mathbf{S})\mathbf{P}, \quad (1.3.26)$$

где $A'(\mathbf{S})$ — транспонированная матрица (1.3.23):

$$\begin{aligned} p_1 &= \frac{1}{s_1} \cdot \frac{s_1 s_2 s_3 s_4}{s_1 s_3 + s_2 s_4}; & p_2 &= \frac{1}{s_2} \cdot \frac{s_1 s_2 s_3 s_4}{s_1 s_3 + s_2 s_4}; \\ p_3 &= \frac{1}{s_3} \cdot \frac{s_1 s_2 s_3 s_4}{s_1 s_3 + s_2 s_4}; & p_4 &= \frac{1}{s_4} \cdot \frac{s_1 s_2 s_3 s_4}{s_1 s_3 + s_2 s_4}. \end{aligned} \quad (1.3.27)$$

Средний штраф, подсчитанный по формуле (1.2.23), равен

$$\begin{aligned} p_{\text{ср}} &= 4 \frac{s_1 s_2 s_3 s_4}{s_1 s_3 + s_2 s_4} = \left[\frac{1}{1 - \Phi^2\left(\frac{\alpha_1 + \alpha_2}{2\sigma}\right)} + \right. \\ &\quad \left. + \frac{1}{1 - \Phi^2\left(\frac{\alpha_1 - \alpha_2}{2\sigma}\right)} \right]^{-1}. \end{aligned} \quad (1.3.28)$$

Легко заметить, что имеет место неравенство

$$0 \leq p_{\text{ср}} \leq \frac{1}{2}. \quad (1.3.29)$$

Отсюда следует, что соответствующий этому алгоритму вероятностный автомат обладает целесообразным поведением. Средний вектор смещения в пространстве \mathbf{X}

$$\begin{aligned} M[\Delta\mathbf{X}] &= (p_1 - p_2 - p_3 + p_4, p_1 + p_2 - p_3 - p_4) = \\ &= \left[\frac{(s_1 + s_4)s_2 s_3 - (s_2 + s_3)s_1 s_4}{s_1 s_3 + s_2 s_4}, \right. \\ &\quad \left. \frac{(s_1 + s_2)s_3 s_4 - (s_3 + s_4)s_1 s_2}{s_1 s_3 + s_2 s_4} \right]. \end{aligned} \quad (1.3.30)$$

Среднее приращение показателя качества

$$M[\Delta Q] = - \frac{\Phi\left(\frac{\alpha_1 + \alpha_2}{2\sigma}\right) \left[1 - \Phi^2\left(\frac{\alpha_1 - \alpha_2}{2\sigma}\right) \right]}{2 - \Phi^2\left(\frac{\alpha_1 + \alpha_2}{2\sigma}\right) - \Phi^2\left(\frac{\alpha_1 - \alpha_2}{2\sigma}\right)} +$$

$$+ \frac{\Phi\left(\frac{\alpha_1 - \alpha_2}{2\sigma}\right) \left[1 - \Phi^2\left(\frac{\alpha_1 + \alpha_2}{2\sigma}\right) \right]}{2 - \Phi^2\left(\frac{\alpha_1 + \alpha_2}{2\sigma}\right) - \Phi^2\left(\frac{\alpha_1 - \alpha_2}{2\sigma}\right)}. \quad (1.3.31)$$

При предельном переходе $\alpha_1 \rightarrow \sqrt{2}/2$, $\alpha_2 \rightarrow \sqrt{2}/2$ (если выполнено условие $\sigma \neq 0$) находим:

$$M[\Delta Q] = - \frac{\Phi\left(\frac{\sqrt{2}}{2\sigma}\right)}{2 - \Phi^2\left(\frac{\sqrt{2}}{2\sigma}\right)}. \quad (1.3.32)$$

Рассмотрим случай, когда $\alpha_1 = 1$; $\alpha_2 = 0$. Тогда из формулы (1.3.25) следует

$$s_1 = s_4 = \frac{1}{2} \left[1 + \Phi\left(\frac{1}{2\sigma}\right) \right]; \quad s_2 = s_3 = \frac{1}{2} \left[1 - \Phi\left(\frac{1}{2\sigma}\right) \right]. \quad (1.3.33)$$

Матрица $A(S)$ имеет вид

$$A(S) = \begin{vmatrix} b_1 & a_1 & a_1 & a_1 \\ a_2 & b_2 & a_2 & a_2 \\ a_2 & a_2 & b_2 & a_2 \\ a_1 & a_1 & a_1 & b_1 \end{vmatrix} \quad (1.3.34)$$

В силу симметрии матрица (1.3.34) распадается и получаем $p_1 = p_4$; $p_2 = p_3$. Окончательное уравнение для определения вектора P приобретает вид

$$p_1 = (b_1 + a_1)p_1 + 2a_2p_2; \quad (1.3.35)$$

$$p_2 = 2a_1p_1 + (b_2 + a_2)p_2.$$

Решая эту систему уравнений, находим:

$$\begin{aligned} p_1 = p_4 &= \frac{1}{2} \cdot \frac{1 - b_2 - a_2}{1 - b_2 - a_2 + 2a_1}; \\ p_2 = p_3 &= \frac{1}{2} \cdot \frac{2a_1}{1 - b_2 - a_2 + 2a_1}. \end{aligned} \quad (1.3.36)$$

Подставляя формулы (1.3.24) в выражение (1.3.6) и учитывая равенства (1.3.33), имеем:

$$\begin{aligned} p_1 = p_4 &= \frac{1}{2} s_2 = \frac{1}{4} \left[1 - \Phi \left(\frac{1}{2\sigma} \right) \right]; \\ p_2 = p_3 &= \frac{1}{2} s_1 = \frac{1}{4} \left[1 + \Phi \left(\frac{1}{2\sigma} \right) \right]. \end{aligned} \quad (1.3.37)$$

По формуле (1.2.23)

$$p_{\text{ср}} = 2s_1s_2 = \frac{1}{2} - \frac{1}{2} \Phi^2 \left(\frac{1}{2\sigma} \right). \quad (1.3.38)$$

Из формулы (1.3.38) следует, что автомат, соответствующий этому алгоритму, обладает целесообразным поведением. Среднее приращение показателя качества за один шаг поиска равно

$$M[\Delta Q] = s_2 - s_1 = -\Phi \left(\frac{1}{2\sigma} \right). \quad (1.3.39)$$

В пределе при $\sigma \rightarrow 0$ ($\sigma \neq 0$) получаем:

$$p_1 = p_4 \rightarrow 0; \quad p_2 = p_3 \rightarrow \frac{1}{2}; \quad p_{\text{ср}} \rightarrow 0; \quad M[\Delta Q] \rightarrow -1. \quad (1.3.40)$$

При $\sigma \rightarrow \infty$ имеем:

$$p_1 = p_4 \rightarrow \frac{1}{4}; \quad p_2 = p_3 \rightarrow \frac{1}{4}; \quad p_{\text{ср}} \rightarrow \frac{1}{2}; \quad M[\Delta Q] \rightarrow 0. \quad (1.3.41)$$

В случае, когда $\sigma = 0$, цепь Маркова, описывающая линейный алгоритм случайного поиска, является неэргодической, в силу чего этот случай требует отдельного анализа.

Из формул (1.3.25) при $\sigma=0$ получаем: $s_1=1$; $s_2=s_3=0$; $s_4=1$.

Матрица (1.2.23) для двумерного случая без воздействия помехи при условии (1.3.14) имеет вид

$$A(\mathbf{S}) = \begin{vmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{vmatrix} \quad (1.3.42)$$

N -я степень этой матрицы

$$A^N(\mathbf{S}) = \begin{vmatrix} \frac{1}{2^{N+1}} & \frac{2^N-1}{2^{N+1}} & \frac{2^N-1}{2^{N+1}} & \frac{1}{2^{N+1}} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \frac{1}{2^{N+1}} & \frac{2^N-1}{2^{N+1}} & \frac{2^N-1}{2^{N+1}} & \frac{1}{2^{N+1}} \end{vmatrix} \quad (1.3.43)$$

По этой матрице получаем вектор вероятностей на N -м шаге

$$p_i^{(N)} = \begin{cases} \frac{1}{2^{N+1}} (p_1^{(0)} + p_4^{(0)}) & \text{при } i=1, 4; \\ \frac{2^N-1}{2^{N+1}} (p_1^{(0)} + p_4^{(0)}) + p_i^{(0)} & \text{при } i=2, 3. \end{cases} \quad (1.3.44)$$

Вектор предельных вероятностей равен

$$p_i = \begin{cases} 0 & \text{при } i=1, 4; \\ p_i^{(0)} + \frac{1}{2} (p_1^{(0)} + p_4^{(0)}) & \text{при } i=2, 3. \end{cases} \quad (1.3.45)$$

Средний вектор смещения в пространстве \mathbf{X}

$$M[\Delta \mathbf{X}] = (-1, -(p_3^{(0)} - p_2^{(0)})). \quad (1.3.46)$$

Для среднего приращения функции качества имеем

$$M[\Delta Q] = -\alpha_1 - (p_3^{(0)} - p_2^{(0)}) \sqrt{1 - \alpha_1^2}. \quad (1.3.47)$$

Особо следует рассмотреть случай $\alpha_1 = \alpha_2 = \sqrt{2}/2$. Тогда в соответствии с определением алгоритма (1.2.24)

$$A(S) = \begin{vmatrix} 1 & 1 & 1 & 1 \\ 4 & 4 & 4 & 4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (1.3.48)$$

Вектор предельных вероятностей равен

$$p_i = \begin{cases} 0, & \text{если } i=1; \\ p_i^{(0)} + \frac{1}{3} p_1^{(0)}, & \text{если } i=2, 3, 4. \end{cases} \quad (1.3.49)$$

Для среднего приращения показателя качества в этом случае имеем

$$M[\Delta Q] = - \left(\frac{1}{3} p_1^{(0)} + p_3^{(0)} \right) \sqrt{2}. \quad (1.3.50)$$

§ 1.4. НЕЛИНЕЙНЫЙ АЛГОРИТМ СЛУЧАЙНОГО ПОИСКА

Для нелинейного алгоритма характерно то, что после удачного шага ($\Delta Q' \leq 0$) выбирается новое случайное направление шага, а после неудачного шага ($\Delta Q' > 0$) делается возврат в предыдущее состояние. В этом параграфе исследуем нелинейный алгоритм случайного поиска, определенный формулой (1.2.3) в § 1.1.

Для этого алгоритма матрицы переходных вероятностей имеют вид [43]:

при нештрафе ($\Delta Q' \leq 0$)

$$A_0 = \left\| \begin{array}{cccc} \frac{1}{m} & \dots & \frac{1}{m} & \\ \cdot & \cdot & \cdot & \\ \frac{1}{m} & \dots & \frac{1}{m} & \end{array} \right\| \quad (1.4.1)$$

где $m = 2^n$;

при штрафе ($\Delta Q' > 0$)

$$A_1 = \left\| \begin{array}{ccc} 0 & \vdots & I \\ \cdot & \cdot & \cdot \\ I & \vdots & 0 \end{array} \right\| \quad (1.4.2)$$

где нулями обозначены блоки, элементы которых равны нулю, а I — единичная матрица.

Далее будем предполагать, что в процессе поиска величины s_i ($i=1, \dots, m$), определенные по формуле (1.2.17), постоянны (например, функция $Q(\mathbf{X})$ является линейной). Тогда функционирование автомата описывается простой однородной цепью Маркова со следующей матрицей переходных вероятностей (см. формулы (1.2.8)):

$$A(\mathbf{S}) = \left\| \begin{array}{cccc} a_1 & a_1 & \dots & a_1 & b_1 & a_1 & a_1 & \dots & a_1 \\ a_2 & a_2 & \dots & a_2 & a_2 & b_2 & a_2 & \dots & a_2 \\ \cdot & \cdot \\ a_{m/2} & a_{m/2} & \dots & a_{m/2} & a_{m/2} & a_{m/2} & a_{m/2} & \dots & b_{m/2} \\ b_{m/2+1} & a_{m/2+1} & \dots & a_{m/2+1} & a_{m/2+1} & a_{m/2+1} & a_{m/2+1} & \dots & a_{m/2+1} \\ \cdot & \cdot \\ a_m & a_m & \dots & b_m & a_m & a_m & a_m & \dots & a_m \end{array} \right\| \begin{array}{l} \left. \vphantom{\begin{array}{c} a_1 \\ a_2 \\ \cdot \\ a_{m/2} \end{array}} \right\} \frac{m}{2} \\ \left. \vphantom{\begin{array}{c} b_{m/2+1} \\ \cdot \\ a_m \end{array}} \right\} \frac{m}{2} \end{array} \quad (1.4.3)$$

где

$$a_i = \frac{1}{m} (1 - s_i); \quad b_i = \frac{1}{m} (1 - s_i) + s_i. \quad (1.4.4)$$

$$p_i = \frac{1 + s_{m/2+i}}{1 - s_i s_{m/2+i}} S \quad \left(i = 1, \dots, \frac{m}{2} \right); \quad (1.4.9)$$

$$p_{i+m/2} = \frac{1 + s_i}{1 - s_i s_{m/2+i}} S \quad \left(i = 1, \dots, \frac{m}{2} \right),$$

где

$$S = \left(\sum_{j=1}^{m/2} \frac{2 + s_j + s_{j+m/2}}{1 - s_j s_{j+m/2}} \right)^{-1}. \quad (1.4.10)$$

Средний штраф равен

$$p_{\text{ср}} = \sum_{i=1}^{m/2} \frac{s_i + 2s_i s_{i+m/2} + s_{i+m/2}}{1 - s_i s_{i+m/2}} S = 1 - mS. \quad (1.4.11)$$

Нетрудно показать, что имеет место неравенство

$$\frac{1}{3} \leq p_{\text{ср}} \leq \frac{1}{2}. \quad (1.4.12)$$

При этом при $\sigma \rightarrow 0$ $p_{\text{ср}} \rightarrow 1/3$ и при $\sigma \rightarrow \infty$ $p_{\text{ср}} \rightarrow 1/2$.

Отсюда следует, что вероятностный автомат, соответствующий рассмотренному алгоритму случайного поиска, в процессе оптимизации при наличии помех с ограниченной дисперсией обладает целесообразным поведением в смысле определения, данного М. Л. Цетлиным [28].

Далее пронумеруем направления смещения $\Delta \mathbf{X}^{(i)}$ в пространстве \mathbf{X} (состояния автомата) в следующем порядке:

$$\begin{aligned} \Delta \mathbf{X}^{(1)} &= (+1, \dots, +1), \quad \Delta \mathbf{X}^{(2)} = (+1, \dots, +1, -1); \quad \Delta \mathbf{X}^{(3)} = \\ &= (+1, \dots, +1, -1, -1); \quad \Delta \mathbf{X}^{(4)} = \\ &= (+1, \dots, +1, -1, -1); \dots; \quad \Delta \mathbf{X}^{(m/2-1)} = \end{aligned}$$

$$\begin{aligned}
 &= (+1, -1, \dots, -1, +1); \Delta \mathbf{X}^{m/2} = \\
 &= (+1, -1, \dots, -1); \Delta \mathbf{X}^{(m/2+i)} = -\Delta \mathbf{X}^{(i)} \quad (1.4.13) \\
 &\quad \left(i=1, \dots, \frac{m}{2} \right).
 \end{aligned}$$

Тогда средний вектор смещения в пространстве $\{\mathbf{X}\}$ равен

$$\begin{aligned}
 M[\Delta \mathbf{X}] = & \left[\left(\frac{S_{1+m/2} - S_1}{1 - S_1 S_{1+m/2}} + \frac{S_{2+m/2} - S_2}{1 - S_2 S_{2+m/2}} + \dots \right. \right. \\
 & \dots + \left. \frac{S_m - S_{m/2}}{1 - S_m S_{m/2}} \right) S, \left(\frac{S_{1+m/2} - S_1}{1 - S_1 S_{1+m/2}} + \dots \right. \\
 & \dots + \frac{S_{m/4+m/2} - S_{m/4}}{1 - S_{m/4} S_{m/4+m/2}} - \frac{S_{m/4+1+m/2} - S_{m/4+1}}{1 - S_{m/4+1} S_{m/4+1+m/2}} - \dots \\
 & \left. \dots - \frac{S_m - S_{m/2}}{1 - S_{m/2} S_m} \right) S, \dots, \left(\frac{S_{1+m/2} - S_1}{1 - S_1 S_{1+m/2}} - \right. \\
 & \left. - \frac{S_{2+m/2} - S_2}{1 - S_2 S_{2+m/2}} + \frac{S_{3+m/2} - S_3}{1 - S_3 S_{3+m/2}} - \frac{S_{4+m/2} - S_4}{1 - S_4 S_{4+m/2}} + \right. \\
 & \left. + \dots - \frac{S_m - S_{m/2}}{1 - S_m S_{m/2}} \right) S \Big], \quad (1.4.14)
 \end{aligned}$$

где S определяется формулой (1.4.10). Среднее смещение в направлении градиента $\text{grad } Q(\mathbf{X}) = (\alpha_1, \alpha_2, \dots, \alpha_n)$ равно

$$\begin{aligned}
 M[\Delta Q] = & S \left[\frac{S_{1+m/2} - S_1}{1 - S_1 S_{1+m/2}} (\alpha_1 + \alpha_2 + \dots + \alpha_n) + \right. \\
 & + \frac{S_{2+m/2} - S_2}{1 - S_2 S_{2+m/2}} (\alpha_1 + \alpha_2 + \dots + \alpha_{n-1} - \alpha_n) + \\
 & + \frac{S_{3+m/2} - S_3}{1 - S_3 S_{3+m/2}} (\alpha_1 + \dots + \alpha_{n-2} - \alpha_{n-1} + \alpha_n) + \\
 & + \frac{S_{4+m/2} - S_4}{1 - S_4 S_{4+m/2}} (\alpha_1 + \dots + \alpha_{n-2} - \alpha_{n-1} - \\
 & - \alpha_n) + \dots + \frac{S_{m-1} - S_{m/2-1}}{1 - S_{m-1} S_{m/2-1}} (\alpha_1 - \dots - \alpha_{n-1} + \alpha_n) + \\
 & \left. + \frac{S_m - S_{m/2}}{1 - S_m S_{m/2}} (\alpha_1 - \alpha_2 - \dots - \alpha_n) \right]. \quad (1.4.15)
 \end{aligned}$$

Теперь предположим, что направление вектора градиента функции $Q(\mathbf{X})$ удовлетворяет условиям

$$\begin{aligned} \alpha_1 > \alpha_2 > \dots > \alpha_n > 0; \\ \alpha_1 > \alpha_2 + \dots + \alpha_n. \end{aligned} \quad (1.4.16)$$

Тогда имеют место следующие неравенства:

$$\Delta Q^{(1)} > 0; \dots; \Delta Q^{(m/2)} > 0; \Delta Q^{(m/2+1)} < 0; \dots; \Delta Q^{(m)} < 0, \quad (1.4.17)$$

где $\Delta Q^{(i)}$ — приращения функции $Q(\mathbf{X})$ по направлению $\Delta \mathbf{X}^{(i)}$. Поэтому при предельном переходе $\sigma \rightarrow 0$ имеем

$$\lim_{\sigma \rightarrow 0} s_i = \begin{cases} 1 & \text{при } i = 1, \dots, \frac{m}{2}; \\ 0 & \text{при } i = \frac{m}{2}, \dots, m; \end{cases} \quad (1.4.18)$$

$$\lim_{\sigma \rightarrow 0} p_i = \begin{cases} \frac{2}{3m} & \text{при } i = 1, \dots, \frac{m}{2}; \\ \frac{4}{3m} & \text{при } i = \frac{m}{2} + 1, \dots, m; \end{cases} \quad (1.4.19)$$

$$\lim_{\sigma \rightarrow 0} p_{\text{ср}} = \frac{1}{3}; \quad \lim_{\sigma \rightarrow 0} M[\Delta \mathbf{X}] = \left(-\frac{1}{3}, 0, \dots, 0 \right); \quad (1.4.20)$$

$$\lim_{\sigma \rightarrow 0} M[\Delta Q] = -\frac{1}{3} \alpha_1. \quad (1.4.21)$$

Нетрудно убедиться, что при предельном переходе $\sigma \rightarrow \infty$ получаем

$$\lim_{\sigma \rightarrow \infty} s_i = \frac{1}{2}; \quad \lim_{\sigma \rightarrow \infty} p_i = \frac{1}{m} \quad (i = 1, \dots, m);$$

$$\lim_{\sigma \rightarrow \infty} p_{\text{ср}} = \frac{1}{2}; \quad \lim_{\sigma \rightarrow \infty} M[\Delta \mathbf{X}] = (0, \dots, 0); \quad (1.4.22)$$

$$\lim_{\sigma \rightarrow \infty} M[\Delta Q] = 0.$$

Далее рассмотрим двумерный случай ($n=2$; $m=4$).

Принимаем следующую нумерацию векторов $\Delta X^{(i)}$:

$$\Delta X^{(1)} = (1, 1); \quad \Delta X^{(2)} = (-1, 1);$$

$$\Delta X^{(3)} = (-1, -1); \quad \Delta X^{(4)} = (1, -1).$$

Пусть функция $Q(X)$ является линейной, т. е.

$$\text{grad } Q(X) = (\alpha_1, \alpha_2) = \text{const}; \quad \alpha_1^2 + \alpha_2^2 = 1.$$

Тогда из формулы (1.2.17) следует:

$$s_1 = \frac{1}{2} \left[1 + \Phi \left(\frac{\alpha_1 + \alpha_2}{2\sigma} \right) \right]; \quad s_2 = \frac{1}{2} \left[1 - \Phi \left(\frac{\alpha_1 - \alpha_2}{2\sigma} \right) \right];$$

$$s_3 = \frac{1}{2} \left[1 - \Phi \left(\frac{\alpha_1 + \alpha_2}{2\sigma} \right) \right]; \quad s_4 = \frac{1}{2} \left[1 + \Phi \left(\frac{\alpha_1 - \alpha_2}{2\sigma} \right) \right],$$

(1.4.23)

где $\Phi(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt$.

Матрица (1.4.3) равна

$$A(S) = \begin{vmatrix} a_1 & a_1 & b_1 & a_1 \\ a_2 & a_2 & a_2 & b_2 \\ b_3 & a_3 & a_3 & a_3 \\ a_4 & b_4 & a_4 & a_4 \end{vmatrix} \quad (1.4.24)$$

где

$$a_i = \frac{1}{4}(1 - s_i); \quad b_i = \frac{1}{4}(1 + 3s_i). \quad (1.4.25)$$

Имеем следующий вектор предельных вероятностей:

$$p_1 = \frac{(1 + s_3)(1 - s_2 s_4)}{3(2 - s_1 s_3 - s_2 s_4)}; \quad p_2 = \frac{(1 + s_4)(1 - s_1 s_3)}{3(2 - s_1 s_3 - s_2 s_4)};$$

$$p_3 = \frac{(1+s_1)(1-s_2s_4)}{3(2-s_1s_3-s_2s_4)}; \quad p_4 = \frac{(1+s_2)(1-s_1s_3)}{3(2-s_1s_3-s_2s_4)}, \quad (1.4.26)$$

где s_i определяется формулами (1.4.23).

Среднее приращение показателя качества равно

$$M[\Delta Q] = \frac{-1}{3 \left[6 + \Phi^2 \left(\frac{\alpha_1 + \alpha_2}{2\sigma} \right) + \Phi^2 \left(\frac{\alpha_1 - \alpha_2}{2\sigma} \right) \right]} \times \\ \times \left\{ \Phi \left(\frac{\alpha_1 + \alpha_2}{2\sigma} \right) \left[3 + \Phi^2 \left(\frac{\alpha_1 - \alpha_2}{2\sigma} \right) \right] (\alpha_1 + \alpha_2) + \right. \\ \left. + \Phi \left(\frac{\alpha_1 - \alpha_2}{2\sigma} \right) \left[3 + \Phi^2 \left(\frac{\alpha_1 + \alpha_2}{2\sigma} \right) \right] (\alpha_1 - \alpha_2) \right\}. \quad (1.4.27)$$

Пусть $\frac{\sqrt{2}}{2} < \alpha_1 \leq 1$. Тогда при предельном переходе $\sigma \rightarrow 0$ получаем:

$$s_1 \rightarrow 1; \quad s_2 \rightarrow 0; \quad s_3 \rightarrow 0; \quad s_4 \rightarrow 1; \quad (1.4.28)$$

$$p_1 \rightarrow \frac{1}{6}; \quad p_2 \rightarrow \frac{1}{3}; \quad p_3 \rightarrow \frac{1}{3}; \quad p_4 \rightarrow \frac{1}{6}; \quad (1.4.29)$$

$$M[\Delta Q] \rightarrow -\frac{\alpha_1}{3}. \quad (1.4.30)$$

Из формул (1.4.30) и (1.4.21) следует, что для рассмотренного алгоритма скорость движения в направлении градиента не зависит от размерности пространства X (при единичном смещении по каждой координате $|\Delta x_i| = 1$).

Рассмотрим случай, когда $\alpha_1 = 1$; $\alpha_2 = 0$. Для этого случая получаем

$$s_1 = s_4 = \frac{1}{2} \left[1 + \Phi \left(\frac{1}{2\sigma} \right) \right]; \\ s_2 = s_3 = \frac{1}{2} \left[1 - \Phi \left(\frac{1}{2\sigma} \right) \right] \quad (1.4.31)$$

Матрица $A(S)$ приобретает вид

$$A(S) = \begin{vmatrix} a_1 & a_1 & b_1 & a_1 \\ a_2 & a_2 & a_2 & b_2 \\ b_2 & a_2 & a_2 & a_2 \\ a_1 & b_1 & a_1 & a_1 \end{vmatrix} \quad (1.4.32)$$

где a_i и b_i ($i=1, 2$) определяются формулами (1.4.25). Из матрицы (1.4.32) видно, что цепь Маркова является эргодической. В силу симметрии матрица (1.4.32) распадается, в результате чего получаем $p_1=p_4$, $p_2=p_3$. В окончательном виде уравнения для определения вектора P записываются просто:

$$\begin{aligned} p_1 &= 2a_1p_1 + (a_2 + b_2)p_2; \\ p_2 &= (a_1 + b_1)p_1 + 2a_2p_2. \end{aligned} \quad (1.4.33)$$

Решая эту систему, находим

$$p_1 = \frac{1}{2} \cdot \frac{1 - 2a_2}{1 + a_1 + b_1 - 2a_2}; \quad p_2 = \frac{a_1 + b_1}{1 + a_1 + b_1 - 2a_2}. \quad (1.4.34)$$

Подставляя формулы (1.4.25) в (1.4.34) и учитывая выражения (1.4.31), имеем:

$$\begin{aligned} p_1 = p_4 &= \frac{1}{6}(1 + s_2) = \frac{3 - \Phi\left(\frac{1}{2\sigma}\right)}{12}; \\ p_2 = p_3 &= \frac{1}{6}(1 + s_1) = \frac{3 + \Phi\left(\frac{1}{2\sigma}\right)}{12}. \end{aligned} \quad (1.4.35)$$

Средний штраф равен

$$p_{\text{ср}} = \frac{1}{3}(1 + 2s_1s_2) = \frac{1}{2} - \frac{1}{6}\Phi^2\left(\frac{1}{2\sigma}\right). \quad (1.4.36)$$

Из этого видно, что вероятностный автомат, соответствующий данному алгоритму случайного поиска, обладает целесообразным поведением.

Средняя длина шага поиска равна

$$M[\Delta x_1] = \frac{1}{3}(s_2 - s_1) = -\frac{1}{3}\Phi\left(\frac{1}{2\sigma}\right). \quad (1.4.37)$$

В пределе при $\sigma \rightarrow 0$ получаем:

$$\begin{aligned} p_1 = p_4 &\rightarrow \frac{1}{6}; & p_2 = p_3 &\rightarrow \frac{1}{3}; \\ p_{\text{ср}} &\rightarrow \frac{1}{3}; & M[\Delta x_1] &\rightarrow -\frac{1}{3}. \end{aligned} \quad (1.4.38)$$

При $\sigma \rightarrow \infty$ имеем

$$\begin{aligned} p_1 = p_4 &\rightarrow \frac{1}{4}; & p_2 = p_3 &\rightarrow \frac{1}{4}; \\ p_{\text{ср}} &\rightarrow \frac{1}{2}; & M[\Delta x_1] &\rightarrow 0. \end{aligned} \quad (1.4.39)$$

Особо необходимо рассмотреть случай $\alpha_1 = \alpha_2 = \sqrt{2}/2$. В этом случае при $\sigma \neq 0$ из формулы (1.4.27) получаем

$$M[\Delta Q] = -\frac{\Phi\left(\frac{\sqrt{2}}{2\sigma}\right)\sqrt{2}}{6 + \Phi^2\left(\frac{\sqrt{2}}{2\sigma}\right)}. \quad (1.4.40)$$

Отсюда при $\sigma \rightarrow 0$ имеем

$$M[\Delta Q] = -\frac{\sqrt{2}}{7}. \quad (1.4.41)$$

Теперь сопоставим линейный и нелинейный алгоритмы случайного поиска. Для этого, используя формулы (1.4.15) и (1.3.13), составим отношение средних приращений показателя качества обоих алгоритмов на одном шаге поиска:

$$\eta = \frac{S^* \frac{s_{i+m/2} - s_i}{s_i s_{i+m/2}}}{S \frac{s_{i+m/2} - s_i}{1 - s_i s_{i+m/2}}} = \frac{S^* (1 - s_i s_{i+m/2})}{S s_i s_{i+m/2}}$$

$$\begin{aligned}
 & 3 \left[3 + \Phi^2 \left(\frac{\Delta Q^{(i)}}{2\sigma} \right) \right] \sum_{j=1}^{m/2} \frac{1}{3 + \Phi^2 \left(\frac{\Delta Q^{(j)}}{2\sigma} \right)} \\
 &= \frac{\left[1 - \Phi^2 \left(\frac{\Delta Q^{(i)}}{2\sigma} \right) \right] \sum_{j=1}^{m/2} \frac{1}{1 - \Phi^2 \left(\frac{\Delta Q^{(j)}}{2\sigma} \right)}}{.}
 \end{aligned}
 \tag{1.4.42}$$

Нетрудно заметить, что при $\sigma \rightarrow \infty$ отношение $\eta \rightarrow 3$. Это означает, что в процессе оптимизации при наличии помех с большой дисперсией линейный алгоритм случайного поиска в три раза эффективнее нелинейного, если функция $Q(\mathbf{X})$ имеет линейный характер.

Сравним эффективность линейного и нелинейного алгоритмов случайного поиска при оптимизации в двумерном пространстве ($n=2$) в случае отсутствия помехи ($\sigma=0$). Для этого составим отношение правых частей выражений (1.3.47) и (1.4.30):

$$\begin{aligned}
 \eta^* &= \frac{-\alpha_1 - (p_3^{(0)} - p_2^{(0)})\sqrt{1 - \alpha_1^2}}{-\frac{1}{3}\alpha_1} = \\
 &= 3 \left[1 + (p_3^{(0)} - p_2^{(0)}) \frac{\sqrt{1 - \alpha_1^2}}{\alpha_1} \right].
 \end{aligned}
 \tag{1.4.43}$$

Видно, что сравнительная эффективность алгоритмов зависит от направления градиента функции $Q(\mathbf{X})$ и от начального распределения вероятностей $\mathbf{P}^{(0)} = (p_1^{(0)}, p_2^{(0)}, p_3^{(0)}, p_4^{(0)})$. Так, например, при $\alpha_1 \rightarrow 1$ или при $p_2^{(0)} \rightarrow p_3^{(0)}$ получаем $\eta^* \rightarrow 3$, а при $\alpha_1 \rightarrow \sqrt{2}/2$ и $p_2^{(0)} \rightarrow 1$ имеем $\eta^* \rightarrow 0$. При $p_3^{(0)} \rightarrow 1$ и $\alpha_1 \rightarrow \sqrt{2}/2$ находим максимальное значение $\eta^* = 6$.

Формула (1.4.43) дает сравнительную оценку линейного и нелинейного алгоритмов случайного поиска, когда направляющие косинусы градиента функции качества удовлетворяют условиям $\sqrt{2}/2 < \alpha_1 \leq 1$; $\alpha_2 = \sqrt{1 - \alpha_1^2}$. В случае, когда $\alpha_1 = \alpha_2 = \sqrt{2}/2$, сравнительная

оценка алгоритмов вычисляется по формуле, полученной из выражений (1.3.50) и (1.4.41),

$$\eta^* = \frac{7(p_1^{(0)} + 3p_3^{(0)})}{3}. \quad (1.4.44)$$

При $p_1^{(0)} \rightarrow 0$ и $p_3^{(0)} \rightarrow 1$ отношение (1.4.44) стремится к максимальному значению ($\eta^* \rightarrow 7$), а при $p_1^{(0)} \rightarrow 0$ и $p_3^{(0)} \rightarrow 0$ — к минимальному ($\eta^* \rightarrow 0$).

§ 1.5. АЛГОРИТМ ГАУССА—ЗЕЙДЕЛЯ И ЕГО СТОХАСТИЧЕСКИЙ ВАРИАНТ

Сущность алгоритма Гаусса—Зейделя состоит в следующем [3]. Процесс оптимизации происходит последовательно по каждой переменной, т. е. состоит из последовательных спусков в направлении координатных осей.

В этом параграфе для простоты исследуем одну модификацию алгоритма Гаусса—Зейделя. Сначала для спуска выбираются подряд все положительные направления координатных осей, а потом — все отрицательные. Если по выбранному направлению спуска приращение функции качества отрицательно ($\Delta Q' < 0$), то производится спуск по этому направлению до тех пор, пока функция качества уменьшается ($\Delta Q' < 0$). Если $\Delta Q' \geq 0$, то происходит переход к следующему направлению спуска.

Из сказанного следует, что автомат этого алгоритма поиска имеет $2n$ состояний, которые пронумеруем в следующем порядке:

$$\begin{aligned} \Delta X^{(1)} &= a e_1; \quad \Delta X^{(2)} = a e_2; \quad \dots; \quad \Delta X^{(n)} = a e_n; \\ \Delta X^{(n+1)} &= -a e_1; \quad \Delta X^{(2)} = -a e_2; \quad \dots; \quad \Delta X^{(2n)} = -a e_n, \end{aligned} \quad (1.5.1)$$

где a — величина шага;

e_1, \dots, e_n — координатные орты.

Далее будем предполагать, что $a=1$. Матрицы переходных вероятностей автомата в соответствии с алгоритмом поиска равны

при нештрафе ($\Delta Q' < 0$)

$$A_0 = \begin{vmatrix} 100 \dots 0 \\ 010 \dots 0 \\ \dots \dots \dots \\ 000 \dots 1 \end{vmatrix} \quad (1.5.2)$$

при штрафе ($\Delta Q' \geq 0$)

$$A_1 = \begin{vmatrix} 0100 \dots 0 \\ 0010 \dots 0 \\ \dots \dots \dots \\ 0000 \dots 1 \\ 1000 \dots 0 \end{vmatrix} \quad (1.5.3)$$

Предполагается, что оптимизация идет при наличии помехи, определяемой формулой (0.1.22), и что функция качества $Q(\mathbf{X})$ линейная. Тогда ее приращение $\Delta Q'_i$ определяется формулой (1.2.15), а вероятности штрафов ($\Delta Q'_i \geq 0$) — формулой (1.2.17). Следовательно, вероятности штрафов действий автомата-поиска определяются выражением

$$s_i = \begin{cases} \frac{1}{2} \left[1 + \Phi \left(\frac{\alpha_i}{2\sigma} \right) \right] & \text{при } i=1, 2, \dots, n; \\ \frac{1}{2} \left[1 - \Phi \left(\frac{\alpha_i}{2\sigma} \right) \right] & \text{при } i=n+1, n+2, \dots, 2n, \end{cases} \quad (1.5.4)$$

где $\Phi(z)$ — функция Лапласа;

α_i — направляющие косинусы вектора гради-

ента функции качества $Q(\mathbf{X})$; $\sum_{i=1}^n \alpha_i^2 = 1$.

Функционирование автомата в среде (1.5.4) описывается цепью Маркова со следующей матрицей переходных вероятностей:

$$A(S) = \begin{vmatrix} 1-s_1 & s_1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1-s_2 & s_2 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 1-s_{m-1} & s_{m-1} \\ s_m & 0 & 0 & 0 & \dots & 0 & 0 & 1-s_m \end{vmatrix} \quad (1.5.5)$$

где $m=2n$.

Поскольку функция качества $Q(X)$ линейна, то, принимая $\sigma \neq 0$, получаем однородную эргодическую цепь Маркова. Следовательно, вектор предельных вероятностей $P = (p_1, \dots, p_m)$ пребывания автомата в состояниях i ($i=1, 2, \dots, m$) находится как решение системы линейных уравнений

$$P = A'(S)P, \quad (1.5.6)$$

где $A'(S)$ — транспонированная матрица (1.5.5).

Решая эту систему, находим

$$p_i = \frac{1}{s_i \sum_{j=1}^m \frac{1}{s_j}} \quad (i=1, 2, \dots, m). \quad (1.5.7)$$

Средний штраф автомата равен

$$p_{cp} = \sum_{i=1}^m s_i p_i = \frac{m}{\sum_{i=1}^m \frac{1}{s_i}}, \quad (1.5.8)$$

а среднее смещение в пространстве параметров $\{X\}$

$$M[\Delta X] = (p_1 - p_{n+1}, p_2 - p_{n+2}, \dots, p_n - p_{2n}) =$$

$$= \left(\frac{p_{\text{ср}}}{s_1 - s_{n-1}}, \dots, \frac{p_{\text{ср}}}{s_n - s_{2n}} \right) \frac{1}{m}. \quad (1.5.9)$$

Среднее приращение функции качества на одном шаге

$$M[\Delta Q] = \frac{1}{\sum_{j=1}^m \frac{1}{s_j}} \sum_{i=1}^n \left(\frac{1}{s_i} - \frac{1}{s_{i+n}} \right) \alpha_i. \quad (1.5.10)$$

Подставляя выражения (1.5.4) в формулы (1.5.8) и (1.5.9), в окончательном виде имеем

$$p_{\text{ср}} = \left[\sum_{i=1}^n \frac{1}{1 - \Phi^2\left(\frac{\alpha_i}{2\sigma}\right)} \right]^{-1}; \quad (1.5.11)$$

$$M[\Delta Q] = - \left[\sum_{j=1}^n \frac{1}{1 - \Phi^2\left(\frac{\alpha_i}{2\sigma}\right)} \right]^{-1} \sum_{i=1}^n \frac{\Phi\left(\frac{\alpha_i}{2\sigma}\right)}{1 - \Phi^2\left(\frac{\alpha_i}{2\sigma}\right)} \alpha_i. \quad (1.5.12)$$

Для двумерного случая при большой помехе из формул (1.5.11) и (1.5.12) находим

$$p_{\text{ср}} = \frac{1}{2} \left(1 - \frac{1}{2\pi\sigma^2} \right); \quad (1.5.13)$$

$$M[\Delta Q] = - \frac{1}{2\sqrt{\pi}\sigma}. \quad (1.5.14)$$

Случай без воздействия помех ($\sigma=0$) необходимо рассмотреть особо. В этом случае при выполнении условия (1.3.14) из формулы (1.5.4) следует:

$$s_1 = \dots = s_n = 1; \quad s_{n+1} = \dots = s_{2n} = 0,$$

и поэтому матрица (1.5.5) принимает вид

$$A(S) = \left(\begin{array}{cccc|cccc|c} 0 & 1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 \\ \cdot & \cdot \\ 0 & 0 & 0 & 0 & \dots & 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 & 0 & 0 & \dots & 0 \\ \cdot & \cdot \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 1 & 0 & \dots & 0 \\ \cdot & \cdot \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 1 \end{array} \right) \quad (1.5.15)$$

$\underbrace{\hspace{10em}}_n \quad \underbrace{\hspace{10em}}_n$

Из этой матрицы видно, что цепь Маркова является неэргодической. Ее состояния от 1 до n являются переходными, а состояния от $n+1$ до $2n$ — поглощающими. Как нетрудно показать, N -я степень матрицы (1.5.15) имеет вид

$$A^N(S) = \left(\begin{array}{cccc|cccc|c} 0 & \dots & 1 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \cdot & \cdot \\ 0 & \dots & 0 & 0 & \dots & 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & \dots & 0 & 1 & 0 & 0 & \dots & 0 \\ \cdot & \cdot \\ 0 & \dots & 0 & 0 & \dots & 0 & 1 & 0 & 0 & \dots & 0 \\ \cdot & \cdot \\ 0 & \dots & 0 & 0 & \dots & 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & \dots & 0 & 0 & 1 & 0 & \dots & 0 \\ \cdot & \cdot \\ 0 & \dots & 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 1 \end{array} \right) \quad (1.5.16)$$

$\underbrace{\hspace{10em}}_n \quad \underbrace{\hspace{10em}}_n$

Из этой матрицы получаем следующее распределение вероятностей цепи Маркова на N -м шаге:

$$\begin{aligned} p_1^{(N)} &= \dots = p_N^{(N)} = 0; \\ p_{N+i}^{(N)} &= p_i^{(0)} \quad (i=1, \dots, n-N); \end{aligned} \tag{1.5.17}$$

$$\begin{aligned} p_{n+1}^{(N)} &= p_{n+1}^{(0)} + \sum_{i=n-N+1}^n p_i^{(0)}; \\ p_{n+i}^{(N)} &= p_{n+i}^{(0)} \quad (i=2, \dots, n), \end{aligned}$$

если $1 \leq N \leq n$.

$$\begin{aligned} p_1^{(N)} &= \dots = p_n^{(N)} = 0; \\ p_{n+1}^{(N)} &= p_{n+1}^{(0)} + \sum_{i=1}^n p_i^{(0)}; \\ p_{n+i}^{(N)} &= p_{n+i}^{(0)} \quad (i=2, \dots, n), \end{aligned} \tag{1.5.18}$$

если $n < N \leq \infty$.

Из формул (1.5.18) видно, что данное распределение вероятностей является предельным распределением ($N \rightarrow \infty$). Оно зависит от начального распределения

$$\mathbf{P}^{(0)} = (p_1^{(0)}, \dots, p_{2n}^{(0)}).$$

Среднее приращение функции качества на одном шаге

$$M[\Delta Q] = - \left(\sum_{i=1}^n p_{n+i}^{(0)} \alpha_i + \sum_{i=1}^n p_i^{(0)} \alpha_1 \right). \tag{1.5.19}$$

Оно также зависит от начальных вероятностей.

Далее рассмотрим стохастический вариант алгоритма Гаусса—Зейделя [18]. Отличие этого алгоритма от его детерминированного варианта заключается в том, что последующая переменная спуска выбирается случайно, т. е. после спуска по переменной x_i с равной вероятностью выбирается для спуска любая из переменных x_j ($j=1, \dots, 2n$). Этот алгоритм соответствует вероятностному автомату со следующими матрицами перехода:

при нештрафе ($\Delta Q' < 0$)

$$A_0 = \begin{vmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \dots & 1 \end{vmatrix} \quad (1.5.20)$$

при штрафе ($\Delta Q' \geq 0$)

$$A_1 = \begin{vmatrix} \frac{1}{m} & \dots & \frac{1}{m} \\ \cdot & \cdot & \cdot \\ \frac{1}{m} & \dots & \frac{1}{m} \end{vmatrix} \quad (1.5.21)$$

где $m = 2n$.

Функционирование этого автомата в случайной среде (1.5.4) описывается цепью Маркова со следующей матрицей переходных вероятностей:

$$A(S) = \begin{vmatrix} b_1 & a_1 & a_1 & \dots & a_1 & a_1 \\ a_2 & b_2 & a_2 & \dots & a_2 & a_2 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{m-1} & a_{m-1} & a_{m-1} & \dots & b_{m-1} & a_{m-1} \\ a_m & a_m & a_m & \dots & a_m & b_m \end{vmatrix} \quad (1.5.22)$$

Здесь

$$a_i = \frac{s_i}{m}; \quad b_i = 1 - s_i + \frac{s_i}{m}, \quad (1.5.23)$$

где s_i определяется формулой (1.5.4).

В § 1.3 показано, что предельное распределение вероятностей цепи Маркова (1.5.22) при $\sigma > 0$ равно

$$p_i = \frac{1}{s_i} \left(\sum_{j=1}^m \frac{1}{s_j} \right)^{-1} \quad (1.5.24)$$

$(i = 1, \dots, 2n).$

Следовательно, стохастический и детерминированный варианты алгоритмов Гаусса—Зейделя при оптимизации с помехами эквивалентны в том смысле, что процесс оптимизации описывается одной и той же цепью Маркова для обоих алгоритмов.

Для случая отсутствия помехи ($\sigma=0$) при учете условия (1.3.14) имеем

$$s_1 = \dots = s_n = 1; s_{n+1} = \dots = s_{2n} = 0$$

и, учитывая формулы (1.5.23), получаем

$$A(s=1) = \left(\begin{array}{cccc} \frac{1}{m} & \dots & \frac{1}{m} & \frac{1}{m} & \dots & \frac{1}{m} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \frac{1}{m} & \dots & \frac{1}{m} & \frac{1}{m} & \dots & \frac{1}{m} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & 1 & 0 & \dots & 0 \\ \dots & \dots \\ 0 & \dots & 0 & 0 & 0 & 0 & \dots & 1 \end{array} \right) \quad (1.5.25)$$

N -я степень этой матрицы имеет вид $A^{(N)}(s=1) =$

$$= \left(\begin{array}{cccc} \frac{1}{2^{N-1}m} & \dots & \frac{1}{2^{N-1}m} & \frac{2^N-1}{2^{N-1}m} & \frac{2^N-1}{2^{N-1}m} & \frac{2^N-1}{2^{N-1}m} & \dots & \frac{2^N-1}{2^{N-1}m} \\ \dots & \dots \\ \frac{1}{2^{N-1}m} & \dots & \frac{1}{2^{N-1}m} & \frac{2^N-1}{2^{N-1}m} & \frac{2^N-1}{2^{N-1}m} & \frac{2^N-1}{2^{N-1}m} & \dots & \frac{2^N-1}{2^{N-1}m} \\ 0 & \dots & 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & 1 & 0 & \dots & 0 \\ \dots & \dots \\ 0 & \dots & 0 & 0 & 0 & 0 & \dots & 1 \end{array} \right) \quad (1.5.26)$$

Распределение вероятностей цепи Маркова на N -м шаге равно

$$p_i^{(N)} = \begin{cases} \frac{1}{2^{N-1}m} \sum_{j=1}^n p_j^{(0)} & \text{при } i=1, \dots, n; \\ \frac{2^{N-1}-1}{2^{N-1}m} \sum_{j=1}^n p_j^{(0)} + p_i^{(0)} & \text{при } i=n+1, \dots, 2n, \end{cases} \quad (1.5.27)$$

а предельное распределение вероятностей

$$p_i = \begin{cases} 0 & \text{при } i=1, \dots, n; \\ p_i^{(0)} + \frac{1}{m} \sum_{j=1}^n p_j^{(0)} & \text{при } i=n+1, \dots, 2n. \end{cases} \quad (1.5.28)$$

Среднее смещение по направлению градиента на одном шаге

$$M[\Delta Q] = - \left(\sum_{i=1}^n p_{i+n}^{(0)} \alpha_i + \frac{1}{m} \sum_{i=1}^n \alpha_i \sum_{j=1}^n p_j^{(0)} \right). \quad (1.5.29)$$

Сравнивая формулы (1.5.19) и (1.5.29), видим, что при поиске в отсутствие помех ($\sigma=0$) преимущество имеет стохастический вариант алгоритма, так как на его быстрдействие меньше влияет направление градиентного вектора минимизируемой функции.

§ 1.6. ДЕТЕРМИНИРОВАННЫЙ АНАЛОГ НЕЛИНЕЙНОГО АЛГОРИТМА СЛУЧАЙНОГО ПОИСКА

Рассмотрим детерминированный аналог нелинейного алгоритма случайного поиска [19]. Предполагается, что поиск производится по координатным осям в пространстве $\{X\}$, т. е. этот поиск является покоординатным (см. формулы (1.5.1)).

Суть этого алгоритма заключается в том, что при $\Delta Q' \geq 0$ происходит возврат в исходное состояние, а при $\Delta Q' < 0$ осуществляется смещение по переменной x_{i+1} , если на предыдущем шаге поиска происходило смещение по переменной x_i ($i=1, \dots, n, n+1, \dots, 2n$), где $j = n+i$ обозначает переменную $-x_i$ [19]. Этот алгоритм поиска задается автоматом со следующими матрицами переходных вероятностей:

при нештрафе ($\Delta Q' < 0$)

$$A_0 = \begin{pmatrix} 0100 \dots 0 \\ 0010 \dots 0 \\ \dots \dots \dots \\ 0000 \dots 1 \\ 1000 \dots 0 \end{pmatrix} \quad (1.6.1)$$

при штрафе ($\Delta Q' \geq 0$)

$$A_1 = \begin{pmatrix} 000 \dots 0100 \dots 0 \\ 000 \dots 0010 \dots 0 \\ \dots \dots \dots \\ 000 \dots 0000 \dots 1 \\ \dots \dots \dots \\ 100 \dots 0000 \dots 0 \\ 010 \dots 0000 \dots 0 \\ \dots \dots \dots \\ 000 \dots 1000 \dots 0 \end{pmatrix} \quad (1.6.2)$$

$\underbrace{\hspace{10em}}_n \quad \underbrace{\hspace{10em}}_n$

Функционирование этого автомата в случайной среде (1.5.4) описывается цепью Маркова с матрицей переходных вероятностей

В случае, когда $\sigma \neq 0$ ($a_i \neq 0$; $b_i \neq 1$), эта цепь Маркова эргодическая. По матрице (1.6.6) для нахождения предельных вероятностей получаем следующую систему алгебраических уравнений:

$$\left. \begin{aligned}
 p_1 &= a_1 p_{n+1} + b_n p_{2n} \\
 p_2 &= a_1 p_1 + a_2 p_{n+2} \\
 p_3 &= a_2 p_2 + a_3 p_{n+3} \\
 &\dots \\
 p_i &= a_{i-1} p_{i-1} + a_i p_{n+i} \\
 &\dots \\
 p_n &= a_{n-1} p_{n-1} + a_n p_{2n}
 \end{aligned} \right\} n$$

$$\left. \begin{aligned}
 p_{n+1} &= b_1 p_1 + a_n p_n \\
 p_{n+2} &= b_2 p_2 + b_1 p_{n+1} \\
 &\dots \\
 p_{n+i} &= b_i p_i + b_{i-1} p_{n+i-1} \\
 &\dots \\
 p_{2n} &= b_n p_n + b_{n-1} p_{2n-1}
 \end{aligned} \right\} n$$
(1.6.7)

Введем обозначения

$$k_1 = b_n p_{2n}; \quad k_2 = a_n p_n. \quad (1.6.8)$$

Тогда из первого и $(n+1)$ -го уравнений этой системы находим

$$p_1 = \frac{a_1 k_2 + k_1}{1 - a_1 b_1} = \frac{g_{12} k_2 + g_{11} k_1}{u_1};$$

$$p_{n+1} = \frac{k_2 + b_1 k_1}{1 - a_1 b_1} = \frac{f_{12} k_2 + f_{11} k_1}{u_1},$$
(1.6.9)

где

$$g_{12} = a_1; \quad g_{11} = 1; \quad f_{12} = 1; \quad f_{11} = b_1; \quad u_1 = 1 - a_1 b_1. \quad (1.6.10)$$

Из i -го и $(n+i)$ -го уравнений получаем

$$p_i = \frac{g_{i2} k_2 + g_{i1} k_1}{v_i}; \quad p_{n+i} = \frac{f_{i2} k_2 + f_{i1} k_1}{v_i}. \quad (1.6.11)$$

$$\text{Здесь } v_i = u_1 u_2 \dots u_i; u_i = 1 - a_i b_i; \quad (1.6.12)$$

$g_{i2}; g_{i1}; f_{i2}; f_{i1}$ определяются по рекуррентным формулам

$$g_{i2} = a_{i-1} g_{i-1,2} + a_i b_{i-1} f_{i-1,2}; \quad (1.6.13)$$

$$g_{i1} = a_{i-1} g_{i-1,1} + a_i b_{i-1} f_{i-1,1};$$

$$f_{i2} = b_i a_{i-1} g_{i-1,2} + b_{i-1} f_{i-1,2}; \quad (1.6.14)$$

$$f_{i1} = b_i a_{i-1} g_{i-1,1} + b_{i-1} f_{i-1,1}$$

$$(i=2, 3, \dots, n).$$

Сравнивая формулы (1.6.10), (1.6.13) и (1.6.14), нетрудно заметить, что выражение для f_{i2} получается из формулы для g_{i1} , а для f_{i1} — из формулы для g_{i2} при замене a_i на b_i и b_i — на a_i ($i=1, 2, \dots, n$). При $i=n$ из формул (1.6.11) получаем

$$p_n = \frac{g_{n2} k_2 + g_{n1} k_1}{v_n}, \quad p_{2n} = \frac{f_{n2} k_2 + f_{n1} k_1}{v_n}. \quad (1.6.15)$$

Из равенств (1.6.8) и (1.6.15) находим соотношение k_1 и k_2 :

$$k_2 = G k_1; k_1 = F k_2, \quad (1.6.16)$$

где

$$G = \frac{a_n g_{n1}}{v_n - a_n g_{n2}}; \quad F = \frac{b_n f_{n2}}{v_n - b_n f_{n1}}; \quad (1.6.17)$$

$$F = \frac{1}{G}. \quad (1.6.18)$$

Отсюда видно, что функция F симметрична G относительно a_i и b_i , т. е. F получается из G , если a_i заменить на b_i . Из условия нормирования $\sum_{i=1}^{2n} p_i = 1$ находим

$$H_2 k_2 + H_1 k_1 = 1, \quad (1.6.19)$$

где

$$H_2 = \sum_{i=1}^n \frac{g_{i2} + f_{i2}}{v_i}; \quad H_1 = \sum_{i=1}^n \frac{g_{i1} + f_{i1}}{v_i}. \quad (1.6.20)$$

Из формул (1.6.16) и (1.6.19) получаем

$$k_1 = \frac{1}{H_2 G + H_1} = \frac{F}{H_2 + H_1 F}; \quad (1.6.21)$$

$$k_2 = \frac{1}{H_2 + H_1 F} = \frac{G}{H_2 G + H_1}. \quad (1.6.22)$$

Следовательно, формулы для определения предельных вероятностей имеют вид

$$p_i = \frac{1}{v_i} \left(\frac{g_{i2}}{H_2 + H_1 F} + \frac{g_{i1}}{H_2 G + H_1} \right) = \frac{g_{i2} G + g_{i1}}{v_i} k_1;$$

$$p_{n+i} = \frac{1}{v_i} \left(\frac{f_{i2}}{H_2 + H_1 F} + \frac{f_{i1}}{H_2 G + H_1} \right) = \frac{f_{i2} G + f_{i1}}{v_i} k_1. \quad (1.6.23)$$

Поскольку H_2 и H_1 симметричны по отношению к a_i и b_i , то p_i и p_{i+n} также симметричны, т. е. p_{i+n} определяется из формул для p_i подстановкой a_i и b_i на место b_i и a_i соответственно.

Среднее приращение функции качества на одном шаге равно

$$M[\Delta Q] = \sum_{i=1}^n (p_i - p_{n+i}) \alpha_i = \sum_{i=1}^n \frac{1}{v_i} \left(\frac{g_{i2} - f_{i2}}{H_2 + H_1 F} + \frac{g_{i1} - f_{i1}}{H_2 G + H_1} \right) \alpha_i. \quad (1.6.24)$$

Далее рассмотрим двумерный случай ($n=2$). Для этого случая матрица (1.6.3) принимает вид

$$A(\mathbf{S}) = \begin{vmatrix} 0 & a_1 & b_1 & 0 \\ 0 & 0 & a_2 & b_2 \\ a_1 & 0 & 0 & b_1 \\ b_2 & a_2 & 0 & 0 \end{vmatrix} \quad (1.6.25)$$

Исходя из этой матрицы имеем следующую систему ал-

гебраических уравнений для определения предельных вероятностей:

$$\begin{aligned} p_1 &= a_1 p_3 + b_2 p_4; \\ p_2 &= a_1 p_1 + a_2 p_4; \\ p_3 &= b_1 p_1 + a_2 p_2; \\ p_4 &= b_2 p_2 + b_1 p_3. \end{aligned} \tag{1.6.26}$$

Решая систему уравнений с учетом формул (1.6.5), находим:

$$\begin{aligned} p_1 &= \frac{b_2 + a_1 a_2^2}{2 + b_2 - a_1 b_1 + a_2}; & p_2 &= \frac{a_1 + a_2 b_1^2}{2 + b_2 - a_1 b_1 + a_2^2}; \\ p_3 &= \frac{a_2 + b_1 b_2^2}{2 + b_2 - a_1 b_1 + a_2^2}; & p_4 &= \frac{b_1 + a_1^2 b_2}{2 + b_2 - a_1 b_1 + a_2^2}. \end{aligned} \tag{1.6.27}$$

Подставляя вместо a_i и b_i величины s_i из формул (1.6.4), получаем:

$$\begin{aligned} p_1 &= \frac{s_2 + s_3 s_4^2}{2 + \sum_{j=1}^3 s_j s_{j+1}^2 + s_4 s_1^2}; & p_2 &= \frac{s_3 + s_4 s_1^2}{2 + \sum_{j=1}^3 s_j s_{j+1}^2 + s_4 s_1^2}; \\ p_3 &= \frac{s_4 + s_1 s_2^2}{2 + \sum_{j=1}^3 s_j s_{j+1}^2 + s_4 s_1^2}; & p_4 &= \frac{s_1 + s_2 s_3^2}{2 + \sum_{j=1}^3 s_j s_{j+1}^2 + s_4 s_1^2}. \end{aligned} \tag{1.6.28}$$

Среднее приращение функции качества равно

$$\begin{aligned} M[\Delta Q] &= \\ &= \frac{(s_2 - s_4 + s_3 s_4^2 - s_1 s_2^2) \alpha_1 + (s_3 - s_1 + s_4 s_1^2 - s_2 s_3^2) \alpha_2}{2 + \sum_{j=1}^3 s_j s_{j+1}^2 + s_4 s_1^2}. \end{aligned} \tag{1.6.29}$$

Рассмотрим случай без воздействия помехи, т. е. когда $\sigma=0$. Тогда по формулам (1.5.4) имеем:

$$s_i = \begin{cases} 1 & \text{при } i=1, \dots, n; \\ 0 & \text{при } i=n+1, \dots, 2n. \end{cases} \quad (1.6.30)$$

По формулам (1.6.4) находим:

$$b_i = \begin{cases} 1 & \text{при } i=1, \dots, n; \\ 0 & \text{при } i=n+1, \dots, 2n; \end{cases} \quad (1.6.31)$$

$$a_i = \begin{cases} 0 & \text{при } i=1, \dots, n; \\ 1 & \text{при } i=n+1, \dots, 2n. \end{cases}$$

Матрица (1.6.6) при условии (1.3.14) имеет вид

$$A(S) = \begin{vmatrix} 00 \dots 1000 \dots 00 \\ 00 \dots 0100 \dots 00 \\ \dots \dots \dots \dots \dots \\ 00 \dots 0000 \dots 10 \\ 00 \dots 0010 \dots 00 \\ \dots \dots \dots \dots \dots \\ 00 \dots 0000 \dots 01 \\ 10 \dots 0000 \dots 00 \end{vmatrix} \quad (1.6.32)$$

Из матрицы видно, что состояния от 2 до n этой цепи Маркова являются невозвратными, а состояние 1 и состояния от $n+1$ до $2n$ — циклическими с периодом $n+1$. Следовательно, предельные вероятности равны

$$p_1 = p_{n+1} = \dots = p_{2n} = \frac{1}{n+1}; \quad (1.6.33)$$

$$p_2 = p_3 = \dots = p_n = 0.$$

Среднее приращение функции качества

$$M[\Delta Q] = -\frac{1}{n+1} \sum_{i=2}^n \alpha_i. \quad (1.6.34)$$

Для двумерного случая ($n=2$) по формулам (1.6.33) и (1.6.34) имеем

$$p_1 = p_3 = p_4 = \frac{1}{3}; \quad p_2 = 0; \quad (1.6.35)$$

$$M[\Delta Q] = -\frac{1}{3} \alpha_2. \quad (1.6.36)$$

§ 1.7. АЛГОРИТМ ПОСЛЕДОВАТЕЛЬНОГО ГРАДИЕНТА

В данном параграфе рассмотрим одну модификацию метода градиента. Эта модификация характеризуется тем, что после пробного шага $g_i > 0$ рабочий шаг, имеющий величину

$$\Delta x_i = -a_i \operatorname{sign} \Delta Q_{ig} \quad (1.7.1)$$

(где ΔQ_{ig} — приращение функции качества в результате пробного шага g_i), делается в i -м направлении. Такая процедура производится по всем координатам. Строго говоря, этот метод нельзя назвать градиентным. Это, скорее, полярный метод. Представим алгоритм как вероятностный автомат.

1. *Одномерный случай* ($n=1$). Состояния автомата пронумеруем в следующем порядке:

$$\Delta X^{(1)} = g; \quad \Delta X^{(2)} = -a; \quad \Delta X^{(3)} = +a. \quad (1.7.2)$$

Здесь g — пробный шаг, а $\pm a$ — рабочие шаги поиска. Пусть вход автомата, как обычно, определяется величиной штрафа

$$c = \operatorname{sgn} \Delta Q' = \begin{cases} 1, & \text{если } \Delta Q' \geq 0; \\ 0, & \text{если } \Delta Q' < 0. \end{cases} \quad (1.7.3)$$

Выход автомата совпадает с его состоянием и образует трехбуквенный алфавит (1.7.2).

Далее предположим, что $a=g=1$. Графы переходов автомата показаны на рис. 1.7.1, а его матрицы переходов имеют вид

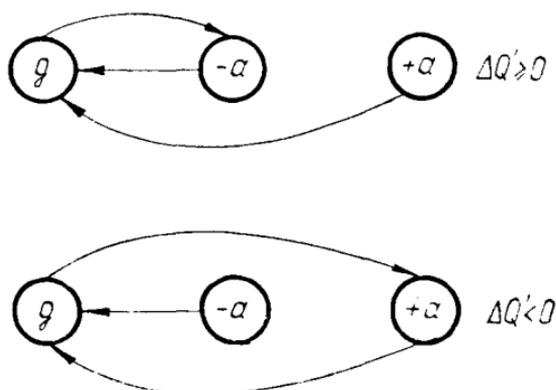


Рис. 1.7.1. Графы переходов автомата для случая $n=1$.

при нештрафе ($\Delta Q' < 0$)

$$A_0 = \begin{vmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{vmatrix} \quad (1.7.4)$$

при штрафе ($\Delta Q' \geq 0$)

$$A_1 = \begin{vmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{vmatrix} \quad (1.7.5)$$

Случайная среда задается через вероятности штрафов действия автомата, которые в предположении о линейности функции качества ($\text{grad } Q(X) = \text{const}$) и нормированности его градиента $\left(\sum_{i=1}^n \alpha_i^2 = 1 \right)$ равны

$$s_1 = s_3 = \frac{1}{2} \left[1 + \Phi \left(\frac{1}{2\sigma} \right) \right];$$

$$s_2 = \frac{1}{2} \left[1 - \Phi \left(\frac{1}{2\sigma} \right) \right]. \quad (1.7.6)$$

Функционирование автомата поиска в случайной среде (1.7.6) описывается цепью Маркова с циклической матрицей переходных вероятностей

$$A(S) = \begin{vmatrix} 0 & s_1 & 1-s_1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{vmatrix} \quad (1.7.7)$$

Предельные вероятности для этой цепи Маркова равны

$$p_1 = \frac{1}{2}; \quad p_2 = \frac{1}{2} s_1; \quad p_3 = \frac{1}{2} (1-s_1). \quad (1.7.8)$$

Средний штраф рабочих шагов

$$p_{\text{ср. п}} = s_1 s_2 = \frac{1}{4} \left[1 - \Phi^2 \left(\frac{1}{2\sigma} \right) \right], \quad (1.7.9)$$

а среднее смещение по направлению градиента

$$M[\Delta Q] = p_3 - p_2 = \frac{1}{2} (1 - 2s_1) = -\frac{1}{2} \Phi \left(\frac{1}{2\sigma} \right). \quad (1.7.10)$$

При $\sigma \rightarrow 0$ имеем

$$p_{\text{ср. п}} \rightarrow 0; \quad M[\Delta Q] \rightarrow -\frac{1}{2}, \quad (1.7.11)$$

а при $\sigma \rightarrow \infty$

$$p_{\text{ср. п}} \rightarrow \frac{1}{4}; \quad M[\Delta Q] \rightarrow 0. \quad (1.7.12)_a$$

2. *Двумерный случай* ($n=2$). Состояния автомата пронумеруем в следующем порядке:

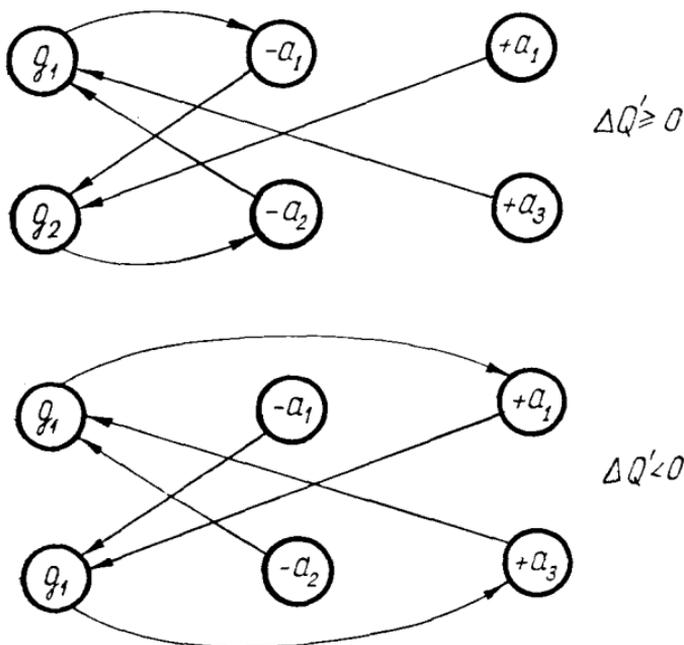


Рис. 1.7.2. Графы переходов автомата для случая $n=2$.

$$\begin{aligned} \Delta X^{(1)} &= g e_1; & \Delta X^{(2)} &= -a e_1; & \Delta X^{(3)} &= a e_1; \\ \Delta X^{(4)} &= g e_2; & \Delta X^{(5)} &= -a e_2; & \Delta X^{(6)} &= a e_2. \end{aligned} \tag{1.7.13}$$

Графы переходов автомата показаны на рис. 1.7.2. Имеем следующие матрицы переходов автомата: при нештрафе ($\Delta Q' < 0$)

$$A_0 = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{1.7.14}$$

при штрафе ($\Delta Q' \geq 0$)

$$A_1 = \begin{vmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{vmatrix} \quad (1.7.15)$$

Функционирование автомата описывается цепью Маркова

$$A(S) = \begin{vmatrix} 0 & s_1 & 1-s_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & s_4 & 1-s_4 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{vmatrix} \quad (1.7.16)$$

где

$$s_1 = s_3 = \frac{1}{2} \left[1 + \Phi \left(\frac{\alpha_1}{2\sigma} \right) \right]; \quad s_2 = \frac{1}{2} \left[1 - \Phi \left(\frac{\alpha_1}{2\sigma} \right) \right];$$

$$s_4 = s_6 = \frac{1}{2} \left[1 + \Phi \left(\frac{\alpha_2}{2\sigma} \right) \right]; \quad (1.7.17)$$

$$s_5 = \frac{1}{2} \left[1 - \Phi \left(\frac{\alpha_2}{2\sigma} \right) \right],$$

α_i — направляющие косинусы вектора-градиента функции

$$Q(X) = \alpha_1 x_1 + \alpha_2 x_2; \quad \alpha_1^2 + \alpha_2^2 = 1. \quad (1.7.18)$$

Предельные вероятности цепи Маркова равны

$$p_1 = p_4 = \frac{1}{4}; \quad p_2 = \frac{1}{4} s_1; \quad p_3 = \frac{1}{4} (1 - s_1);$$

(1.7.19)

$$p_5 = \frac{1}{4} s_4; \quad p_6 = \frac{1}{4} (1 - s_4).$$

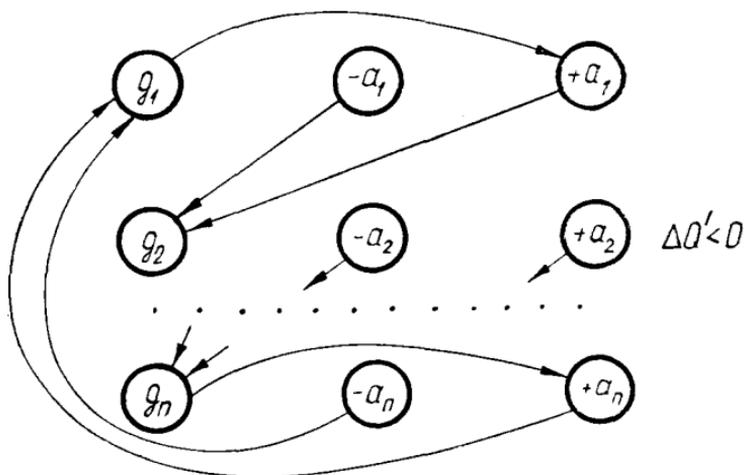
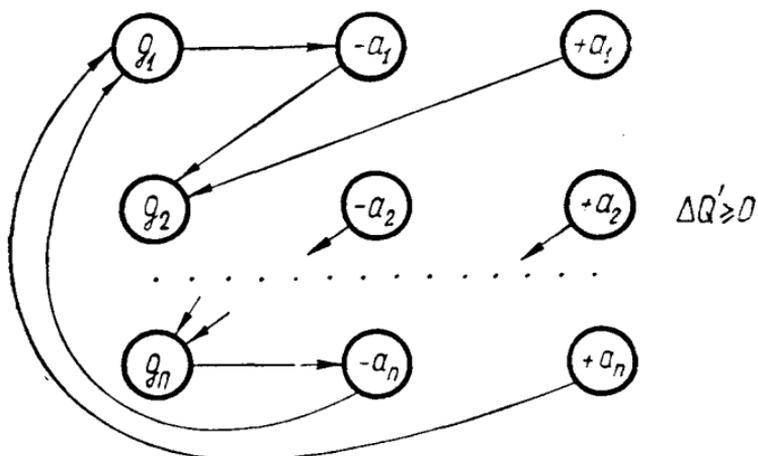


Рис. 1.7.3. Графы переходов автомата для случая $n > 2$.

Средний штраф рабочих шагов

$$p_{\text{ср. р}} = \frac{1}{2} (s_1 s_2 + s_4 s_5) = \frac{1}{8} \left[2 - \Phi^2 \left(\frac{\alpha_1}{2\sigma} \right) - \Phi^2 \left(\frac{\alpha_2}{2\sigma} \right) \right], \quad (1.7.20)$$

а среднее приращение показателя качества

$$M[\Delta Q] = \frac{1}{4} [(1 - 2s_1)\alpha_1 + (1 - 2s_4)\alpha_2] = -\frac{1}{4} \left[\Phi \left(\frac{\alpha_1}{2\sigma} \right) \alpha_1 + \Phi \left(\frac{\alpha_2}{2\sigma} \right) \alpha_2 \right]. \quad (1.7.21)$$

При $\sigma \rightarrow 0$ имеем

$$p_{\text{ср. р}} \rightarrow 0; \quad M[\Delta Q] \rightarrow -\frac{1}{4} (\alpha_1 + \alpha_2), \quad (1.7.22)$$

а при $\sigma \rightarrow \infty$

$$p_{\text{ср. р}} \rightarrow \frac{1}{4}; \quad M[\Delta Q] \rightarrow 0. \quad (1.7.23)$$

3. Многомерный случай ($n > 2$).

$$Q(\mathbf{X}) = \sum_{i=1}^n \alpha_i x_i; \quad \sum_{i=1}^n \alpha_i^2 = 1. \quad (1.7.24)$$

Графы переходов автомата для этого случая показаны на рис. 1.7.3. Матрицы переходов имеют вид при нештрафе ($\Delta Q' < 0$)

$$A_0 = \left\| \begin{array}{cccccc} B_3 & B_1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & B_3 & B_1 & 0 & \cdots & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & \cdots & B_3 & B_1 \\ B_1 & 0 & 0 & 0 & \cdots & 0 & B_3 \end{array} \right\| \quad (1.7.25)$$

при штрафе ($\Delta Q' \geq 0$)

$$A_1 = \begin{vmatrix} B_2 & B_1 & 0 & 0 & \dots & 0 & 0 \\ 0 & B_2 & B_1 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & B_3 & B_1 \\ B_1 & 0 & 0 & 0 & \dots & 0 & B_3 \end{vmatrix} \quad (1.7.26)$$

где

$$B_1 = \begin{vmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{vmatrix} \quad B_2 = \begin{vmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix} \quad B_3 = \begin{vmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix} \quad (1.7.27)$$

Функционирование автомата в случайной среде, заданной функцией (1.7.24), описывается цепью Маркова

$$A(S) = \begin{vmatrix} B(s_1) & B_1 & 0 & 0 & \dots & 0 & 0 \\ 0 & B(s_2) & B_1 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & B(s_{n-1}) & B_1 \\ B_1 & 0 & 0 & 0 & \dots & 0 & B(s_n) \end{vmatrix} \quad (1.7.28)$$

Здесь

$$B(s_i) = \begin{vmatrix} 0 & s_i & 1-s_i \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix} \quad (1.7.29)$$

$$s_i = \frac{1}{2} \left[1 + \Phi \left(\frac{\alpha_i}{2\sigma} \right) \right], \quad (1.7.30)$$

где Φ — интеграл Лапласа. Обозначим через $P_i = (p_{zi-2}, p_{zi-1}, p_{zi})$ вектор предельных вероятностей, соответствующий i -й переменной функции (1.7.24). Для определения этих векторов по матрице (1.7.28) составим систему векторных уравнений:

$$\begin{aligned}
 \mathbf{P}_1 &= \mathbf{P}_1 B(s_1) + \mathbf{P}_n B_1; \\
 \mathbf{P}_2 &= \mathbf{P}_1 B_1 + \mathbf{P}_2 B(s_2); \\
 \mathbf{P}_3 &= \mathbf{P}_2 B_1 + \mathbf{P}_3 B(s_3); \\
 &\dots \\
 \mathbf{P}_i &= \mathbf{P}_{i-1} B_1 + \mathbf{P}_i B(s_i); \\
 &\dots \\
 \mathbf{P}_n &= \mathbf{P}_{n-1} B_1 + \mathbf{P}_n B(s_n).
 \end{aligned} \tag{1.7.31}$$

Из i -го векторного уравнения этой системы следует

$$\mathbf{P}_i \begin{vmatrix} 1 & -s_i & -1+s_i \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix} = \mathbf{P}_{i-1} \begin{vmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{vmatrix} \tag{1.7.32}$$

Отсюда для координат векторов \mathbf{P}_i и \mathbf{P}_{i-1} получаем соотношения

$$\begin{aligned}
 p_{3i-2} &= p_{3(i-1)-1} + p_{3(i-1)} = p_{3i-4} + p_{3i-3} = p_{3i-5}; \\
 p_{3i-1} &= p_{3i-2} s_i; \\
 p_{3i} &= p_{3i-2} (1 - s_i).
 \end{aligned} \tag{1.7.33}$$

Из первого уравнения системы (1.7.31) имеем:

$$\begin{aligned}
 p_1 &= p_{3n-1} + p_{3n}; \\
 p_2 &= s_1 p_1; \\
 p_3 &= (1 - s_1) p_1.
 \end{aligned} \tag{1.7.34}$$

После нормировки величин p_j ($j=1, 2, \dots, 3n$) находим

$$\begin{aligned}
 p_{3i-2} &= \frac{1}{2n} \quad (i=1, \dots, n); \\
 p_{3i-1} &= \frac{s_i}{2n} \quad (i=1, \dots, n); \\
 p_{3i} &= \frac{1-s_i}{2n} \quad (i=1, \dots, n).
 \end{aligned} \tag{1.7.35}$$

Средний штраф рабочих шагов равен

$$\rho_{\text{ср. р}} = \sum_{i=1}^n \left[\frac{1-s_i}{2n} s_i + \frac{(1-s_i)s_i}{2n} \right] = \sum_{i=1}^n \frac{(1-s_i)s_i}{n}, \quad (1.7.36)$$

а среднее смещение по направлению градиента

$$\begin{aligned} M[\Delta Q] &= \sum_{i=1}^n (p_{3i} - p_{3i-1}) \alpha_i = \sum_{i=1}^n \frac{1-2s_i}{2n} \alpha_i = \\ &= -\frac{1}{2n} \sum_{i=1}^n \Phi\left(\frac{\alpha_i}{2\sigma}\right) \alpha_i. \end{aligned} \quad (1.7.37)$$

Для случая отсутствия помех ($\sigma=0$) имеем

$$\rho_{\text{ср. р}} = 0; \quad M[\Delta Q] = -\frac{1}{2n} \sum_{i=1}^n \alpha_i, \quad (1.7.38)$$

а при $\sigma \rightarrow \infty$

$$\rho_{\text{ср. р}} \rightarrow \frac{1}{4}; \quad M[\Delta Q] \rightarrow 0. \quad (1.7.39)$$

§ 1.8. АЛГОРИТМ ПАРАЛЛЕЛЬНОГО ГРАДИЕНТА

Рассмотрим параллельную модификацию метода. Этот алгоритм представляет собой поиск по вершинам гиперкуба. По всем координатам делаются пробные шаги и определяется полярное направление градиента. Это полярное направление соответствует направлению на вершину гиперкуба, которая наиболее близка к направлению градиента. Схема алгоритма параллельного градиента показана на рис. 1.8.1. Алгоритм в терминах теории автоматов представляется в виде двух автоматов — автомата пробных шагов $A^{(n)}$, имеющего $n+1$ внутренних состояний, и автомата рабочих шагов $A^{(p)}$, имеющего $m+1$ ($m=2^n$) внутренних состояний.

Пронумеруем возможные рабочие шаги $\Delta X^{(i)}$ в следующем порядке:

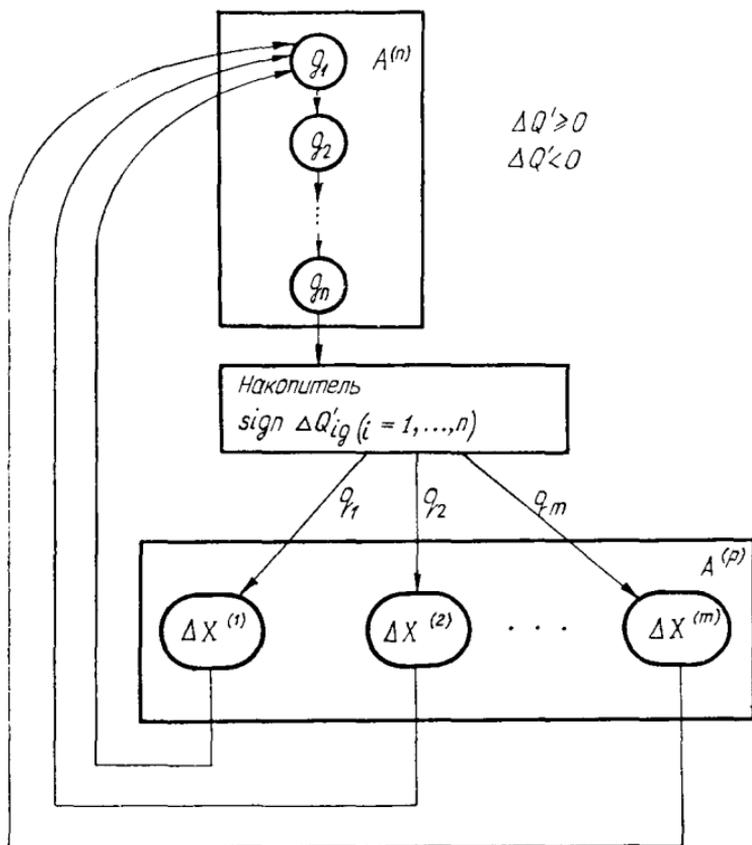


Рис. 1.8.1. Схема оптимизации методом параллельного градиента.

$$\begin{aligned}
 \Delta X^{(1)} &= (1, \dots, 1); \quad \Delta X^{(2)} = (1, \dots, 1, -1); \quad \Delta X^{(3)} = \\
 &= (1, \dots, 1, -1, 1); \quad \Delta X^{(4)} = (1, \dots, 1, -1, -1); \quad \dots \\
 &\dots; \quad \Delta X^{(m/2-1)} = (1, -1, \dots, -1, 1); \quad \dots; \quad \Delta X^{m/2} = \\
 &= (1, -1, \dots, -1); \quad \Delta X^{(m+i)} = -\Delta X^{(i)} \quad (1.8.1) \\
 &\quad \left(i = 1, 2, \dots, \frac{m}{2} \right).
 \end{aligned}$$

У автомата пробных шагов внутренними состояниями являются пробные шаги $g_i e_i$ (их число равно n) и одно фиктивное состояние — автомат рабочих шагов. У автомата рабочих шагов внутренними состояниями явля-

ются рабочие шаги $\Delta X^{(i)}$ (их число равно $m=2^n$) и одно фиктивное состояние — автомат пробных шагов.

Определим множество входов автоматов. Пусть входом автомата пробных шагов $A^{(n)}$ является знак приращения функции качества. Входы автомата рабочих шагов $A^{(p)}$ определим следующим образом. Пусть в состояниях от 1 до m автомат принимает на вход только числа $c=0$ и $c=1$, а в состоянии $m+1$ (фиктивное состояние) — только векторы $\mathbf{C}=(c_1, \dots, c_n)$, координаты c_i которых могут иметь значения 0 или 1. При этом i -я координата вектора \mathbf{C} равна нулю, если автомат пробных шагов $A^{(n)}$ в i -м состоянии не получил штрафа ($c=0$), и равна единице, если получил штраф ($c=1$). Следовательно, вектор $\mathbf{C}=(c_1, \dots, c_n)$ представляет собой набор величин c , полученных при пробных шагах. Векторы $\mathbf{C}_j=(c_1^{(j)}, \dots, c_n^{(j)})$, количество которых равно 2^n , пронумеруем в том же порядке, как рабочие шаги $\Delta X^{(i)}$, т. е.

$$\begin{aligned} \mathbf{C}_1 &= (0, \dots, 0); \mathbf{C}_2 = (0, \dots, 0, 1); \mathbf{C}_3 = (0, \dots, 0, 1, 0); \dots \\ &\dots; \mathbf{C}_{m/2-1} = (0, 1, 1, \dots, 1, 0); \mathbf{C}_{m/2} = (0, 1, \dots \\ &\dots, 1); \dots; \mathbf{C}_{m-1} = (1, 1, \dots, 1, 0); \mathbf{C}_m = (1, \dots, 1). \end{aligned} \quad (1.8.2)$$

Нетрудно заметить, что векторы \mathbf{C}_i ($i=1, \dots, 2^n$) образуются некоторым неоднородным автоматом «памяти» со следующими, зависящими от номера шага N матрицами переходов:

$$\begin{aligned} A_0(N) &= \left\| \begin{array}{cc} \underbrace{I}_{2^{N-1}} & 0 \\ 0 & \underbrace{I}_{2^{n-2^{N-1}}} \end{array} \right\| \left. \begin{array}{l} 2^{N-1} \\ 2^{n-2^{N-1}} \end{array} \right\} \\ & \hspace{15em} - \\ A_1(N) &= \left\| \begin{array}{ccc} 0 & I & 0 \\ 0 & 0 & 0 \end{array} \right\| \left. \begin{array}{l} 2^{N-1} \\ 2^{n-2^{N-1}} \end{array} \right\} \end{aligned} \quad (1.8.3)$$

где I — единичная матрица порядка 2^{N-1} ($N=1, 2, \dots, n$).

Таким образом, автомат рабочих шагов $A^{(p)}$ представляет собой автомат с ограничениями на входе.

На рис. 1.8.2 показаны графы переходов из одного состояния в другое для автомата пробных шагов $A^{(n)}$ и для автомата рабочих шагов $A^{(p)}$ при $c=0$; $c=1$; $C_j = (c_1^{(j)}, \dots, c_n^{(j)})$. Из этого рисунка получаем следующие матрицы переходных вероятностей:

$$A_1^{(n)} = A_0^{(n)} = \underbrace{\begin{pmatrix} 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & 0 & 0 & \dots & 0 & 0 \end{pmatrix}}_{n+1} \quad \left. \vphantom{\begin{pmatrix} 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & 0 & 0 & \dots & 0 & 0 \end{pmatrix}} \right\} n+1 \quad (1.8.4)$$

2) для автомата рабочих шагов

$$A_j^{(p)} = \underbrace{\begin{pmatrix} 0 & 0 & \dots & 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 & 1 \\ \dots & \dots \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & \dots & 1 & 0 & \dots & 0 & 0 \end{pmatrix}}_{\substack{j \\ m+1-j}} \quad \left. \vphantom{\begin{pmatrix} 0 & 0 & \dots & 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 & 1 \\ \dots & \dots \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & \dots & 1 & 0 & \dots & 0 & 0 \end{pmatrix}} \right\} m+1 \quad (1.8.5)$$

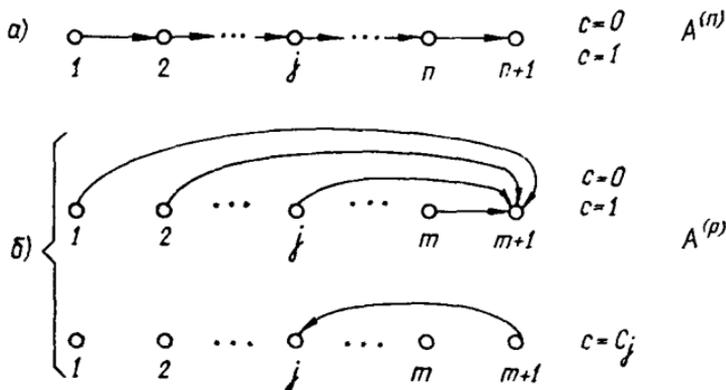


Рис. 1.8.2. Графы переходов автомата пробных шагов (а) и автомата рабочих шагов (б).

Здесь индекс j обозначает состояние рабочего автомата, на которое он переходит из фиктивного состояния. При этом j совпадает с номером вектора штрафа $C_j = (c_1^{(j)}, \dots, c_n^{(j)})$.

Объединяя автомат пробных и автомат рабочих шагов, получаем автомат, количество состояний которого равно $n+t$ и переходные матрицы имеют вид

$$A_j = \left(\begin{array}{cccccccc} 0 & \dots & 0 & 0 & \dots & 1 & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & \dots & 1 & 0 & 0 & \dots & 0 \\ \cdot & \cdot \\ 0 & \dots & 0 & 0 & \dots & 1 & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ \cdot & \cdot \\ 0 & \dots & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 1 \\ 0 & \dots & 1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \end{array} \right) \quad (1.8.6)$$

$\underbrace{\hspace{10em}}_m$
 $\underbrace{\hspace{10em}}_n$
 $\underbrace{\hspace{10em}}_m$
 $\underbrace{\hspace{10em}}_n$

Среда задается вероятностями появления векторов $C_j = (c_1^{(j)}, \dots, c_n^{(j)})$, т. е. вероятностями q_j выбора j -го рабочего шага после окончания пробных шагов. Вектор вероятностей $Q = (q_1, \dots, q_{2^n})$, согласно нумерации векторов $\Delta X^{(j)}$ (1.8.1) и переходных матриц (1.8.3), определяется формулой

$$Q = Q^{(0)} \prod_{N=1}^n [s_N A_1(N) + (1 - s_N) A_0(N)]. \quad (1.8.7)$$

Здесь $Q^{(0)}$ — 2^n -мерный начальный вектор состояний автомата «памяти» (1.8.3); $Q^{(0)} = (1, 0, \dots, \dots, 0)$;

s_N — вероятность штрафа N -го пробного шага;

$$s_N = \text{Вер} (\Delta Q'_N \geq 0) = \frac{1}{2} \left[1 + \Phi \left(\frac{g \alpha_N}{2\sigma} \right) \right], \quad (1.8.8)$$

где α_N ($N=1, \dots, n$) — направляющие косинусы градиента функции $Q(X)$.

Функционирование объединенного автомата в случайной среде (1.8.8) описывается цепью Маркова со следующей циклической переходной матрицей:

$$A_0 = A_1 = \begin{pmatrix} 0 & \dots & 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 1 & 0 & 0 & \dots & 0 \\ \dots & \dots \\ 0 & \dots & 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & 1 & 0 & \dots & 0 \\ \dots & \dots \\ 0 & \dots & 0 & 0 & 0 & 0 & \dots & 1 \\ \underbrace{q_1 \dots q_m}_m & \underbrace{0 \ 0 \ 0 \ \dots \ 0}_n & & & & & & \end{pmatrix} \quad (1.8.9)$$

Предельные вероятности этой цепи находятся как решение системы алгебраических уравнений

$$p_i = q_i p_{m+n} \quad (i = 1, \dots, m);$$

$$p_{m+1} = \sum_{i=1}^m p_i; \quad (1.8.10)$$

$$p_{m+2} = p_{m+1}; \dots \cdot p_{m+n} = p_{m+n-1}.$$

Решением этой системы является

$$p_i = \begin{cases} \frac{q_i}{1+n} & (i = 1, \dots, m); \\ \frac{1}{1+n} & (i = m+1, \dots, m+n). \end{cases} \quad (1.8.11)$$

Тогда средний вектор рабочего шага в пространстве $\{X\}$ равен

$$M[\Delta X] = \sum_{i=1}^m p_i \Delta X^{(i)} = \left\{ [(q_{1+m/2} - q_1) + \right. \\ \left. + (q_{2+m/2} - q_2) + \dots + (q_m - q_{m/2})] \frac{1}{1+n}, \right.$$

$$\begin{aligned}
& [(q_{1+m/2} - q_1) + \dots + (q_{m/4+m/2} - q_{m/2}) - \dots \\
& \dots - (q_{m/4+1+m/2} - q_{m/4+1}) - \dots - (q_m - q_{m/2})] \frac{1}{1+n}, \dots \\
& \dots, [(q_{1+m/2} - q_1) - (q_{2+m/2} - q_2) + (q_{3+m/2} - q_3) - \\
& - (q_{4+m/2} - q_4) + \dots - (q_m - q_{m/2})] \frac{1}{1+n} \} .
\end{aligned}
\tag{1.8.12}$$

Среднее изменение показателя качества равно

$$\begin{aligned}
M[\Delta Q] = & - \frac{1}{1+n} [(q_{1+m/2} - q_1) (\alpha_1 + \alpha_2 + \dots + \alpha_n) + \\
& + (q_{2+m/2} - q_2) (\alpha_1 + \alpha_3 + \dots + \alpha_{n-1} - \alpha_n) + \\
& + (q_{3+m/2} - q_3) (\alpha_1 + \dots + \alpha_{n-2} - \alpha_{n-1} + \alpha_n) + \\
& + (q_{4+m/2} - q_4) (\alpha_1 + \dots + \alpha_{n-2} - \alpha_{n-1} - \alpha_n) + \dots \\
& \dots + (q_{m-1} - q_{m/2-1}) (\alpha_1 - \dots - \alpha_{n-1} + \alpha_n) + \\
& + (q_m - q_{m/2}) (\alpha_1 - \dots - \alpha_{n-1} - \alpha_n)].
\end{aligned}
\tag{1.8.13}$$

Для двумерного случая ($n=2$) из формул (1.8.11) и (1.8.7) имеем:

$$\begin{aligned}
p_1 &= \frac{1}{3} (1-s_1) (1-s_2); \quad p_2 = \frac{1}{3} (1-s_1) s_2; \\
p_3 &= \frac{1}{3} s_1 s_2; \quad p_4 = \frac{1}{3} s_1 (1-s_2); \\
p_5 &= \frac{1}{3}; \quad p_6 = \frac{1}{3};
\end{aligned}
\tag{1.8.14}$$

$$\begin{aligned}
p_{\text{ср}} &= \frac{1}{3} [s'_1 (1-s_1) (1-s_2) + s'_2 (1-s_1) s_2 + \\
& + s'_3 s_1 s_2 + s'_4 s_1 (1-s_2)];
\end{aligned}
\tag{1.8.15}$$

$$M[\Delta Q] = -\frac{1}{3} [(s_1 + s_2 - 1)(\alpha_1 + \alpha_2) + (s_1 - s_2)(\alpha_1 - \alpha_2)], \quad (1.8.16)$$

где

$$\begin{aligned} s'_1 &= \frac{1}{2} \left[1 + \Phi \left(\frac{\alpha_1 + \alpha_2}{2\sigma} \right) \right]; \\ s'_2 &= \frac{1}{2} \left[1 + \Phi \left(\frac{\alpha_1 - \alpha_2}{2\sigma} \right) \right]; \\ s'_3 &= \frac{1}{2} \left[1 - \Phi \left(\frac{\alpha_1 + \alpha_2}{2\sigma} \right) \right]; \\ s'_4 &= \frac{1}{2} \left[1 - \Phi \left(\frac{\alpha_1 - \alpha_2}{2\sigma} \right) \right]; \end{aligned} \quad (1.8.17)$$

s_1, s_2 определяются формулой (1.8.8).

Далее рассмотрим случай, когда отсутствует помеха ($\sigma=0$). Предположим, что направление вектора градиента функции $Q(\mathbf{X})$ удовлетворяет условию

$$\begin{aligned} \alpha_1 > \alpha_2 > \dots > \alpha_n > 0; \\ \alpha_1 > \alpha_2 + \dots + \alpha_n. \end{aligned} \quad (1.8.18)$$

Тогда из формул (1.8.17) и (1.8.8) следует:

$$\begin{aligned} s'_1 = s'_2 = \dots = s'_{m/2} = 1; \quad s'_{m/2+1} = \dots = s'_m = 0; \\ s_1 = s_2 = \dots = s_n = 1; \end{aligned} \quad (1.8.19)$$

Поэтому по формуле (1.8.7) получаем:

$$\begin{aligned} q_1 = q_2 = \dots = q_{m/2} = 0; \quad q_{m/2+1} = 1; \\ q_{m/2+2} = \dots = q_m = 0, \end{aligned} \quad (1.8.20)$$

а по формулам (1.8.11) имеем:

$$p_i = \begin{cases} \frac{1}{1+n} & \text{при } i = \frac{m}{2} + 1, m+1, \dots, m+n; \\ 0 & \text{при } i = 1, 2, \dots, \frac{m}{2}, \frac{m}{2} + 2, \dots, m. \end{cases} \quad (1.8.21)$$

Средний штраф рабочих шагов и среднее изменение функции качества равны

$$p_{\text{ср. р}} = 0; M[\Delta Q] = - \frac{1}{1+n} \sum_{i=1}^n \alpha_i. \quad (1.8.22)$$

Сравнение формул (1.8.22) и (1.7.38) показывает, что при отсутствии помехи алгоритм параллельного градиента по быстрдействию в $2/(1/n+1)$ раза превосходит алгоритм последовательного градиента.

Таким образом, в пределе при $n \rightarrow \infty$ это отношение стремится к 2.

§ 1.9. АЛГОРИТМ НАИСКОРЕЙШЕГО СПУСКА

Этот алгоритм, так же как и алгоритм параллельного градиента, представляет собой поиск по вершинам гиперкуба. Метод наискорейшего спуска отличается от метода параллельного градиента тем, что после удачного рабочего шага не происходит перехода к пробным шагам, а продолжается спуск по направлению удачного рабочего шага до первого неудачного шага, после чего снова совершаются пробные шаги [3]. Схема метода наискорейшего спуска показана на рис. 1.9.1. Аналогично методу градиента метод наискорейшего спуска может быть представлен в виде двух автоматов: автомата пробных шагов $A^{(n)}$ и автомата рабочих шагов $A^{(p)}$. Автомат пробных шагов имеет $n+1$ состояний: n состояний — рабочие шаги, одно фиктивное состояние — автомат пробных шагов.

Пусть входами автомата пробных шагов $A^{(n)}$ и автомата рабочих шагов $A^{(p)}$ являются числа $c=0$ или $c=1$ и векторы $C_j = (c_1^{(j)}, \dots, c_n^{(j)})$, определяемые формулами (1.8.2). Тогда автомат, соответствующий алгоритму наискорейшего спуска, представляет собой автомат с ограничениями на входе. Переходы из одного состояния в другое для автомата пробных шагов $A^{(n)}$ и автомата рабочих шагов $A^{(p)}$ показаны на рис. 1.9.2. По этому рисунку получаем следующие переходные матрицы:

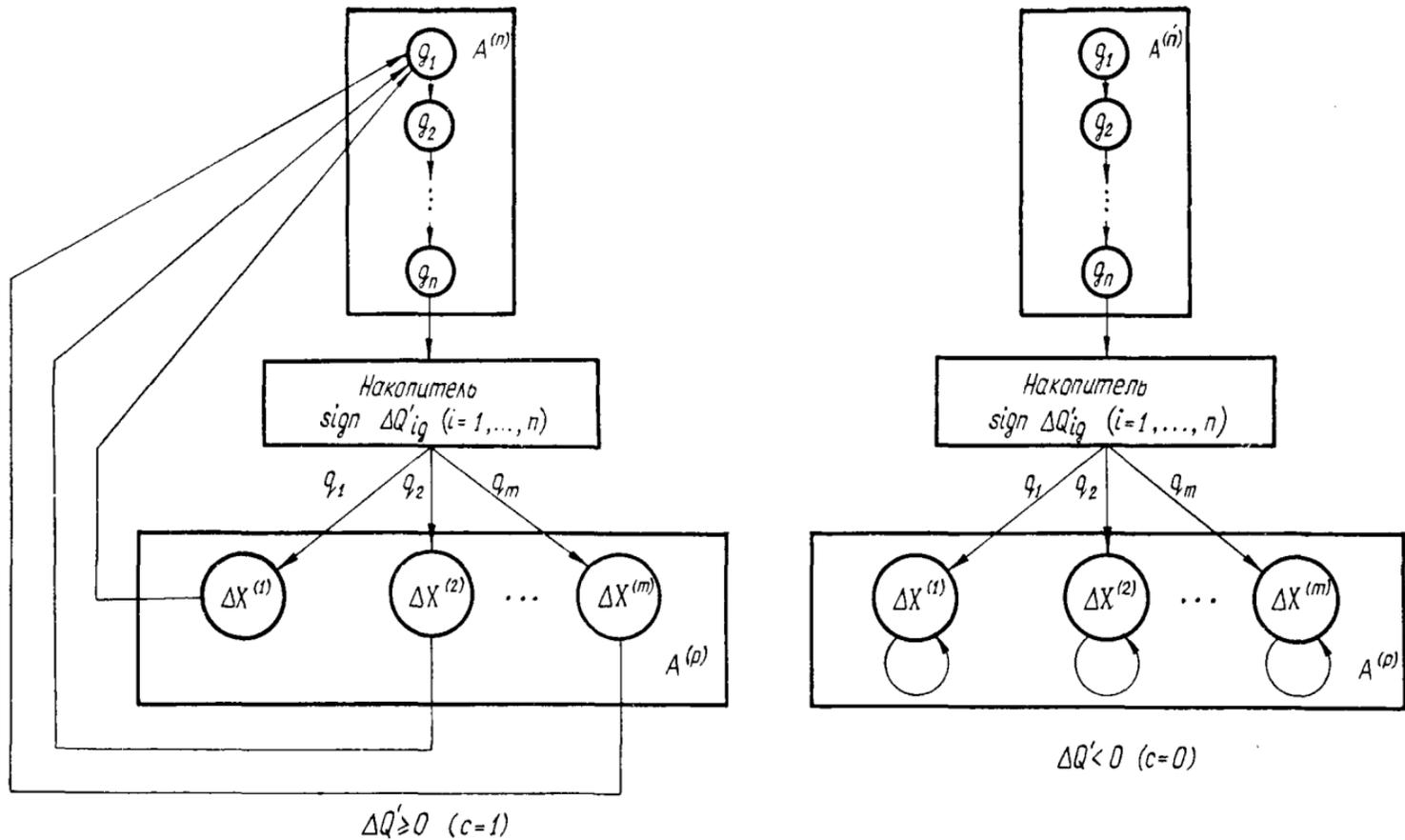


Рис. 1.9.1. Схема оптимизации методом наискорейшего спуска.

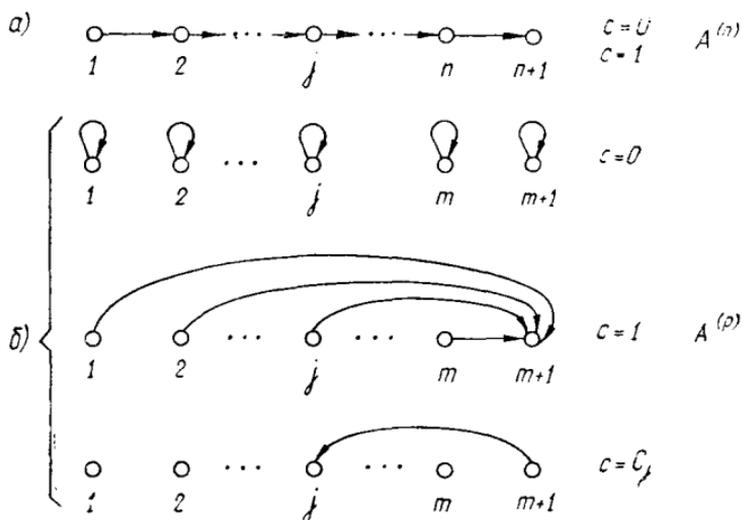


Рис. 1.9.2. Графы переходов автомата пробных шагов (а) и автомата рабочих шагов (б).

1) для автомата пробных шагов

$$A_0^{(n)} = A_1^{(n)} = \underbrace{\left\| \begin{array}{ccccccc} 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & \dots & 1 & 0 \\ 1 & 0 & 0 & 0 & \dots & 0 & 0 \end{array} \right\|}_{n+1}}^{n+1} \quad (1.9.1)$$

2) для автомата рабочих шагов

$$A_{0j}^{(p)} = \left\| \begin{array}{cccccccc} 1 & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\ \cdot & \cdot \\ 0 & 0 & 0 & \dots & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & 1 & 0 & \dots & 0 & 0 \end{array} \right\|_{m+1}$$

$\underbrace{\hspace{10em}}_j \quad \underbrace{\hspace{10em}}_{m+1-j}$

$$A_{1j}^{(p)} = \left\{ \begin{array}{cccc} 0 \dots 0 & 0 & 0 \dots 1 \\ 0 \dots 0 & 0 & 0 \dots 1 \\ \cdot & \cdot & \cdot & \cdot \\ 0 \dots 0 & 0 & 0 \dots 1 \\ 0 \dots 1 & 0 & 0 \dots 0 \end{array} \right\}_{m+1} \quad (j=1, 2, \dots, 2^n). \quad (1.9.2)$$

$\underbrace{\hspace{10em}}_j \quad \underbrace{\hspace{10em}}_{m+1-j}$

Объединяя матрицы при $\Delta Q' < 0$ автомата пробных шагов $A^{(n)}$ и автомата рабочих шагов $A^{(p)}$, имеем переходную матрицу

$$A_{0j} = \left\{ \begin{array}{cccc} 1 & 0 & 0 \dots 0 & 0 \dots 0 & 0 & 0 & 0 & 0 & 0 & 0 \dots 0 \\ 0 & 1 & 0 \dots 0 & 0 \dots 0 & 0 & 0 & 0 & 0 & 0 & 0 \dots 0 \\ \cdot & \cdot \\ 0 & 0 & 0 \dots 0 & 0 \dots 1 & 0 & 0 & 0 & 0 & 0 \dots 0 \\ 0 & 0 & 0 \dots 0 & 0 \dots 0 & 0 & 1 & 0 & 0 & 0 \dots 0 \\ 0 & 0 & 0 \dots 0 & 0 \dots 0 & 0 & 0 & 1 & 0 & 0 \dots 0 \\ \cdot & \cdot \\ 0 & 0 & 0 \dots 0 & 0 \dots 0 & 0 & 0 & 0 & 0 & 0 \dots 1 \\ 0 & 0 & 0 \dots 1 & 0 \dots 0 & 0 & 0 & 0 & 0 & 0 \dots 0 \end{array} \right\} \begin{array}{l} m \\ n \end{array} \quad (1.9.3)$$

$\underbrace{\hspace{10em}}_j \quad \underbrace{\hspace{10em}}_{m-j} \quad \underbrace{\hspace{10em}}_n$

Объединяя матрицы $A_{1^{\Pi}}$ и $A_{1j}^{(p)}$, получаем переходную матрицу автомата при штрафе

$$A_{1j} = \left\{ \begin{array}{cccc} 0 \dots 0 & 0 & 0 \dots 1 & 0 & 0 & 0 & 0 \dots 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 \dots 0 & 0 & 0 \dots 1 & 0 & 0 & 0 & 0 \dots 0 \\ 0 \dots 0 & 0 & 0 \dots 0 & 1 & 0 & 0 & 0 \dots 0 \\ 0 \dots 0 & 0 & 0 \dots 0 & 0 & 1 & 0 & 0 \dots 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 \dots 0 & 0 & 0 \dots 0 & 0 & 0 & 0 & 0 \dots 1 \\ 0 \dots 1 & 0 & 0 \dots 0 & 0 & 0 & 0 & 0 \dots 0 \end{array} \right\} \begin{array}{l} m \\ n \end{array} \quad (1.9.4)$$

$\underbrace{\hspace{10em}}_j \quad \underbrace{\hspace{10em}}_{m-j} \quad \underbrace{\hspace{10em}}_n$

Функционирование автомата в случайной среде описывается цепью Маркова с матрицей переходных вероятностей

$$A(S) = \left(\begin{array}{cccccccc} 1-s'_1 & 0 & 0 & \dots & 0 & s'_1 & 0 & 0 & \dots & 0 \\ 0 & 1-s'_2 & 0 & \dots & 0 & s'_2 & 0 & 0 & \dots & 0 \\ \dots & \dots \\ 0 & 0 & 0 & \dots & 1-s'_m & s'_m & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 1 & 0 & \dots & 0 \\ \dots & \dots \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 1 \\ \underbrace{q_1 \quad q_2 \quad q_3 \quad \dots \quad q_m}_m & \underbrace{0 \quad 0 \quad 0 \quad \dots \quad 0}_n & & & & & & & & \end{array} \right) \left. \begin{array}{l} \\ \\ \\ \\ \\ \\ \\ \end{array} \right\} \begin{array}{l} m \\ \\ \\ n \end{array} \quad (1.9.5)$$

где s'_j ($j=1, \dots, m$) — вероятность штрафа j -го рабочего шага; q_j определяется формулой (1.8.8). Предельные вероятности этой цепи находятся как решение системы алгебраических уравнений

$$p_i = (1 - s'_i) p_i + q_i p_{m+n} \quad (i=1, \dots, m);$$

$$p_{m+1} = \sum_{i=1}^m s'_i p_i; \quad (1.9.6)$$

$$p_{m+2} = p_{m+1}; \dots; p_{m+n} = p_{m+n-1}.$$

Решением этой системы является

$$p_i = \begin{cases} \frac{q_i}{s'_i} \left(\sum_{j=1}^m \frac{q_j}{s'_j} + n \right)^{-1} & \text{при } i=1, \dots, m; \\ \left(\sum_{j=1}^m \frac{q_j}{s'_j} + n \right)^{-1} & \text{при } i=m+1, \dots, m+n. \end{cases} \quad (1.9.7)$$

Средний штраф рабочих шагов равен

$$\rho_{\text{ср. п}} = \sum_{i=1}^m q_i \left(\sum_{j=1}^m \frac{q_j}{s'_j} + n \right)^{-1}. \quad (1.9.8)$$

Среднее смещение по направлению градиента определяем по формуле (1.8.11), заменяя в этой формуле q_i

($i=1, \dots, m+n$) на q_i/s'_i и $(1+n)$ на $\sum_{i=1}^m q_i/s'_i + n$:

$$\begin{aligned} M[\Delta Q] = & - \left(\sum_{i=1}^m \frac{q_i}{s'_i} + n \right)^{-1} \left[\left(\frac{q_{1+m/2}}{s'_{1+m/2}} - \frac{q_1}{s'_1} \right) \times \right. \\ & \times (\alpha_1 + \alpha_2 + \dots + \alpha_n) + \left(\frac{q_{2+m/2}}{s'_{2+m/2}} - \frac{q_2}{s'_2} \right) (\alpha_1 + \\ & + \alpha_2 + \dots + \alpha_{n-1} - \alpha_n) + \left(\frac{q_{3+m/2}}{s'_{3+m/2}} - \frac{q_3}{s'_3} \right) \times \\ & \times (\alpha_1 \alpha_2 + \dots + \alpha_{n-2} - \alpha_{n-1} + \alpha_n) + \\ & + \left(\frac{q_{4+m/2}}{s'_{4+m/2}} - \frac{q_4}{s'_4} \right) (\alpha_1 + \alpha_2 + \dots + \alpha_{n-2} - \alpha_{n-1} - \\ & - \alpha_n) + \dots + \left(\frac{q_{m-1}}{s'_{m-1}} - \frac{q_{m/2-1}}{s'_{m/2-1}} \right) (\alpha_1 - \alpha_2 - \dots \\ & \dots - \alpha_{n-1} + \alpha_n) + \left(\frac{q_m}{s'_m} - \frac{q_{m/2}}{s'_{m/2}} \right) \times \\ & \left. \times (\alpha_1 - \alpha_2 - \dots - \alpha_{n-1} - \alpha_n) \right]. \quad (1.9.9) \end{aligned}$$

Для двумерного случая ($n=2$) по этой формуле имеем

$$M[\Delta Q] = \frac{\left(\frac{q_1}{s'_1} - \frac{q_3}{s'_3}\right)(\alpha_1 + \alpha_2) + \left(\frac{q_2}{s'_2} - \frac{q_4}{s'_4}\right)(\alpha_1 - \alpha_2)}{\sum_{i=1}^4 \frac{q_i}{s'_i} + 2} \quad (1.9.10)$$

Далее рассмотрим случай, когда отсутствует помеха ($\sigma=0$). Тогда имеют место формулы (1.8.18) и (1.8.19). Следовательно, матрица переходных вероятностей цепи Маркова (1.9.5) приобретает вид

$$A(S) = \begin{pmatrix} 0 \dots 0 & 0 \dots 0 & 1 & 0 & 0 & 0 & \dots & 0 \\ \dots & \dots \\ 0 \dots 0 & 0 \dots 0 & 1 & 0 & 0 & 0 & \dots & 0 \\ 0 \dots 1 & 0 \dots 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ \dots & \dots \\ 0 \dots 0 & 0 \dots 1 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 \dots 0 & 0 \dots 0 & 0 & 1 & 0 & 0 & \dots & 0 \\ 0 \dots 0 & 0 \dots 0 & 0 & 0 & 1 & 0 & \dots & 0 \\ \dots & \dots \\ 0 \dots 0 & 0 \dots 0 & 0 & 0 & 0 & 0 & \dots & 1 \\ 0 \dots 1 & 0 \dots 0 & 0 & 0 & 0 & 0 & \dots & 0 \end{pmatrix} \quad (1.9.11)$$

$\underbrace{\hspace{1.5cm}}_{\frac{m}{2}} \quad \underbrace{\hspace{1.5cm}}_{\frac{m}{2}} \quad \underbrace{\hspace{1.5cm}}_n$

Из этой матрицы видно, что у цепи Маркова состояния от 1 до $m/2$ и от $m+1$ до $m+n$ — невозвратные, а от $m/2+1$ до m — поглощающие. Поэтому она является неэргодической и ее предельные вероятности зависят от ее начальных вероятностей $p_i^{(0)}$, $i=1, \dots, m+n$. Вероятности перехода за N шагов и предельные вероятности определяются при помощи N -й степени матрицы (1.9.11). Можно показать, что эта степень равна следующим матрицам:

где нулями обозначены блоки, элементы которых равны нулю.

Предельные вероятности цепи Маркова равны ее полным вероятностям на N -м шаге при $N > n$:

$$p_i = p_i^N (N > n) = \begin{cases} 0 & \text{при } i = 1, \dots, \frac{m}{2}; \\ \sum_{j=1}^{m/2+1} p_j^{(0)} + \sum_{j=m+1}^{m+n} p_j^{(0)} & \text{при } i = \frac{m}{2} + 1; \\ p_i^{(0)} & \text{при } i = \frac{m}{2} + 2, \dots, m; \\ 0 & \text{при } i = m + 1, \dots, m + n. \end{cases} \quad (1.9.14)$$

Средний штраф рабочих шагов равен

$$p_{\text{ср. п}} = \sum_{i=1}^m s_i p_i = 0. \quad (1.9.15)$$

Среднее смещение в направлении градиента определяется по формуле (1.9.9) с использованием вероятностей (1.9.14):

$$M[\Delta Q] = - \left[\left(\sum_{j=1}^{m/2+1} p_j^{(0)} + \sum_{j=m+1}^{m+n} p_j^{(0)} \right) (\alpha_1 + \alpha_2 + \dots \right. \\ \dots + \alpha_n) + p_{2+m/2}^{(0)} (\alpha_1 + \alpha_2 + \dots \\ \dots + \alpha_{n-1} - \alpha_n) + p_{3+m/2}^{(0)} (\alpha_1 + \alpha_2 + \dots + \alpha_{n-2} - \\ - \alpha_{n-1} + \alpha_n) + p_{4+m/2}^{(0)} (\alpha_1 + \alpha_2 + \dots + \alpha_{n-2} - \alpha_{n-1} - \\ - \alpha_n) + \dots + p_{m-1}^{(0)} (\alpha_1 - \alpha_2 - \dots - \alpha_{n-1} + \alpha_n) + \\ \left. + p_m^{(0)} (\alpha_1 - \alpha_2 - \dots - \alpha_{n-1} - \alpha_n) \right]. \quad (1.9.16)$$

Для двумерного случая ($n=2$) по формулам (1.9.14) и (1.9.16) получаем:

$$p_1 = p_2 = 0; \quad p_3 = p_1^{(0)} + p_2^{(0)} + p_3^{(0)} + p_5^{(0)} + p_6^{(0)};$$

$$p_4 = p_4^{(0)}; \quad p_5 = p_6 = 0; \quad (1.9.17)$$

$$M[\Delta Q] = -[(p_1^{(0)} + p_2^{(0)} + p_3^{(0)} + p_5^{(0)} + p_6^{(0)}) (\alpha_1 + \alpha_2) +$$

$$+ p_4^{(0)} (\alpha_1 - \alpha_2)]. \quad (1.9.18)$$

§ 1.10. СОПОСТАВЛЕНИЕ НЕПРЕРЫВНОГО И АВТОМАТНОГО АЛГОРИТМОВ СЛУЧАЙНОГО ПОИСКА

Непрерывный вариант случайного поиска характеризуется тем, что шаг в n -мерном пространстве параметров определяется как радиус-вектор, конец которого равномерно распределен по поверхности n -мерной гиперсферы, а начало лежит в ее центре.

В многоканальных оптимизаторах [11, 43] используется дискретный, т. е. автоматный, вариант случайного поиска, при котором конец радиуса-вектора распределяется равномерно только по вершинам n -мерного гиперкуба, а начало вектора по-прежнему лежит в центре n -мерной гиперсферы, описывающей гиперкуб.

Рассмотрим, как введение дискретности повлияет на эффективность и статистические свойства поиска [44].

Под эффективностью поиска будем понимать его быстроедействие [3], измеряемое средней величиной благоприятного приращения функции качества объекта, приходящейся на один шаг поиска. Быстроедействие поиска характеризует скорость смещения к экстремуму функции в направлении ее градиента. Быстроедействие поиска будем исследовать, используя линейную модель функции качества

$$Q(\mathbf{X}) = Q(0) + [\text{grad } Q(\mathbf{X}), \mathbf{X}], \quad (1.10.1)$$

где \mathbf{X} — n -мерный вектор параметров;
 $[\text{grad } Q(\mathbf{X}), \mathbf{X}]$ — скалярное произведение вектора градиента на вектор \mathbf{X} .

Интересующее нас приращение функции качества за один шаг легко вычисляется на основании (1.10.1):

$$\Delta Q_i = Q(\mathbf{X}_i + \Delta \mathbf{X}_i) - Q(\mathbf{X}_i) = |\text{grad } Q(\mathbf{X})| |\Delta \mathbf{X}_i| \cos \varphi_i. \quad (1.10.2)$$

Здесь $|\text{grad } Q(\mathbf{X})|$ и $|\Delta \mathbf{X}|$ — модули $\text{grad } Q(\mathbf{X}) = (\alpha_1, \dots, \alpha_n)$ и вектора шага $\Delta \mathbf{X}_i$; φ_i — угол между направлением градиента и направлением шага.

Из формулы (1.10.2) следует, что эффективность случайного поиска определяется «удачностью» выбора направления случайного шага (при фиксированных $|\text{grad } Q(\mathbf{X})|$ и $|\Delta \mathbf{X}_i|$).

Так, при поиске $\max Q(\mathbf{X})$ шаги будут удачными, если $\cos \varphi_i > 0$. При неудачном шаге происходит автоматический возврат в предыдущую удачную точку и снова определяется случайное направление шага. При удачном шаге также делается случайный шаг, что отличает процедуру нелинейного алгоритма. В линейном алгоритме за неудачным шагом следует повторение шага в том же направлении.

Таким образом, эффективность алгоритмов обоого вида существенно зависит от результативности одного случайного шага. Поэтому изучим свойства автоматного случайного поиска на одном случайном шаге.

При расчете эффективности случайного шага, для того чтобы получить оценку в сопоставимых единицах, необходимо, как принято выше, полагать, что

$$|\text{grad } Q(\mathbf{X})| = 1; \quad |\Delta \mathbf{X}| = 1. \quad (1.10.3)$$

Тогда координаты вектора шага $\Delta \mathbf{X}_i = (\Delta x_{i1}, \dots, \Delta x_{in})$ имеют вид

$$\Delta x_{ij} = \pm \frac{1}{\sqrt{n}} \quad (j=1, 2, \dots, n). \quad (1.10.4)$$

Другими словами, вектор $\Delta \mathbf{X}_i$ есть вершина n -мерного гиперкуба и случайный механизм выбора шага заключается в выборе любой вершины этого гиперкуба равновероятно, т. е. с вероятностью $1/2^n$.

Так же без ограничения общности можно положить, что производится поиск $\max Q(\mathbf{X})$ и задача заключается

в том, чтобы найти условное математическое ожидание смещения к цели

$$M(\Delta Q) = \frac{1}{2^m} \sum_{i=1}^m \Delta Q_i, \quad (1.10.5)$$

где m — число неблагоприятных направлений шага ΔX .

В варианте случайного поиска на гиперсфере $M[\Delta Q]$ не зависит от выбора направления $\text{grad } Q$. Но при дискретном варианте ситуация резко меняется. Например, при выборе координат градиента для случая $n \geq 2$ в виде

$$\alpha_i = 1; \alpha_j = 0 \quad (1.10.6)$$

$$(i \neq j = 1, 2, \dots, n)$$

имеем 2^{n-1} значений $\Delta Q_i = 1/\sqrt{n}$, 2^{n-1} значений $\Delta Q_i = -1/\sqrt{n}$ и $M[\Delta Q] = 1/(2\sqrt{n}) = m^*$. При выборе координат градиента в виде

$$\alpha_i = \alpha_j = \frac{1}{\sqrt{2}}; \alpha_k = 0 \quad (1.10.7)$$

$$(k = 1, 2, \dots, n; k \neq i \neq j)$$

имеем 2^{n-2} чисел $\Delta Q_i = 2/\sqrt{2n}$, 2^{n-2} чисел $\Delta Q_i = -2/\sqrt{2n}$, 2^{n-1} чисел $\Delta Q_i = 0$ и $M[\Delta Q] = 1/(2\sqrt{2n}) = m_*$.

Докажем, что при $n \geq 2$

$$m_* \leq M[\Delta Q] \leq m^*. \quad (1.10.8)$$

Оценка сверху получается следующим образом. Представим $M[\Delta Q]$ как среднее арифметическое 2^n чисел:

$$M[\Delta Q] = \frac{1}{2^n} \sum_{\Delta Q_i \geq 0} \Delta Q_i = \frac{1}{2} 2^{-n} \sum_{i=1}^{2^n} |\Delta Q_i|. \quad (1.10.9)$$

Последнее равенство справедливо ввиду того, что для каждого $\Delta Q_i > 0$ в силу симметричности вершины X_i

гиперкуба и линейности объекта имеется равное по модулю, но обратное по знаку приращение $\Delta Q'_i$. Используя известное неравенство, имеем

$$M[\Delta Q] = \frac{1}{2} 2^{-n} \sum_{i=1}^{2^n} |\Delta Q_i| \leq \frac{1}{2} 2^{-n} \sqrt{\sum_{i=1}^{2^n} \Delta Q_i^2}, \quad (1.10.10)$$

причем равенство достигается только тогда, когда все $|\Delta Q_i|$, кроме одного, равны нулю. Легко проверить, что такая ситуация возможна лишь при выборе градиента $Q(\mathbf{X})$ в виде (1.10.6). Для нахождения оценки снизу необходимо рассмотрение $M[\Delta Q]$ как среднего арифметического m положительных чисел ΔQ_i :

$$M[\Delta Q] = \frac{1}{2^n} \sum_{\Delta Q_i > 0} \Delta Q_i = m 2^{-n} \frac{1}{m} \sum_{i=1}^m \Delta Q_i, \quad (1.10.11)$$

для которого известно другое неравенство:

$$M[\Delta Q] \geq m 2^{-n} \sqrt{\prod_{i=1}^m \Delta Q_i}. \quad (1.10.12)$$

Здесь равенство достигается при условии, что все ΔQ_i равны друг другу. Несложно показать, что равенство возможно лишь при выборе градиента $Q(\mathbf{X})$ в виде (1.10.7). Таким образом, неравенства (1.10.8) доказаны. В работе [3] потери на поиск k_{Π} определены следующим образом:

$$k_{\Pi} = \frac{l \Delta Q_{\max}}{M[\Delta Q]}, \quad (1.10.13)$$

где l — число определений функции качества на один рабочий шаг.

При случайном поиске с пересчетом, который был рассмотрен выше, $l=1$, а при градиентном методе поиска $l=n+1$. С учетом (1.10.2) и (1.10.3) для случайного поиска имеем

$$\Delta Q_{\max} = |\text{grad } Q(\mathbf{X})| |\Delta \mathbf{X}_i| = 1 \text{ и } k_{\Pi} = \frac{1}{M[\Delta Q]}.$$

Теперь сравним потери на поиск при различных методах оптимизации. Как показано в работе [3], потери на поиск равны:

- а) для метода градиента $k_{\Pi} = n + 1$;
 б) для варианта случайного поиска с пересчетом при шаге, равномерно распределенном по гиперсфере,

$$k_{\Pi} = \frac{\pi(n-1)\Gamma\left(\frac{n-1}{2}\right)}{\Gamma\left(\frac{n}{2}\right)}, \quad (1.10.14)$$

где $\Gamma(n)$ — гамма-функция.

Для варианта случайного поиска на гиперкубе используем оценки (1.10.8).

На рис. 1.10.1 видно, что максимальные потери случайного поиска на гиперкубе меньше потерь при использовании градиентного метода для $n > 6$.

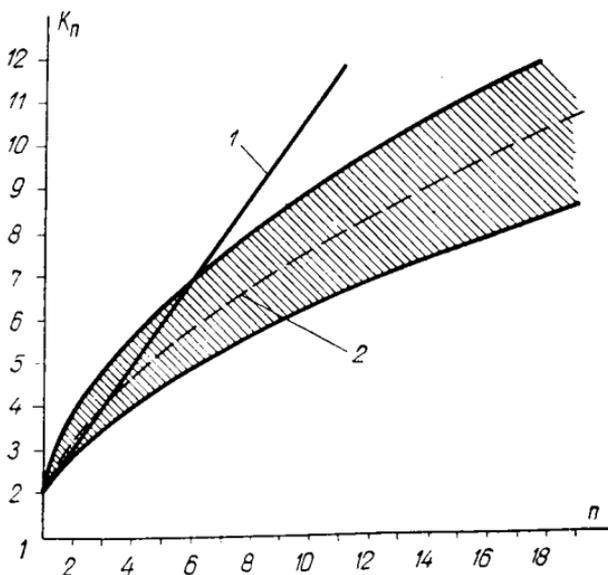


Рис. 1.10.1. Потери на поиск:

1 — метод градиента; 2 — непрерывный случайный поиск; зона потерь случайного поиска на гиперкубе заштрихована.

Рассмотрим некоторые вероятностные характеристики случайного поиска на гиперкубе. Отметим тот факт, что $M[\Delta Q]$ при равномерном распределении направлений градиента совпадает с математическим ожиданием приращения функции качества случайного поиска на гиперсфере. Это вытекает из того, что случайная величина φ_i из (1.10.2) будет иметь точно такое же распределение, как и при случайном поиске на гиперсфере. Однако если направление градиента зафиксировано, то распределение угла φ_i при случайном поиске на гиперкубе дискретно и зависит от выбора направлений градиента, что непосредственно видно из выражения для $\cos \varphi_i$:

$$\cos \varphi_i = \frac{\sum_{j=1}^n \alpha_j \Delta x_{ij}}{|\text{grad } Q(\mathbf{X})| |\Delta \mathbf{X}_i|} = \sum_{j=1}^n \alpha_j \Delta x_{ij} \quad (1.10.15)$$

$(i=1, 2, \dots, 2^n; j=1, 2, \dots, n).$

Вид закона распределения угла φ зависит от направления градиента $Q(\mathbf{X})$. Так, при выборе координат градиента в виде (1.10.6) имеем:

$$\begin{aligned} \cos \varphi_1 &= \frac{1}{\sqrt{n}}; P(\varphi_1) = \frac{1}{2}; \\ \cos \varphi_2 &= -\frac{1}{\sqrt{n}}; P(\varphi_2) = \frac{1}{2}. \end{aligned} \quad (1.10.16)$$

При выборе координат градиента в виде (1.10.7) имеем 2^{n-2} чисел $\cos \varphi_1 = 2/\sqrt{2n}$, 2^{n-1} чисел $\cos \varphi_2 = 0$ и 2^{n-2} чисел $\cos \varphi_3 = -2/\sqrt{2n}$, т. е.

$$\begin{aligned} \cos \varphi_1 &= \frac{2}{\sqrt{2n}}; P(\varphi_1) = \frac{1}{4}; \\ \cos \varphi_2 &= 0; P(\varphi_2) = \frac{1}{2}; \\ \cos \varphi_3 &= -\frac{2}{\sqrt{2n}}; P(\varphi_3) = \frac{1}{4}. \end{aligned} \quad (1.10.17)$$

Если выбрать координаты градиента в виде

$$\alpha_1 = \alpha_2 = \dots = \alpha_n = \frac{1}{\sqrt{n}}, \quad (1.10.18)$$

то будет иметь место следующий закон распределения угла φ :

$$\cos \varphi_m = \frac{(n-m) - m}{n} = 1 - \frac{2m}{n}; \quad P(\varphi_m) = \frac{C_n^m}{2^n}, \quad (1.10.19)$$

где $m = 0, 1, 2, \dots, n$; $C_n^m = \frac{n!}{m!(n-m)!}$.

В последнем случае число разных по величине углов φ_m между векторами шагов и вектором градиента равно $n+1$, т. е. увеличивается с ростом n . Попутно можно отметить, что при оптимизации n -мерной функции качества ситуация, при которой выполняется условие (1.10.6), может иметь место $2n$ раз, ситуация, когда выполняется

условие (1.10.7), будет иметь место $4 \sum_{r=1}^{n-1} r = 2n(n-1)$

раз и ситуация, когда выполняется условие (1.10.18), может встретиться 2^n раз, т. е. при больших n условие (1.10.18) является наиболее возможным.

Случайный поиск на гиперкубе, при котором на каждом шаге производится случайный равновероятный выбор одного из множества дискретных направлений поиска, можно считать своего рода аппроксимацией случайного поиска на гиперсфере. Качество аппроксимации можно оценивать, например, по величине среднеквадратического отклонения функции распределения $F^*(\varphi)$ случайной величины угла φ при случайном поиске на гиперкубе от функции распределения $F(\varphi)$ при случайном поиске на гиперсфере, т. е.

$$\delta = \sqrt{\sum_m P(\varphi_m) [F^*(\varphi_m) - F(\varphi_m)]^2}. \quad (1.10.20)$$

Здесь

$$F^*(\varphi_m) = P(0 \leq \varphi < \varphi_m) = \sum_{k=1}^m P(\varphi_k)$$

матрицей полного факторного эксперимента типа 2^n и обладает свойством ортогональности [45]:

$$\sum_{i=1}^{2^n} \Delta x_{ij} \Delta x_{ik} = \begin{cases} 0 & \text{при } j \neq k; \\ \frac{2^n}{n} & \text{при } j = k. \end{cases} \quad (1.10.26)$$

Следовательно,

$$D[\Delta Q] = \frac{1}{2^n} \sum_{j=1}^n \alpha_j^2 - M^2[\Delta Q] \quad (1.10.27)$$

и в силу условия (1.10.3)

$$D[\Delta Q] = \frac{1}{2n} - M^2[\Delta Q]. \quad (1.10.28)$$

Отсюда при $n \geq 2$ из неравенств (1.10.8) следуют оценки (1.10.22).

Из рис. 1.10.3 видно, что в принятых терминах потери на поиск неравенства (1.10.22) можно интерпретировать так: минимальным потерям соответствует минимальная дисперсия, а максимальным потерям — максимальная дисперсия.

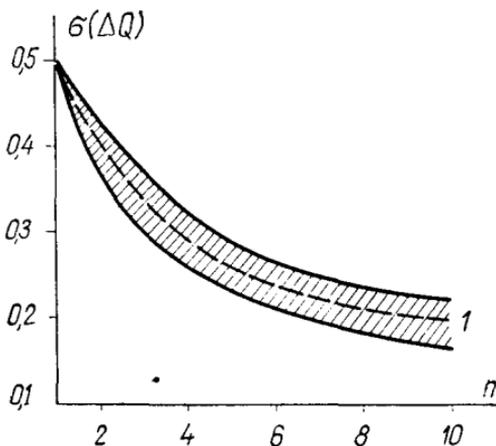


Рис. 1.10.3. График зависимости от n среднеквадратических отклонений величины относительного смещения к цели. Кривая 1 — метод непрерывного случайного поиска.

§ 1. 11. АВТОМАТНАЯ ОПТИМИЗАЦИЯ ПРИ НАЛИЧИИ ОГРАНИЧЕНИЙ

Выше рассматривались процессы оптимизации без ограничения на область поиска. Однако в реальных случаях на область поиска накладываются различного рода ограничения. Чаще всего в виде неравенств:

$$S : h_i(\mathbf{X}) \geq 0 \\ (i=1, \dots, k).$$

Учет ограничений может быть сделан двояко. Во-первых, методом штрафных функций, т. е. путем образования новой функции качества вида

$$\tilde{Q}(\mathbf{X}) = \begin{cases} Q(\mathbf{X}) & \text{при } \mathbf{X} \in S; \\ Q(\mathbf{X}) + q(\mathbf{X}) & \text{при } \mathbf{X} \notin S, \end{cases}$$

где $q(\mathbf{X})$ — штраф за нарушение ограничений, который строится следующим образом:

$$q(\mathbf{X}) = 0 \quad \text{при } \mathbf{X} \in S; \\ q(\mathbf{X}) > 0 \quad \text{при } \mathbf{X} \notin S;$$

например,

$$q(\mathbf{X}) = -\delta \min \{0, h_1(\mathbf{X}), \dots, h_k(\mathbf{X})\},$$

где δ — достаточно большой коэффициент штрафа.

Минимизируя функцию $\tilde{Q}(\mathbf{X})$ в открытой области, т. е. решая задачу

$$\tilde{Q}(\mathbf{X}) \rightarrow \min, \\ \mathbf{X} \in E^n$$

получаем \mathbf{X}^* , который при достаточно большом δ совпадает с решением исходной задачи (0.1.5), т. е.

$$\tilde{\mathbf{X}}^* = \mathbf{X}^*.$$

Такой подход, как это очевидно, не требует коррекции алгоритма, но допускает заход в область $\mathbf{X} \notin S$.

Другой подход к решению оптимизационных задач с ограничениями связан с организацией реакции на заход в недопустимую область. Например, нарушение ограничений отождествляется с неудачным шагом:

$$(\mathbf{X} \notin S) \equiv (\Delta Q \geq 0).$$

В этом случае использование нелинейного алгоритма (0.2.4), записанного в виде

$$\Delta \mathbf{X}_N = \begin{cases} a\Xi & \text{при } (\Delta Q_{N-1} < 0) \wedge (\mathbf{X}_{N-1} \in S); \\ -\Delta \mathbf{X}_{N-1} & \text{при } (\Delta Q_{N-1} \geq 0) \vee (\mathbf{X}_{N-1} \notin S), \end{cases}$$

гарантирует решение поставленной задачи (здесь знак \wedge — логическое «И»; \vee — логическое «ИЛИ»).

Линейный алгоритм (0.2.2) запишется в виде

$$\Delta \mathbf{X}_N = \begin{cases} \Delta \mathbf{X}_{N-1} & \text{при } (\Delta Q_{N-1} < 0) \wedge (\mathbf{X}_{N-1} \in S); \\ a\Xi & \text{при } (\Delta Q_{N-1} \geq 0) \wedge (\mathbf{X}_{N-1} \in S); \\ -\Delta \mathbf{X}_{N-1} & \text{при } \mathbf{X}_{N-1} \notin S. \end{cases} \quad (1.11.1)$$

Здесь обратный шаг введен как реакция на нарушение ограничений.

Рассмотрим работу последнего алгоритма для случая, когда множество возможных направлений $\{\Delta \mathbf{X}\}$ образуется координатными направлениями (1.5.1), т. е. имеет место стохастический вариант алгоритма Гаусса—Зейделя (см. § 1.5):

$$\Delta \mathbf{X}^{(i)} = \begin{cases} a\mathbf{e}_i & \text{при } i \leq n; \\ -a\mathbf{e}_{i-n} & \text{при } n < i \leq 2n, \end{cases}$$

где \mathbf{e}_i — i -й орт. Здесь случайный вектор Ξ имеет дискретное распределение

$$p(i) = \frac{1}{2n} \quad (i=1, \dots, 2n).$$

Ограничимся для простоты случаем $n=2$, который рассмотрел И. Л. Антонов [17].

Схема рассмотренного им алгоритма такова. В начальный момент с равной вероятностью $\frac{1}{2}$ выбирается один

из регулируемых параметров x_1 или x_2 и с вероятностью $\frac{1}{2}$ по этому параметру делается шаг в ту или другую сторону. Если этот шаг удачен, т. е. если приращение показателя качества имеет желаемый знак, то система продолжает движение по той же координате и в том же направлении до тех пор, пока указанное приращение не сменит знак. После этого снова производится случайный выбор направления движения. Если первый шаг был неудачен, то процедура выбора направления движения повторяется сразу после этого шага.

Будем считать, что переход системы из точки в точку происходит мгновенно, но в каждой точке плоскости, куда система попадает после n -го шага, она задерживается на единицу времени (например, на 1 сек).

Определим поведение системы в окрестности точек, являющихся запрещенными, следующим образом. Система, попав в процессе поиска в точку, принадлежащую запрещенной области, остается там в течение 1 сек, после чего с вероятностью 1 возвращается в точку, исходную для последнего шага. Если при «отражении» знак приращения показателя качества будет таким, какой нужен, система будет продолжать двигаться в том же направлении до момента, пока не сменится знак приращения, после чего указанным образом производится выбор нового направления движения. Если при «отражении» приращение показателя качества имеет нежелательный знак, то выбор нового направления движения производится сразу.

В § 1.5 было показано, что такой процесс поиска посредством его автоматного представления сводится к некоторой цепи Маркова, т. е. представляет собой случайное блуждание по плоской решетке с шагом, равным единице. Легко заметить, что вероятности переходов системы из точки в точку будут зависеть только от того, в какой точке система находится в данный момент, т. е. процесс переходов также будет марковским. Точнее говоря, этот процесс будет описываться однородной марковской цепью со счетным множеством состояний.

Назовем переход системы из точки в точку между двумя последовательными остановками для выбора направления движения тактом и рассмотрим процесс по-

иска, принимая в качестве аргумента этого процесса число тактов.

Будем считать, что показатель качества системы можно представить в виде

$$Q(x_1, x_2) = f_1(x_1) + f_2(x_2),$$

где $f_1(x_1)$ и $f_2(x_2)$ — произвольные функции, имеющие единственный строгий минимум.

Будем считать, что минимум показателя качества $Q(x_1, x_2)$ находится в точке $(0, 0)$.

Назовем окрестностью минимума точки $(1, 1)$, $(1, -1)$, $(-1, 1)$, $(-1, -1)$ и найдем среднее число тактов, необходимое системе, чтобы из некоторой точки плоскости попасть впервые в одну из точек окрестности минимума.

Поскольку нас интересует среднее число тактов, необходимое системе для первого попадания в окрестность минимума, мы можем считать точки, принадлежащие этой окрестности, поглощающими, т. е. такими, что вероятность перехода из этих точек в самих себя равна 1, и, следовательно, можем для определения искомого среднего числа тактов воспользоваться следующей формулой [47]:

$$N = \sum_{q=1}^{\infty} P_{q-1}(X_i, S), \quad (1.11.2)$$

где X_i — исходное состояние системы;

S — множество всех возможных состояний системы, кроме состояний, принадлежащих окрестности минимума;

$P_q(X_i, S)$ — вероятность того, что, выходя из состояния X_i , система после q тактов окажется в одном из состояний множества S , не заходя при этом ни разу в точки, принадлежащие окрестности минимума;

$$P_q(X_i, S) = \sum_{X_j \in S} p_{ij}^{(q)}.$$

Для определения этих вероятностей разобьем множество на однородные подмножества.

Разбиением на однородные подмножества назовем такое разбиение множества S , когда для любых двух состояний X_j и $X_j \in S_k$ и для любого подмножества S_l имеет место равенство

$$\sum_{X_s \in S_l} p_{is} = \sum_{X_s \in S_l} p_{js} \quad (1.11.3)$$

и подмножества S_i попарно не пересекаются. По крайней мере одно такое разбиение всегда существует: каждое возможное состояние является однородным подмножеством.

Обозначим $\sum_{X_j \in S_l} p_{ij}$ для $X_i \in S_k$ через $p(S_k, S_l)$. Ис-

пользуя то, что подмножества S_1, \dots, S_m не пересекаются, можем записать

$$P_q(X_i, S) = \sum_{k=1}^m P(X_i, S_k). \quad (1.11.4)$$

Вводя производящую функцию $A(s)$ последовательности a_q

$$A(s) = \sum_{q=1}^{\infty} a_q s^q, \quad (1.11.5)$$

умножая выражения (1.11.4) на s^q и суммируя их для всех q от 1 до ∞ , получим

$$P(s, S) = \sum_{k=1}^m P(s, S_k), \quad (1.11.6)$$

где $P(s, S)$ и $P(s, S_k)$ — соответственно производящие функции последовательностей $P_q(X_i, S)$ и $P_q(X_i, S_k)$.

Предположим, что $P_0(X_i, S) = 1$, т. е. что $X_i = S$. Тогда формулу (1.11.2) можно записать так:

$$N = 1 + P(1, S) = 1 + \sum_{k=1}^m P(1, S_k). \quad (1.11.7)$$

Задача, таким образом, свелась к отысканию чисел $P(1, S_k)$. Для сокращения записи в дальнейшем будем писать вместо $P(1, S_k)$ просто $P(S_k)$.

Теперь найдем вероятность $P(S_k)$. Согласно определению переходной вероятности за q шагов

$$P_q(\mathbf{X}_i, S_l) = \sum_{\mathbf{X}_j \in S_l} p_{ij}^{(q)}. \quad (1.11.8)$$

Используя уравнение Колмогорова—Чепмена, получим

$$\begin{aligned} P_q(\mathbf{X}_i, S_l) &= \sum_{\mathbf{X}_j \in S_l} p_{ik}^{(q-1)} \sum_{\mathbf{X}_h \in S} p_{hj} = \\ &= \sum_{s=1}^m \sum_{\mathbf{X}_j \in S_l} p_{ik}^{(q-1)} \sum_{\mathbf{X}_h \in S_s} p_{hj}, \end{aligned} \quad (1.11.9)$$

или, применяя введенные обозначения,

$$P_q(\mathbf{X}_i, S_l) = \sum_{s=1}^m p(S_s, S_l) P_{q-1}(\mathbf{X}_i, S_s). \quad (1.11.10)$$

Переходя к уравнениям в производящих функциях и полагая в них $s=1$, получим

$$P(S_l) = \sum_{s=1}^m p(S_s, S_l) P(S_s) + \sum_{s=1}^m p(S_s, S_l) P_0(S_s) \quad (l=1, \dots, m), \quad (1.11.11)$$

где $P_0(S_s)$ есть вероятность того, что в начальный момент $\mathbf{X}_l \in S_s$.

Решая полученную линейную неоднородную систему уравнений, найдем $P(S_l)$ ($l=1, \dots, m$). Подставляя полученные значения в (1.11.7), определим среднее число тактов, необходимое системе для первого попадания в окрестность экстремума.

С точки зрения сравнения детерминированных и случайных методов поиска интересно рассмотреть ограниченные запрещенные области, закрывающие прямой путь в окрестность экстремума из какой-либо части плоскости.

Найдем N для запрещенных областей двух типов:

1) запрещенной области типа I: множество точек $X = (x_1, x_2)$ с координатами, удовлетворяющими неравенствам

$$0 \leq x_1 \leq m; \quad -l \leq x_2 \leq l; \quad (1.11.12)$$

2) запрещенной области типа II: множество точек, координаты которых удовлетворяют неравенствам

$$x_1 - x_2 \leq m + 1; \quad x_2 \leq 0; \quad x_1 \geq 0, \quad (1.11.13)$$

за исключением точек $(m+1, 0)$, $(0, -m-1)$.

Запрещенные области типов I и II представлены на рис. 1.11.1 в виде заштрихованных участков.

Определим N для ограничений, создаваемых областью типа I.

В этом случае окрестность минимума состоит из двух точек $(-1, 1)$, $(-1, -1)$, а множество возможных состояний можно разбить на однородные подмножества (указываются только точки однородного подмножества, расположенные в верхней полуплоскости; точки в нижней полуплоскости определяются из соображений симметрии):

1) множества A_i ($i=2, 3, \dots, l$);

$$A_i = \{\emptyset: (-1, i)\};$$

через \emptyset обозначена точка плоскости $X = (x_1, x_2)$;

2) множества B_i ($i=1, 2, \dots, l$);

$$B_i = \{\emptyset: (-k, i) (k=2, 3, \dots, \infty)\};$$

3) множества C_i ($i=1, 2, \dots, m$);

$$C_i = \{\emptyset: (i, k) (k=l+1, l+2, \dots, \infty)\};$$

4) множества D_i ($i=1, 2, \dots, l$);

$$D_i = \{\emptyset: (k, i) (k=m+1, m+2, \dots, \infty)\};$$

5) множество E :

$$E = \{\emptyset: (-1, k) (k=l+1, l+2, \dots, \infty)\};$$

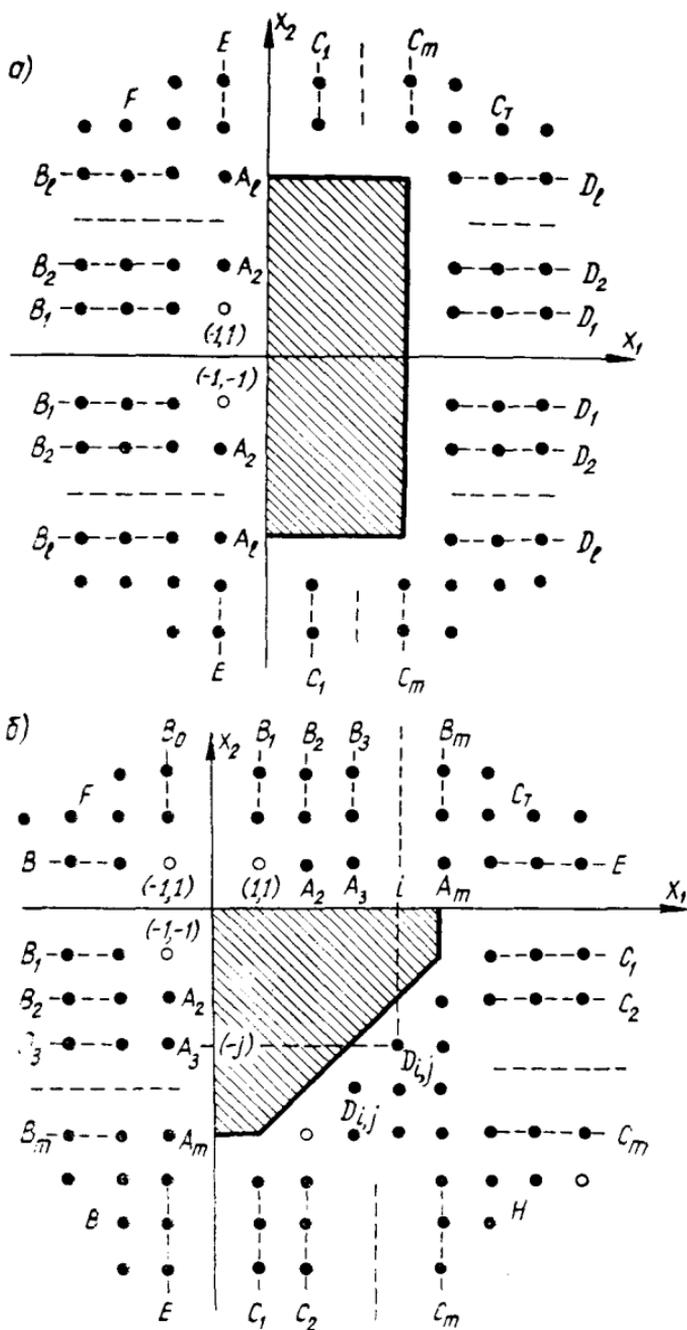


Рис. 1.11.1. Запрещенные области типа I (а) и типа II (б).

6) множество F :

$$F = \{\emptyset: (i, j) \ (i = -2, -3, \dots, -\infty; j = l+1, l+2, \dots, \infty)\};$$

7) множество G :

$$G = \{\emptyset: (i, j) \ (i = m+1, m+2, \dots, \infty; j = l+1, l+2, \dots, \infty)\}.$$

Эти множества содержат все возможные состояния системы (т. е. все точки плоскости, в которых система может остановиться для выбора нового направления движения) и действительно являются однородными.

Предположим сначала, что исходное состояние i находится в множестве G . Запишем для указанных множеств систему уравнений (1.11.11):

$$\alpha \begin{cases} 3P(A_k) - P(B_k) - P(A_{k-1}) = 0; \\ -P(A_k) + 3P(B_k) - P(B_{k-1}) = 0 \end{cases} \quad (k=2, 3, \dots, l);$$

$$3P(B_1) - \sum_{k=1}^l P(B_k) - P(F) = 0;$$

$$2P(C_1) - P_l(F) - P(E) = 0;$$

$$2P(D_1) - \sum_{k=1}^l P(D_k) - P(G) = 1; \tag{1.11.14}$$

$$3P(E) - P(A_l) - \sum_{k=1}^m P(C_k) - P(G) = 1;$$

$$(\beta) \ 2P(C_k) - P(C_{k-1}) = 0 \quad (k=2, 3, \dots, m);$$

$$(\gamma) \ 2P(D_k) - P(D_{k-1}) = 0 \quad (k=2, 3, \dots, l);$$

$$2P(F) - P(B_l) - P(E) = 0;$$

$$2P(G) - P(D_l) - P(C_m) = 2.$$

$P(A_1) \equiv 0$ введено для единообразия записи подсистемы (α) . Решая отдельно от остальных уравнений подсистемы (α) , (β) и (γ) , получим:

$$\begin{aligned}
 P(A_k) &= \frac{4^{l-k} + 2^{l-k}}{2} P(A_l) - \frac{4^{l-k} - 2^{l-k}}{2} P(B_l); \\
 P(B_k) &= \frac{4^{l-k} - 2^{l-k}}{2} P(A_l) + \frac{4^{l-k} + 2^{l-k}}{2} P(B_l); \\
 P(B_l) &= \frac{2^{l-1} + 1}{2^{l-1} - 1} P(A_l); \\
 P(C_k) &= 2^{m-k} P(C_m); \\
 P(D_k) &= 2^{l-k} P(D_l).
 \end{aligned}
 \tag{1.11.15}$$

Подставляя полученные соотношения в оставшиеся уравнения системы (1.11.14) и решая их, получим:

$$\begin{aligned}
 P(A_l) &= \frac{2(2^{l-1} - 1)}{4^l}; \quad P(B_l) = \frac{2(2^{l-1} + 1)}{4^l}; \\
 P(C_m) &= \frac{2(2^{l+1} + 1)}{2^{m+l}}; \quad P(D_l) = 4 + \frac{2(2^{l+1} + 1)}{2^{m+l}}; \\
 P(E) &= \frac{8}{3} + \frac{3 \cdot 2^l - 2}{3 \cdot 4^l}; \quad P(F) = \frac{4}{3} + \frac{3 \cdot 2^l + 2}{3 \cdot 4^l}; \\
 P(G) &= 3 + \frac{2(2^{l+1} + 1)}{2^{m+l}}.
 \end{aligned}
 \tag{1.11.16}$$

Используя формулы (1.11.15) и (1.11.16), найдем выражения для всех $p(\sigma)$ (σ — однородное подмножество); подставляя их в формулу (1.11.7), получим

$$N_G = 10 + 4 \cdot 2^l + 4 \cdot \frac{2^l}{2^m} + 2 \left(\frac{1}{2^l} - \frac{1}{2^m} - \frac{1}{2^{m+l}} \right).
 \tag{1.11.17}$$

N_G — среднее число тактов, необходимое системе для первого попадания в окрестность минимума, если начальное состояние $i \in G$.

Аналогично для случая, когда начальное состояние $i \in F$, получим систему уравнений, отличающуюся от (1.11.14) лишь правыми частями. Решая эту систему и подставляя результаты в (1.11.7), получим

$$N_F = 12 + 4 \cdot \frac{2^l}{2^m} + 4 \left(\frac{1}{2^l} - \frac{1}{2^{m+l}} \right).
 \tag{1.11.18}$$

Рассмотрим случай ограничений, создаваемых областью типа II.

Окрестность минимума состоит теперь из трех точек: $(1, 1)$, $(-1, 1)$, $(-1, -1)$. Однородными множествами будут:

1) множества A_i ($i=2, 3, \dots, m$);

$$A_i = \{\emptyset: (i, 1), (-1, i)\};$$

2) множества B_0 :

$$B_0 = \{\emptyset: (-1, i), (-i, 1) (i=2, 3, \dots, \infty)\};$$

3) множества B_i ($i=1, 2, 3, \dots, m$);

$$B_i = \{\emptyset: (i, k), (-k, -i) (k=2, 3, \dots, \infty)\};$$

4) множества C_i ($i=1, 2, 3, \dots, m$);

$$C_i = \{\emptyset: (i, -k), (k, -i) (k=m+1, m+2, \dots, \infty)\};$$

5) множества D_{ij} .

Множества D_{ij} в совокупности образуют множество точек, координаты которых удовлетворяют следующим неравенствам:

$$2 \leq x_1 \leq m; \quad -m \leq x_2 \leq -2; \quad x_1 - x_2 \geq m + 2.$$

Пусть $2 \leq i \leq m$ и $\frac{m+1}{2} \leq j \leq m$ для m нечетного и $m/2 \leq j \leq m$ для четного m . Тогда

$$D_{ij} = \{\emptyset: (i, -j), (j, -i)\};$$

6) множество E :

$$E = \{\emptyset: (k, 1), (-1, -k) (k=m+1, m+2, \dots, \infty)\};$$

7) множество F :

$$F = \{\emptyset: (i, k) (i=-2, -3, \dots, -\infty; k=2, 3, \dots, \infty)\};$$

8) множество G :

$$G = \{\emptyset: (i, k), (-k, -i) (i=m+1, m+2, \dots, \infty; k=2, 3, \dots, \infty)\};$$

9) множество H :

$$H = \{ \emptyset: (i, -k) (i = m+1, m+2, \dots, \infty; k = m+1, m+2, \dots, \infty) \}.$$

В предположении, что $P_0(H) = 1$, получим следующую систему уравнений для искомых величин:

$$(\alpha) \begin{cases} 3P(A_i) - P(B_i) - P(A_{i-1}) = 0; \\ -P(A_i) + 3P(B_i) - P(B_{i-1}) = 0 \end{cases} \\ (i = 2, 3, \dots, m; P(A_1) \equiv 0);$$

$$3P(B_0) - \sum_{i=1}^m P(B_i) - P(G) = 0;$$

$$3P(B_1) - P(B_0) - 2P(F) = 0;$$

$$2P(C_1) - P(E) - P(G) = 0;$$

$$(\beta) \begin{cases} 3P(C_i) - P(C_{i-1}) - P(D_{i,m}) = 0 \\ \qquad \qquad \qquad (i = 2, 3, \dots, m); \\ 4P(D_{i,j}) - P(D_{i-1,j}) - P(D_{i,j-1}) = 0 \\ \qquad \qquad \qquad (i > j+1; i+j > m+2); \\ 4P(D_{i,j}) - 2P(D_{i-1,j}) - P(D_{i,j-1}) = 0 \\ \qquad \qquad \qquad (i = j+1; i+j > m+2); \\ 4P(D_{i,j}) - P(D_{i-1,j}) = 0 \\ \qquad \qquad \qquad (i = j; i+j > m+2); \\ 2P(D_{i,j}) - \sum_{k=j+1}^m (1 + \delta_{ik})P(D_{i,k}) - \sum_{k=i+1}^m P(D_{k,j}) - \\ - P(C_i) - P(C_j) = 0 \\ \qquad \qquad \qquad \left(i+j = m+2; \delta_{ik} = \begin{cases} 1, & i=k \\ 0, & i \neq k \end{cases} \right); \end{cases}$$

$$3P(E) - \sum_{i=1}^m P(C_i) - 2P(H) - P(A_m) = 2;$$

$$2P(F) - P(B_0) = 0;$$

$$2P(G) - P(B_m) - P(E) = 0;$$

$$2P(H) - P(C_m) = 2.$$

(1.11.19)

Из подсистем (α) и (β) найдем:

$$P(A_k) = \frac{4^{l-k} + 2^{l-k}}{2} P(A_l) - \frac{4^{l-k} - 2^{l-k}}{2} P(B_l);$$

$$P(B_k) = - \frac{4^{l-k} - 2^{l-k}}{2} P(A_l) + \frac{4^{l-k} + 2^{l-k}}{2} P(B_l);$$

$$P(B_l) = \frac{2^{l-1} + 1}{2^{l-1} - 1} P(A_l);$$

$$P(D_{k,m}) = 2^{m-k} P(C_m); \quad P(C_k) = 2^{m-k} P(C_m). \quad (1.11.20)$$

После подстановки соотношений (1.11.20) в оставшиеся уравнения системы (1.11.19) и решения полученной системы найдем:

$$P(A_m) = \frac{2^{m-1} - 1}{17 \cdot 4^{m-1}}; \quad P(B_0) = \frac{12}{17};$$

$$P(C_m) = \frac{17 \cdot 2^{m-2} + 1}{17 \cdot 4^{m-2}}; \quad P(E) = \frac{34 \cdot 4^{m-1} + 3 \cdot 2^{m-1} - 1}{51 \cdot 4^{m-2}};$$

$$P(F) = \frac{6}{17}; \quad P(G) = \frac{17 \cdot 4^{m-1} + 3 \cdot 2^{m-1} + 1}{51 \cdot 4^{m-2}}. \quad (1.11.21)$$

Находя с помощью соотношений (1.11.20) и (1.11.21) все необходимые величины и подставляя их в (1.11.7), получим

$$N_H = 10 + m + \frac{m+2}{17 \cdot 2^{m-2}}. \quad (1.11.22)$$

Для случаев, когда начальное состояние расположено в множествах G и F , тем же методом соответственно получим:

$$N_G = 10 + m + \frac{m+2}{17 \cdot 2^{m-2}}; \quad N_F = 8 + \frac{9(m+2)}{17 \cdot 2^{m-1}}. \quad (1.11.23)$$

Определим теперь время, необходимое системе для первого попадания в окрестность минимума. Назовем такт правильным, если система в конце данного такта не удалилась от точки минимума, т. е. приблизилась к ней или осталась на том же расстоянии, на котором она находилась в начале такта. В противном случае будем называть такт неправильным.

Пусть до того, как система попала в окрестность минимума, прошло n тактов и пусть правильных было k . Число неправильных тактов равно тогда $n - k$. При данном методе поиска потеря времени на каждый неправильный такт равна 2 сек. В самом деле, 1 сек уходит на то, чтобы совершить этот неправильный такт, и 1 сек на то, чтобы потом снова сделать шаг в правильном направлении, так как после n тактов система должна оказаться в окрестности минимума.

Из k правильных тактов 2 необходимы (при сделанных предположениях) для попадания в указанную окрестность. Время, требующееся для осуществления этих тактов, обозначим t' . Остается $k - 2$ тактов, которые система затрачивает на «колебания» около осей x_1 и x_2 , являющихся (при предположениях, сделанных относительно показателя качества) линиями остановки, и на «отражение» от границ запрещенных множеств. В обоих случаях на каждый такой такт затрачивается 2 сек и, следовательно, всего $2(k - 2)$ сек.

Таким образом, время, соответствующее первому попаданию в окрестность минимума за n тактов, равно

$$t = t' + 2(k - 2) + 2(n - k). \quad (1.11.24)$$

Усредняя это выражение по возможным реализациям, получим

$$T = t' + 2(N - 2). \quad (1.11.25)$$

Подставляя вместо N его значения, соответствующие случаям (1.11.17), (1.11.18), (1.11.22), (1.11.23), получим зависимость времени поиска от формы и размеров запрещенных областей и от расположения начальной точки на плоскости.

Полученные результаты дают возможность говорить о преимуществе алгоритма (1.11.1) и аналогичных ему случайных методов поиска перед детерминированными при указанных условиях. В самом деле, легко можно указать области, из которых система при детерминированном методе поиска и указанных ограничениях никогда не попадет в окрестность минимума, в то время как формулы (1.11.17), (1.11.18), (1.11.22), (1.11.23) и (1.11.25) показывают, что при применении случайного поиска цель может быть достигнута за вполне приемлемое время.

ГЛАВА II

ПОИСК С САМООБУЧЕНИЕМ

§ 2.1. АЛГОРИТМЫ САМООБУЧЕНИЯ ПРИ СЛУЧАЙНОМ ПОИСКЕ КАК ВЕРОЯТНОСТНЫЕ АВТОМАТЫ

В предыдущей главе были исследованы алгоритмы случайного и детерминированного поисков как вероятностные автоматы в случайных средах. В настоящей главе исследуем с использованием такой интерпретации различные алгоритмы самообучения.

Под самообучением при поиске понимается изменение вероятностных характеристик процесса поиска. Пусть вероятность выбора шага ΔX в пространстве переменных $X = (x_1, \dots, x_n)$ зависит от векторного параметра W , который назовем вектором «памяти» и число координат которого равно числу координат вектора ΔX , т. е.

$$W = (\omega_1, \dots, \omega_n). \quad (2.1.1)$$

Вероятность выбора N -го шага поиска ΔX_N задается как условная вероятность

$$p_N(\Delta X) = p(\Delta X_N / W_N), \quad (2.1.2)$$

где W_N — значение вектора памяти W на N -м шаге поиска.

Значение вектора памяти W_N на N -м шаге поиска определяется как некоторая векторная функция от значения вектора памяти W_{N-1} на $(N-1)$ -м шаге, от сделанного шага ΔX_{N-1} и от приращения функции качества $\Delta Q'_{N-1}$, полученного на $(N-1)$ -м шаге:

$$\mathbf{W}_N = \Psi(\mathbf{W}_{N-1}, \Delta \mathbf{X}_{N-1}, \Delta Q'_{N-1}). \quad (2.1.3)$$

В общем случае эта функция может быть и случайной. На изменение длины вектора памяти \mathbf{W} накладывается ограничение

$$|\mathbf{W}| \leq d\sqrt{n} \quad (2.1.4)$$

($d = \text{const}$),

где n — размерность пространства $\{\mathbf{W}\}$.

Далее предполагается, что вектор памяти \mathbf{W} , а также смещение $\Delta \mathbf{X}$ могут принимать значения из конечных множеств

$$\mathbf{W} \in \{\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \dots, \mathbf{W}^{(m)}\}; \quad (2.1.5)$$

$$\Delta \mathbf{X} \in \{\Delta \mathbf{X}^{(1)}, \Delta \mathbf{X}^{(2)}, \dots, \Delta \mathbf{X}^{(r)}\}. \quad (2.1.6)$$

Что касается приращения функции качества $\Delta Q'$, то нас будут интересовать только два значения:

$$\text{sgn } \Delta Q' = c = \begin{cases} 1, & \text{если } \Delta Q' \geq 0; \\ 0, & \text{если } \Delta Q' < 0. \end{cases} \quad (2.1.7)$$

Поиск с самообучением определим как вероятностный автомат следующим образом. Пусть состоянием автомата является вектор памяти \mathbf{W} , выходом автомата — вектор смещения $\Delta \mathbf{X}$, а его входом $\text{sgn } \Delta Q' = c$. Тогда множество состояний, множество выходов и множество входов автомата будут определяться соответственно формулами (2.1.5), (2.1.6) и (2.1.7). Функция переходов автомата будет определяться выражением (2.1.3), а функция выходов — соотношением (2.1.2). Отметим, что по формуле (2.1.3) изменение состояний автомата $\mathbf{W}_{N-1} \rightarrow \mathbf{W}_N$ зависит также от выхода автомата $\Delta \mathbf{X}_{N-1}$ на $(N-1)$ -м шаге. Таким образом, с учетом предположений (2.1.5) и (2.1.6) алгоритмы самообучения при случайном поиске описываются как вероятностные автоматы с конечным числом состояний и конечным числом выходов. При этом вероятности выходов автомата зависят только от его состояния и не зависят от его входа, а вероятности

переходов автомата из одного состояния в другое зависят кроме состояния и входа автомата также от его выхода на предыдущем шаге.

Введем следующие обозначения:

$$p_{i_1, i_2}^{(c)}(j) = \text{Вер} (\mathbf{W}_N^{(i_1)} \rightarrow \mathbf{W}_{N+1}^{(i_2)} / \Delta \mathbf{X}_N = \Delta \mathbf{X}^{(j)}, c) \quad (2.1.8)$$

$$(i_1, i_2 = 1, \dots, m; c = 0, 1);$$

$$q_{ij} = \text{Вер} (\Delta \mathbf{X}_N = \Delta \mathbf{X}^{(j)} / \mathbf{W}_N = \mathbf{W}^{(i)}) \quad (2.1.9)$$

$$(i = 1, \dots, m; j = 1, \dots, v);$$

$$s_j = \text{Вер} (c = 1 / \Delta \mathbf{X} = \Delta \mathbf{X}^{(j)}) = \frac{1}{2} \left[1 + \Phi \left(\frac{\Delta Q^{(j)}}{2\sigma} \right) \right] \quad (2.1.10)$$

$$(j = 1, \dots, v);$$

$$\Phi(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt \quad \text{— интеграл вероятности. Здесь}$$

$p_{i_1, i_2}^{(c)}(j)$ обозначает вероятность перехода автомата из состояния $\mathbf{W}^{(i_1)}$ в состояние $\mathbf{W}^{(i_2)}$, если на предыдущем шаге поиска на выходе автомата появился сигнал $\Delta \mathbf{X}^{(j)}$, а потом на его вход поступил входной сигнал c ; q_{ij} — вероятность появления на выходе сигнала $\Delta \mathbf{X}^{(j)}$ при нахождении автомата в состоянии $\mathbf{W}^{(i)}$; $s^{(j)}$ — вероятность штрафа j -го действия (выхода) автомата.

Переходную функцию автомата будем задавать при помощи $2v$ следующих стохастических матриц: для $c = 0$

$$A_0(j) = \left\| \begin{array}{ccc} p_{11}^{(0)}(j) & \cdots & p_{1m}^{(0)}(j) \\ \cdot & \cdot & \cdot \\ p_{m1}^{(0)}(j) & \cdots & p_{mm}^{(0)}(j) \end{array} \right\| \quad (2.1.11)$$

для $c = 1$

$$A_1(j) = \left\| \begin{array}{ccc} p_{11}^{(1)}(j) & \cdots & p_{1m}^{(1)}(j) \\ \cdot & \cdot & \cdot \\ p_{m1}^{(1)}(j) & \cdots & p_{mm}^{(1)}(j) \end{array} \right\| \quad (2.1.12)$$

Выходную функцию автомата задаем при помощи прямоугольной матрицы порядка $m \times v$

$$Q_{ij} = \begin{vmatrix} q_{11} & \cdots & q_{1v} \\ \cdot & \cdot & \cdot \\ q_{v1} & \cdots & q_{vv} \\ \cdot & \cdot & \cdot \\ q_{m1} & \cdots & q_{mv} \end{vmatrix} \quad (2.1.13)$$

Далее, используя формулы (2.1.8) и (2.1.9), находим совместную вероятность события, состоящего в том, что у автомата появился выход $\Delta X^{(j)}$ и автомат потом переходит из состояния $\mathbf{W}^{(i_1)}$ в состояние $\mathbf{W}^{(i_2)}$ при условии, что на его вход поступил сигнал c :

$$\text{Вер} (\mathbf{W}_N^{(i_1)} \rightarrow \mathbf{W}_{N+1}^{(i_2)}, \Delta X_N = \Delta X^{(j)} / c) = q_{i_1 j} p_{i_1 i_2}(j). \quad (2.1.14)$$

Совместная вероятность следующих трех событий: $\mathbf{W}_N^{(i_1)} \rightarrow \mathbf{W}_{N+1}^{(i_2)}$; $\Delta X_N = \Delta X^{(j)}$ и $c=1$ равна

$$\text{Вер} (\mathbf{W}_N^{(i_1)} \rightarrow \mathbf{W}_{N+1}^{(i_2)}, \Delta X_N = \Delta X^{(j)}, c=1) = s_j q_{i_1 j} p_{i_1 i_2}^{(1)}(j). \quad (2.1.15)$$

Аналогично для $c=0$ имеем

$$\begin{aligned} \text{Вер} (\mathbf{W}_N^{(i_1)} \rightarrow \mathbf{W}_{N+1}^{(i_2)}, \Delta X_N = \Delta X^{(j)}, c=0) = \\ = (1 - s_j) q_{i_1 j} p_{i_1 i_2}^{(0)}(j). \end{aligned} \quad (2.1.16)$$

Суммируя эти вероятности, получаем совместную вероятность перехода автомата из состояния $\mathbf{W}^{(i_1)}$ в состояние $\mathbf{W}^{(i_2)}$ при появлении сигнала $\Delta X^{(j)}$ на его выходе:

$$\begin{aligned} \text{Вер} (\mathbf{W}_N^{(i_1)} \rightarrow \mathbf{W}_{N+1}^{(i_2)}, \Delta X_N = \Delta X^{(j)}) = p_{i_1 i_2}(j) = \\ = q_{i_1 j} [(1 - s_j) p_{i_1 i_2}^{(0)}(j) + s_j p_{i_1 i_2}^{(1)}(j)]. \end{aligned} \quad (2.1.17)$$

Суммируя вероятности $p_{i_1 i_2}(j)$ по всем выходам $\Delta X^{(j)}$ ($j=1, \dots, v$), находим полную вероятность перехода автомата из состояния $\mathbf{W}^{(i_1)}$ в состояние $\mathbf{W}^{(i_2)}$:

$$p_{i_1 i_2} = \sum_{j=1}^v q_{i_1, j} [(1-s_j) p_{i_1 i_2}^{(0)}(j) + s_j p_{i_1 i_2}^{(1)}(j)] \quad (2.1.18)$$

$$(i_1, i_2 = 1, \dots, m).$$

Следовательно, полное функционирование автомата в случайной среде (2.1.10) описывается цепью Маркова с переходными вероятностями (2.1.18). Запишем эту цепь Маркова в виде стохастической матрицы порядка $m \times m$

$$A = \begin{vmatrix} p_{11} & \dots & p_{1m} \\ \dots & \dots & \dots \\ p_{m1} & \dots & p_{mm} \end{vmatrix} \quad (2.1.19)$$

где $p_{i_1 i_2}$ ($i_1, i_2 = 1, \dots, m$) определяются формулой (2.1.18). Предельные вероятности этой цепи Маркова, если она однородная и эргодическая, находятся как решение системы алгебраических уравнений

$$\mathbf{P} = A' \mathbf{P}, \quad (2.1.20)$$

где A' — транспонированная матрица (2.1.19);

\mathbf{P} — вектор предельных вероятностей; $\mathbf{P} = (p_1, \dots, \dots, p_m)$.

В некоторых случаях для исследования матрицы (2.1.19) удобнее представить ее при помощи более простых матриц. Для этого вводим квадратные матрицы $Q_i(j)$ и $S(j)$ порядка m :

$$Q_i(j) = \begin{vmatrix} q_{1j} & 0 & 0 & \dots & 0 & 0 \\ 0 & q_{2j} & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 0 & q_{mj} \end{vmatrix} \quad (2.1.21)$$

$$S(j) = \begin{vmatrix} s_j & 0 & 0 & \dots & 0 & 0 \\ 0 & s_j & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 0 & s_j \end{vmatrix} \quad (2.1.22)$$

При помощи введенных матриц (2.1.21) и (2.1.22) и матриц (2.1.11), (2.1.12) матрица A представляется в виде

$$A = \sum_{j=1}^v [(I - S(j)) Q_i(j) A_0(j) + S(j) Q_i(j) A_1(j)], \quad (2.1.23)$$

где I — единичная матрица порядка m .

Отметим, что при помощи конечных вероятностных автоматов, переходные функции которых зависят от их выходов, представляется любой алгоритм дискретного самообучения (векторы \mathbf{W} и $\Delta \mathbf{X}$ принимают значения из дискретного множества векторов). Рассмотрим некоторые частные случаи автоматов, функционирование которых описывается цепью Маркова с матрицей переходных вероятностей (2.1.23).

1. *Вероятностный автомат, переходная функция которого не зависит от его выходов.* В этом случае все матрицы $A_c(j)$ при фиксированном c равны между собой, т. е.

$$A_0(j) = A_0; \quad A_1(j) = A_1 \quad (j=1, \dots, v). \quad (2.1.24)$$

Матрица A равна

$$A = (I - S^*) A_0 + S^* A_1, \quad (2.1.25)$$

где

$$S^* = \sum_{j=1}^v S(j) Q_i(j) = \begin{vmatrix} \sum_{j=1}^v q_{1j} S_j & 0 & \dots & 0 \\ 0 & \sum_{j=1}^v q_{2j} S_j & \dots & 0 \\ 0 & 0 & \dots & \sum_{j=1}^v q_{mj} S_j \end{vmatrix} \quad (2.1.26)$$

Замеченные ошибки

Стр.	Строка или № формулы	Напечатано	Должно быть
24	Форм. (0.3.29)	$T_1(y_1/x)B$	$T_1(y/x)B$
51	Форм. (0.3.11), 4-я строка	$\Delta X^{(m/2+1)}$	$\Delta X^{(m/2+i)}$
63	2-я строка снизу	$= (+1, \dots, +1, -1, -1);$	$= (+1, \dots, +1, -1, +1);$
71	Форм. (1.5.1), 2-я строка	$\Delta X^{(2)} = -ae_2; \dots$	$\Delta X^{(n+2)} = -ae_2; \dots$
94	Матрица (1.7.26)	$\dots B_3 B_1$ $\dots 0 B_3$	$\dots B_2 B_1$ $\dots 0 B_2$
111	Матрица (1.9.13)	$.010 \dots 01.$	$.000 \dots 01.$
112	Форм. (1.9.14), 2-я строка	$+ \sum_{j=m+1}^{m+n} p_j^{(0)}$	$+ \sum_{j=m+1}^{m+n} p_j^{(0)}$
186	Табл. 2.2.2, 2-я строка снизу	$\frac{1}{4} (1 - k\delta)$	$\frac{1}{4} (1 + k\delta)$
191	Форм. (2.2.54), 1-я строка	$= p_{2^{n-1}} =$	$= p_{2^{n-1}} =$
294	1-я строка снизу	$(\sigma = 1)$	$(\delta = 1)$
334	2-я строка текста снизу	Тогда выражения (3.5.11) и (3.5.12)	Тогда выражения (3.5.11) и (3.5.13)
340	6-я и 7-я строки сверху	Случайный поиск в многопараметрических задачах.	Проблемы случайного поиска, вып. 1.
341	9-я и 10-я строки снизу	Сопоставление непрерывного и автоматного случайного поиска. 118	Сопоставление непрерывного и автоматного алгоритмов случайного поиска . . . 113
341	6—8-я строки снизу	Переходный процесс в двумерной экстремальной системе при наличии запрещенных областей и случайном методе поиска . 123	Автоматная оптимизация при наличии ограничений . . . 123

Здесь сумма

$$s^{(i)} = \sum_{j=1}^v q_{ij}s_j, \tag{2.1.27}$$

где s_j ($j=1, \dots, v$) определяется формулой (2.1.10), а q_{ij} — элементы матрицы (2.1.13), представляющие собой вероятность, с какой штрафуются автомат, если он находится в состоянии $\mathbf{W}^{(i)}$ ($i=1, 2, \dots, m$). Эта формула аналогична формуле, рассмотренной в главе II для случая, когда выход автомата совпадает с его состоянием.

2. *Вероятностные автоматы со случайными выходами и детерминированными переходами.* Функционирование этого автомата в случайной среде описывается частным случаем цепи Маркова (2.1.25). Для этого случая матрицы A_0 и A_1 являются стохастическими булевыми матрицами, т. е. матрицами, у которых в каждой строке один элемент равен единице, а все другие равны нулю. Вероятности штрафов для таких автоматов рассматриваются относительно его состояний и определяются формулой (2.1.26).

3. *Вероятностные автоматы со случайными переходами и детерминированными выходами.* Функционирование такого автомата в случайной среде (2.1.10) представляет собой процесс оптимизации при помощи коллектива независимых автоматов [12, 13]. Его полное функционирование описывается частным случаем цепи Маркова (2.1.25). В этом случае матрицы A_0 и A_1 являются стохастическими, а матрица Q_{ij} (формула (2.1.13)) — булевой стохастической матрицей. Матрицы $Q_i(j)$ ($j=1, \dots, v$) имеют вид

$$Q_i(j) = \left[\begin{array}{cccccc} 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \end{array} \right] \tag{2.1.28}$$

$\underbrace{\hspace{10em}}_{j_i} \quad \underbrace{\hspace{10em}}_{m-j_i}$

где номер j_i соответствует действию $\Delta X^{(j_i)}$, которое производит автомат в состоянии $\mathbf{W}^{(i)}$. С учетом матрицы (2.1.28) и формулы (2.1.27) находим матрицу вероятностей штрафов

$$S^* = \begin{pmatrix} s_{j_1} & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\ 0 & s_{j_2} & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\ \dots & \dots \\ 0 & 0 & 0 & \dots & s_{j_i} & 0 & \dots & 0 & 0 \\ \dots & \dots \\ 0 & 0 & 0 & \dots & 0 & 0 & \dots & 0 & s_{j_m} \end{pmatrix} \quad (2.1.29)$$

где s_{j_i} ($j_i=1, \dots, v$) — вероятность штрафа того действия автомата $\Delta X^{(j_i)}$, которое соответствует состоянию $\mathbf{W}^{(i)}$.

Рассмотрим, как для автомата, заданного матрицами (2.1.11) и (2.1.12), определить выходную функцию (2.1.2). Так как вектор ΔX принимает значение из конечного множества (2.1.6), то и координаты Δx_l ($l=1, \dots, n$) вектора ΔX принимают значения из конечных множеств

$$\Delta x_l \in \{\Delta x_l^{(1)}, \dots, \Delta x_l^{(v_l)}\} \quad (2.1.30)$$

$$(l=1, 2, \dots, n),$$

где v_l — конечное число.

Выходную функцию автомата $p_N(\Delta X)$, определенную формулой (2.1.2), задаем при помощи выходной функции для каждой координаты Δx_l вектора ΔX , т. е. используя n функций, не зависящих от номера шага N :

$$p_l^{(N)}(\Delta x_l) = p_l(\Delta x_l^{(N)}/\mathbf{W}_N) \quad (2.1.31)$$

$$(l=1, 2, \dots, n),$$

где N — номер шага поиска.

Рассмотрим алгоритмы самообучения при поиске по вершинам гиперкуба. Предположим, что координаты Δx_l ($l=1, \dots, n$) вектора ΔX могут принимать только одно из двух значений: +1 или -1. Тогда множество действий автомата имеет

$$v = 2^n \quad (2.1.32)$$

элементов, т. е. имеются 2^n возможных смещений $\Delta \mathbf{X}^{(j)}$ в пространстве $\{\mathbf{X}\}$. Предположим, что l -я функция (2.1.31) зависит только от l -й координаты ω_l вектора памяти \mathbf{W} . Тогда формулу (2.1.31) можно задать в виде следующих двух формул:

$$\text{Вер} (\Delta x_l^{(N)} = 1/\omega_l^{(N)}) = F_l(q_l, \omega_l); \quad (2.1.33)$$

$$\text{Вер} (\Delta x_l^{(N)} = -1/\omega_l^{(N)}) = 1 - F_l(q_l, \omega_l) \quad (2.1.34)$$

$$(l=1, 2, \dots, n; \quad q_l = \text{const}, 0 < q_l \leq 1).$$

Далее будем требовать, чтобы функции $F_l(q_l, \omega_l)$ были монотонно возрастающими и чтобы они удовлетворяли условию

$$0 \leq F_l(q_l, \omega_l) \leq 1 \quad (2.1.35)$$

$$(l=1, \dots, n).$$

Функция $F_l = F_l(q_l, \omega_l)$ может иметь следующий вид:

$$F_l = \begin{cases} 1, & \text{если } \omega_l > q; \\ \frac{1}{2} \left(1 + \frac{\omega_l}{q} \right), & \text{если } |\omega_l| \leq q; \\ 0, & \text{если } \omega_l < -q. \end{cases} \quad (2.1.36)$$

Вид этой функции показан на рис. 2.1.1. На изменение

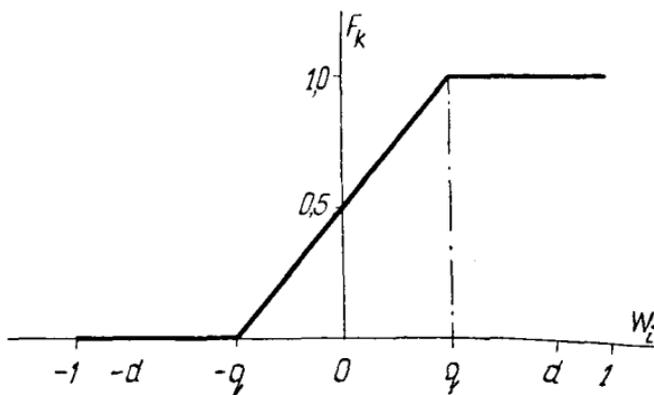


Рис. 2.1.1. Функция выхода автомата.

координат x_l вектора \mathbf{W} будем накладывать ограничение

$$\tilde{\omega}_l = \begin{cases} d, & \text{если } \omega_l > d; \\ \omega_l, & \text{если } -d \leq \omega_l \leq d; \\ -d, & \text{если } \omega_l < -d. \end{cases} \quad (2.1.37)$$

Теперь рассмотрим, как, используя формулы (2.1.33) и (2.1.34), определить вероятность q_{ij} , т. е. вероятность появления выхода $\Delta \mathbf{X}^{(j)}$ при условии, что автомат находится в состоянии $\mathbf{W}^{(i)}$. По определению (2.1.9) положено

$$q_{ij} = \text{Вер} (\Delta \mathbf{X} = \Delta \mathbf{X}^{(j)} / \mathbf{W} = \mathbf{W}^{(i)}).$$

Каждый выход $\Delta \mathbf{X}^{(j)}$ характеризуется набором своих координат, т. е.

$$\Delta \mathbf{X}^{(j)} = (\Delta x_1^{(j)}, \dots, \Delta x_n^{(j)}), \quad (2.1.38)$$

где

$$\Delta x_l^{(j)} = \begin{cases} +1 & \text{с вероятностью } F_l(q_l, \omega_l^{(i)}); \\ -1 & \text{с вероятностью } 1 - F_l(q_l, \omega_l^{(i)}). \end{cases}$$

Вероятности $F_l^{(i)} = F_l(q_l, \omega_l^{(i)})$ и $1 - F_l(q_l, \omega_l^{(i)}) = 1 - F_l^{(i)}$ зависят от состояния $\mathbf{W}^{(i)}$, в котором находится вектор \mathbf{W} , т. е. от координат $\omega_l^{(i)}$ вектора $\mathbf{W}^{(i)}$. Эта зависимость выражается формулами (2.1.33) и (2.1.34). Объединяя формулы (2.1.33) и (2.1.34), имеем:

$$\text{Вер} (\Delta x_l = \Delta x_l^{(j)} / \omega_l = \omega_l^{(i)}) = \frac{1}{2} [1 - (1 - 2F_l^{(i)}) \text{sign } \Delta x_l^{(j)}], \quad (2.1.39)$$

где

$$\text{sign } z = \begin{cases} 1, & \text{если } z > 0; \\ 0, & \text{если } z = 0; \\ -1, & \text{если } z < 0. \end{cases} \quad (2.1.40)$$

При $\Delta x_l^{(i)} = 1$ из этого соотношения получаем формулу (2.1.33), а при $\Delta x_l^{(i)} = -1$ — формулу (2.1.34). Следовательно,

$$q_{ij} = \text{Вер} (\Delta x_1 = \Delta x_1^{(j)}, \dots, \Delta x_n = \Delta x_n^{(j)} / \mathbf{W} = \mathbf{W}^{(i)}) = \left(\frac{1}{2}\right)^n \prod_{l=1}^n [1 - (1 - 2F_l^{(i)}) \text{sign } \Delta x_l^{(j)}], \quad (2.1.41)$$

где

$$\mathbf{W}^{(i)} = (w_1^{(i)}, \dots, w_n^{(i)}). \quad (2.1.42)$$

Теперь в общем виде определены все величины, необходимые для составления матрицы переходных вероятностей (2.1.23). После нахождения вектора предельных вероятностей $\mathbf{P} = (p_1, \dots, p_n)$ для этой цепи Маркова определяется средний выход автомата, т. е. среднее смещение в пространстве \mathbf{X} на одном шаге поиска:

$$M[\Delta\mathbf{X}] = \sum_{i=1}^m p_i \left(\sum_{j=1}^{2^n} q_{ij} \Delta\mathbf{X}^{(j)} \right). \quad (2.1.43)$$

В этой формуле сумма

$$M[\Delta\mathbf{X}/\mathbf{W}^{(i)}] = \sum_{j=1}^{2^n} q_{ij} \Delta\mathbf{X}^{(j)} \quad (2.1.44)$$

является средним выходом автомата при его нахождении в состоянии $\mathbf{W}^{(i)}$, т. е. средним смещением в пространстве \mathbf{X} при значении вектора памяти $\mathbf{W} = \mathbf{W}^{(i)}$. Среднее смещение по направлению градиента определяется как скалярное произведение векторов $M[\Delta\mathbf{X}]$ и градиентного вектора $\alpha = (\alpha_1, \dots, \alpha_n)$ оптимизируемой функции $Q(\mathbf{X})$.

Далее рассмотрим несколько конкретных видов формулы самообучения (2.1.3). Пусть функция $\Psi = \Psi(\mathbf{W}_{N-1}, \Delta\mathbf{X}_{N-1}, \Delta Q'_{N-1})$ является детерминированной и имеет вид

$$\mathbf{W}_N = \begin{cases} \mathbf{W}_{N-1} - \delta \operatorname{sign} \Delta Q'_{N-1} \operatorname{sign} \Delta\mathbf{X}_{N-1}, & \text{если } |\mathbf{W}_N| \leq d\sqrt{n}; \\ \frac{\mathbf{W}_N d\sqrt{n}}{|\mathbf{W}_N|}, & \text{если } |\mathbf{W}_N| > d\sqrt{n}, \end{cases} \quad (2.1.45)$$

где вектор $\operatorname{sign} \Delta\mathbf{X}$ определяется как

$$\operatorname{sgn} \Delta\mathbf{X} = (\operatorname{sgn} \Delta x_1, \dots, \operatorname{sgn} \Delta x_n); \quad (2.1.46)$$

$$\operatorname{sgn} z = \begin{cases} -1, & \text{если } z < 0; \\ +1, & \text{если } z \geq 0. \end{cases} \quad (2.1.47)$$

Постоянная δ здесь определяется формулой

$$\delta = \frac{d}{k}, \quad (2.1.48)$$

где k — любое фиксированное натуральное число ($k = 1, 2, 3, \dots$); $d \geq 0$. Предполагается, что каждая координата ω_l ($l = 1, \dots, n$) вектора памяти $\mathbf{W} = (\omega_1, \dots, \omega_n)$ может находиться в одном из следующих $2k+1$ состояний:

$$\omega_l = d, (k-1)\delta, (k-2)\delta, \dots, \delta, 0, -\delta, -2\delta, \dots, - (k-1)\delta, -d. \quad (2.1.49)$$

Тогда вектор памяти \mathbf{W} может находиться в одном из $(2k+1) \times (2k+1)$ состояний, т. е. множество внутренних состояний автомата (2.1.5) содержит $(2k+1) \times (2k+1)$ элементов. Самообучение заключается в переходе вектора \mathbf{W} из одного состояния $\mathbf{W}^{(i)}$ в другое $\mathbf{W}^{(j)}$ в соответствии с формулой (2.1.45).

Видно, что формула (2.1.45) определяет вероятностный автомат, вероятности переходов которого зависят от его выходов. Переходы этого автомата определяются стохастическими матрицами (2.1.11) и (2.1.12). Построим эти матрицы. Для удобства перепишем формулу (2.1.45) в виде двух формул в отдельности для случая нештрафа ($\Delta Q'_N < 0$) и для случая штрафа ($\Delta Q'_N \geq 0$):
для $\Delta Q'_N < 0$

$$\mathbf{W}_N = \begin{cases} \mathbf{W}_{N-1} + \delta \operatorname{sign} \Delta \mathbf{X}_{N-1}, & \text{если } |\mathbf{W}_N| < d\sqrt{n}; \\ \frac{\mathbf{W}_N d\sqrt{n}}{|\mathbf{W}_N|}, & \text{если } |\mathbf{W}_N| > d\sqrt{n}; \end{cases} \quad (2.1.50)$$

для $\Delta Q'_N \geq 0$ имеем

$$\mathbf{W}_N = \begin{cases} \mathbf{W}_{N-1} - \delta \operatorname{sign} \Delta \mathbf{X}_{N-1}, & \text{если } |\mathbf{W}_N| \leq d\sqrt{n}; \\ \frac{\mathbf{W}_N d\sqrt{n}}{|\mathbf{W}_N|}, & \text{если } |\mathbf{W}_N| > d\sqrt{n}. \end{cases} \quad (2.1.51)$$

Первая из полученных формул (2.1.50) задает матрицы $A_0(j)$ ($j=1, \dots, v$), а вторая — (2.1.51) — матрицы $A_1(j)$ ($j=1, \dots, v$).

На рис. 2.1.2 показаны пространства $\{X\}$ и $\{W\}$ для

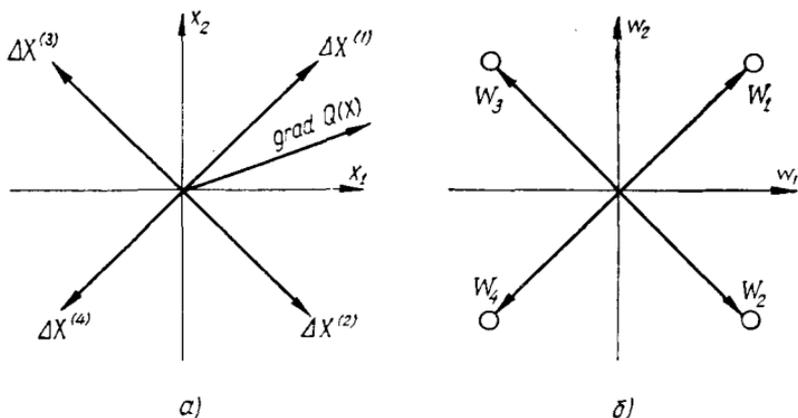


Рис. 2.1.2. Пространство выходов $\Delta X^{(i)}$ и пространство состояний $W^{(j)}$ автомата.

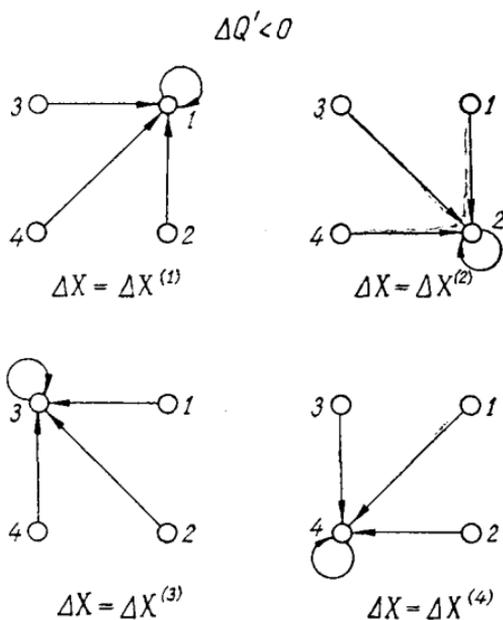


Рис. 2.1.3. Переходы автомата при нештрафе ($\Delta Q' < 0$).

случая, когда $n=2$ и когда количество состояний $\mathbf{W}^{(i)}$ вектора \mathbf{W} равно количеству возможных смещений $\Delta\mathbf{X}^{(j)}$ в пространстве $\{\mathbf{X}\}$, т. е. когда $m=2^n=4$. Там же показана нумерация векторов $\Delta\mathbf{X}^{(j)}$ и $\mathbf{W}^{(i)}$. На рис. 2.1.3 представлены полученные по формуле (2.1.50) переходы автомата из состояния $\mathbf{W}^{(i_1)}$ в состояние $\mathbf{W}^{(i_2)}$ при нештрафе ($\Delta Q' < 0$) для различных значений выхода $\Delta\mathbf{X}^{(j)}$ ($j=1, 2, 3, 4$) автомата. Исходя из данного на рис. 2.1.3 графа переходов при $\delta \geq d$ можно написать следующие матрицы $A_0(j)$ переходных вероятностей:

$$\begin{aligned}
 A_0(1) &= \begin{vmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{vmatrix} & A_0(2) &= \begin{vmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{vmatrix} \\
 A_0(3) &= \begin{vmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{vmatrix} & A_0(4) &= \begin{vmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{vmatrix}
 \end{aligned} \tag{2.1.52}$$

На рис. 2.1.4 показаны полученные по формуле (2.1.51) переходы автомата из одного состояния в другое при штрафе ($\Delta Q' \geq 0$) для различных $\Delta\mathbf{X}^{(j)}$. Исходя из данного на рисунке графа имеем следующие матрицы переходных вероятностей $A_1(j)$ ($j=1, 2, 3, 4$):

$$\begin{aligned}
 A_1(1) &= \begin{vmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{vmatrix} & A_1(2) &= \begin{vmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{vmatrix} \\
 A_1(3) &= \begin{vmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{vmatrix} & A_1(4) &= \begin{vmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{vmatrix}
 \end{aligned} \tag{2.1.53}$$

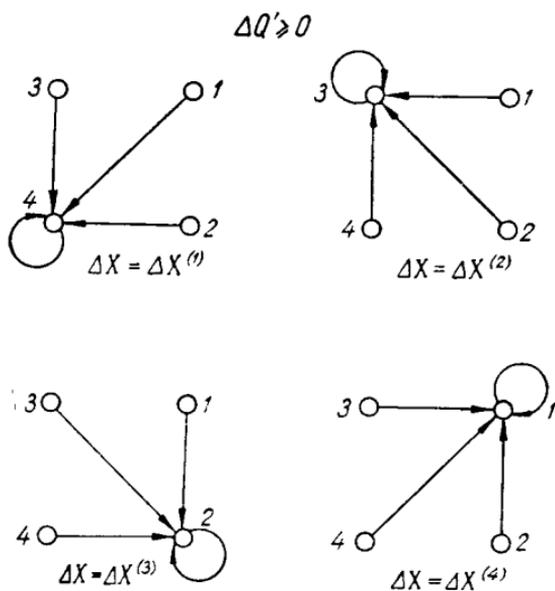


Рис. 2.1.4. Переходы автомата при штрафе ($\Delta Q' \geq 0$).

Матрицы $A_0(j)$ и $A_1(j)$ удовлетворяют равенствам

$$A_1(1) = A_0(4); \quad A_1(2) = A_0(3); \quad (2.1.54)$$

$$A_1(3) = A_0(2); \quad A_1(4) = A_0(1).$$

Определим вид следующей матрицы:

$$A(j) = Q_i(j)[(I - S(j))A_0(j) + S(j)A_1(j)] \quad (2.1.55)$$

$$(j = 1, 2, 3, 4).$$

Учитывая равенства (2.1.54), имеем:

$$A(1) = Q_i(1)[(I - S(1))A_0(1) + S(1)A_0(4)];$$

$$A(2) = Q_i(2)[(I - S(2))A_0(2) + S(2)A_0(3)];$$

$$A(3) = Q_i(3)[(I - S(3))A_0(3) + S(3)A_0(2)]; \quad (2.1.56)$$

$$A(4) = Q_i(4)[(I - S(4))A_0(4) + S(4)A_0(1)].$$

Отметим, что если во вторых слагаемых этих формул матрица $S(j)$ зависит от номера j , то матрица A_0 зависит от номера $(5-j)$. Суммируя равенства формулы (2.1.56), находим

$$A = \sum_{j=1}^4 [Q_i(j) (I - S(j)) + Q_i(5-j) S(5-j)] A_0(j). \quad (2.1.57)$$

Поскольку матрицы $Q_i(j)$ и $S(j)$ диагональные, матрица

$$\bar{Q}(j) = Q_i(j) (I - S(j)) + Q_i(5-j) S(5-j) \quad (2.1.58)$$

также является диагональной. Ее элементы, как не трудно понять, вычисляются по формуле

$$\bar{q}_i(j) = q_{ij}(1 - s_j) + q_{i,(v+1-j)} s_{v+1-j}, \quad (2.1.59)$$

где v — количество выходов автомата;
 q_{ij} ; $q_{i,(v+1-j)}$ — элементы матрицы (2.1.13);
 s_j и s_{v+1-j} определяются формулой (2.1.10).

Используя формулы (2.1.59) и (2.1.57), находим матрицу (2.1.60) переходных вероятностей цепи Маркова, описывающей функционирование алгоритма самообучения (2.1.45) в случайной среде (2.1.10). В этой матрице q_{ij} ($i, j=1, \dots, 4$) — элементы матрицы Q_{ij} ; s_j ($j=1, \dots, 4$) — вероятность штрафа действия $\Delta X^{(j)}$ автомата, определяемая формулой (2.1.10).

$$A = \begin{vmatrix} q_{11}(1-s_1) + q_{14}s_4 & q_{12}(1-s_2) + q_{13}s_3 & & \\ q_{21}(1-s_1) + q_{24}s_4 & q_{22}(1-s_2) + q_{23}s_3 & & \\ q_{31}(1-s_1) + q_{34}s_4 & q_{32}(1-s_2) + q_{33}s_3 & & \\ q_{41}(1-s_1) + q_{44}s_4 & q_{42}(1-s_2) + q_{43}s_3 & & \\ & q_{13}(1-s_3) + q_{12}s_2 & q_{14}(1-s_4) + q_{11}s_1 & \\ & q_{23}(1-s_3) + q_{22}s_2 & q_{24}(1-s_4) + q_{21}s_1 & \\ & q_{33}(1-s_3) + q_{32}s_2 & q_{34}(1-s_4) + q_{31}s_1 & \\ & q_{43}(1-s_3) + q_{42}s_2 & q_{44}(1-s_4) + q_{41}s_1 & \end{vmatrix} \quad (2.1.60)$$

На рис. 2.1.5 показана нумерация внутренних состояний автомата для случая, когда $m > 2^n$, т. е. когда количество внутренних состояний $W^{(i)}$ автомата больше, чем количество его выходов $\Delta X^{(j)}$. На рис. 2.1.6 показаны найденные по формулам (2.1.50) и (2.1.51) переходы такого автомата из одного состояния в другое в зависимости от его выходов $\Delta X^{(j)}$ при нештрафе ($\Delta Q' < 0$). Под каждым графом в скобках указан выход, которому соответствует этот граф переходов в случае штрафования этого выхода. По изображенному на рис. 2.1.6 графу получаем следующие матрицы переходных вероятностей автомата: при нештрафе

$$A_0(j) = \left\{ \begin{array}{cccccc} P(j) & 0 & 0 & \dots & 0 & 0 \\ P(j) & 0 & 0 & \dots & 0 & 0 \\ 0 & P(j) & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & P(j) & 0 \end{array} \right\}^{2k+1}$$

$(j=1, 2);$

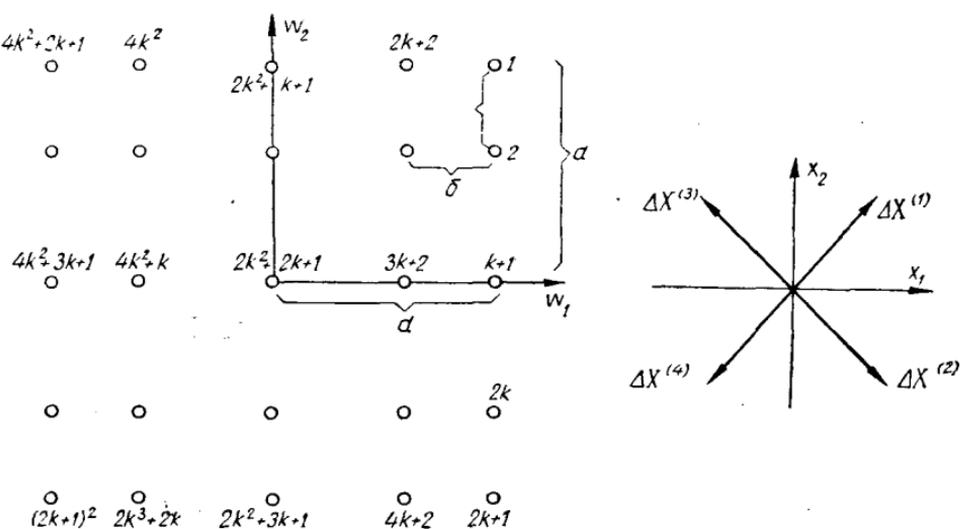


Рис. 2.1.5. Нумерация состояний $W^{(i)}$ и выходов $\Delta X^{(j)}$ автомата при $n=2$ и $m > 2^n$.

$$A_0(j) = \left(\begin{array}{cccccc} 0 & P(j) & 0 & 0 & \dots & 0 \\ 0 & 0 & P(j) & 0 & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & \dots & P(j) \\ 0 & 0 & 0 & 0 & \dots & P(j) \end{array} \right)_{2k+1} \quad (j=3, 4); \quad (2.1.61)$$

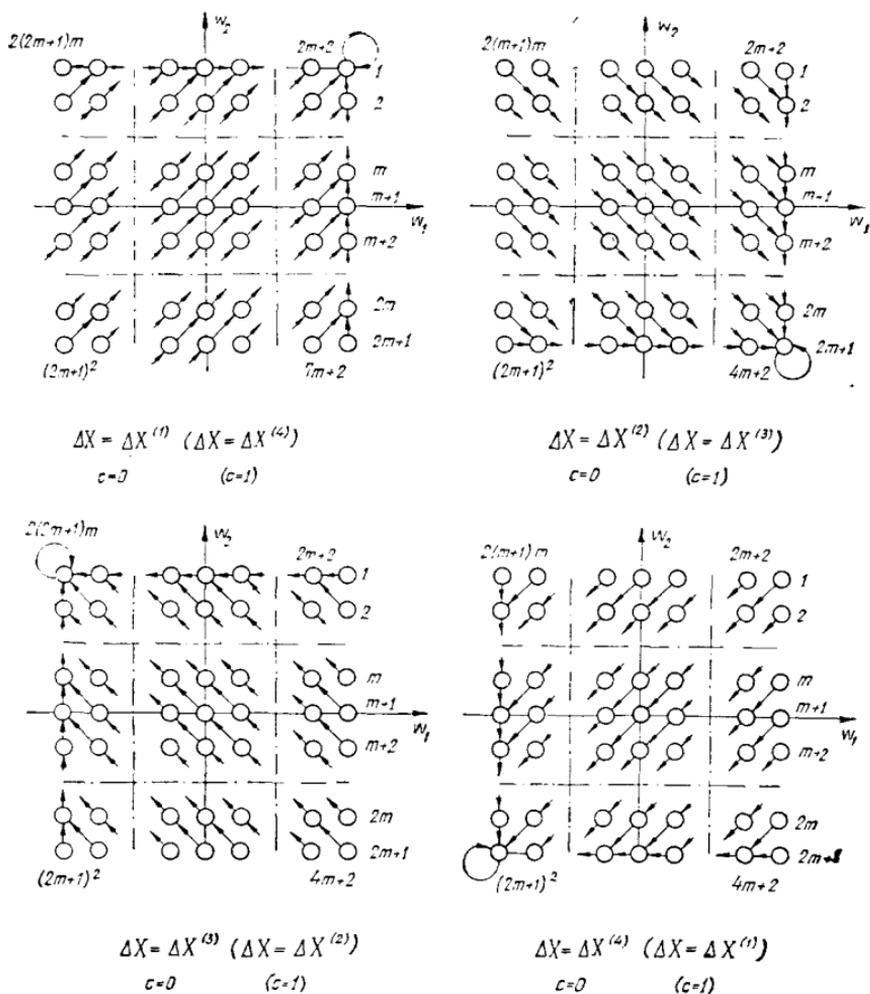


Рис. 2.1.6. Переходы автомата для $m > 2^n$ ($n=2$).

при штрафе

$$\begin{aligned} A_1(1) &= A_0(4); & A_1(2) &= A_0(3); \\ A_1(3) &= A_0(2); & A_1(4) &= A_0(1). \end{aligned} \quad (2.1.62)$$

В матрицах $A_c(j)$ ($c=0, 1; j=1, \dots, 4$) нулями обозначены матрицы порядка $(2k+1) \times (2k+1)$, элементы которых равны нулю; $P(j)$ ($j=1, 2, 3, 4$) — матрицы порядка $(2k+1) \times (2k+1)$ следующего вида:

$$P(1) = P(3) = \begin{vmatrix} 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \dots & 1 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 & 0 \end{vmatrix} \quad (2.1.63)$$

$$P(2) = P(4) = \begin{vmatrix} 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 \end{vmatrix} \quad (2.1.64)$$

В матрице $A_0(j)$ физический смысл блоков $P(j)$ таков, что он характеризует переход за один шаг из u -го столбца состояний на v -й столбец графа, изображенного на рис. 2.1.6. Если блок $P(j)$ равен нулю, это означает, что за один шаг невозможен переход из u -го столбца состояния на v -й столбец состояния.

Элементы i -й строки блока $P(j)$ являются вероятностями перехода из i -го состояния u -го столбца в состоянии, принадлежащие v -му столбцу.

Теперь построим матрицу A . Поскольку выполнены условия (2.1.62), то имеют место равенства (2.1.56) и матрицу A можно строить по формуле (2.1.57). Введем диагональные матрицы (2.1.65) порядка $(2k+1) \times (2k+1)$, элементы которых вычисляются по формуле (2.1.59).

$$\tilde{Q}_u(j) =$$

$$= \left\| \begin{array}{cccc} \tilde{q}_{(u-1)(2k+1)+1}(j) & 0 & 0 \dots & 0 \\ 0 & \tilde{q}_{(u-1)(2k+1)+2}(j) & 0 \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 0 \dots & \tilde{q}_{(u-1)(2k+1)+2k+1}(j) \end{array} \right\| \quad (2.1.65)$$

Затем, используя блоки $\tilde{Q}_u(j)$, матрицу $\tilde{Q}(j)$ перепишем в виде

$$\tilde{Q}(j) = \left\| \begin{array}{cccc} \tilde{Q}_1(j) & 0 & 0 \dots & 0 \\ 0 & \tilde{Q}_2(j) & 0 \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 0 \dots & \tilde{Q}_{2k+1}(j) \end{array} \right\| \quad (2.1.66)$$

По формуле (2.1.57), учитывая вид матрицы $A_0(j)$ и блочное представление матрицы $\tilde{Q}(j)$, находим матрицу

$$A = \left\| \begin{array}{ccccccccc} P_{11} & P_{12} & 0 & 0 & 0 \dots & 0 & 0 & 0 & 0 \\ P_{21} & 0 & P_{23} & 0 & 0 \dots & 0 & 0 & 0 & 0 \\ 0 & P_{32} & 0 & P_{34} & 0 \dots & 0 & 0 & 0 & 0 \\ \dots & \dots \\ 0 & 0 & 0 & 0 & 0 \dots & P_{2k,2k-1} & 0 & P_{2k,2k+1} & \\ 0 & 0 & 0 & 0 & 0 \dots & 0 & P_{2k+1,2k} & P_{2k+1,2k+1} & \end{array} \right\| \quad (2.1.67)$$

где блоки P_{uv} ($u, v = 1, \dots, 2k+1$) равны

$$P_{11} = \tilde{Q}_1(1)P(1) + \tilde{Q}_1(2)P(2);$$

$$P_{u,u-1} = \tilde{Q}_u(1)P(1) + \tilde{Q}_u(2)P(2) \quad (u=2, \dots, 2k+1); \quad (2.1.68)$$

$$P_{u,u+1} = \tilde{Q}_u(3)P(3) + \tilde{Q}_u(4)P(4) \quad (u=1, \dots, 2k);$$

$$P_{2k+1,2k+1} = \tilde{Q}_{2k+1}(3)P(3) + \tilde{Q}_{2k+1}(4)P(4).$$

Учитывая матрицы (2.1.63) и (2.1.64) имеем матрицы (2.1.69) — (2.1.72). Элементы этих матриц $q_{(u-1)(2k+1)+i}(j)$ ($i=1, \dots, 2k; j=1, \dots, 4$) определяются по формуле (2.1.59). Следовательно, матрицы $P_{u,u-1}$ имеют вид (2.1.73) и (2.1.74). Матрицы $P_{u,u+1}$ и $P_{2k+1,2k+1}$ можно

$$\begin{aligned} & \tilde{Q}_u(1)P(1) = \\ = & \left\| \begin{array}{cccccc} \tilde{q}_{(u-1)(2k+1)+1}(1) & 0 & \dots & 0 & 0 \\ \tilde{q}_{(u-1)(2k+1)+2}(1) & 0 & \dots & 0 & 0 \\ 0 & \tilde{q}_{(u-1)(2k+1)+3}(1) & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \tilde{q}_{(u-1)(2k+1)+2k+1}(1) & 0 \end{array} \right\| \end{aligned} \quad (2.1.69)$$

$$\begin{aligned} & \tilde{Q}_u(3)P(3) = \\ = & \left\| \begin{array}{cccccc} \tilde{q}_{(u-1)(2k+1)+1}(3) & 0 & 0 \dots & 0 & 0 \\ \tilde{q}_{(u-1)(2k+1)+2}(3) & 0 & 0 \dots & 0 & 0 \\ 0 & \tilde{q}_{(u-1)(2k+1)+3}(3) & 0 \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 \dots & \tilde{q}_{(u-1)(2k+1)+2k+1}(3) & 0 \end{array} \right\| \end{aligned} \quad (2.1.70)$$

$$\tilde{Q}_u(2)P(2) =$$

$$= \begin{vmatrix} 0 & \tilde{q}_{(u-1)(2k+1)+1}(2) & 0 & 0 \dots & 0 \\ 0 & 0 & \tilde{q}_{(u-1)(2k+1)+2}(2) & 0 \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 \dots & \tilde{q}_{(u-1)(2k+1)+2k}(2) \\ 0 & 0 & 0 & 0 \dots & \tilde{q}_{(u-1)(2k+1)+2k+1}(2) \end{vmatrix} \quad (2.1.71)$$

$$\tilde{Q}_u(4)P(4) =$$

$$= \begin{vmatrix} 0 & \tilde{q}_{(u-1)(2k+1)+1}(4) & 0 & 0 \dots & 0 \\ 0 & 0 & \tilde{q}_{(u-1)(2k+1)+2}(4) & 0 \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 \dots & \tilde{q}_{(u-1)(2k+1)+2k}(4) \\ 0 & 0 & 0 & 0 \dots & \tilde{q}_{(u-1)(2k+1)+2k+1}(4) \end{vmatrix} \quad (2.1.72)$$

$$P_{u,u-1} =$$

$$= \begin{vmatrix} \tilde{q}_{(u-1)(2k+1)+1}(1) & \tilde{q}_{(u-1)(2k+1)+1}(2) & 0 & 0 \\ \tilde{q}_{(u-1)(2k+1)+2}(1) & 0 & \tilde{q}_{(u-1)(2k+1)+2}(2) & 0 \\ 0 & \tilde{q}_{(u-1)(2k+1)+3}(1) & 0 & \tilde{q}_{(u-1)(2k+1)+3}(2) \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 \dots & 0 & 0 & 0 \\ 0 \dots & 0 & 0 & 0 \\ 0 \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots \\ 0 \dots & \tilde{q}_{(u-1)(2k+1)+2k}(1) & 0 & \tilde{q}_{(u-1)(2k+1)+2k}(2) \\ 0 \dots & 0 & \tilde{q}_{(u-1)(2k+1)+2k+1}(1) & \tilde{q}_{(u-1)(2k+1)+2k+1}(2) \end{vmatrix} \quad (2.1.73)$$

$$P_{11} = \begin{vmatrix} \tilde{q}_1(1) & \tilde{q}_1(2) & 0 & 0 & 0 \dots & 0 & 0 & 0 \\ \tilde{q}_2(1) & 0 & \tilde{q}_2(2) & 0 & 0 \dots & 0 & 0 & 0 \\ 0 & \tilde{q}_3(1) & 0 & \tilde{q}_3(2) & 0 \dots & 0 & 0 & 0 \\ \dots & \dots \\ 0 & 0 & 0 & 0 & 0 \dots & \tilde{q}_{2k}(1) & 0 & \tilde{q}_{2k}(2) \\ 0 & 0 & 0 & 0 & 0 \dots & 0 & \tilde{q}_{2k+1}(1) & \tilde{q}_{2k+1}(2) \end{vmatrix} \quad (2.1.74)$$

найти из матриц $P_{u, u-1}$ и P_{11} путем замены в последних элемента $q_{(u-1)(2k+1)+i}$ (1) элементом $q_{(u-1)(2k+1)+i}$ (3) и элемента $q_{(n-1)(2k+1)+i}$ (2) — элементом $q_{(u-1)(2k+1)+i}$ (4) ($i = 1, \dots, 2k+1$).

Далее рассмотрим тот вид самообучения, когда в формуле (2.1.3) переходная функция $\Psi = \Psi(\mathbf{W}_{N-1}, \Delta Q_{N-1}, \Delta \mathbf{X}_{N-1})$ не зависит от выхода автомата, т. е. когда

$$\Psi = \Psi(\mathbf{W}_{N-1}, \Delta Q'_{N-1}) \quad (2.1.75)$$

и $\Psi(\mathbf{W}_{N-1}, \Delta Q'_{N-1})$ — детерминированная функция. Такой алгоритм самообучения наиболее удобно задать при помощи формулы

$$\mathbf{W}_N = \begin{cases} \mathbf{W}_{N-1} - \delta \operatorname{sign} \Delta Q'_{N-1} \mathbf{I}, & \text{если } |\mathbf{W}_N| \leq d\sqrt{n}; \\ \frac{\mathbf{W}_N d\sqrt{n}}{|\mathbf{W}_N|}, & \text{если } |\mathbf{W}_N| > d\sqrt{n}, \end{cases} \quad (2.1.76)$$

где \mathbf{I} — n -мерный вектор с координатами $-1, 0$ или $+1$, соответствующий некоторому фиксированному направлению шага;

$$\mathbf{I} = (i_1, \dots, i_n). \quad (2.1.77)$$

В формуле (2.1.76) последующие значения вектора памяти \mathbf{W} зависят не от сделанного шага, а от полученного приращения $\Delta Q'_{N-1}$ оптимизируемой функции. Далее будет показана работоспособность алгоритма при решении задач оптимизации. Такой алгоритм самообучения представляет собой вероятностный автомат с детерминированными переходами и случайными выходами. Полное

функционирование такого автомата описывается цепью Маркова с матрицей переходных вероятностей (2.1.25), которую можно построить, если известны матрицы переходных вероятностей автомата при нештрафе A_0 и при штрафе A_1 и матрица вероятностей штрафов S^* автомата по отношению к его состояниям.

Построим матрицы переходных вероятностей A_0 и A_1 для этого автомата. Напишем формулу (2.1.76) отдельно для случаев

$$\text{sign } \Delta Q'_{N-1} = \pm 1, \quad (2.1.78)$$

т. е. при $\Delta Q' < 0$

$$\mathbf{W}_N = \begin{cases} \mathbf{W}_{N-1} + \delta \mathbf{I}^{(0)}, & \text{если } |\mathbf{W}_N| \leq d\sqrt{n}; \\ \tilde{\mathbf{W}}_N = \frac{\mathbf{W}_N d\sqrt{n}}{|\mathbf{W}_N|}, & \text{если } |\mathbf{W}_N| > d\sqrt{n}, \end{cases} \quad (2.1.79)$$

а при $\Delta Q' \geq 0$

$$\mathbf{W}_N = \begin{cases} \mathbf{W}_{N-1} - \delta \mathbf{I}^{(1)}, & \text{если } |\mathbf{W}_N| \leq d\sqrt{n}; \\ \tilde{\mathbf{W}}_N = \frac{\mathbf{W}_N d\sqrt{n}}{|\mathbf{W}_N|}, & \text{если } |\mathbf{W}_N| > d\sqrt{n}. \end{cases} \quad (2.1.80)$$

Здесь $\mathbf{I}^{(0)}$ и $\mathbf{I}^{(1)}$ — n -мерные векторы, координаты которых равны ± 1 , т. е. они направлены вдоль биссектрис пространственных углов, образованных координатными квадрантами.

Пусть

$$\mathbf{I}^{(0)} = (i_1^{(0)}, \dots, i_n^{(0)}); \quad (2.1.81)$$

$$\mathbf{I}^{(1)} = (i_1^{(1)} \dots i_n^{(1)}). \quad (2.1.82)$$

Рассмотрим случай, когда

$$\mathbf{I}^{(0)} = \mathbf{I}^{(1)} = (\text{sgn } \omega_1, \dots, \text{sgn } \omega_n). \quad (2.1.83)$$

Функция sgn определена выше (2.1.47). По формуле (2.1.79) получаем переходы автомата при нештрафе ($c=0$), а по формуле (2.1.80) — переходы при штрафе ($c=1$). Эти переходы представлены на рис. 2.1.7.

Матрицы A_0 и A_1 , характеризующие реакцию автомата соответственно при нештрафе и штрафе, имеют вид

$$A_0 = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad A_1 = \begin{vmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{vmatrix} \quad (2.1.84)$$

Для случая $m > 2^n$ переходы автомата, определяемые формулами (2.1.79) и (2.1.80), при учете формулы (2.1.83), показаны на рис. 2.1.8. Переходная матрица A_0 имеет блочный вид

$$A_0 = \begin{vmatrix} P_{11}^{(0)} & 0 & 0 \dots & 0 \\ P_{21}^{(0)} & 0 & 0 \dots & 0 \\ 0 & P_{32}^{(0)} & 0 \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 0 \dots & P_{2k-1,2k}^{(0)} \\ 0 & 0 & 0 \dots & P_{2k,2k}^{(0)} \end{vmatrix} \quad (2.1.85)$$

где нулями обозначены матрицы, элементы которых равны нулю; $P_{ij}^{(0)}$ — матрицы порядка $(2k) \times (2k)$, равные между собой:

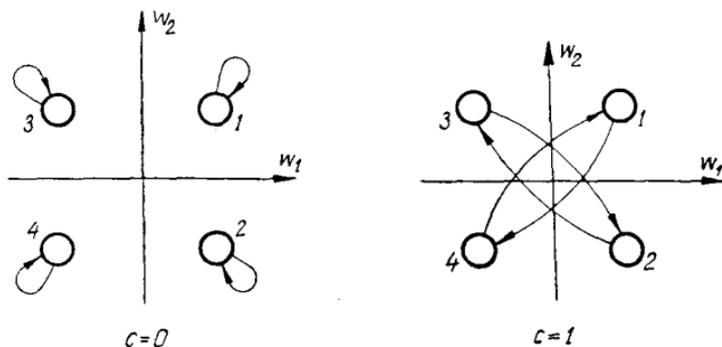


Рис. 2.1.7. Граф переходов автомата, переходы которого не зависят от выходной функции, для случая $m=2^n$ ($n=2$).

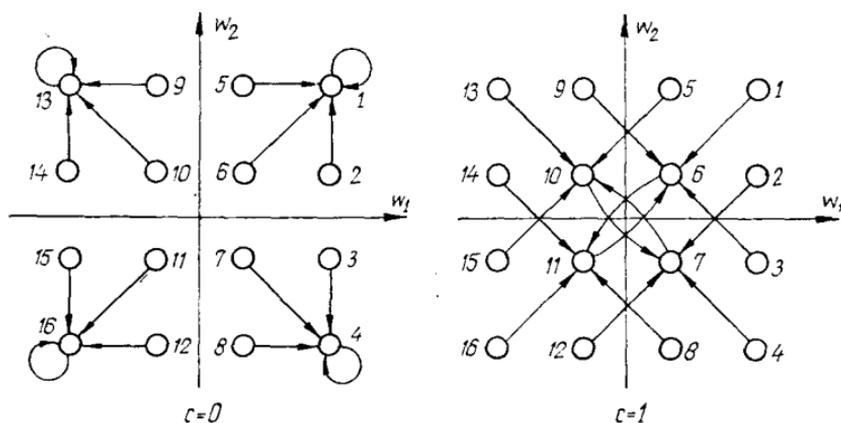


Рис. 2.1.8. Граф переходов автомата, переходы которого не зависят от выходной функции, для случая $m > 2^n$ ($n=2, m=16$).

$$P_{ij}^{(0)} = P^{(0)} = \underbrace{\begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & 0 & \dots & 0 & 1 \end{pmatrix}}_{2k} \quad (2.1.86)$$

Переходная матрица A_1 равна следующей блочной матрице:

$$A_1 = \begin{pmatrix} 0 & P_{12}^{(1)} & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & P_{23}^{(1)} & 0 & \dots & 0 & 0 & 0 \\ \cdot & \cdot \\ 0 & 0 & 0 & 0 & \dots & P_{2k-1, 2k-2}^{(1)} & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & P_{2k, 2k-1}^{(1)} & 0 \end{pmatrix} \quad (2.1.87)$$

где $P_{ij}^{(1)}$ — матрицы порядка $(2k) \times (2k)$, равные между собой:

$$P_{ij}^{(1)} = P^{(1)} = \left(\begin{array}{cccccccccccc} 0 & 1 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 1 & 0 & 0 \end{array} \right) \quad (2.1.88)$$

$(i=1, \dots, 2k; j=1, \dots, 2k).$

Функционирование автомата в случайной среде описывается цепью Маркова с матрицей переходных вероятностей вида

$$A = \left(\begin{array}{cccccccccccc} P_{11} & P_{12} & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 \\ P_{21} & 0 & P_{23} & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & P_{32} & 0 & P_{34} & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 \\ \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots & P_{2k-1,2k-2} & 0 & P_{2k-1,2k} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & P_{2k,2k-1} & P_{2k,2k} & 0 & 0 & 0 \end{array} \right) \quad (2.1.89)$$

Здесь P_{ij} равны матрицам

$$\begin{aligned} P_{11} &= (I - S^*_1)A_0; \dots; P_{i,i-1} = (I - S^*_i)A_0; \\ P_{i,i+1} &= S^*_i A_1; \dots; P_{2k,2k} = S^*_{2k} A_1, \end{aligned} \quad (2.1.90)$$

где

$$S^*_i = \left(\begin{array}{cccc} s^{((i-1)2k+1)} & 0 & 0 & \dots & 0 \\ 0 & s^{((i-1)2k+2)} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & s^{((i-1)2k+2l)} \end{array} \right)$$

Далее рассмотрим такой алгоритм самообучения, когда распределение выходов $\Delta \mathbf{X}^{(j)}$, определяемое формулой (2.1.2), представляет δ -функцию, т. е. когда каждому состоянию $\mathbf{W}^{(i)}$ автомата соответствует один определенный его выход $\Delta \mathbf{X}^{(j)}$ и функция (2.1.3), определяющая переходы внутренних состояний автомата, является случайной и не зависит от вектора $\Delta \mathbf{X}$:

$$p_N(\mathbf{W}) = p(\mathbf{W}_N / \mathbf{W}_{N-1}, \Delta Q'_{N-1}). \quad (2.1.91)$$

Эта зависимость реализуется заданием двух стохастических матриц A_0 и A_1 . Конкретный алгоритм самообучения, т. е. конкретный автомат, будет задан, если будут заданы конкретные матрицы A_0 и A_1 . Такой алгоритм самообучения представляет собой вероятностный автомат со случайными переходами и детерминированными выходами.

Пусть каждая координата ω_l ($l=1, \dots, n$) вектора памяти $\mathbf{W} = (\omega_1, \dots, \omega_n)$ может находиться в одном из $2k$ состояний:

$$\omega_l^{(j)} = d - (j-1)\delta \quad (2.1.92)$$

$$(j=1, \dots, 2k),$$

где $\omega_l^{(j)}$ — j -е состояние памяти по l -й координате.

Здесь

$$\delta = \frac{2d}{2k-1} \quad (2.1.93)$$

$$(d > 0)$$

d — число, задающее верхнюю и нижнюю границу изменения координат ω_l вектора \mathbf{W} ;

k — фиксированное натуральное число, характеризующее количество дискрет обучения по одной координате.

Тогда вектор памяти \mathbf{W} может находиться в одном из $(2k)^2$ состояний, т. е. множество внутренних состояний автомата содержит $(2k)^2$ элементов. Этот случай соответствует $n=2$.

Часто удобнее задавать не автомат, описывающий переходы вектора \mathbf{W} , а коллектив из n независимых автоматов, где каждый автомат коллектива описывает переходы по одной координате ω_l ($l=1, 2, \dots, n$) вектора \mathbf{W} [12—16]. В этом случае задаются переходные матрицы каждого автомата из коллектива. Потом по этим матрицам строятся матрицы переходных вероятностей всего коллектива автоматов, представляющего более сложный автомат и задающего переходы вектора \mathbf{W} из одних состояний в другие. Пусть стохастические матрицы порядка $(2k) \times (2k)$

$$R_0(l) = \begin{pmatrix} r_{11}^{(0)}(l) & \dots & r_{1,2k}^{(0)}(l) \\ \dots & \dots & \dots \\ r_{2k,1}^{(0)}(l) & \dots & r_{2k,2k}^{(0)}(l) \end{pmatrix} \quad (2.1.94)$$

$$R_1(l) = \begin{pmatrix} r_{11}^{(1)}(l) & \dots & r_{1,2k}^{(1)}(l) \\ \dots & \dots & \dots \\ r_{2k,1}^{(1)}(l) & \dots & r_{2k,2k}^{(1)}(l) \end{pmatrix} \quad (2.1.95)$$

$$(l=1, \dots, n)$$

являются переходными матрицами l -го автомата указанного коллектива ($R_0(l)$ — при нештрафе; $R_1(l)$ — при штрафе). Обозначим через $\mathbf{W}^{(i_1, \dots, i_l, \dots, i_n)}$ состояние вектора памяти \mathbf{W} , l -я координата ($l=1, \dots, n$) которого находится в состоянии i_l , а через $\mathbf{W}^{(j_1, \dots, j_l, \dots, j_n)}$ — состояние, l -я координата которого находится в состоянии j_l . Тогда вероятность перехода $\mathbf{W}^{(i_1, \dots, i_n)} \rightarrow \mathbf{W}^{(j_1, \dots, j_n)}$ будет определяться по формуле

$$P(\mathbf{W}^{(i_1, \dots, i_n)} \rightarrow \mathbf{W}^{(j_1, \dots, j_n)}) = P(\omega_1^{(i_1)} \rightarrow \omega_1^{(j_1)}, \dots, \omega_n^{(i_n)} \rightarrow \omega_n^{(j_n)}) = \prod_{l=1}^n r_{i_l j_l}(l) \quad (2.1.96)$$

$$(i_l, j_l = 1, \dots, 2l).$$

Рассмотрим более подробно случай, когда $n=2$. Из формулы (2.1.96) получаем при нештрафе ($\Delta Q' < 0$)

$$P(\mathbf{W}^{((\beta_1-1)2k+\gamma_1)} \rightarrow \mathbf{W}^{((\beta_2-1)2k+\gamma_2)/c=0}) = P(\omega_1^{(\beta_1)} \rightarrow \omega_1^{(\beta_2)});$$

$$\begin{aligned} \omega_2^{(\gamma_1)} \rightarrow \omega_2^{(\gamma_2)} / c = 0 &= p_{(\beta_1-1)2k+\gamma_1, (\beta_2-1)2k+\gamma_2}^{(0)} = \\ &= r_{\beta_1\beta_2}^{(0)} (1) r_{\gamma_1\gamma_2}^{(0)} (2), \end{aligned} \quad (2.1.97)$$

где

$$\beta_1 = 1, 2, \dots, 2k; \quad \beta_2 = 1, 2, \dots, 2k;$$

$$\gamma_1 = 1, 2, \dots, 2k; \quad \gamma_2 = 1, 2, \dots, 2k.$$

При штрафе ($\Delta Q' \geq 0$) имеем аналогичное выражение:

$$p_{(\beta_1-1)2k+\gamma_1, (\beta_2-1)2k+\gamma_2} = r_{\beta_1\beta_2}^{(1)} (1) r_{\gamma_1\gamma_2}^{(1)} (2). \quad (2.1.98)$$

Следовательно, для вектора \mathbf{W} имеем блочные переходные матрицы

$$A_0 = \left\| \begin{array}{ccc} P_{11}^{(0)} & \dots & P_{1,2k}^{(0)} \\ \dots & \dots & \dots \\ P_{2k,1}^{(0)} & \dots & P_{2k,2k}^{(0)} \end{array} \right\| \quad (2.1.99)$$

$$A_1 = \left\| \begin{array}{ccc} P_{11}^{(1)} & \dots & P_{1,2k}^{(1)} \\ \dots & \dots & \dots \\ P_{2k,1}^{(1)} & \dots & P_{2k,2k}^{(1)} \end{array} \right\| \quad (2.1.100)$$

где блоки $P_{ij}^{(0)}$ и $P_{ij}^{(1)}$ ($i, j = 1, \dots, 2k$) равны следующим матрицам порядка $(2k) \times (2k)$:

$$P_{ij}^{(0)} = r_{ij}^{(0)} (1) R_0 (2); \quad (2.1.101)$$

$$P_{ij}^{(1)} = r_{ij}^{(1)} (1) R_1 (2). \quad (2.1.102)$$

Здесь $R_0 (2)$ и $R_1 (2)$ — матрицы (2.1.94) и (2.1.95) при $l=2$; $r_{ij}^{(0)} (1)$ и $r_{ij}^{(1)} (1)$ — элементы тех же матриц при $l=1$.

В некоторых случаях коллектив состоит из одинаковых автоматов, которые переходят из данного состояния только в соседние состояния. При этом переходы автомата такие, что при штрафе ($\Delta Q' \geq 0$) автомат старается менять свое действие, т. е. с большей вероятностью переходит в ту сторону, где происходит смена его действия (выхода ΔX); при нештрафе ($\Delta Q^{(l)} < 0$) автомат старается

Эти матрицы должны быть стохастическими, т. е.

$$r_1 + r_2 = 1. \quad (2.1.105)$$

При этом предполагается, что имеет место неравенство

$$r_1 > r_2. \quad (2.1.106)$$

Из формул (2.1.102) и (2.1.101) и матриц (2.1.103) и (2.1.104) следует, что в этом случае матрицы A_0 и A_1 имеют блочный вид (см. матрицы (2.1.107) и (2.1.108)). В этих матрицах подматрицы $P_1^{(0)}$, $P_2^{(0)}$ и $P_1^{(1)}$, $P_2^{(1)}$ равны

$$P_1^{(0)} = r_1 R^{(0)}; \quad P_2^{(0)} = r_2 R^{(0)}; \quad (2.1.109)$$

$$P_1^{(1)} = r_1 R^{(1)}; \quad P_2^{(1)} = r_2 R^{(1)}, \quad (2.1.110)$$

$$A_0 = \begin{pmatrix} \overbrace{P_1^{(0)} & P_2^{(0)} & 0 & 0 \dots 0 & 0}^k & \\ P_1^{(0)} & 0 & P_2^{(0)} & 0 \dots 0 & 0 & \\ \dots & \dots & \dots & \dots & \dots & \\ 0 & 0 & 0 & 0 \dots P_1^{(0)} & 0 & \\ 0 & 0 & 0 & 0 \dots 0 & P_2^{(0)} & \\ \dots & \dots & \dots & \dots & \dots & \\ 0 & 0 & 0 & 0 \dots 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \dots 0 & 0 & 0 \end{pmatrix}$$

$$A_1 = \begin{pmatrix} \overbrace{P_2^{(1)} & P_1^{(1)} & 0 & 0 \dots 0 & 0}^k & \\ P_2^{(1)} & 0 & P_1^{(1)} & 0 \dots 0 & 0 & \\ \dots & \dots & \dots & \dots & \dots & \\ 0 & 0 & 0 & 0 \dots P_2^{(1)} & 0 & \\ 0 & 0 & 0 & 0 \dots 0 & P_1^{(1)} & \\ \dots & \dots & \dots & \dots & \dots & \\ 0 & 0 & 0 & 0 \dots 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \dots 0 & 0 & 0 \end{pmatrix}$$

где $R^{(0)}$ и $R^{(1)}$ — матрицы (2.1.103) и (2.1.104); r_1 и r_2 — отличные от нуля элементы этих матриц, удовлетворяющие условиям (2.1.105) и (2.1.106).

Таким образом, в двумерном случае ($n=2$) переходные матрицы автомата поиска выражаются через переходные матрицы одномерных автоматов поиска, умноженных на переходные вероятности этих автоматов. Аналогичным свойством обладают переходные матрицы автомата поиска в n -мерном случае, если этот автомат образован из коллектива одинаковых независимых автоматов, действующих по всем параметрам оптимизируемого объекта. Это обстоятельство облегчает исследование процесса оптимизации в многомерном пространстве.

$$\begin{pmatrix} \overbrace{0 & 0 & 0 \dots 0 & 0 & 0}^k & \\ 0 & 0 & 0 \dots 0 & 0 & 0 & \\ \dots & \dots & \dots & \dots & \dots & \\ P_2^{(0)} & 0 & 0 \dots 0 & 0 & 0 & 0 \\ 0 & P_1^{(0)} & 0 \dots 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \\ 0 & 0 & 0 \dots P_2^{(0)} & 0 & P_1^{(0)} & \\ 0 & 0 & 0 \dots 0 & P_2^{(0)} & P_1^{(0)} & \end{pmatrix} \quad (2.1.107)$$

$$\begin{pmatrix} \overbrace{0 & 0 & 0 \dots 0 & 0 & 0}^k & \\ 0 & 0 & 0 \dots 0 & 0 & 0 & \\ \dots & \dots & \dots & \dots & \dots & \\ P_1^{(1)} & 0 & 0 \dots 0 & 0 & 0 & 0 \\ 0 & P_2^{(1)} & 0 \dots 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \\ 0 & 0 & 0 \dots P_1^{(1)} & 0 & P_2^{(1)} & \\ 0 & 0 & 0 \dots 0 & P_1^{(1)} & P_2^{(1)} & \end{pmatrix} \quad (2.1.108)$$

Далее построим матрицы Q_{ij} и S^* для случая, когда выходы ΔX автомата детерминированные, т. е. когда любому состоянию $W^{(i)}$ автомата соответствует один определенный его выход $\Delta X^{(j)}$. Очевидно, что соответствующий состоянию $W^{(i)}$ выход $\Delta X^{(j)}$ будет зависеть от того, как пронумерованы состояния $W^{(i)}$ и выходы $\Delta X^{(i)}$. Рассмотрим для наглядности двумерный случай ($n=2$). Пусть координата вектора ΔX принимает значения $\Delta x_l = +1$ — когда координата w_l вектора W находится в состояниях от d до $\frac{\delta}{2}$, и $\Delta x_l = -1$ — когда координата w_l находится в состояниях от $-\frac{\delta}{2}$ до $-d$.

Представим номер состояния автомата i в виде формулы

$$i = (\beta - 1)2k + \gamma, \quad (2.1.111)$$

где γ и β принимают значения натуральных чисел от 1 до $2k$: $\gamma = 1, \dots, 2k$; $\beta = 1, \dots, 2k$. При такой записи номера состояния автомата значение переменной γ указывает в сетке состояний $W^{(i)}$ номер горизонтальной прямой, а значение переменной β указывает номер вертикальной прямой.

Пронумеруем выходы $\Delta X^{(j)}$ автомата (их всего 4 для $n=2$) в следующем порядке:

$$\begin{aligned} \Delta X^{(1)} &= (1, 1); \quad \Delta X^{(2)} = (1, -1); \quad \Delta X^{(3)} = (-1, 1); \\ \Delta X^{(4)} &= (-1, -1). \end{aligned} \quad (2.1.112)$$

Находим, при каких значениях вектора $W^{(i)}$ ($i = 1, \dots, (2k)^2$) вектор ΔX принимает указанные ниже значения. Если в формуле (2.1.111) i подсчитано

$$\text{при } \begin{cases} \gamma = 1, \dots, k; \\ \beta = 1, \dots, k, \end{cases} \text{ то } \Delta X = \Delta X^{(1)}; \quad (2.1.113)$$

$$\text{если при } \begin{cases} \gamma = k+1, \dots, 2k; \\ \beta = 1, \dots, k, \end{cases} \text{ то } \Delta X = \Delta X^{(2)}; \quad (2.1.114)$$

$$\text{если при } \begin{cases} \gamma = 1, \dots, k; \\ \beta = k+1, \dots, 2k, \end{cases} \text{ то } \Delta X = \Delta X^{(3)}; \quad (2.1.115)$$

если при $\begin{cases} \gamma = k+1, \dots, 2k; \\ \beta = k+1, \dots, 2k, \end{cases}$ то $\Delta \mathbf{X} = \Delta \mathbf{X}^{(4)}$. (2.1.116)

Теперь для рассматриваемого случая можно построить матрицу Q_{ij} (2.1.13). Используя формулы (2.1.113) — (2.1.116), находим, что она имеет следующий блочный вид:

$$Q_{ij} = \left(\begin{array}{c|c} \left. \begin{array}{c} Q_1 \\ \vdots \\ Q_1 \end{array} \right\}^k & \\ \left. \begin{array}{c} Q_2 \\ \vdots \\ Q_2 \end{array} \right\}^k & \end{array} \right) \quad (2.1.117)$$

где блоки Q_1 и Q_2 — прямоугольные матрицы $4 \times 2k$

$$Q_1 = \left(\begin{array}{c|c} \left. \begin{array}{cccc} 1 & 0 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot \end{array} \right\}^k & \\ \left. \begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 1 & 0 & 0 \end{array} \right\}^k & \end{array} \right) \quad Q_2 = \left(\begin{array}{c|c} \left. \begin{array}{cccc} 0 & 0 & 1 & 0 \\ \cdot & \cdot & \cdot & \cdot \end{array} \right\}^k & \\ \left. \begin{array}{cccc} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 1 \end{array} \right\}^k & \end{array} \right) \quad (2.1.118)$$

Матрица S^* штрафов автомата по состояниям $\mathbf{W}^{(i)}$ находится по формуле (2.1.27) с учетом формул (2.1.117) и (2.1.118):

$$S^* = \left(\begin{array}{c|c} \left. \begin{array}{cccccc} S^*_1 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & S^*_1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \cdot & \cdot \\ 0 & 0 & 0 & \dots & S^*_1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 & S^*_2 & 0 & \dots & 0 \\ \cdot & \cdot \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & S^*_2 \end{array} \right\}^k & \end{array} \right) \quad (2.1.119)$$

где блоки S^*_1 и S^*_2 являются диагональными матрицами

$$S^*_1 = \left(\begin{array}{cccc|cccc} s_1 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & s_1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \cdot & \cdot \\ 0 & 0 & 0 & \dots & s_1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 & s_2 & 0 & \dots & 0 \\ \cdot & \cdot \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & s_2 \end{array} \right) \left. \begin{array}{l} \vphantom{\left(} \right. \\ \vphantom{\left(} \right.} \end{array} \right\} \begin{array}{l} k \\ k \\ k \\ k \\ k \\ k \\ k \end{array} \right) \quad (2.1.120)$$

$$S^*_2 = \left(\begin{array}{cccc|cccc} s_3 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & s_3 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \cdot & \cdot \\ 0 & 0 & 0 & \dots & s_3 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 & s_4 & 0 & \dots & 0 \\ \cdot & \cdot \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & s_4 \end{array} \right) \left. \begin{array}{l} \vphantom{\left(} \right. \\ \vphantom{\left(} \right.} \end{array} \right\} \begin{array}{l} k \\ k \\ k \\ k \\ k \\ k \\ k \end{array} \right)$$

Здесь s_j ($j=1, \dots, 4$) — вероятности штрафов действия $\Delta X^{(j)}$ автомата, определяемые формулой (2.1.10).

§ 2.2. СВОЙСТВА ПОКООРИНАТНОГО САМООБУЧЕНИЯ ПРИ ОПТИМИЗАЦИИ В ОТСУТСТВИЕ ПОМЕХ

В этом параграфе исследуем алгоритм покоординатного самообучения при случайном поиске для случая, когда на оптимизируемую функцию не накладывается помеха ($\sigma=0$).

Покоординатное самообучение в процессе случайного поиска вводится в виде изменения вероятностных свойств двоичного выбора движения системы вдоль каждой из координат [9]. Пусть вероятность выбора шага вдоль по-

ложительного направления оси x_i на N -м шаге определяется кусочно-линейной функцией параметра w_i :

$$\bar{p}_i = \begin{cases} 0, & \text{если } w_i \leq -1; \\ \frac{1}{2} (1 + w_i), & \text{если } -1 \leq w_i \leq 1; \\ 1, & \text{если } 1 \leq w_i. \end{cases} \quad (2.2.1)$$

Для удобства введем n -мерное пространство $\{\mathbf{W}\}$, координаты которого образуются величинами w_i ($i=1, \dots, n$). В этом пространстве вектор памяти \mathbf{W} определяет направление преимущественного движения системы. В процессе одного шага поиска система смещается в пространстве параметров вдоль вектора

$$\Delta \mathbf{X} = \mathbf{X}_N - \mathbf{X}_{N-1}, \quad (2.2.2)$$

модуль которого предполагается постоянным и равным единице: $|\Delta \mathbf{X}| = 1$. Координаты этого вектора Δx_i ($i=1, \dots, n$) определяются следующим образом:

$$\Delta x_i = \begin{cases} \frac{1}{\sqrt{n}} \text{ с вероятностью } \bar{p}_i; \\ -\frac{1}{\sqrt{n}} \text{ с вероятностью } 1 - \bar{p}_i. \end{cases} \quad (2.2.3)$$

Будем рассматривать следующий алгоритм покоординатного самообучения для случая максимизации:

$$w_i^{(N+1)} = w_i^{(N)} + \delta \operatorname{sign} (\Delta Q_N \Delta x_i^{(N)}), \quad (2.2.4)$$

где δ — шаг изменения параметра w_i , определяющий интенсивность обучения ($\delta > 0$);

ΔQ_N — приращение функции качества на N -м шаге поиска; $\Delta Q_N = Q_N - Q_{N-1}$.

При самообучении возможно $w_i \gg 1$, что приведет к нежелательному «детерминированию» поиска. Поэтому имеет смысл изменение величин w_i ограничить следующими пределами:

$$w_i = \begin{cases} -d, & \text{если } w_i \leq -d; \\ w_i, & \text{если } -d < w_i < d; \\ d, & \text{если } d \leq w_i. \end{cases} \quad (2.2.5)$$

Изучение динамики такого самообучения возможно и целесообразно с привлечением теории марковских цепей, которые хорошо описывают процессы подобного рода. В процессе обучения вектор памяти \mathbf{W} переходит из одного состояния в другое в соответствии с алгоритмом обучения (2.2.4) и ограничениями (2.2.5). Пусть общее число возможных состояний этого вектора конечно и равно m . Пусть также вероятность перехода из i -го состояния в j -е определяется числом p_{ij} . Тогда изменение вектора \mathbf{W} в процессе самообучения

$$\rightarrow \mathbf{W}_{N-1} \rightarrow \mathbf{W}_N \rightarrow \mathbf{W}_{N+1} \rightarrow ,$$

как это будет показано далее, образует простую цепь Маркова. Будет также показано, что в линейном поле эта цепь будет однородной. (В нелинейном случае переходные вероятности зависят от положения системы, т. е. от времени, что вносит неоднородность.)

Таким образом, матрица $m \times m$ переходных вероятностей p_{ij} определяет процесс самообучения, и задача заключается в определении этих вероятностей с последующим привлечением общей теории марковских цепей.

1. *Случай $n=2$; $\delta \geq 2d$.* Исследуем сначала простейшую схему обучения в двумерном пространстве. В этом случае вектор $\mathbf{W} = (\omega_1, \omega_2)$ может находиться в одном из

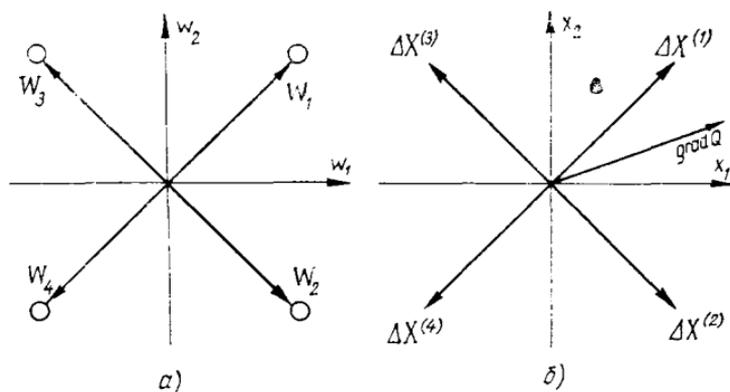


Рис. 2.2.1. Пространства $\{\mathbf{W}\}$ и $\{\mathbf{X}\}$ для $n=2$.

четырёх состояний, которые занумеруем в следующем порядке:

$$\begin{aligned} \mathbf{W}_1 &= (d, d); \quad \mathbf{W}_2 = (d, -d); \\ \mathbf{W}_3 &= (-d, d); \quad \mathbf{W}_4 = (-d, -d). \end{aligned} \tag{2.2.6}$$

Эти состояния вектора \mathbf{W} показаны на плоскости обучения на рис. 2.2.1, *а*.

На рис. 2.2.1, *б* показаны возможные направления движения оптимизируемой системы в двумерном пространстве параметров. Таких направлений также четыре. Выбор направления движения определяется двумя совместными событиями, а именно движением по обеим координатам. Поэтому вероятности выбора указанных направлений движения следующие:

$$\begin{aligned} P(\Delta\mathbf{X}^{(1)}) &= \bar{p}_1 \bar{p}_2; \\ P(\Delta\mathbf{X}^{(2)}) &= \bar{p}_1 (1 - \bar{p}_2); \\ P(\Delta\mathbf{X}^{(3)}) &= (1 - \bar{p}_1) \bar{p}_2; \\ P(\Delta\mathbf{X}^{(4)}) &= (1 - \bar{p}_1) (1 - \bar{p}_2), \end{aligned} \tag{2.2.7}$$

где \bar{p}_i ($i=1, 2$) определяются формулой (2.2.1).

Вероятности \bar{p}_1 и \bar{p}_2 зависят от состояния вектора \mathbf{W} , что и создает марковость процесса обучения.

Пусть вектор \mathbf{W} находится в состоянии \mathbf{W}_1 , т. е. $\mathbf{W} = (d, d)$. Тогда по формулам (2.2.7) и при учете ограничений (2.2.5) получаем следующие условные вероятности:

$$\begin{aligned} P(\Delta\mathbf{X}^{(1)}/\mathbf{W}_1) &= \frac{1}{4} (1+d) (1+d) = q_{11}; \\ P(\Delta\mathbf{X}^{(2)}/\mathbf{W}_1) &= \frac{1}{4} (1+d) (1-d) = q_{12}; \\ P(\Delta\mathbf{X}^{(3)}/\mathbf{W}_1) &= \frac{1}{4} (1-d) (1+d) = q_{13}; \\ P(\Delta\mathbf{X}^{(4)}/\mathbf{W}_1) &= \frac{1}{4} (1-d) (1-d) = q_{14}, \end{aligned} \tag{2.2.8}$$

где $P(\Delta\mathbf{X}^{(j)}/\mathbf{W}_i)$ обозначает вероятность выбора направления $\Delta\mathbf{X}^{(j)}$ при $\mathbf{W} = \mathbf{W}_i$.

Аналогично подсчитываем условные вероятности

$P(\Delta\mathbf{X}^{(j)}/\mathbf{W}_i)$ для других состояний вектора \mathbf{W} . Полученные значения показаны в табл. 2.2.1.

Таблица 2.2.1

$\mathbf{W}_i \backslash \Delta\mathbf{X}^{(j)}$	$\Delta\mathbf{X}^{(1)}$	$\Delta\mathbf{X}^{(2)}$	$\Delta\mathbf{X}^{(3)}$	$\Delta\mathbf{X}^{(4)}$
\mathbf{W}_1	$\frac{1}{4} (1+d)^2$	$\frac{1}{4} (1-d^2)$	$\frac{1}{4} (1-d^2)$	$\frac{1}{4} (1-d)^2$
\mathbf{W}_2	$\frac{1}{4} (1-d^2)$	$\frac{1}{4} (1+d)^2$	$\frac{1}{4} (1-d)^2$	$\frac{1}{4} (1-d^2)$
\mathbf{W}_3	$\frac{1}{4} (1-d^2)$	$\frac{1}{4} (1-d)^2$	$\frac{1}{4} (1+d)^2$	$\frac{1}{4} (1-d^2)$
\mathbf{W}_4	$\frac{1}{4} (1-d)^2$	$\frac{1}{4} (1-d^2)$	$\frac{1}{4} (1-d^2)$	$\frac{1}{4} (1+d)^2$

Теперь обратимся к алгоритму самообучения. Пусть для определенности градиентное направление расположено между направлениями $\Delta\mathbf{X}^{(1)}$ и $\Delta\mathbf{X}^{(2)}$ плоскости параметров (см. рис. 2.2.1, б). Тогда

$$\Delta Q^{(j)} = [\text{grad } Q, \Delta\mathbf{X}^{(j)}] \quad (2.2.9)$$

$$(j=1, \dots, 4);$$

$$\Delta Q^{(1)} > 0; \Delta Q^{(2)} > 0; \Delta Q^{(3)} < 0; \Delta Q^{(4)} < 0. \quad (2.2.10)$$

Переходим к автоматному представлению этого алгоритма самообучения, т. е. рассмотрим вектор \mathbf{W} как состояние автомата, а $\Delta\mathbf{X}$ — как его выход. Тогда таблица 2.2.1 представляет матрицу вероятностей выходов автомата Q_{ij} . Случайная среда, с которой взаимодействует автомат, задается модифицированной формулой (2.1.10) предыдущего параграфа:

$$s_j = \text{Вер} (\Delta Q \leq 0 / \Delta\mathbf{X} = \Delta\mathbf{X}^{(j)}) = \frac{1}{2} \left[1 - \Phi \left(\frac{\Delta Q^{(j)}}{2\sigma} \right) \right], \quad (2.2.11)$$

где приращения функции $\Delta Q^{(j)}$ определяются формулой (2.2.9).

При таком определении вероятности штрафа действия

автомата учитывается то, что происходит поиск максимума $Q(\mathbf{X})$. Предположим, что функция $Q(\mathbf{X})$ является линейной, т. е. приращения $\Delta Q^{(j)}$ ($j=1, 2, 3, 4$) не меняются в процессе поиска. В этом параграфе будем исходить из предположения, что на функцию $Q(\mathbf{X})$ не накладывается помеха ($\sigma=0$). Тогда по формуле (2.2.11) с учетом формул (2.2.10) имеем:

$$s_1=s_2=0; s_3=s_4=1. \tag{2.2.12}$$

Функционирование автомата в стационарной случайной среде (2.2.12) описывается цепью Маркова с матрицей переходных вероятностей (2.1.60). Подставляя в эту матрицу значения q_{ij} из таблицы 2.2.1 и значения s_j из формулы (2.2.12), получаем матрицу переходных вероятностей для вектора \mathbf{W} :

$$A(\mathbf{W}_j/\mathbf{W}_i) = \begin{vmatrix} \frac{1}{2}(1+d^2) & \frac{1}{2}(1-d^2) & 0 & 0 \\ \frac{1}{2}(1-d^2) & \frac{1}{2}(1+d^2) & 0 & 0 \\ \frac{1}{2}(1-d^2) & \frac{1}{2}(1+d^2) & 0 & 0 \\ \frac{1}{2}(1+d^2) & \frac{1}{2}(1-d^2) & 0 & 0 \end{vmatrix} \tag{2.2.13}$$

Возможные переходы вектора \mathbf{W} из одного состояния в другое показаны на рис. 2.2.2.

Из матрицы видно, что переходы вектора \mathbf{W} из одного состояния в другое образуют простую, однородную цепь Маркова. Состояния 3 и 4 являются несущественными; значение имеют только состояния 1 и 2, т. е. такие, первая координата которых положительна. Это вызвано тем, что в состояниях 1 и 2 приращение функции качества максимально (напомним, что рассматривается случай максимизации показателя качества объекта).

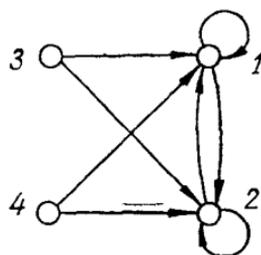


Рис. 2.2.2. Граф вероятностей перехода для вектора памяти (направление градиента расположено между биссектрисами первого и четвертого квадрантов).

Из матрицы (2.2.13), используя формулу Перрона [48] и учитывая таблицу 2.2.1, получаем матрицу переходных вероятностей за N шагов для вектора памяти \mathbf{W} :

$$A^N(\mathbf{W}_j/\mathbf{W}_i) = \begin{pmatrix} \frac{1}{2}(1+d^{2N}) & \frac{1}{2}(1-d^{2N}) & 0 & 0 \\ \frac{1}{2}(1-d^{2N}) & \frac{1}{2}(1+d^{2N}) & 0 & 0 \\ \frac{1}{2}(1-d^{2N}) & \frac{1}{2}(1+d^{2N}) & 0 & 0 \\ \frac{1}{2}(1+d^{2N}) & \frac{1}{2}(1-d^{2N}) & 0 & 0 \end{pmatrix} \quad (2.2.14)$$

Пусть $p_i^{(0)}$ обозначает вероятность того, что в начале поиска вектор \mathbf{W} находится в состоянии i , $p_i^{(N)}$ — вероятность того, что на N -м шаге поиска вектор \mathbf{W} находится в состоянии i , и p_i — вероятность того, что в пределе при $N \rightarrow \infty$ вектор \mathbf{W} будет принимать значение \mathbf{W}_i . Тогда по матрице (2.2.14) получаем:

$$\begin{aligned} p_1^{(N)} &= \frac{1}{2} + \frac{1}{2} d^{2N} (p_1^{(0)} + p_4^{(0)}) - \frac{1}{2} d^{2N} (p_2^{(0)} + p_3^{(0)}); \\ p_2^{(N)} &= \frac{1}{2} - \frac{1}{2} d^{2N} (p_1^{(0)} + p_4^{(0)}) + \frac{1}{2} d^{2N} (p_2^{(0)} + p_3^{(0)}); \\ p_3^{(N)} &= p_4^{(N)} = 0. \end{aligned} \quad (2.2.15)$$

Рассмотрим случай, когда выполняется условие $0 < d < 1$. Тогда из (2.2.15) при предельном переходе $N \rightarrow \infty$ получаем:

$$\begin{aligned} p_1 &= \lim_{N \rightarrow \infty} p_1^{(N)} = \frac{1}{2}; \\ p_2 &= \lim_{N \rightarrow \infty} p_2^{(N)} = \frac{1}{2}. \end{aligned} \quad (2.2.16)$$

Из выражения (2.2.16) следует, что в этом случае цепь Маркова является эргодической, т. е. предельное рас-

пределение вектора памяти \mathbf{W} не зависит от его начального распределения. Из этого же выражения видно, что в пределе при $N \rightarrow \infty$ вектор \mathbf{W} равновероятно пребывает в существенных состояниях 1 и 2.

Изложенный прием определения предельных вероятностей p_1, p_2 требует вычисления корней характеристического уравнения и миноров матрицы переходных вероятностей $A(\mathbf{W}_j/\mathbf{W}_i)$, что весьма сложно для матриц больших размерностей. Эти же предельные вероятности можно получить стандартным образом, решая систему линейных уравнений

$$\mathbf{P} = A' \mathbf{P}, \quad (2.2.17)$$

где A' — транспонированная матрица (2.2.13);

\mathbf{P} — вектор предельных вероятностей.

Определим теперь среднее приращение функции $Q(x_1, x_2)$ на N -м шаге поиска:

$$M_N[\Delta Q] = \sum_{i=1}^4 p_i^{(N)} M[\Delta Q/\mathbf{W}_i], \quad (2.2.18)$$

где $M[\Delta Q/\mathbf{W}_i]$ — среднее приращение функции качества при $\mathbf{W} = \mathbf{W}_i$; суммирование ведется по всем состояниям i . Среднее приращение $M[\Delta Q/\mathbf{W}_i]$ вычисляется по формуле

$$M[\Delta Q/\mathbf{W}_i] = \sum_{j=1}^4 P(\Delta \mathbf{X}^{(j)}/\mathbf{W}_i) \Delta Q^{(j)}, \quad (2.2.19)$$

где $P(\Delta \mathbf{X}^{(j)}/\mathbf{W}_i)$ определяется таблицей 2.2.1, $\Delta Q^{(j)}$ — формулой (2.2.9). Суммирование ведется по всем возможным направлениям движения оптимизируемой системы в плоскости параметров. Из формул (2.2.9) и (2.2.10) и таблицы 2.2.1 следует, что

$$M[\Delta Q/\mathbf{W}_i] = d \Delta Q^{(i)}. \quad (2.2.20)$$

Поэтому

$$\begin{aligned} M_N[\Delta Q] &= \frac{1}{\sqrt{2}} d \alpha_1 + \\ &+ \frac{1}{\sqrt{2}} d^{2N+1} \alpha_2 (p_1^{(0)} + p_4^{(0)} - p_2^{(0)} - p_3^{(0)}), \end{aligned} \quad (2.2.21)$$

где α_1 и α_2 — направляющие косинусы градиентного вектора; $\alpha_1^2 + \alpha_2^2 = 1$. Видно, что при больших N $M_N[\Delta Q]$ мало зависит от начальных вероятностей $p_i^{(0)}$. В предельном случае ($N \rightarrow \infty$) получаем

$$M[\Delta Q] = \lim_{N \rightarrow \infty} M_N[\Delta Q] = \frac{1}{\sqrt{2}} d \alpha_1. \quad (2.2.22)$$

Отметим, что максимальное значение $M[\Delta Q]$ принимает в случае, если $\alpha_1 = 1$, т. е. когда направление градиента совпадает с координатной осью x_1 . Минимальное значение $M[\Delta Q]$ принимает при $\alpha_1 \rightarrow \frac{1}{\sqrt{2}}$, т. е. когда направление градиента стремится к направлению $\Delta X^{(1)}$ или $\Delta X^{(2)}$. Тогда

$$M[\Delta Q] = \frac{1}{2} d. \quad (2.2.23)$$

Следовательно, среднее приращение функции качества в зависимости от различного направления градиента изменяется в пределах

$$\frac{1}{2} \leq \frac{M[\Delta Q]}{d} \leq \frac{1}{\sqrt{2}}. \quad (2.2.24)$$

Рассмотрим частный случай, когда $d=1$. Тогда из матрицы (2.2.14), подставляя $d=1$, получаем следующую матрицу переходных вероятностей за N шагов для вектора \mathbf{W} :

$$A^N(\mathbf{W}_j/\mathbf{W}_i) = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{vmatrix}. \quad (2.2.25)$$

Как видно из матрицы, вероятности перехода вектора \mathbf{W} из одного состояния в другое не зависят от номера шага N . Состояния 1 и 2 матрицы (2.2.25) являются поглощающими.

Подставляя в (2.2.15) $d=1$, получаем:

$$\begin{aligned} p_1^{(N)} &= p_1^{(0)} + p_4^{(0)}; \\ p_2^{(N)} &= p_2^{(0)} + p_3^{(0)}. \end{aligned} \quad (2.2.26)$$

В этом случае распределение вектора \mathbf{W} по существенным состояниям 1 и 2 не зависит от номера шага поиска N . Оно зависит лишь от начальных вероятностей $p_i^{(0)}$. В пределе при $N \rightarrow \infty$ получаем:

$$\begin{aligned} p_1 &= \lim_{N \rightarrow \infty} p_1^{(N)} = p_1^{(0)} + p_4^{(0)}; \\ p_2 &= \lim_{N \rightarrow \infty} p_2^{(N)} = p_2^{(0)} + p_3^{(0)}. \end{aligned} \quad (2.2.27)$$

В этом случае цепь Маркова является незергодической. Если $d=1$, то в соответствии с формулой (2.2.21) среднее приращение функции качества на N -м шаге поиска не зависит от N и равно

$$M[\Delta Q] = \frac{1}{\sqrt{2}} \alpha_1 + (p_1^{(0)} + p_4^{(0)} - p_2^{(0)} - p_3^{(0)}) \frac{1}{\sqrt{2}} \alpha_2. \quad (2.2.28)$$

Дальше рассмотрим особый случай, когда направление градиента оптимизируемой системы совпадает с одной из биссектрис, например с биссектрисой $\Delta \mathbf{X}^{(1)}$, т. е. когда $\alpha_1 = \alpha_2 = 1/\sqrt{2}$. Тогда по формуле (2.2.9) получаем:

$$\begin{aligned} \Delta Q^{(1)} &= |\mathbf{grad} Q|; \quad \Delta Q^{(2)} = \Delta Q^{(3)} = 0; \\ \Delta Q^{(4)} &= -|\mathbf{grad} Q|. \end{aligned} \quad (2.2.29)$$

Используя эти неравенства и формулу самообучения (2.2.4), запишем матрицу переходных вероятностей для вектора памяти \mathbf{W} в следующем виде:

$$A(\mathbf{W}_j/\mathbf{W}_i) = \begin{vmatrix} 1 & 0 & 0 & 0 \\ q_{12} + q_{13} & q_{11} + q_{14} & 0 & 0 \\ q_{12} + q_{13} & 0 & q_{11} + q_{14} & 0 \\ q_{11} + q_{14} & 0 & 0 & q_{12} + q_{13} \end{vmatrix} \quad (2.2.30)$$

Переходы из одного состояния в другое показаны на рис. 2.2.3 в виде графа.

Из матрицы (2.2.30), используя формулу Перрона, получаем матрицу переходных вероятностей за N шагов для вектора памяти \mathbf{W} :

$$A(W_j/W_i) = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 1 - (q_{11} + q_{14})^N & (q_{11} + q_{14})^N & 0 & 0 \\ 1 - (q_{11} + q_{14})^N & 0 & (q_{11} + q_{14})^N & 0 \\ 1 - (q_{12} + q_{13})^N & 0 & 0 & (q_{12} + q_{13})^N \end{vmatrix} \quad (2.2.31)$$

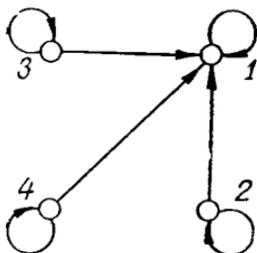


Рис. 2.2.3. Граф вероятностей перехода для вектора памяти (направление градиента совпадает с биссектрисой первого квадранта).

По этой матрице находим распределение вектора \mathbf{W} на N -м шаге поиска:

$$p_1^{(N)} = 1 - (q_{11} + q_{14})^N p_2^{(0)} - (q_{11} + q_{14})^N p_3^{(0)} - (q_{12} + q_{13})^N p_4^{(0)};$$

$$p_2^{(N)} = (q_{11} + q_{14})^N p_2^{(0)};$$

$$p_3^{(N)} = (q_{11} + q_{14})^N p_3^{(0)};$$

$$p_4^{(N)} = (q_{12} + q_{13})^N p_4^{(0)},$$

(2.2.32)

где

$$q_{11} + q_{14} = \frac{1}{2} (1 + d^2); \quad q_{12} + q_{13} = \frac{1}{2} (1 - d^2).$$

Рассмотрим случай, когда выполняется условие $0 < d < 1$. Тогда при предельном переходе $N \rightarrow \infty$ получаем

$$p_i = \lim_{N \rightarrow \infty} p_i^{(N)} = \begin{cases} 1 & \text{при } i = 1; \\ 0 & \text{при } i = 2, 3, 4. \end{cases} \quad (2.2.33)$$

В этом случае цепь Маркова является эргодической и в пределе вектор \mathbf{W} с вероятностью 1 принимает значение \mathbf{W}_1 , т. е. система обучается идеально. Среднее приращение функции качества на N -м шаге поиска равно

$$M_N[\Delta Q] = \left[1 - \left(\frac{1 + d^2}{2} \right)^N (p_2^{(0)} + p_3^{(0)}) - 2 \left(\frac{1 - d^2}{2} \right)^N p_4^{(0)} \right] d. \quad (2.2.34)$$

В пределе при $N \rightarrow \infty$ получим

$$M[\Delta Q] = \lim_{N \rightarrow \infty} M_N[\Delta Q] = d. \quad (2.2.35)$$

Из формулы (2.2.35) видно, что в случае, когда $\alpha_1 = 1/\sqrt{2}$, результат почти вдвое превышает приращение функции качества. Отметим, что такой случай маловероятен и поэтому результат (2.2.35) имеет узко локальный характер.

Дальше рассмотрим частный случай, когда $d=1$. Тогда из матрицы (2.2.31) получаем следующую матрицу переходных вероятностей за N шагов:

$$A^N(\mathbf{W}_j/\mathbf{W}_i) = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{vmatrix} \quad (2.2.36)$$

Из этой матрицы видно, что состояния 1, 2, 3 являются поглощающими, а состояние 4 — несущественным. В пределе при $N \rightarrow \infty$ получаем:

$$\begin{aligned} p_1 &= \lim_{N \rightarrow \infty} p_1^{(N)} = p_1^{(0)} + p_4^{(0)}; \\ p_2 &= \lim_{N \rightarrow \infty} p_2^{(N)} = p_2^{(0)}; \\ p_3 &= \lim_{N \rightarrow \infty} p_3^{(N)} = p_3^{(0)}; \\ p_4 &= \lim_{N \rightarrow \infty} p_4^{(N)} = 0. \end{aligned} \quad (2.2.37)$$

Таким образом, установлено, что для $d=1$ цепь Маркова также является незргодической.

2. *Случай $n=2$; $\delta = \frac{d}{k}$.* Выше рассмотрен простейший случай покоординатного обучения в процессе случайного поиска, когда имеется всего два уровня памяти для каждой координаты. Обычно количество уровней памяти больше двух. В этом случае вектор может находиться в одном из $(2k+1)^2$ состояний, где k — целое число. Эти состояния пронумерованы и показаны на рис. 2.2.4.

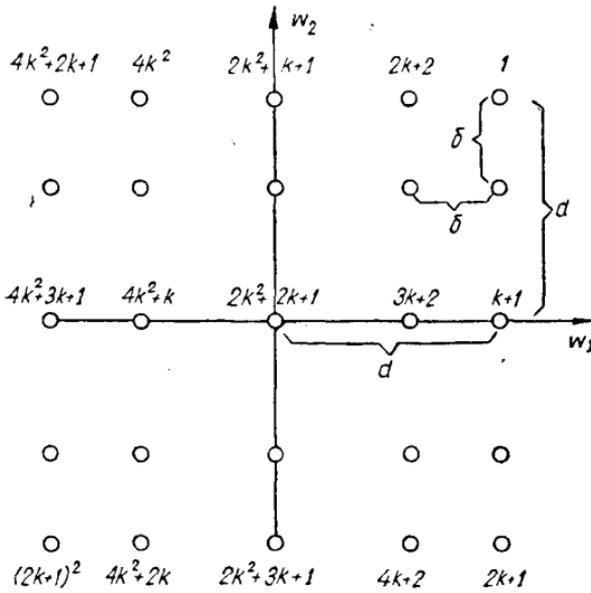


Рис. 2.2.4. Плоскость обучения $\left(\delta = \frac{d}{k}\right)$.

Пусть вектор градиента функции качества оптимизируемой системы удовлетворяет условию

$$\alpha_1 > \frac{1}{\sqrt{2}}, \tag{2.2.38}$$

где α_1 — направляющий косинус вектора градиента.

В этом случае имеем автомат с $(2k+1)^2$ состояниями и 4 выходами $\Delta X^{(j)}$. Функционирование такого автомата (переходы вектора \mathbf{W} из одного состояния в другое) описывается цепью Маркова с блочной матрицей (2.1.67) из предыдущего параграфа. Блоки $P_{u,v}$ этой матрицы равны матрицам (2.1.73) и (2.1.74), элементы которых определяются формулой (2.1.59). Так как на функцию $Q(\mathbf{X})$ не накладывается помеха ($\sigma=0$), то имеют место равенства (2.2.12) и из формул (2.1.59) и (2.1.68) получаем, что в матрице A блоки $P_{u,u+1}$ и $P_{2k+1,2k+1}$ равны нулю, т. е.

$$P_{u,u+1} = 0 \quad (u = 1, 2, \dots, 2k); \quad (2.2.39)$$

$$P_{2k+1,2k+1} = 0.$$

Следовательно, матрица A принимает вид

$$A(W_j/W_i) = \begin{vmatrix} P_{11} & 0 & 0 \dots & 0 & 0 & 0 & 0 \\ P_{21} & 0 & 0 \dots & 0 & 0 & 0 & 0 \\ 0 & P_{32} & 0 \dots & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 \dots & P_{2k,2k-1} & 0 & 0 & 0 \\ 0 & 0 & 0 \dots & 0 & P_{2k+1,2k} & 0 & 0 \end{vmatrix} \quad (2.2.40)$$

В этой матрице нулями обозначены блоки, элементы которых равны нулю, $P_{u,v}$ являются квадратными матрицами Якоби [49] — частными случаями матриц (2.1.73) и (2.1.74). Так, например, для $i=j=1$ имеем матрицу

$$P_{11} = \begin{vmatrix} p_{11} & p_{12} & 0 & 0 & 0 \dots & 0 & 0 & 0 & 0 \\ p_{21} & 0 & p_{23} & 0 & 0 \dots & 0 & 0 & 0 & 0 \\ 0 & p_{32} & 0 & p_{34} & 0 \dots & 0 & 0 & 0 & 0 \\ \dots & \dots \\ 0 & 0 & 0 & 0 & 0 \dots & p_{2k-1,2k-2} & 0 & p_{2k-1,2k} & 0 \\ 0 & 0 & 0 & 0 & 0 \dots & 0 & p_{2k,2k-1} & 0 & p_{2k,2k+1} \\ 0 & 0 & 0 & 0 & 0 \dots & 0 & 0 & p_{2k+1,2k} & p_{2k+1,2k+1} \end{vmatrix} \quad (2.2.41)$$

Элементы p_{ij} последней матрицы с учетом формулы (2.1.59) определяются из таблицы 2.2.2 условных вероятностей выбора направлений движения оптимизируемой системы в плоскости параметров. Элементы таблицы вычислены по формулам (2.2.7). При помощи таблицы можно показать, что элементы матрицы (2.2.41) удовлетворяют соотношениям

$W_i \backslash \Delta X^{(j)}$	$\Delta X^{(1)}$	$\Delta X^{(2)}$
W_1	$\frac{1}{4} (1+k\delta) (1+k\delta)$	$\frac{1}{4} (1+k\delta) (1-k\delta)$
W_2	$\frac{1}{4} (1+k\delta) (1+(k-1)\delta)$	$\frac{1}{4} (1+k\delta) (1-(k-1)\delta)$
...
W_k	$\frac{1}{4} (1+k\delta) (1+\delta)$	$\frac{1}{4} (1+k\delta) (1-\delta)$
W_{k+1}	$\frac{1}{4} (1+k\delta)$	$\frac{1}{4} (1+k\delta)$
W_{k+2}	$\frac{1}{4} (1+k\delta) (1-\delta)$	$\frac{1}{4} (1+k\delta) (1+\delta)$
...
W_{2k}	$\frac{1}{4} (1-k\delta) (1-(k-1)\delta)$	$\frac{1}{4} (1+k\delta) (1+(k-1)\delta)$
W_{2k+1}	$\frac{1}{4} (1+k\delta) (1-k\delta)$	$\frac{1}{4} (1+k\delta) (1+k\delta)$

$$\begin{aligned}
 p_{11} &= p_{2k+1,2k+1}; \quad p_{12} = p_{2k+1,2k}; \quad p_{23} = p_{2k,2k-1}; \quad p_{34} = \\
 &= p_{2k-1,2k-2}; \quad \dots; \quad p_{k,k+1} = p_{k+2,k+1}; \quad p_{k+1,k+2} = p_{k+1,k}; \\
 & p_{k+2,k+3} = p_{k,k-1}; \quad \dots; \quad p_{2k-1,2k} = p_{32}; \quad p_{2k,2k+1} = p_{21}.
 \end{aligned}$$

(2.2.42)

Построенные по матрицам (2.2.40) и (2.2.41) переходы вектора W из одного состояния в другое показаны на рис. 2.2.5. Из рисунка видно, что вектор памяти W за $(2k+1)$ шагов, независимо от того, в каком состоянии он находился в начале поиска, достигает правого ребра сетки. Потом происходит блуждание по состояниям

Таблица 2.2.2

$\Delta X^{(3)}$	$\Delta X^{(4)}$
$\frac{1}{4} (1-k\delta) (1+k\delta)$	$\frac{1}{4} (1-k\delta) (1-k\delta)$
$\frac{1}{4} (1-k\delta) (1+(k-1)\delta)$	$\frac{1}{4} (1-k\delta) (1-(k-1)\delta)$
.....
$\frac{1}{4} (1-k\delta) (1+\delta)$	$\frac{1}{4} (1-k\delta) (1-\delta)$
$\frac{1}{4} (1-k\delta)$	$\frac{1}{4} (1-k\delta)$
$\frac{1}{4} (1-k\delta) (1-\delta)$	$\frac{1}{4} (1-k\delta) (1+\delta)$
.....
$\frac{1}{4} (1-k\delta) (1-(k-1)\delta)$	$\frac{1}{4} (1-k\delta) (1+(k-1)\delta)$
$\frac{1}{4} (1-k\delta) (1-k\delta)$	$\frac{1}{4} (1-k\delta) (1+k\delta)$

этого ребра. Поэтому основной интерес представляет анализ переходных вероятностей за такое число шагов N , которое превосходит $2k+1$ ($N \geq 2k+1$). Для таких N из матрицы (2.2.40) получаем следующую блочную матрицу переходных вероятностей за N шагов для вектора памяти \mathbf{W} :

$$A^N(\mathbf{W}_j/\mathbf{W}_i) = \begin{vmatrix} P_{11}^{(N)} & & 0 & \dots & 0 \\ P_{21} & P_{11}^{N-1} & & 0 & \dots & 0 \\ P_{32} & P_{21} & P_{11}^{N-2} & & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ P_{2k+1,2k} & \dots & P_{21} & P_{11} & 0 & \dots & 0 \end{vmatrix} \quad (2.2.43)$$

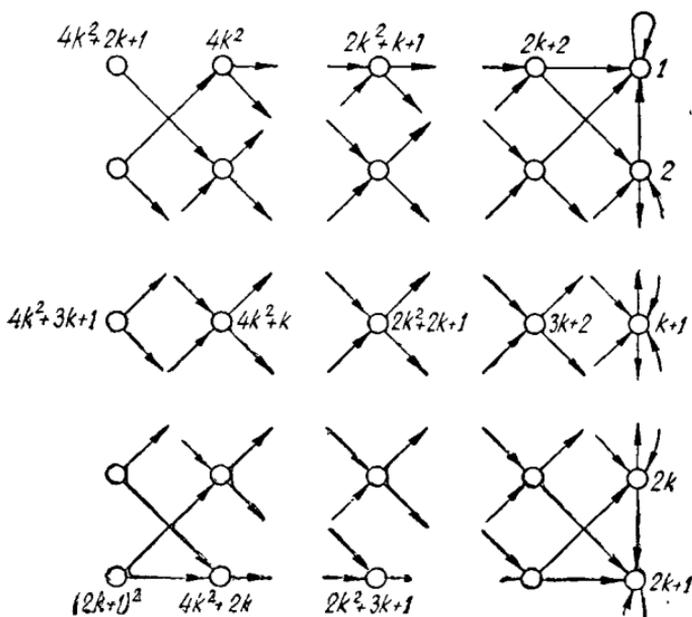


Рис. 2.2.5. Возможные переходы вектора \mathbf{W} на плоскости обучения $\left(\delta = \frac{d}{k}\right)$.

где нулями обозначены блоки, элементы которых равны нулю; $P_{i+1,i} \dots P_{21} P_{11}^{N-i}$ ($i=0, \dots, 2k$) — произведение матриц; P_{11}^{N-i} — $(N-i)$ -я степень матрицы P_{11} ; эту степень можно определить, используя формулу Перрона.

Из матрицы (2.2.43) видно, что состояния от $2k+2$ до $(2k+1)^2$ являются несущественными. Поэтому

$$p_{2k+2} = \dots = p_{(2k+1)^2} = 0. \quad (2.2.44)$$

Теперь находим предельные вероятности p_i . Из матрицы (2.2.40), учитывая (2.2.44), получаем

$$\mathbf{P} = P'_{11} \mathbf{P}, \quad (2.2.45)$$

где P'_{11} — транспонированная матрица (2.2.41);

\mathbf{P} — вектор предельных вероятностей.

Решая систему (2.2.45), получаем:

$$p_{2k+1} = p_1;$$

$$p_{2k} = p_2 = \frac{p_{12}}{p_{21}} p_1;$$

$$p_{2k-1} = p_3 = \frac{p_{23}}{p_{32}} \cdot \frac{p_{12}}{p_{21}} p_1;$$

(2.2.46)

.....

$$p_{k+2} = p_k = \prod_{j=2}^k \frac{p_{j-1,j}}{p_{j,j-1}} p_1;$$

$$p_{k+1} = \prod_{j=2}^{k+1} \frac{p_{j-1,j}}{p_{j,j-1}} p_1,$$

где

$$p_1 = \left(2 + 2 \sum_{m=2}^k \prod_{j=2}^m \frac{p_{j-1,j}}{p_{j,j-1}} + \prod_{j=2}^{k+1} \frac{p_{j-1,j}}{p_{j,j-1}} \right)^{-1}. \quad (2.2.47)$$

Эти предельные вероятности определяют дискретное распределение блуждания системы в процессе поиска вдоль правого ребра квадрата на рис. 2.2.5. В качестве

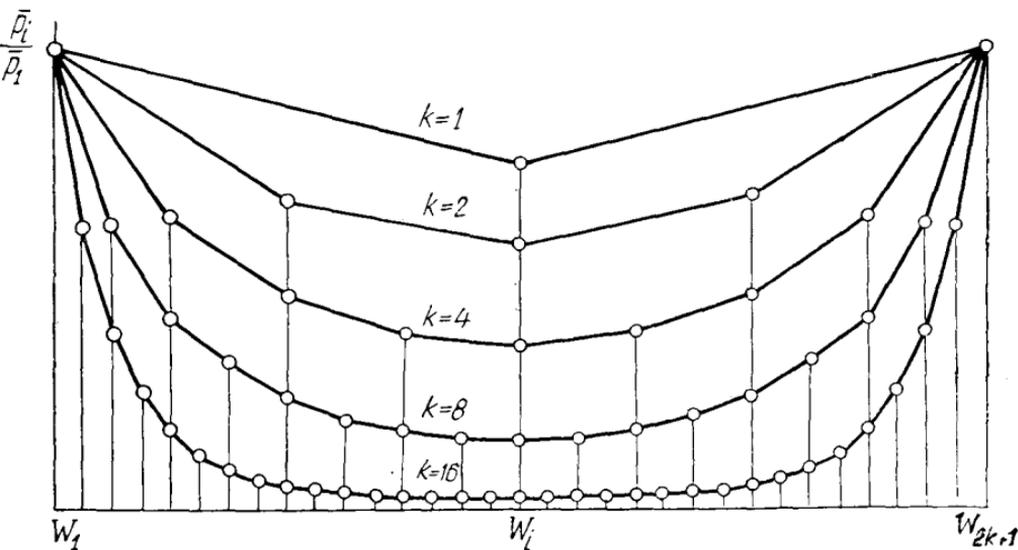


Рис. 2.2.6. Предельное распределение вектора W по существенным состояниям для $k=1; 2; 4; 8; 16$.

примера на рис. 2.2.6 показано такое дискретное распределение, рассчитанное по формулам (2.2.46) для случая $d = \frac{1}{2}$; $k = 1; 2; 4; 8; 16$. Характерно относительное увеличение вероятности на концах ребра по сравнению с его серединой. Это, по-видимому, является результатом по-координатного характера обучения, в процессе которого система движется вдоль диагоналей.

Далее определим среднее приращение функции качества при $\mathbf{W} = \mathbf{W}_i$:

$$M[\Delta Q/\mathbf{W}_i] = \sum_{j=1}^4 P(\Delta X^{(j)}/\mathbf{W}_i) \Delta Q^{(j)} = [P(\Delta X^{(1)}/\mathbf{W}_i) - P(\Delta X^{(4)}/\mathbf{W}_i)] \Delta Q^{(1)} + [P(\Delta X^{(2)}/\mathbf{W}_i) - P(\Delta X^{(3)}/\mathbf{W}_i)] \Delta Q^{(2)}, \quad (2.2.48)$$

где $\Delta Q^{(j)}$ определены формулой (2.2.10). По формулам, приведенным в таблице 2.2.2, получаем:

$$P(\Delta X^{(1)}/\mathbf{W}_i) - P(\Delta X^{(4)}/\mathbf{W}_i) = \frac{1}{2} (2k + 1 - i) \delta; \quad (2.2.49)$$

$$P(\Delta X^{(2)}/\mathbf{W}_i) - P(\Delta X^{(3)}/\mathbf{W}_i) = \frac{1}{2} (i - 1) \delta.$$

Поэтому

$$M[\Delta Q/\mathbf{W}_i] = \frac{1}{2} [(2k + 1 - i) \delta \Delta Q^{(1)} + (i - 1) \delta \Delta Q^{(2)}]. \quad (2.2.50)$$

Усредняя это выражение по всем возможным значениям i , в предельном случае $N \rightarrow \infty$ получаем

$$M[\Delta Q] = \frac{1}{\sqrt{2}} d \alpha_1, \quad (2.2.51)$$

что совпадает с результатом, полученным для случая $\delta \geq 2d$ (2.2.22). Это доказывает, что при статистической оптимизации с обучением в двумерном пространстве параметров в случае линейной функции качества оптимизируемой системы скорость оптимизации при $N \rightarrow \infty$ не

§ 2.3. СВОЙСТВА ПОКООРИНАТНОГО САМООБУЧЕНИЯ ПРИ СТАТИСТИЧЕСКОЙ ОПТИМИЗАЦИИ С ПОМЕХАМИ

Пусть на функцию качества оптимизируемой системы $Q(x_1, \dots, x_n)$ накладывается помеха $\varepsilon(\sigma)$, которая является нормально распределенной случайной величиной с математическим ожиданием, равным нулю, и дисперсией $D = \sigma^2$. Тогда оптимизация системы проводится по следующей величине:

$$Q'(x_1, \dots, x_n) = Q(x_1, \dots, x_n) + \varepsilon(\sigma). \quad (2.3.1)$$

Алгоритм покоординатного самообучения можно представить в виде

$$\omega_i^{(N+1)} = \omega_i^{(N)} + \delta \operatorname{sign}(\Delta Q'_N \Delta x_i^{(N)}), \quad (2.3.2)$$

где δ — шаг изменения параметра ω_i , определяющий интенсивность обучения ($\delta > 0$);

$\Delta Q'_N$ — приращение функции $Q'(x_1, \dots, x_n)$ на N -м шаге поиска; $\Delta Q'_N = Q'_N - Q'_{N-1}$.

Как и в предыдущем параграфе, полагаем $|\Delta X| = 1$, а на изменение величин ω_i накладываем обычное ограничение $|\omega_i| < d$ ($i = 1, \dots, n$).

1. *Двумерный случай* ($n = 2$; $\delta \geq 2d$). Плоскость параметров оптимизируемой системы и плоскость обучения имеют тот же вид, что и в предыдущем параграфе (см. рис. 2.2.1). Из формулы (2.3.1) следует, что для единичного шага вдоль направления $\Delta X^{(j)}$ имеем приращение функции качества в виде

$$\Delta Q'_j = \Delta Q^{(j)} + \varepsilon(\sqrt{2}\sigma) = (\Delta X^{(j)}, \operatorname{grad} Q) + \varepsilon(\sqrt{2}\sigma), \quad (2.3.3)$$

где $\varepsilon(\sigma\sqrt{2})$ — нормально распределенная случайная величина с математическим ожиданием, равным нулю, и дисперсией $2\sigma^2$.

Для линейной функции $Q(\mathbf{X})$, вектор градиента которой равен

$$\operatorname{grad} Q(\mathbf{X}) = (\alpha_1, \alpha_2) \quad (2.3.4)$$

$$(\alpha_1^2 + \alpha_2^2 = 1),$$

получаем следующие выражения для приращения функции качества по направлениям $\Delta X^{(j)}$:

$$\begin{aligned} \Delta Q^{(1)} &= \alpha_1 + \alpha_2; \quad \Delta Q^{(2)} = \alpha_1 - \alpha_2; \\ \Delta Q^{(3)} &= -(\alpha_1 - \alpha_2); \quad \Delta Q^{(4)} = -(\alpha_1 + \alpha_2). \end{aligned} \quad (2.3.5)$$

Алгоритм покоординатного самообучения представляет собой автомат, функционирование которого описывается цепью Маркова с матрицей переходных вероятностей (2.1.60). Вероятности q_{ij} ($i, j=1, \dots, n$) этой матрицы задаются таблицей 2.2.1, а вероятности штрафов s_j — формулой (2.2.11).

Используя равенства (2.3.5), находим

$$s_4 = 1 - s_1; \quad s_3 = 1 - s_2. \quad (2.3.6)$$

Вводим обозначения

$$\Phi_i = 1 - s_i = \frac{1}{2} \left[1 + \Phi \left(\frac{\Delta Q^{(i)}}{2\sigma} \right) \right]; \quad (2.3.7)$$

$$A_1 = q_{11} + q_{14} = \frac{1}{2} (1 + d^2); \quad (2.3.8)$$

$$A_2 = q_{12} + q_{13} = \frac{1}{2} (1 - d^2),$$

где $\Phi(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt$ — интеграл вероятности. Тогда из матрицы (2.1.60), учитывая формулы (2.3.6) — (2.3.8), получаем следующую матрицу переходных вероятностей для вектора \mathbf{W} [49]:

$$A = \begin{vmatrix} A_1\Phi_1 & A_2\Phi_2 & A_2\Phi_3 & A_1\Phi_4 \\ A_2\Phi_1 & A_1\Phi_2 & A_1\Phi_3 & A_2\Phi_4 \\ A_2\Phi_1 & A_1\Phi_2 & A_1\Phi_3 & A_2\Phi_4 \\ A_1\Phi_1 & A_2\Phi_2 & A_2\Phi_3 & A_1\Phi_4 \end{vmatrix} \quad (2.3.9)$$

Видно, что при $d \neq 1$ и $\sigma \neq 0$ из любого состояния возможен переход в любое другое состояние, т. е. цепь Маркова является неприводимой. Из матрицы (2.3.9), используя формулу Перрона, получаем матрицу переходных вероятностей за N шагов для вектора \mathbf{W} :

$$\begin{aligned}
 A^N = & \left(\begin{array}{cccc}
 \frac{1}{2} \Phi_1 (1 + d^{2N}) \frac{1}{2} \Phi_2 (1 - d^{2N}) \frac{1}{2} \Phi_3 (1 - d^{2N}) \frac{1}{2} \Phi_4 (1 + d^{2N}) \\
 \frac{1}{2} \Phi_1 (1 - d^{2N}) \frac{1}{2} \Phi_2 (1 + d^{2N}) \frac{1}{2} \Phi_3 (1 + d^{2N}) \frac{1}{2} \Phi_4 (1 - d^{2N}) \\
 \frac{1}{2} \Phi_1 (1 - d^{2N}) \frac{1}{2} \Phi_2 (1 + d^{2N}) \frac{1}{2} \Phi_3 (1 + d^{2N}) \frac{1}{2} \Phi_4 (1 - d^{2N}) \\
 \frac{1}{2} \Phi_1 (1 + d^{2N}) \frac{1}{2} \Phi_2 (1 - d^{2N}) \frac{1}{2} \Phi_3 (1 - d^{2N}) \frac{1}{2} \Phi_4 (1 + d^{2N})
 \end{array} \right) \\
 & \qquad \qquad \qquad (2.3.10)
 \end{aligned}$$

Пусть $p_i^{(0)}$ обозначает вероятность того, что в начале поиска вектор \mathbf{W} находится в состоянии i ; p_i^N — вероятность того, что на N -м шаге поиска вектор \mathbf{W} находится в состоянии i , и p_i — вероятность того, что в пределе при $N \rightarrow \infty$ вектор \mathbf{W} будет принимать значение \mathbf{W}_i . Тогда по матрице (2.3.10) имеем:

$$\begin{aligned}
 p_1^{(N)} &= \frac{1}{2} \Phi_1 + \frac{1}{2} \Phi_1 d^{2N} (p_1^{(0)} + p_4^{(0)}) - \\
 &\quad - \frac{1}{2} \Phi_1 d^{2N} (p_2^{(0)} + p_3^{(0)}); \\
 p_2^{(N)} &= \frac{1}{2} \Phi_2 - \frac{1}{2} \Phi_2 d^{2N} (p_1^{(0)} + p_4^{(0)}) + \\
 &\quad + \frac{1}{2} \Phi_2 d^{2N} (p_2^{(0)} + p_3^{(0)}); \\
 p_3^{(N)} &= \frac{1}{2} \Phi_3 - \frac{1}{2} \Phi_3 d^{2N} (p_1^{(0)} + p_4^{(0)}) + \\
 &\quad + \frac{1}{2} \Phi_3 d^{2N} (p_2^{(0)} + p_3^{(0)}); \\
 p_4^{(N)} &= \frac{1}{2} \Phi_4 + \frac{1}{2} \Phi_4 d^{2N} (p_1^{(0)} + p_4^{(0)}) - \\
 &\quad - \frac{1}{2} \Phi_4 d^{2N} (p_2^{(0)} + p_3^{(0)}).
 \end{aligned}
 \tag{2.3.11}$$

Рассмотрим случай, когда выполняется условие $0 < d < 1$. Тогда из (2.3.11) при предельном переходе $N \rightarrow \infty$ получаем

$$p_i = \frac{1}{2} \Phi_i \quad (i=1, \dots, 4). \quad (2.3.12)$$

В этом случае цепь Маркова является эргодической, т. е. предельное распределение вектора \mathbf{W} не зависит от его начального распределения. Оно зависит от величины поехи σ . Из выражения (2.3.12), учитывая (2.3.7), находим

$$\lim_{\sigma \rightarrow \infty} p_i = \frac{1}{4} \quad \text{при } i=1, \dots, 4; \quad (2.3.13)$$

$$\lim_{\sigma \rightarrow 0} p_i = \begin{cases} \frac{1}{2} & \text{при } i=1, 2; \\ 0 & \text{при } i=3, 4. \end{cases}$$

Из формул (2.3.13) видно, что при $\sigma \rightarrow \infty$ вектор \mathbf{W} равномерно распределяется по всем состояниям i ($i=1, \dots, 4$), а при $\sigma \rightarrow 0$ вектор \mathbf{W} с одинаковой вероятностью $1/2$ принимает состояния 1 и 2 и с вероятностью 0 — состояния 3 и 4.

Среднее приращение функции качества $Q(x_1, x_2)$ на N -м шаге поиска равно

$$M_N[\Delta Q] = \sum_{i=1}^4 p_i^{(N)} M[\Delta Q / \mathbf{W}_i], \quad (2.3.14)$$

где

$$M[\Delta Q / \mathbf{W}_i] = d \Delta Q^{(i)}. \quad (2.3.15)$$

Подставляя выражения (2.3.11) и (2.3.15) в соотношение (2.3.14), получаем

$$M_N[\Delta Q] = \frac{1}{2\sqrt{2}} d \left\{ \Phi \left(\frac{\Delta Q^{(1)}}{2\sigma} \right) [1 + d^{2N} (p_1^{(0)} + p_4^{(0)} - p_2^{(0)} - p_3^{(0)})] (\alpha_1 + \alpha_2) + \Phi \left(\frac{\Delta Q^{(2)}}{2\sigma} \right) [1 - d^{2N} (p_1^{(0)} + p_4^{(0)} - p_2^{(0)} - p_3^{(0)})] (\alpha_1 - \alpha_2) \right\}, \quad (2.3.16)$$

где α_1 и α_2 являются направляющими косинусами градиентного вектора, причем $\alpha_1^2 + \alpha_2^2 = 1$.

В предельном случае, когда $N \rightarrow \infty$, из (2.3.16) получаем

$$M[\Delta Q] = \lim_{N \rightarrow \infty} M_N[\Delta Q] = \frac{1}{2\sqrt{2}} d \left[\Phi \left(\frac{\Delta Q^{(1)}}{2\sigma} \right) (\alpha_1 + \alpha_2) + \Phi \left(\frac{\Delta Q^{(2)}}{2\sigma} \right) (\alpha_1 - \alpha_2) \right]. \quad (2.3.17)$$

Отсюда имеем:

$$\lim_{\sigma \rightarrow 0} M[\Delta Q] = \frac{1}{\sqrt{2}} d \alpha_1; \quad (2.3.18)$$

$$\lim_{\sigma \rightarrow \infty} M[\Delta Q] = 0. \quad (2.3.19)$$

На рис. 2.3.1 представлено поведение среднего приращения функции качества в зависимости от уровня помехи σ для случаев, когда направление градиента совпадает с осью x_1 ($\alpha_1 = 1; \alpha_2 = 0$) и когда направление градиента совпадает с биссектрисой $\Delta \mathbf{X}^{(1)}$ ($\alpha_1 = \alpha_2 = 1/\sqrt{2}$). Очевидно, что при малых помехах ($\sigma \leq 1$) система лучше оптимизируется тогда, когда направление градиента совпадает с осью x_1 , хуже, — когда направление градиента совпадает с биссектрисой $\Delta \mathbf{X}^{(1)}$. Для помех $\sigma > 1,5$ среднее увеличение функции качества практически не зависит от направления градиента оптимизируемой системы.

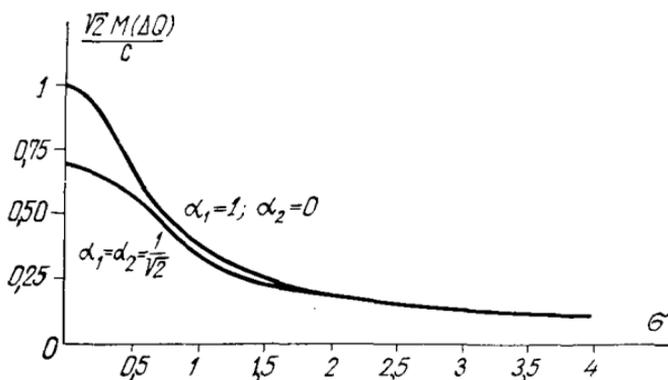


Рис. 2.3.1. Характеристика изменения среднего приращения функции качества на одном шаге поиска для $n=2$ и $\delta \geq 2d$ при $N \rightarrow \infty$ в зависимости от уровня помехи σ .

Далее рассмотрим частный случай, когда $d=1$. Подставляя в матрицах (2.3.9) и (3.3.10) $d=1$, получаем

$$A^N = A = \begin{vmatrix} \Phi_1 & 0 & 0 & \Phi_4 \\ 0 & \Phi_2 & \Phi_3 & 0 \\ 0 & \Phi_2 & \Phi_3 & 0 \\ \Phi_1 & 0 & 0 & \Phi_4 \end{vmatrix} \quad (2.3.20)$$

В этом случае вероятности перехода вектора \mathbf{W} из одного состояния в другое не зависят от номера шага N . Эти переходы показаны на рис. 2.3.2. Видно, что состояния цепи Маркова образуют два замкнутых множества [50]. В одно множество входят состояния 1 и 4, во второе — состояния 2 и 3.

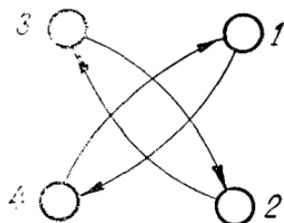


Рис. 2.3.2. Граф переходов вектора \mathbf{W} при $d=1$.

Подставляя в формулу (2.3.11) $d=1$, получаем:

$$\begin{aligned} p_1^{(N)} &= \Phi_1(p_1^{(0)} + p_4^{(0)}); \\ p_2^{(N)} &= \Phi_2(p_2^{(0)} + p_3^{(0)}); \\ p_3^{(N)} &= \Phi_3(p_2^{(0)} + p_3^{(0)}); \\ p_4^{(N)} &= \Phi_4(p_1^{(0)} + p_4^{(0)}). \end{aligned} \quad (2.3.21)$$

Следовательно, распределение вектора \mathbf{W} по состояниям i также не зависит от номера шага поиска N . Это распределение зависит лишь от начальных вероятностей $p_i^{(0)}$.

В пределе при $N \rightarrow \infty$ из (2.3.21) получаем:

$$p_i = \begin{cases} \Phi_i(p_1^{(0)} + p_4^{(0)}) & (i=1, 4); \\ \Phi_i(p_2^{(0)} + p_3^{(0)}) & (i=2, 3). \end{cases} \quad (2.3.22)$$

Подставляя в выражение (2.3.16) $d=1$, имеем

$$\begin{aligned} M[\Delta Q] = M_N[\Delta Q] = & \frac{1}{\sqrt{2}} \left[\Phi \left(\frac{\Delta Q^{(1)}}{2\sigma} \right) (p_1^{(0)} + p_4^{(0)}) (\alpha_1 + \right. \\ & \left. + \alpha_2) + \Phi \left(\frac{\Delta Q^{(2)}}{2\sigma} \right) (p_2^{(0)} + p_3^{(0)}) (\alpha_1 - \alpha_2) \right]. \end{aligned} \quad (2.3.23)$$

Из этой формулы находим:

$$\lim_{\sigma \rightarrow 0} M[\Delta Q] = \frac{1}{\sqrt{2}} \alpha_1 + \frac{1}{\sqrt{2}} (p_1^{(0)} + p_4^{(0)} - p_2^{(0)} - p_3^{(0)}) \alpha_2; \quad (2.3.24)$$

$$\lim_{\sigma \rightarrow \infty} M[\Delta Q] = 0. \quad (2.3.25)$$

2. *Многомерный случай* ($n > 2$; $\delta \geq 2d$). Пусть направляющие косинусы градиентного вектора α_i ($i = 1, \dots, n$) удовлетворяют условию

$$\begin{aligned} \alpha_1 > \alpha_2 > \dots > \alpha_n > 0; \\ \alpha_1 > \alpha_2 + \dots + \alpha_n. \end{aligned} \quad (2.3.26)$$

Тогда при нумерации состояний W_i вектора W (2.2.52) предыдущего параграфа имеем:

$$\begin{aligned} \Delta Q^{(1)}, \Delta Q^{(2)}, \dots, \Delta Q^{(2^{n-1})} > 0; \\ \Delta Q^{(2^{n-1}+1)}, \dots, \Delta Q^{(2^n)} < 0. \end{aligned} \quad (2.3.27)$$

Проделав те же самые выкладки, что и для $n=2$, получаем предельное распределение вектора W по состояниям:

$$\begin{aligned} p_i = \frac{1}{2^{n-1}} \Phi_i \\ (i = 1, \dots, 2^n). \end{aligned} \quad (2.3.28)$$

Среднее приращение функции качества $Q(x_1, \dots, x_n)$ на один шаг поиска в предельном случае $N \rightarrow \infty$ равно

$$\begin{aligned} M[\Delta Q] = d \cdot 2^{1-n} \sum_{i=1}^{2^n} \Phi_i \Delta Q^{(i)} = d \cdot 2^{1-n} n^{-\frac{1}{2}} \times \\ \times \left\{ \left[\Phi \left(\frac{\Delta Q^{(1)}}{2\sigma} \right) + \dots + \Phi \left(\frac{\Delta Q^{(2^{n-1})}}{2\sigma} \right) \right] \alpha_1 + \right. \\ \left. + \left[\Phi \left(\frac{\Delta Q^{(1)}}{2\sigma} \right) + \dots + \Phi \left(\frac{\Delta Q_{2^{n-2}}}{2\sigma} \right) - \right. \right. \end{aligned}$$

$$\begin{aligned}
 & -\Phi\left(\frac{\Delta Q^{(2^{n-2}+1)}}{2\sigma}\right) - \dots - \Phi\left(\frac{\Delta Q^{(2^{n-1})}}{2\sigma}\right) \Big] \alpha_2 + \dots \\
 & \dots + \left[\Phi\left(\frac{\Delta Q^{(1)}}{2\sigma}\right) + \Phi\left(\frac{\Delta Q^{(2)}}{2\sigma}\right) - \Phi\left(\frac{\Delta Q^{(3)}}{2\sigma}\right) - \right. \\
 & \left. - \Phi\left(\frac{\Delta Q^{(4)}}{2\sigma}\right) - \dots - \Phi\left(\frac{\Delta Q^{(2^{n-1})}}{2\sigma}\right) \right] \alpha_{n-1} + \\
 & + \left[\Phi\left(\frac{\Delta Q^{(1)}}{2\sigma}\right) - \Phi\left(\frac{\Delta Q^{(2)}}{2\sigma}\right) + \Phi\left(\frac{\Delta Q^{(3)}}{2\sigma}\right) - \dots \right. \\
 & \left. \dots - \Phi\left(\frac{\Delta Q^{(2^{n-1})}}{2\sigma}\right) \right] \alpha_n \Big\}. \tag{2.3.29}
 \end{aligned}$$

В предельном случае из этой формулы следует:

$$\lim_{\sigma \rightarrow 0} M[\Delta Q] = \frac{d}{\sqrt{n}} \alpha_1; \tag{2.3.30}$$

$$\lim_{\sigma \rightarrow \infty} M[\Delta Q] = 0. \tag{2.3.31}$$

§ 2.4. СВОЙСТВА ПОКООРИНАТНОГО САМООБУЧЕНИЯ С МАЛОЙ ИНТЕНСИВНОСТЬЮ ПРИ СТАТИСТИЧЕСКОЙ ОПТИМИЗАЦИИ С ПОМЕХАМИ

В предыдущем параграфе был рассмотрен случай покоординатного самообучения, когда вектор памяти \mathbf{W} может принимать одно из $2^n = 4$ возможных значений, т. е. когда количество состояний автомата совпадает с количеством его выходов ($m = 2^n$). В этом параграфе рассмотрим случай, когда $m = (2k+1)^n$, т. е. когда $m > 2^n$. Пусть функция $Q(\mathbf{X})$ обладает теми же свойствами, которые были заданы формулами (2.3.1) — (2.3.8). Тогда работа алгоритма как вероятностного автомата, функционирующего в случайной среде, описывается

$$A = \begin{pmatrix} P_{11} & P_{12} & 0 & 0 & 0 & \dots & 0 \\ P_{21} & 0 & P_{23} & 0 & 0 & \dots & 0 \\ 0 & P_{32} & 0 & P_{34} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots & P_{2k-1,2k-2} \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 \end{pmatrix}$$

цепью Маркова с блочной матрицей переходных вероятностей [51], т. е. матрицей (2.4.1).

В этой матрице нулями обозначены блоки, элементы которых равны нулю, а P_{ij} — матрицы типа Якоби $(2k+1)$ -го порядка, определяемые формулами (2.1.73) и (2.1.74) (например, матрица (2.4.2), где p_{ij} — вероятность перехода вектора \mathbf{W} из состояния i в состояние j).

$P_{11} =$

$$= \begin{pmatrix} p_{11} & p_{12} & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ p_{21} & 0 & p_{23} & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & p_{32} & 0 & p_{34} & 0 & \dots & 0 & 0 & 0 & 0 \\ \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots & p_{2k-1,2k-2} & 0 & p_{2k-1,2k} & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & p_{2k,2k-1} & 0 & p_{2k,2k+1} \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & p_{2k+1,2k} & p_{2k+1,2k+1} \end{pmatrix} \quad (2.4.2)$$

Элементы p_{ij} ($i=1, \dots, (2k+1)^2$; $j=1, \dots, 4$) определяются формулой (2.1.59) при учете формул (2.3.6) и (2.3.7), т. е.

$$p_{ij} = (q_{ij} + q_{i,5-j}) \Phi_j, \quad (2.4.3)$$

где

$$q_{ij} = \frac{1}{4} \prod_{l=1}^2 (1 + \omega_l^{(i)} \text{sign } \Delta x_l^{(j)}). \quad (2.4.4)$$

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \dots & \dots & \dots \\ 0 & P_{2k-1,2k} & 0 \\ P_{2k,2k-1} & 0 & P_{2k,2k+1} \\ 0 & P_{2k+1,2k} & P_{2k+1,2k+1} \end{pmatrix} \quad (2.4.1)$$

Элемент q_{ij} получен из формулы (2.1.41) с использованием конкретного вида функции $F(q_i, \omega_i)$, заданной формулой (2.2.1). Нумерация состояний и переходы вектора \mathbf{W} из одного состояния в другое для этой цепи Маркова показаны на рис. 2.4.1. Учитывая матрицу (2.4.1), получаем систему векторных уравнений для нахождения предельных вероятностей p_i ($i=1, \dots, (k+1)^2$):

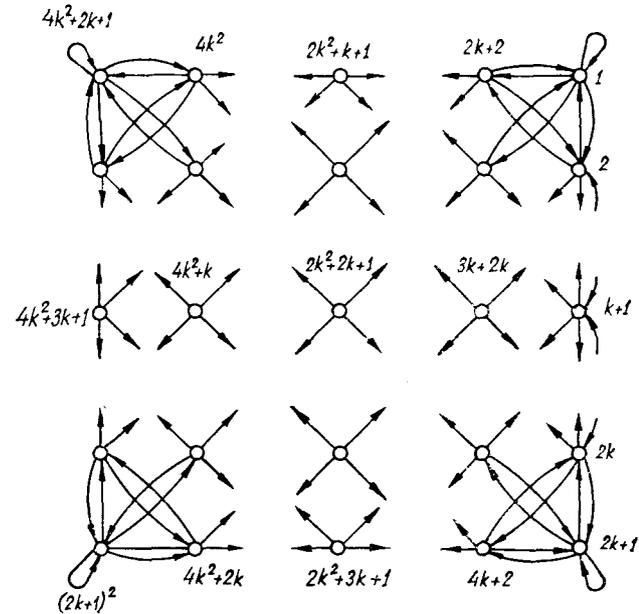


Рис. 2.4.1. Граф переходов вектора \mathbf{W} при $\delta = \frac{d}{k}$.

$$\begin{aligned}
 \mathbf{P}_1 &= P'_{11} \mathbf{P}_1 + P'_{21} \mathbf{P}_2; \\
 \mathbf{P}_2 &= P'_{12} \mathbf{P}_1 + P'_{32} \mathbf{P}_3; \\
 &\dots \\
 \mathbf{P}_{k+1} &= P'_{k,k+1} \mathbf{P}_k + P'_{k+2,k+1} \mathbf{P}_{k+2}; \\
 &\dots \\
 \mathbf{P}_{2k} &= P'_{2k-1,2k} \mathbf{P}_{2k-1} + P'_{2k+1,2k} \mathbf{P}_{2k+1}; \\
 \mathbf{P}_{2k+1} &= P'_{2k,2k+1} \mathbf{P}_{2k} + P'_{2k+1,2k+1} \mathbf{P}_{2k+1},
 \end{aligned} \tag{2.4.5}$$

где P'_{ij} — транспонированная матрица P_{ij} ($i, j=1, \dots, 2k+1$);
 \mathbf{P}_i — $(2k+1)$ -мерные векторы предельных вероятностей;

$$\mathbf{P}_i = (p_{(2k+1)i-2k}, \dots, p_{(2k+1)i}) \quad (i=1, \dots, 2k+1).$$

Решая систему векторных уравнений (2.4.5), получаем уравнение для определения координат вектора \mathbf{P}_{k+1} [52]:

$$\begin{aligned}
 \mathbf{P}_{k+1} &= \{P'_{k,k+1} (I - P'_{k-1,k} (I - \dots - P'_{12} (I - P'_{11})^{-1} P'_{21} \dots \\
 &\quad \dots P'_{k,k-1})^{-1} P'_{k+1,k} + P'_{k+2,k+1} (I - P'_{k+1,k} (I - \dots \\
 &\quad \dots - P'_{2k+1,2k} (I - P'_{2k+1,2k+1})^{-1} P'_{2k,2k+1} \dots \\
 &\quad \dots P'_{k,k+1})^{-1} P'_{k+1,k+2}) \} \mathbf{P}_{k+1}.
 \end{aligned} \tag{2.4.6}$$

Векторы \mathbf{P}_i ($i=1, \dots, k, k+2, \dots, 2k+1$) определяются из следующих рекуррентных выражений:

$$\begin{aligned}
 \mathbf{P}_k &= (I - P'_{k-1,k} (I - \dots - P'_{12} (I - P'_{11})^{-1} P'_{21} \dots \\
 &\quad \dots P'_{k,k-1})^{-1} P'_{k+1,k} \mathbf{P}_{k+1}; \\
 \mathbf{P}_2 &= (I - P'_{12} (I - P'_{11})^{-1} P'_{21}) P'_{32} \mathbf{P}_3; \\
 \mathbf{P}_1 &= (I - P'_{11})^{-1} P'_{21} \mathbf{P}_2; \\
 \mathbf{P}_{k+2} &= (I - P'_{k+3,k+2} (I - \dots - P'_{2k+1,2k} (I - \dots \\
 &\quad \dots - P'_{2k+1,2k+1}) P'_{2k,2k+1} \dots P'_{k+3,k+4})^{-1} P'_{k+2,k+3})^{-1} \times \\
 &\quad \times P'_{k+1,k+2} \mathbf{P}_{k+1}; \\
 \mathbf{P}_{2k} &= (I - P'_{2k+1,2k} (I - P'_{2k+1,2k+1})^{-1} P'_{2k,2k+1})^{-1} \times \\
 &\quad \times P'_{2k-1,2k} \mathbf{P}_{2k-1}; \\
 \mathbf{P}_{2k+1} &= (I - P'_{2k+1,2k+1})^{-1} P'_{2k,2k+1} \mathbf{P}_{2k}.
 \end{aligned} \tag{2.4.7}$$

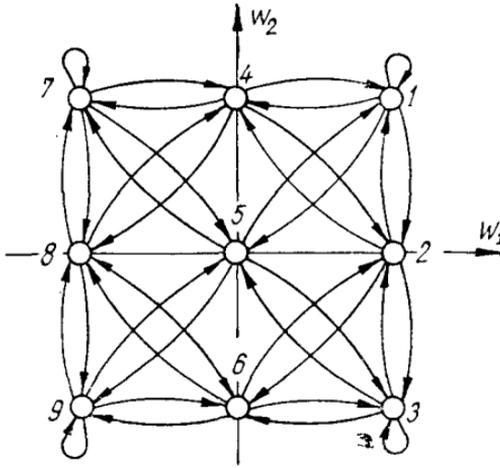


Рис. 2.4.2. Граф переходов вектора \mathbf{W} при $\delta = d$.

Далее более подробно рассмотрим случай, когда $\delta = d$. В этом случае вектор памяти \mathbf{W} может находиться в одном из девяти состояний. Нумерация состояний и переходов для этого случая показаны на рис. 2.4.2. Для определения предельного распределения вектора \mathbf{W} по состояниям $i = 1, \dots, 9$ из формул (2.4.6) и (2.4.7) получаем следующие выражения:

$$\begin{aligned}
 \mathbf{P}_2 &= [P'_{12}(I - P'_{11})^{-1}P'_{21} + P'_{32}(I - P'_{33})^{-1}P'_{23}]\mathbf{P}_2; \\
 \mathbf{P}_1 &= (I - P'_{11})^{-1}P'_{21}\mathbf{P}_2; \\
 \mathbf{P}_3 &= (I - P'_{33})^{-1}P'_{23}\mathbf{P}_2,
 \end{aligned}
 \tag{2.4.8}$$

где

$$\begin{aligned}
 P'_{11} &= \begin{vmatrix} A_1\Phi_1 & B\Phi_1 & 0 \\ A_2\Phi_2 & 0 & A_2\Phi_1 \\ 0 & B\Phi_2 & A_1\Phi_2 \end{vmatrix} & P'_{12} &= \begin{vmatrix} A_2\Phi_3 & B\Phi_3 & 0 \\ A_1\Phi_4 & 0 & A_1\Phi_3 \\ 0 & B\Phi_4 & A_2\Phi_4 \end{vmatrix} \\
 P'_{21} &= \begin{vmatrix} B\Phi_1 & \Phi_1 & 0 \\ B\Phi_2 & 0 & B\Phi_1 \\ 0 & \Phi_2 & B\Phi_2 \end{vmatrix} & P'_{23} &= \begin{vmatrix} B\Phi_3 & \Phi_3 & 0 \\ B\Phi_4 & 0 & B\Phi_3 \\ 0 & \Phi_4 & B\Phi_4 \end{vmatrix}
 \end{aligned}
 \tag{2.4.9}$$

$$P'_{32} = \left\| \begin{array}{ccc} A_2\Phi_1 & B\Phi_1 & 0 \\ A_1\Phi_2 & 0 & A_1\Phi_1 \\ 0 & B\Phi_2 & A_2\Phi_2 \end{array} \right\| \quad P'_{33} = \left\| \begin{array}{ccc} A_1\Phi_3 & B\Phi_3 & 0 \\ A_2\Phi_4 & 0 & A_2\Phi_3 \\ 0 & B\Phi_4 & A_1\Phi_4 \end{array} \right\|$$

$$P_1 = \left\| \begin{array}{c} p_1 \\ p_2 \\ p_3 \end{array} \right\| \quad P_2 = \left\| \begin{array}{c} p_4 \\ p_5 \\ p_6 \end{array} \right\| \quad P_3 = \left\| \begin{array}{c} p_7 \\ p_8 \\ p_9 \end{array} \right\| \quad (2.4.10)$$

$$A_1 = \frac{1}{2} (1+d^2); \quad A_2 = \frac{1}{2} (1-d^2); \quad B = \frac{1}{2};$$

$$\Phi_i = \frac{1}{2} \left[1 + \Phi \left(\frac{\Delta Q^{(i)}}{2\sigma} \right) \right] \quad (2.4.11)$$

$$(i=1, 2, 3, 4);$$

$$\Phi_1 + \Phi_4 = 1; \quad \Phi_2 + \Phi_3 = 1; \quad A_1 + A_2 = 1;$$

$\Phi \left(\frac{\Delta Q^{(i)}}{2\sigma} \right)$ — интеграл вероятности,

I — единичная матрица.

Для краткости введем обозначения

$$H_1 = (I - P'_{11})^{-1} P'_{21}; \quad H_2 = (I - P'_{33})^{-1} P'_{23}; \quad (2.4.12)$$

$$P'_{12}H_1 + P'_{32}H_2 = U_1 + U_2 = U. \quad (2.4.13)$$

Из первого уравнения системы (2.4.8) следует, что

$$p_4 = \frac{u_{21}(1-u_{33}) + u_{13}u_{32}}{(1-u_{11})(1-u_{33}) - u_{13}u_{31}} p_5 = F_1 p_5;$$

$$p_6 = \frac{u_{32}(1-u_{11}) + u_{31}u_{12}}{(1-u_{11})(1-u_{33}) - u_{13}u_{31}} p_5 = F_2 p_5, \quad (2.4.14)$$

где u_{ij} ($i, j=1, 2, 3$) определяются формулой (2.4.13).

Вычислим элементы матриц H_i ($i=1, 2$). После обращения матрицы $I - P'_{11}$ получаем матрицу

$$\begin{aligned} (I - P'_{11})^{-1} &= \\ &= \frac{1}{D_1} \left\| \begin{array}{cc} 1 - A_1\Phi_2 - BA_2\Phi_1\Phi_2 & B\Phi_1(1 - A_1\Phi_2) \\ A_2\Phi_2(1 - A_1\Phi_2) & (1 - A_1\Phi_1)(1 - A_1\Phi_2) \\ BA_2\Phi_2^2 & B\Phi_2(1 - A_1\Phi_1) \\ & BA_2\Phi_1^2 \\ & A_2\Phi_1(1 - A_1\Phi_1) \\ & 1 - A_1\Phi_1 - BA_2\Phi_1\Phi_2 \end{array} \right\| \quad (2.4.15) \end{aligned}$$

где

$$D_1 = (1 - A_1\Phi_1)(1 - A_1\Phi_2) - BA_2\Phi_1\Phi_2[2 - A_1(\Phi_1 + \Phi_2)]. \quad (2.4.16)$$

В соответствии с формулами (2.4.12) имеем матрицу

$$\begin{aligned} H_1 &= \frac{1}{D_1} \left\| \begin{array}{cc} B\Phi_1[1 - B\Phi_2(d^2 + A_1\Phi_2 + A_2\Phi_1)] \\ B(1 - A_1\Phi_2)(1 - d\Phi_1)\Phi_2 \\ B^2\Phi_2^2(1 - d^2)\Phi_1 \\ (1 - A_1\Phi_2)\Phi_1 & B^2\Phi_1^2(1 - d^2\Phi_2) \\ A_2\Phi_1\Phi_2[2 - A_1(\Phi_1 + \Phi_2)] & B(1 - A_2\Phi_1)(1 - d\Phi_2)\Phi_1 \\ (1 - A_1\Phi_1)\Phi_2 & B\Phi_2[1 - B\Phi_1(d^2 + A_1\Phi_1 + A_2\Phi_2)] \end{array} \right\| \quad (2.4.17) \end{aligned}$$

$(I - P'_{33})$, D_3 и H_2 находим, подставляя в формулы (2.4.15), (2.4.16) и (2.4.17) Φ_3 и Φ_4 вместо Φ_1 и Φ_2 . Из выражений (2.4.8) при помощи матрицы (2.4.17) и равенств (2.4.14) получаем:

$$\begin{aligned}
p_1 &= \Phi_1 \left\{ (1 - A_1 \Phi_2) \left[\frac{1}{2} \left(1 + \frac{1}{2} \Phi_2 \right) F_1 + 1 \right] - \right. \\
&\quad \left. - \frac{1}{4} A_2 \Phi_1 \Phi_2 F_1 + \frac{1}{4} \Phi_1 (1 - d^2 \Phi_2) F_2 \right\} \frac{p_5}{D_1}; \\
p_2 &= \left\{ A_2 \Phi_1 \Phi_2 [2 - A_1 (\Phi_1 + \Phi_2)] + \frac{1}{2} [\Phi_2 (1 - A_1 \Phi_2) \times \right. \\
&\quad \left. \times (1 - d^2 \Phi_1) F_1 + \Phi_1 (1 - A_1 \Phi_1) (1 - d^2 \Phi_2) F_2] \right\} \frac{p_5}{D_1}; \\
p_3 &= \Phi_2 \left\{ \frac{1}{4} \Phi_2 (1 - d^2 \Phi_1) F_1 + (1 - A_1 \Phi_1) \left[\frac{1}{2} \left(1 + \right. \right. \right. \\
&\quad \left. \left. + \frac{1}{2} \Phi_1 \right) F_2 + 1 \right] - \frac{1}{4} A_2 \Phi_1 \Phi_2 F_2 \right\} \frac{p_5}{D_1}; \\
p_7 &= \Phi_3 \left\{ (1 - A_1 \Phi_4) \left[\frac{1}{2} \left(1 + \frac{1}{2} \Phi_4 \right) F_1 + 1 \right] - \right. \\
&\quad \left. - \frac{1}{4} A_2 \Phi_3 \Phi_4 F_1 + \frac{1}{4} \Phi_3 (1 - d^2 \Phi_2) F_2 \right\} \frac{p_5}{D_3}; \\
p_8 &= \left\{ A_2 \Phi_3 \Phi_4 [2 - A_1 (\Phi_3 + \Phi_4)] + \frac{1}{2} [\Phi_4 (1 - A_1 \Phi_4) \times \right. \\
&\quad \left. \times (1 - d^2 \Phi_3) F_1 + \Phi_3 (1 - A_1 \Phi_3) (1 - d^2 \Phi_4) F_2] \right\} \frac{p_5}{D_3}; \\
p_9 &= \Phi_4 \left\{ \frac{1}{4} \Phi_4 (1 - d^2 \Phi_3) F_1 + (1 - A_1 \Phi_3) \left[\frac{1}{2} \left(1 + \right. \right. \right. \\
&\quad \left. \left. + \frac{1}{2} \Phi_3 \right) F_2 + 1 \right] - \frac{1}{4} A_2 \Phi_3 \Phi_4 F_2 \right\} \frac{p_5}{D_3},
\end{aligned} \tag{2.4.18}$$

где D_3 можно определить из формулы (2.4.16), если заменить Φ_1 и Φ_2 на Φ_3 и Φ_4 . Нормируя p_i ($i=1, \dots, 9$), находим

$$p_5 = \left\{ \left[\sum_{i=1,3} \left[\frac{1}{2} \Phi_i (1 - A_1 \Phi_{i+1}) \left(1 + \frac{1}{2} \Phi_{i+1} \right) + \right. \right. \right.$$

$$\begin{aligned}
 & + \frac{1}{2} \Phi_{i+1} (1 - d^2 \Phi_i) \left(1 - \frac{1}{2} d^2 \Phi_{i+1} \right) - \\
 & - \frac{1}{4} A_2 \Phi_i^2 \Phi_{i+1} \left] \frac{1}{D_i} + 1 \right] F_1 + \left[\sum_{i=1,3} \left[\frac{1}{2} \Phi_{i+1} (1 - \right. \right. \\
 & - A_1 \Phi_i) \left(1 + \frac{1}{2} \Phi_i \right) + \frac{1}{2} \Phi_i (1 - d^2 \Phi_{i+1}) \left(1 - \right. \\
 & - \left. \frac{1}{2} d^2 \Phi_i \right) - \left. \frac{1}{4} A_2 \Phi_i^2 \Phi_{i+1} \right] D_{i+1} \left. \right] F_2 + \\
 & + \sum_{i=1,3} [d^2 \Phi_i \Phi_{i+1} + (1 - A_1 A_2 \Phi_i \Phi_{i+1}) \times \\
 & \times (\Phi_i + \Phi_{i+1})] + 1 \left. \right\}^{-1}. \tag{2.4.19}
 \end{aligned}$$

Из (2.4.13) следует:

$$\begin{aligned}
 u_{11}^{(1)} &= \frac{1}{2} \Phi_3 \left\{ (1 - A_1 \Phi_2) \left[A_2 \Phi_1 + \frac{1}{2} \Phi_2 + \frac{1}{2} \Phi_1 \Phi_2 (A_2 - \right. \right. \\
 & - \left. \left. d^2) \right] - \frac{1}{2} A_2^2 \Phi_1^2 \Phi_2 \right\} \frac{1}{D_1}; \\
 u_{33}^{(1)} &= \frac{1}{2} \Phi_4 \left\{ (1 - A_1 \Phi_1) \left[A_2 \Phi_2 + \frac{1}{2} \Phi_1 + \right. \right. \\
 & + \left. \frac{1}{2} \Phi_1 \Phi_2 (A_2 - d^2) \right] - \frac{1}{2} A_2^2 \Phi_1 \Phi_2^2 \left. \right\} \frac{1}{D_1}; \tag{2.4.20}
 \end{aligned}$$

$$u_{12}^{(1)} = \Phi_1 \Phi_3 A_2 \left[1 + \Phi_2 \left(A_2 - A_1 \frac{\Phi_1 + \Phi_2}{2} \right) \right] \frac{1}{D_1};$$

$$u_{32}^{(1)} = \Phi_2 \Phi_4 A_2 \left[1 + \Phi_1 \left(A_2 - A_1 \frac{\Phi_1 + \Phi_2}{2} \right) \right] \frac{1}{D_1}.$$

$$u_{13}^{(1)} = \frac{1}{4} \Phi_1 \Phi_3 (1 - d^2 \Phi_1) (1 - d^2 \Phi_2) \frac{1}{D_1};$$

$$u_{31}^{(1)} = \frac{1}{4} \Phi_2 \Phi_4 (1 - d^2 \Phi_1) (1 - d^2 \Phi_2) \frac{1}{D_1}.$$

$u_{ij}^{(2)}$ ($i, j=1, 2, 3$) получается при замене $\Phi_1, \Phi_2, \Phi_3, \Phi_4$ на $\Phi_3, \Phi_4, \Phi_1, \Phi_2$ соответственно. Подставляя $u_{ij} = u_{ij}^{(1)} + u_{ij}^{(2)}$ ($i, j=1, 2, 3$) в выражение (2.4.14), получаем выражения для F_1 и F_2 . По найденным формулам (2.4.14), (2.4.18) и (2.4.19) с учетом (2.4.20) определяем предельное распределение вектора \mathbf{W} по состояниям на плоскости обучения в зависимости от величины помехи σ и от параметра «ограниченности» обучения d .

Рассмотрим некоторые частные случаи.

1. Пусть $\sigma \rightarrow 0$. Тогда, используя формулы (2.4.11), получаем:

$$\lim_{\sigma \rightarrow 0} \Phi_1 = \lim_{\sigma \rightarrow 0} \Phi_2 = 1; \quad \lim_{\sigma \rightarrow 0} \Phi_3 = \lim_{\sigma \rightarrow 0} \Phi_4 = 0;$$

$$\lim_{\sigma \rightarrow 0} D_1 = 0; \quad \lim_{\sigma \rightarrow 0} D_3 = 1; \tag{2.4.21}$$

$$\lim_{\sigma \rightarrow 0} \frac{p_5}{D_1} = \frac{1}{(1+A_2)A_2(F_1+F_2+2)}.$$

Подставляя выражение (2.4.21) в формулы (2.4.18) и (2.4.19), имеем:

$$p_1 = p_3 = \frac{1}{2(1+A_2)}; \quad p_2 = \frac{A_2}{1+A_2}; \quad p_4 = \dots = p_9 = 0.$$

$$\tag{2.4.22}$$

2. Пусть $\sigma \rightarrow \infty$. Тогда, используя формулы (2.4.11), находим

$$\lim_{\sigma \rightarrow \infty} \Phi_1 = \lim_{\sigma \rightarrow \infty} \Phi_2 = \lim_{\sigma \rightarrow \infty} \Phi_3 = \lim_{\sigma \rightarrow \infty} \Phi_4 = \frac{1}{2};$$

$$\lim_{\sigma \rightarrow \infty} F_1 = \lim_{\sigma \rightarrow \infty} F_2 = \frac{2A_2}{A_1}; \tag{2.4.23}$$

$$\lim_{\sigma \rightarrow \infty} \frac{p_5}{D_1} = \frac{8A_1}{(5+7A_2)(2+A_2)(1+A_2)}.$$

Подставив выражение (2.4.23) в формулы (2.4.14), (2.4.18) и (2.4.19), получаем:

$$p_1 = p_3 = p_7 = p_9 = \frac{1}{5 + 7A_2}; \quad p_5 = \frac{A_1}{5 + 7A_2}; \quad p_2 = p_4 = p_6 =$$

$$= p_8 = \frac{2A_2}{5 + 7A_2}. \quad (2.4.24)$$

3. Пусть $d \rightarrow 1$. Тогда

$$\lim_{d \rightarrow 1} A_1 = 1; \quad \lim_{d \rightarrow 1} A_2 = 0; \quad \lim_{d \rightarrow 1} F_1 = \lim_{d \rightarrow 1} F_2 = 0;$$

$$\lim_{d \rightarrow 1} D_1 = \Phi_3 \Phi_4; \quad \lim_{d \rightarrow 1} D_2 = \Phi_1 \Phi_2. \quad (2.4.25)$$

Подставляя это выражение в формулы (2.4.14), (2.4.18) и (2.4.19), получаем:

$$p_1 = \frac{\Phi_1}{\Phi_4} p_5; \quad p_3 = \frac{\Phi_2}{\Phi_3} p_5; \quad p_7 = \frac{\Phi_3}{\Phi_2} p_5; \quad p_9 = \frac{\Phi_4}{\Phi_1} p_5;$$

$$p_5 = \left[1 + \frac{\Phi_1}{\Phi_4} + \frac{\Phi_2}{\Phi_3} + \frac{\Phi_3}{\Phi_2} + \frac{\Phi_4}{\Phi_1} \right]^{-1}; \quad (2.4.26)$$

$$p_2 = p_4 = p_6 = p_8 = 0.$$

Из равенств (2.4.26) следует, что состояния цепи 2, 4, 6 и 8 образуют множество невозвратных состояний, а 1, 3, 5, 7, 9 — множество возвратных состояний.

Среднее приращение функции качества $M[\Delta Q]$ равно

$$M[\Delta Q] = \sum_{i=1}^9 p_i M[\Delta Q / W_i] = [p_1 - p_9 + \frac{1}{2}(p_2 - p_8 + p_4 -$$

$$- p_6)] d\Delta Q^{(1)} + \left[p_3 - p_7 + \frac{1}{2}(p_2 - p_8 - p_4 + p_6) \right] d\Delta Q^{(2)}, \quad (2.4.27)$$

где p_i ($i=1, \dots, 9$) определяются формулами (2.4.18) и (2.4.19).

Из выражения (2.4.27) с учетом равенств (2.4.22) и (2.4.24) следует:

$$\lim_{\sigma \rightarrow 0} M[\Delta Q] = \frac{1}{\sqrt{2}} d\alpha_1; \quad \lim_{\sigma \rightarrow \infty} M[\Delta Q] = 0. \quad (2.4.28)$$

Для $d=1$ из выражения (2.4.27), учитывая равенства (2.4.26), получаем

$$\begin{aligned} M[\Delta Q] = & \frac{\Phi_1 - \Phi_4}{\Phi_1 \Phi_4} \left(1 + \frac{\Phi_1^2 + \Phi_4^2}{\Phi_1 \Phi_4} + \frac{\Phi_2^2 + \Phi_3^2}{\Phi_2 \Phi_3} \right)^{-1} \Delta Q^{(1)} + \\ & + \frac{\Phi_2 - \Phi_3}{\Phi_2 \Phi_3} \left(1 + \frac{\Phi_1^2 + \Phi_4^2}{\Phi_1 \Phi_4} + \frac{\Phi_2^2 + \Phi_3^2}{\Phi_2 \Phi_3} \right)^{-1} \Delta Q^{(2)}. \end{aligned} \quad (2.4.29)$$

Для случая, когда $\alpha_1=1$; $\alpha_2=0$ (вектор градиента функции качества совпадает с осью x_1), $\Phi_1=\Phi_2$; $\Phi_3=\Phi_4$;

$\Delta Q^{(1)}=\Delta Q^{(2)}=1/\sqrt{2}$. Из (2.4.29) следует:

$$M[\Delta Q] = \sqrt{2} \frac{\Phi_1 - \Phi_4}{\Phi_1 \Phi_4} \left(1 + 2 \frac{\Phi_1^2 + \Phi_4^2}{\Phi_1 \Phi_4} \right)^{-1}. \quad (2.4.30)$$

Если же $\alpha_1=\alpha_2=1/\sqrt{2}$ (вектор градиента функции качества совпадает с биссектрисой первого квадранта), то

$$\Phi_2 = \Phi_3 = \frac{1}{2}; \quad \Delta Q^{(1)} = 1; \quad \Delta Q^{(2)} = 0. \quad (2.4.31)$$

Из (2.4.29) получаем

$$M[\Delta Q] = \frac{\Phi_1 - \Phi_4}{\Phi_1 \Phi_4} \left(3 + \frac{\Phi_1^2 + \Phi_4^2}{\Phi_1 \Phi_4} \right)^{-1}. \quad (2.4.32)$$

Можно показать, что если α_1 соответствует условию $1/\sqrt{2} \leq \alpha_1 \leq 1$, то $M[\Delta Q]$ удовлетворяет неравенству

$$\begin{aligned} \frac{\Phi_1 - \Phi_4}{\Phi_1 \Phi_4} \left(3 + \frac{\Phi_1^2 + \Phi_4^2}{\Phi_1 \Phi_4} \right)^{-1} & \leq M[\Delta Q] \leq \\ & \leq \sqrt{2} \frac{\Phi_1 - \Phi_4}{\Phi_1 \Phi_4} \left(1 + \frac{\Phi_1^2 + \Phi_4^2}{\Phi_1 \Phi_4} \right)^{-1}. \end{aligned} \quad (2.4.33)$$

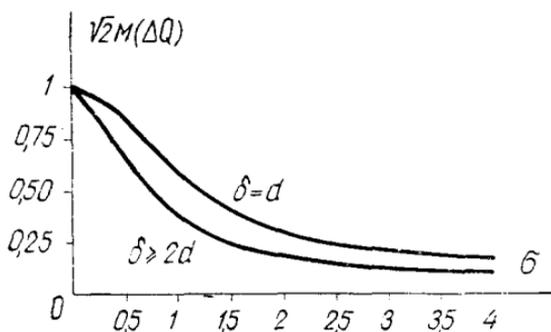


Рис. 2.4.3. Сравнительная характеристика изменения среднего приращения функции качества за один шаг поиска в зависимости от помехи σ при обучении для случаев $\delta = d$ и $\delta \geq 2d$ ($d = 1$).

На рис. 2.4.3 сравнивается изменение $\sqrt{2}M[\Delta Q]$, вычисленное по формуле (2.4.30), с изменением $\sqrt{2}M[\Delta Q]$, вычисленным для $\alpha_1 = 1$; $\alpha_2 = 0$ при $d = 1$, в случае, когда $\delta \geq 2d$. Видно, что для случая $\delta = d$ среднее приращение функции качества на один шаг поиска примерно на $1/3$ больше, чем для случая $\delta \geq 2d$. Это показывает, что в обстановке помех для оптимизации системы целесообразно применять самообучение с меньшей интенсивностью. Этот вывод согласуется с экспериментально полученным результатом, приведенным в работе [6]: уменьшение скорости обучения улучшает свойства поиска.

§ 2.5. СВОЙСТВА ПОКООРИНАТНОГО САМООБУЧЕНИЯ С ДЕТЕРМИНИРОВАННОЙ ПЕРЕХОДНОЙ ФУНКЦИЕЙ, НЕ ЗАВИСЯЩЕЙ ОТ ВЫХОДА АВТОМАТА

Такое самообучение задается формулой (2.1.76) параграфа 2.1 и представляет собой автомат с детерминированными переходами и случайными выходами.

1. *Случай* $m=2^n=4$ ($\delta \geq 2d$). Матрицы переходов для этого случая задаются формулами (2.1.84). Полное функционирование автомата в случайной среде описывается цепью Маркова с матрицей переходных вероятностей (2.1.25). Элементы матрицы штрафов (2.1.26) определяются по формуле (2.1.27). По матрице (2.1.25) находим

$$A = \begin{vmatrix} 1-s^{(1)} & 0 & 0 & s^{(1)} \\ 0 & 1-s^{(2)} & s^{(2)} & 0 \\ 0 & s^{(3)} & 1-s^{(3)} & 0 \\ s^{(4)} & 0 & 0 & 1-s^{(4)} \end{vmatrix} \quad (2.5.1)$$

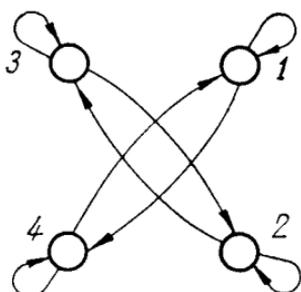


Рис. 2.5.1. Переходы вектора \mathbf{W} для случая $m=4$; $n=2$.

Множество состояний и переходы из одного состояния в другое для этой цепи показаны на рис. 2.5.1. Видно, что состояния 1 и 4 образуют одно замкнутое множество, а состояния 2 и 3 — второе замкнутое множество. Исходя из матрицы (2.5.1) имеем систему уравнений для определения предельных вероятностей

$$\begin{aligned} s^{(1)}p_1 &= p_4s^{(4)}; \\ s^{(2)}p_2 &= s^{(3)}p_3. \end{aligned} \quad (2.5.2)$$

Поскольку состояния цепи Маркова распадаются на два замкнутых класса, то можно рассматривать две цепи Маркова, отдельно для каждого замкнутого множества состояний. Из системы уравнений (2.5.2), используя начальные вероятности состояний $p_1^{(0)}$, $p_2^{(0)}$, $p_3^{(0)}$, $p_4^{(0)}$ и учитывая условия нормировки

$$p_1 + p_4 = 1; \quad p_2 + p_3 = 1, \quad (2.5.3)$$

находим предельные вероятности

$$\begin{aligned} p_1 &= s^{(4)}(p_1^{(0)} + p_4^{(0)}); & p_2 &= s^{(3)}(p_2^{(0)} + p_3^{(0)}); \\ p_3 &= s^{(3)}(p_2^{(0)} + p_3^{(0)}); & p_4 &= s^{(1)}(p_1^{(0)} + p_4^{(0)}). \end{aligned} \quad (2.5.4)$$

Теперь по формуле (2.1.27) определим вероятности штрафов $s^{(i)}$ ($i=1, \dots, 4$). Вероятности q_{ij} ($j, i=$

$= 1, \dots, 4$), содержащиеся в этой формуле, даны в таблице 2.2.1. Вероятности штрафов s_j действий $\Delta \mathbf{X}^{(j)}$ определяются по формуле (2.2.11):

$$s_1 = \frac{1}{2} \left[1 - \Phi \left(\frac{\alpha_1 + \alpha_2}{2\sigma} \right) \right]; \quad s_2 = \frac{1}{2} \left[1 - \Phi \left(\frac{\alpha_1 - \alpha_2}{2\sigma} \right) \right]; \quad (2.5.5)$$

$$s^{(3)} = \frac{1}{2} \left[1 + \Phi \left(\frac{\alpha_1 + \alpha_2}{2\sigma} \right) \right]; \quad s_4 = \frac{1}{2} \left[1 + \Phi \left(\frac{\alpha_1 - \alpha_2}{2\sigma} \right) \right].$$

Следовательно, для вероятностей штрафов $s^{(i)}$ ($i = 1, 2, 3, 4$) имеем:

$$\begin{aligned} s^{(1)} &= \frac{1}{2} [1 + d(s_1 - s_2)]; & s^{(2)} &= \frac{1}{2} [1 + d(s_2 - s_3)]; \\ s^{(3)} &= \frac{1}{2} [1 - d(s_2 - s_3)]; & s^{(4)} &= \frac{1}{2} [1 - d(s_1 - s_4)]. \end{aligned} \quad (2.5.6)$$

Отсюда видно, что

$$s^{(1)} + s^{(4)} = 1; \quad s^{(2)} + s^{(3)} = 1. \quad (2.5.7)$$

Среднее приращение функции качества $Q(\mathbf{X})$ определяется по формуле

$$M[\Delta Q] = \sum_{i=1}^m p_i M[\Delta Q/\mathbf{W}^{(i)}]. \quad (2.5.8)$$

Если функция $Q(\mathbf{X})$ линейная, то в формуле (2.5.8)

$$M[\Delta Q/\mathbf{W}^{(i)}] = \sum_{j=1}^4 q_{ij} \Delta Q^{(j)} = d \Delta Q^{(i)},$$

где $\Delta Q^{(i)}$ — приращение функции $Q(\mathbf{X})$ (на одном шаге) по направлению $\Delta \mathbf{X}^{(i)}$. Следовательно, среднее приращение функции качества

$$\begin{aligned} M[\Delta Q] &= d[(s^{(4)} - s^{(1)}) (p_1^{(0)} + p_4^{(0)}) (\alpha_1 + \alpha_2) + (s^{(3)} - s^{(2)}) \times \\ &\times (p_2^{(0)} + p_3^{(0)}) (\alpha_1 - \alpha_2)] = d^2 \left[\Phi \left(\frac{\alpha_1 + \alpha_2}{2\sigma} \right) (p_1^{(0)} + \right. \end{aligned}$$

$$+ p_4^{(0)} (\alpha_1 + \alpha_2) + \Phi \left(\frac{\alpha_1 - \alpha_2}{2\sigma} \right) (p_2^{(0)} + p_3^{(0)}) \times$$

$$\times (\alpha_1 - \alpha_2) \Big]. \quad (2.5.9)$$

Рассмотрим случай, когда $\sigma \rightarrow 0$, т. е. когда отсутствует помеха. Тогда $s_1 \rightarrow 0$, $s_2 \rightarrow 0$, $s_3 \rightarrow 1$, $s_4 \rightarrow 1$. Следовательно,

$$s^{(1)} \rightarrow \frac{1}{2}(1-d); \quad s^{(2)} \rightarrow \frac{1}{2}(1-d);$$

$$s^{(3)} \rightarrow \frac{1}{2}(1+d); \quad s^{(4)} \rightarrow \frac{1}{2}(1+d).$$

(2.5.10)

Из формулы (2.5.9) имеем

$$M[\Delta Q] = d^2 [\alpha_1 + (p_1^{(0)} + p_4^{(0)} - p_2^{(0)} - p_3^{(0)}) \alpha_2]. \quad (2.5.11)$$

2. Случай $m=4^2=16$ ($\delta=2d/3$). В этом случае по матрице (2.1.25) находим матрицу (2.5.12).

$$A = \begin{pmatrix} 1-s^{(1)} & 0 & 0 & 0 & 0 & s^{(1)} & 0 & 0 & 0 \\ 1-s^{(2)} & 0 & 0 & 0 & 0 & 0 & s^{(2)} & 0 & 0 \\ 0 & 0 & 0 & 1-s^{(3)} & 0 & s^{(3)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1-s^{(4)} & 0 & 0 & s^{(4)} & 0 & 0 \\ 1-s^{(5)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1-s^{(6)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1-s^{(7)} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1-s^{(8)} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & s^{(9)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & s^{(10)} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & s^{(11)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & s^{(12)} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

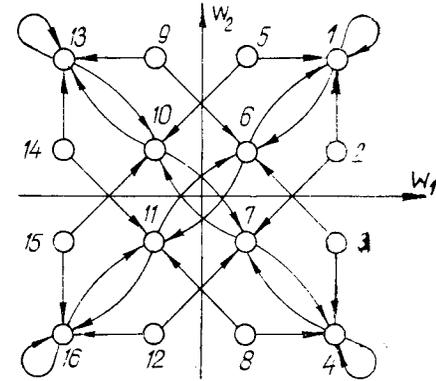


Рис. 2.5.2. Переходы вектора W для случая $m=16$; $n=2$.

Переходы этой цепи Маркова показаны на рис. 2.5.2. Из рисунка видно, что состояния цепи Маркова образуют два замкнутых множества. В первое множество

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s^{(5)} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & s^{(6)} & 0 & 0 & 0 & 0 & 0 \\ s^{(7)} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & s^{(8)} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1-s^{(9)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1-s^{(10)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1-s^{(11)} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1-s^{(12)} \\ s^{(13)} & 0 & 0 & 1-s^{(13)} & 0 & 0 & 0 \\ 0 & s^{(14)} & 0 & 1-s^{(14)} & 0 & 0 & 0 \\ s^{(15)} & 0 & 0 & 0 & 0 & 0 & 1-s^{(15)} \\ 0 & s^{(15)} & 0 & 0 & 0 & 0 & 1-s^{(16)} \end{pmatrix} \quad (2.5.12)$$

входят состояния 1, 6, 11, 16, во второе — состояния 4, 7, 10, 13. Другие состояния являются невозвратными. Предельные вероятности невозвратных состояний равны нулю, т. е.

$$p_2 = p_3 = p_5 = p_8 = p_9 = p_{12} = p_{14} = p_{15} = 0. \quad (2.5.13)$$

Из матрицы (2.5.12) получаем уравнения для нахождения предельных вероятностей состояний 1, 6, 11, 16, принадлежащих первому замкнутому множеству:

$$\begin{aligned} s^{(1)}p_1 &= (1 - s^{(6)})p_6; \\ s^{(6)}p_6 &= s^{(11)}p_{11}; \\ (1 - s^{(11)})p_{11} &= s^{(16)}p_{16}, \end{aligned} \quad (2.5.14)$$

и для состояний 4, 7, 10, 13, принадлежащих второму замкнутому множеству:

$$\begin{aligned} s^{(4)}p_4 &= (1 - s^{(7)})p_7; \\ s^{(7)}p_7 &= s^{(10)}p_{10}; \\ (1 - s^{(10)})p_{10} &= s^{(13)}p_{13}. \end{aligned} \quad (2.5.15)$$

Решая систему уравнений (2.5.14), с учетом условий нормировки и начальных вероятностей получаем:

$$\begin{aligned} p_1 &= \frac{s^{(16)}s^{(11)}(1 - s^{(6)})}{S^{(1)}} P_1^{(0)}; \\ p_6 &= \frac{s^{(16)}s^{(11)}s^{(1)}}{S^{(1)}} P_1^{(0)}; \\ p_{11} &= \frac{s^{(16)}s^{(6)}s^{(1)}}{S^{(1)}} P_1^{(0)}; \\ p_{16} &= \frac{(1 - s^{(11)})s^{(6)}s^{(1)}}{S^{(1)}} P_1^{(0)}, \end{aligned} \quad (2.5.16)$$

где

$$S^{(1)} = s^{(16)}s^{(11)}(s^{(1)} + s^{(11)}) + s^{(1)}s^{(6)}(s^{(6)} + s^{(16)}). \quad (2.5.17)$$

Здесь $P_1^{(0)}$ — начальная вероятность пребывания системы в первом замкнутом множестве;

$$\begin{aligned}
 P_1^{(0)} = & p_1^{(0)} + p_6^{(0)} + p_{11}^{(0)} + p_{16}^{(0)} + p_2^{(0)}p_{21} + p_3^{(0)}p_{36} + \\
 & + p_5^{(0)}p_{51} + p_8^{(0)}p_{8,11} + p_9^{(0)}p_{9,6} + p_{12}^{(0)}p_{12,11} + \\
 & + p_{14}^{(0)}p_{14,11} + p_{15}^{(0)}p_{15,16}.
 \end{aligned} \tag{2.5.18}$$

Решая систему уравнений (2.5.15), получаем аналогичные формулы для предельных вероятностей p_4, p_7, p_{10}, p_{13} :

$$\begin{aligned}
 p_4 = & \frac{s^{(13)}s^{(10)}(1-s^{(7)})}{S^{(2)}} P_2^{(0)}; \\
 p_7 = & \frac{s^{(13)}s^{(10)}s^{(4)}}{S^{(2)}} P_2^{(0)}; \\
 p_{10} = & \frac{s^{(13)}s^{(7)}s^{(4)}}{S^{(2)}} P_2^{(0)}; \\
 p_{13} = & \frac{(1-s^{(10)})s^{(7)}s^{(1)}}{S^{(2)}} P_2^{(0)},
 \end{aligned} \tag{2.5.19}$$

где

$$S^{(2)} = s^{(13)}s^{(10)}(s^{(4)} + s^{(10)}) + s^{(4)}s^{(7)}(s^{(13)} + s^{(7)}). \tag{2.5.20}$$

Здесь $P_2^{(0)}$ — начальная вероятность пребывания системы во втором замкнутом множестве;

$$\begin{aligned}
 P_2^{(0)} = & p_4^{(0)} + p_7^{(0)} + p_{10}^{(0)} + p_{13}^{(0)} + p_2^{(0)}p_{27} + p_3^{(0)}p_{34} + \\
 & + p_5^{(0)}p_{5,10} + p_8^{(0)}p_{8,4} + p_9^{(0)}p_{9,13} + p_{12}^{(0)}p_{12,7} + \\
 & + p_{14}^{(0)}p_{14,13} + p_{15}^{(0)}p_{15,20}.
 \end{aligned} \tag{2.5.21}$$

В формулах (2.5.12) — (2.5.21) входящие вероятности штрафов s^j ($j=1, \dots, 16$) определяются по формуле (2.1.27) с учетом формул (2.1.41) и (2.2.11).

По формуле 2.1.41 определяем вероятности q_{ij} появления на выходе $\Delta X^{(j)}$ при условии, что вектор памяти \mathbf{W} находится в состоянии $\mathbf{W}^{(i)}$. Эти вероятности для

различных состояний приведены в таблице 2.2.1. Определенные по формуле (2.1.27) вероятности $s^{(j)}$ штрафов состояний следующие:

$$\begin{aligned} s^{(1)} &= \frac{1}{2} [1 + d(s_1 - s_4)]; & s^{(6)} &= \frac{1}{2} [1 + \frac{2}{3} d(s_1 - s_4)]; \\ s^{(11)} &= \frac{1}{2} [1 - \frac{2}{3} d(s_1 - s_4)]; & s^{(16)} &= \frac{1}{2} [1 - d(s_1 - s_4)]; \end{aligned} \quad (2.5.22)$$

$$\begin{aligned} s^{(4)} &= \frac{1}{2} [1 + d(s_2 - s_3)]; & s^{(7)} &= \frac{1}{2} [1 + \frac{2}{3} d(s_2 - s_3)]; \\ s^{(10)} &= \frac{1}{2} [1 - \frac{2}{3} d(s_2 - s_3)]; & s^{(13)} &= \frac{1}{2} [1 - d(s_2 - s_3)]. \end{aligned}$$

Из этих формул находим равенства

$$\begin{aligned} s^{(1)} + s^{(16)} &= 1; & s^{(6)} + s^{(11)} &= 1; \\ s^{(4)} + s^{(13)} &= 1; & s^{(7)} + s^{(10)} &= 1. \end{aligned} \quad (2.5.23)$$

Следовательно,

$$\begin{aligned} S^{(1)} &= [s^{(16)} + (s^{(6)})^2] s^{(1)}; & S^{(2)} &= [s^{(13)} + (s^{(7)})^2] s^{(4)}; \\ p_1 &= \frac{s^{(16)} (s^{(11)})^2}{S^{(1)}} P_1^{(0)}; & p_6 &= \frac{s^{(16)} s^{(11)} s^{(1)}}{S^{(1)}} P_1^{(0)}; \\ p_{11} &= \frac{s^{(16)} s^{(6)} s^{(1)}}{S^{(1)}} P_1^{(0)}; & p_{16} &= \frac{(s^{(6)})^2 s^{(1)}}{S^{(1)}} P_1^{(0)}; \\ p_4 &= \frac{s^{(13)} (s^{(10)})^2}{S^{(2)}} P_2^{(0)}; & p_7 &= \frac{s^{(13)} s^{(10)} s^{(4)}}{S^{(2)}} P_2^{(0)}; \\ p_{10} &= \frac{s^{(13)} s^{(7)} s^{(4)}}{S^{(2)}} P_2^{(0)}; & p_{13} &= \frac{(s^{(7)})^2 s^{(1)}}{S^{(2)}} P_2^{(0)}. \end{aligned} \quad (2.5.24)$$

Среднее приращение функции $Q(\mathbf{X})$ при нахождении вектора \mathbf{W} в состоянии $\mathbf{W}^{(i)}$ равно

$$M[\Delta Q/W^{(i)}] = \sum_{j=1}^4 q_{ij} \Delta Q^{(j)} = \begin{cases} d\Delta Q^{(1)} & \text{при } i=1; \\ \frac{2}{3} d\Delta Q^{(1)} & \text{при } i=6; \\ \frac{2}{3} d\Delta Q^{(3)} & \text{при } i=11; \\ d\Delta Q^{(3)} & \text{при } i=16; \\ d\Delta Q^{(4)} & \text{при } i=4; \\ \frac{2}{3} d\Delta Q^{(4)} & \text{при } i=7; \\ \frac{2}{3} d\Delta Q^{(2)} & \text{при } i=10; \\ d\Delta Q^{(2)} & \text{при } i=13, \end{cases} \quad (2.5.25)$$

где

$$\Delta Q^{(1)} = (\alpha_1 + \alpha_2); \quad \Delta Q^{(2)} = (\alpha_1 - \alpha_2);$$

$$\Delta Q^{(3)} = -(\alpha_1 + \alpha_2); \quad \Delta Q^{(4)} = -(\alpha_1 - \alpha_2).$$

Среднее приращение функции $Q(\mathbf{X})$ при учете всех состояний $\mathbf{W}^{(i)}$ вектора памяти \mathbf{W} равно

$$M[\Delta Q] = \sum_{i=1}^{16} p_i M[\Delta Q/W^{(i)}] = \left[(p_1 - p_{16}) + \frac{2}{3} (p_6 - p_{11}) \right] \times \\ \times d\Delta Q^{(1)} + \left[(p_4 - p_{13}) + \frac{2}{3} (p_7 - p_{10}) \right] d\Delta Q^{(2)}, \quad (2.5.26)$$

где предельные вероятности p_i определены формулами (2.5.24).

Рассмотрим случай, когда отсутствует помеха ($\sigma=0$). В этом случае $s_1=s_2=0$; $s_3=s_4=1$. Следовательно,

$$s^{(1)}=s^{(4)}=\frac{1}{2}(1-d); \quad s^{(13)}=s^{(16)}=\frac{1}{2}(1+d);$$

$$s^{(6)} = s^{(7)} = \frac{1}{2} \left(1 - \frac{2}{3} d \right); \quad s^{(10)} = s^{(11)} = \frac{1}{2} \left(1 + \frac{2}{3} d \right). \quad (2.5.27)$$

Предельные вероятности равны:

$$p_1 = \frac{s^{(16)} (s^{(11)})^2}{S^{(1)}} P_1^{(0)}; \quad p_4 = \frac{s^{(16)} (s^{(11)})^2}{S^{(1)}} P_2^{(0)};$$

$$p_{16} = \frac{(s^{(6)})^2 s^{(1)}}{S^{(1)}} P_1^{(0)}; \quad p_{13} = \frac{(s^{(16)})^2 s^{(1)}}{S^{(1)}} P_2^{(0)}; \quad (2.5.28)$$

$$p_6 = \frac{s^{(16)} s^{(11)} s^{(1)}}{S^{(1)}} P_1^{(0)}; \quad p_7 = \frac{s^{(16)} s^{(11)} s^{(1)}}{S^{(1)}} P_2^{(0)};$$

$$p_{11} = \frac{s^{(16)} s^{(6)} s^{(1)}}{S^{(1)}} P_1^{(0)}; \quad p_{10} = \frac{s^{(16)} s^{(6)} s^{(1)}}{S^{(1)}} P_2^{(0)},$$

где

$$S^{(1)} = S^{(2)} = s^{(16)} s^{(11)} (s^{(1)} + s^{(11)}) + s^{(6)} s^{(1)} (s^{(16)} + s^{(6)}). \quad (2.5.29)$$

Подставляя эти вероятности и выражения (2.5.25) в формулу (2.5.26) и учитывая формулы (2.5.27), находим среднее приращение функции

$$M[\Delta Q] = \frac{25d^2}{18 + 7d^2} [\alpha_1 + (P_1^{(0)} - P_2^{(0)}) \alpha_2], \quad (2.5.30)$$

где α_1, α_2 — направляющие косинусы градиентного вектора функции $Q(\mathbf{X})$; $P_1^{(0)}$ и $P_2^{(0)}$ определяются формулами (2.5.19) и (2.5.21). Из этой формулы имеем

$$\lim_{d \rightarrow 1} M[\Delta Q] = \alpha_1 + (P_1^{(0)} - P_2^{(0)}) \alpha_2 \quad (2.5.31)$$

и

$$\lim_{d \rightarrow 0} M[\Delta Q] = 0. \quad (2.5.32)$$

Аналогичным образом можно исследовать случай, когда $m > 16$ ($\delta < 2d/3$).

§ 2.6. СТАТИСТИЧЕСКИЕ СВОЙСТВА КОЛЛЕКТИВА ОПТИМИЗИРУЮЩИХ АВТОМАТОВ И ИХ СРАВНЕНИЕ СО СВОЙСТВАМИ ПОКООРИНАТНОГО САМООБУЧЕНИЯ

В настоящем параграфе исследованы статистические свойства оптимизирующих автоматов с детерминированными выходами и проведено их сравнение со свойствами алгоритма покоординатного самообучения [14, 15]. Структурная схема системы оптимизации изображена на рис. 2.6.1. Здесь A_1, \dots, A_n — независимые вероятностные автоматы, перерабатывающие значения функции $Q(\mathbf{X})$ в значения переменных x_1, \dots, x_n . Каждая из переменных x_j ($j=1, \dots, n$) управляется своим автоматом A_j . Входной переменной всех управляющих автоматов является знак приращения функции $\text{sign } \Delta Q(\mathbf{X})$. Внутреннее состояние автомата A_j определяет изменение переменной согласно соотношениям

$$x_j^{(N+1)} = x_j^{(N)} + \Delta x_j^{(N+1)}, \quad (2.6.1)$$

где $\Delta x_j^{(N+1)}$ — выход j -го автомата, равный $+1$ или -1 в зависимости от того, находится ли j -й автомат на

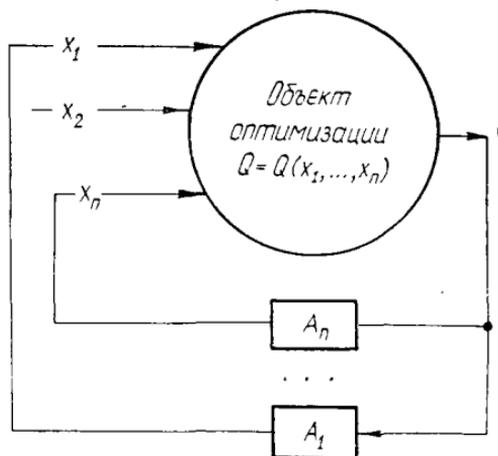


Рис. 2.6.1. Схема системы оптимизации коллективом независимых автоматов.

$(N+1)$ -м такте в одном из первых m состояний или в одном из последних m состояний.

Смена состояний вероятностного автомата с линейной тактикой с $2m$ состояниями показана на рис. 2.6.2.

Систему оптимизирующих вероятностных автоматов будем сопоставлять с алгоритмом покоординатного самообучения при случайном поиске. Напомним сущность этого алгоритма: Пусть в формуле (2.6.1) $\Delta x_j^{(N+1)}$ равно $+1$ с вероятностью \tilde{p}_j и -1 с вероятностью $1 - \tilde{p}_j$, причем \tilde{p}_j определяется по следующей формуле:

$$\tilde{p}_j = \frac{1}{2}(1 + \omega_j). \quad (2.6.2)$$

Здесь ω_j — параметр памяти, который в процессе поиска меняется от шага к шагу в зависимости от накопленного опыта по следующей рекуррентной формуле:

$$\omega_j^{(N+1)} = \omega_j^{(N)} + \delta \operatorname{sign}(\Delta Q_N \Delta x_j^{(N)}), \quad (2.6.3)$$

где δ — постоянная ($\delta > 0$).

Чтобы не допустить нежелательного детерминирования поиска, на изменение ω_j накладывается ограничение

$$\omega_j = \begin{cases} -d, & \text{если } \omega_j \leq -d; \\ \omega_j, & \text{если } -d < \omega_j < d; \\ d, & \text{если } d \leq \omega_j, \end{cases} \quad (2.6.4)$$

где $0 \leq d \leq 1$.

Нетрудно заметить, что система случайного поиска с покоординатным самообучением равносильна вышеизложенной схеме оптимизации, если вместо автомата A_j по-

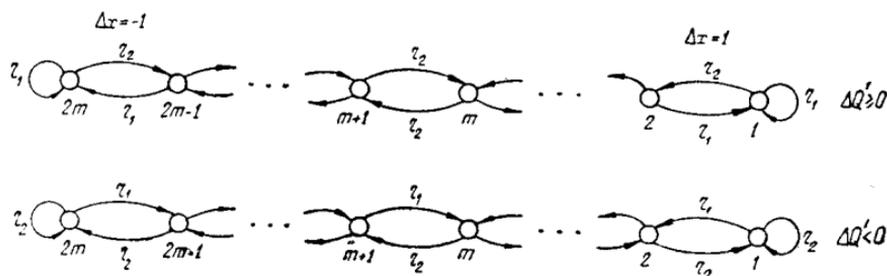


Рис. 2.6.2. Графы переходов оптимизирующих автоматов.

ставить такие автоматы со случайными выходами, внутренним состоянием которых является параметр ω_j , определяющий вероятности \bar{p}_j ; $1 - \bar{p}_j$ выходов автомата. Переходные функции этих автоматов зависят от их выходов. Предположим, что функция $Q(\mathbf{X})$ является линейной, т. е.

$$Q(\mathbf{X}) = Q(\mathbf{X}_0) + (\mathbf{X}, \text{grad } Q(\mathbf{X})), \quad (2.6.5)$$

где

$$\text{grad } Q(\mathbf{X}) = \text{const.}$$

Будем рассматривать задачу отыскания максимума функции $Q(\mathbf{X})$. Рассмотрим случай, когда оптимизацию производит коллектив автоматов A_j . Предположим, что все автоматы A_j коллектива одинаковы и что их переходы из одного состояния в другое описываются матрицей (2.1.103) при нештрафе ($c=0$) и матрицей (2.1.104) при штрафе ($c=1$). Далее предположим, что на объект оптимизации не воздействует помеха, т. е. $\sigma=0$. Сначала анализ проводим для случая одномерного пространства $\{\mathbf{X}\}$ ($n=1$). Тогда имеем следующие вероятности s_i штрафов действий Δx_i автомата ($i=1, 2$):

$$s_1=0; \quad s_2=1 \quad (2.6.6)$$

и получаем следующую матрицу вероятностей штрафов (см. формулу (2.1.120)):

$$S^*_1 = \left(\begin{array}{cccccccc} 0 & \dots & 0 & 0 & 0 & \dots & 0 & \\ \cdot & \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 & \\ 0 & \dots & 0 & 1 & 0 & \dots & 0 & \\ \cdot & \\ \cdot & \\ 0 & \dots & 0 & 0 & 0 & \dots & 1 & \end{array} \right) \left. \begin{array}{l} \vphantom{\left(\right.} \right\} m \\ \vphantom{\left(\right.} \right\} m \end{array} \right) \quad (2.6.7)$$

По формуле (2.1.25) находим матрицу переходных вероятностей цепи Маркова, описывающей функционирование автомата,

$$A = \begin{pmatrix} r_1 & r_2 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ r_1 & 0 & r_2 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & r_1 & 0 & r_2 & 0 & \dots & 0 & 0 & 0 & 0 \\ \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots & r_1 & 0 & r_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & r_1 & 0 & r_2 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & r_1 & r_2 \end{pmatrix} \quad (2.6.8)$$

Вектор предельных вероятностей определяется решением системы линейных уравнений

$$P = A'P, \quad (2.6.9)$$

где A' — транспонированная матрица A ;

P — вектор предельных вероятностей; $P = (p_1, \dots, \dots, p_{2m})$.

Из системы (2.6.9) находим предельные вероятности

$$p_i = r_1^{2m-i} r_2^{i-1} \frac{r_1 - r_2}{r_1^{2m} - r_2^{2m}} \quad (2.6.10)$$

$$(i = 1, 2, \dots, 2m).$$

Среднее приращение функции $Q(x_1)$ на одном шаге

$$M[\Delta Q] = \sum_{j=1}^m p_j - \sum_{i=m+1}^{2m} p_i = \frac{r_1^m - r_2^m}{r_1^m + r_2^m}. \quad (2.6.11)$$

Для случайного поиска с покоординатным самообучением переходы параметра памяти ω_j (переходы автомата

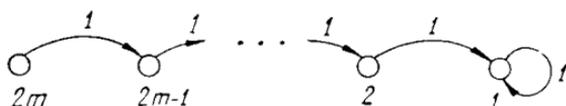


Рис. 2.6.3. Граф переходов параметра ω_j алгоритма покоординатного самообучения.

из одного состояния в другое) для случая $\delta = d/m$ показаны на рис. 2.6.3. Эти переходы образуют цепь Маркова с матрицей переходных вероятностей

$$A = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & 0 \end{pmatrix} \quad (2.6.12)$$

Предельные вероятности равны

$$p_1 = 1; \quad p_2 = p_3 = \dots = p_{2m} = 0 \quad (2.6.13)$$

и среднее приращение функции $Q(x_1)$ на одном шаге

$$M[\Delta Q] = p_1 - (1 - p_1) = 2p_1 - 1 = d. \quad (2.6.14)$$

Сопоставляя формулы (2.6.11) и (2.6.14), видим, что выражению (2.6.11) соответствует параметр d в алгоритме случайного поиска с покоординатным самообучением.

Формула (2.6.10) дает решение для функций, зависящих от одного переменного $Q(\mathbf{X}) = Q(x_1)$.

Далее рассмотрим двумерный случай, т. е. функции, зависящие от двух переменных $Q = Q(x_1, x_2)$. Тогда в процедуре оптимизации участвуют два автомата A_1 и A_2 , система которых $A = \{A_1, A_2\}$ образует один более сложный автомат A . Внутренним состоянием автомата A является вектор $\mathbf{W} = (\omega_1, \omega_2)$, где ω_1 и ω_2 — внутренние состояния автоматов A_1 и A_2 соответственно, а его выходом является вектор $\Delta \mathbf{X} = (\Delta x_1, \Delta x_2)$, где Δx_1 и Δx_2 — выходы автоматов A_1 и A_2 соответственно. Входом автомата A является $\text{sign } \Delta Q$.

Пусть далее вектор градиента функции $Q(x_1, x_2)$ находится между биссектрисами первого и четвертого квадранта плоскости $\mathbf{X}(x_1, x_2)$ (рис. 2.6.4) и пусть каждый из автоматов A_1 и A_2 имеет два состояния $\omega_1^{(i)}$ и $\omega_2^{(i)}$

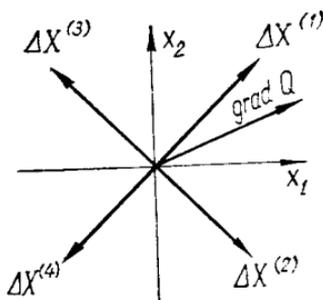


Рис. 2.6.4. Плоскость независимых переменных оптимизируемой функции $Q(x_1, x_2)$.

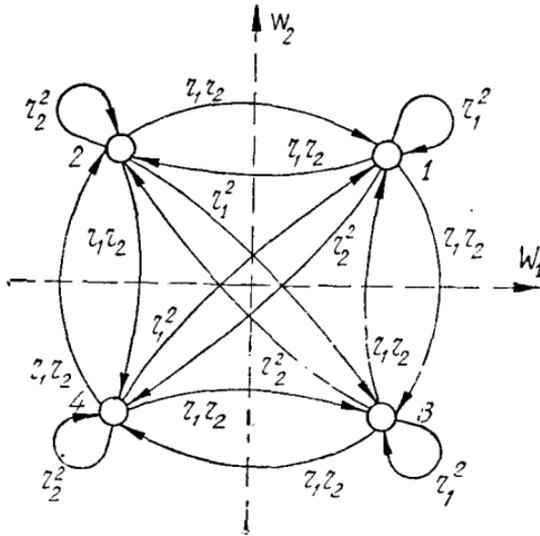


Рис. 2.6.5. Граф переходов цепи Маркова, описывающей переходы системы автоматов $A = \{A_1, A_2\}$ при $m_1 = m_2 = 1$.

($i=1, 2$), т. е. $m_1 = m_2 = 1$. Тогда автомат A имеет переходные матрицы A_0 и A_1 типа (2.1.107) и (2.1.108). По формуле (2.1.25) находим, что переходы автомата A образуют однородную цепь Маркова (рис. 2.6.5) со следующей матрицей переходных вероятностей:

$$A = \begin{vmatrix} r_1^2 & r_1 r_2 & r_1 r_2 & r_2^2 \\ r_1 r_2 & r_2^2 & r_1^2 & r_1 r_2 \\ r_1 r_2 & r_2^2 & r_1^2 & r_1 r_2 \\ r_1^2 & r_1 r_2 & r_1 r_2 & r_2^2 \end{vmatrix} \quad (2.6.15)$$

Предельные вероятности этой цепи Маркова равны

$$p_1 = p_3 = \frac{1}{2} r_1; \quad (2.6.16)$$

$$p_2 = p_4 = \frac{1}{2} r_2.$$

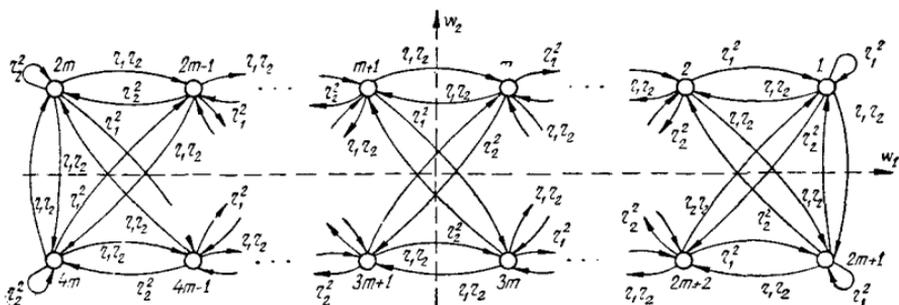


Рис. 2.6.6. Граф переходов цепи Маркова, описывающей переходы системы автоматов $A = \{A_1, A_2\}$ при $m_1 = m$; $m_2 = 1$.

Среднее приращение функции $Q(x_1, x_2)$ на одном шаге

$$M[\Delta Q] = \frac{1}{\sqrt{2}}(r_1 - r_2)\alpha_1, \quad (2.6.17)$$

где $\alpha_1 = \cos(x_1, \text{grad } Q)$.

Для алгоритма случайного поиска с покоординатным самообучением в § 2.2 было получено:

$$M[\Delta Q] = \frac{1}{\sqrt{2}} d \alpha_1. \quad (2.6.18)$$

Следовательно, параметр d соответствует разности вероятностей $r_1 - r_2$.

Когда автомат A_1 имеет $2m$ состояний, а автомат A_2 — два состояния, тогда по матрице (2.1.25) находим, что переходы автомата $A = \{A_1, A_2\}$ образуют цепь Маркова (рис. 2.6.6) со следующей матрицей переходных вероятностей:

$$A = \begin{vmatrix} T_1 & T_2 \\ T_2 & T_1 \end{vmatrix} \quad (2.6.19)$$

где блоки T_1 и T_2 — матрицы (2.6.20) и (2.6.21) типа Якоби.

$$T_1 = \left(\begin{array}{cccccccccccccccc} r_1^2 & r_1 r_2 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ r_1^2 & 0 & r_1 r_2 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & r_1^2 & 0 & r_1 r_2 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots & r_1^2 & 0 & r_1 r_2 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & r_1 r_2 & 0 & r_2^2 & 0 & \dots & 0 & 0 & 0 & 0 \\ \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & r_1 r_2 & 0 & r_2^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & r_1 r_2 & 0 & r_2^2 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & r_1 r_2 & r_2^2 \end{array} \right) \left. \begin{array}{l} \dots \\ \dots \\ \dots \\ \dots \\ \dots \\ \dots \end{array} \right\} m$$

(2.6.20)

Предельные вероятности для этой цепи Маркова равны

$$p_i = p_{i+2m} = \frac{1}{2} r_1^{2m-i} r_2^{i-1} \frac{r_1 - r_2}{r_1^{2m} - r_2^{2m}} \quad (2.6.22)$$

$(i = 1, \dots, 2m),$

а среднее приращение функции $Q(\mathbf{X})$ определяется по формуле

$$M[\Delta Q] = \frac{1}{\sqrt{2}} \cdot \frac{r_1^m - r_2^m}{r_1^m + r_2^m} \alpha_1. \quad (2.6.23)$$

Следовательно,

$$\lim_{m \rightarrow \infty} M[\Delta Q] = \frac{\alpha_1}{\sqrt{2}}. \quad (2.6.24)$$

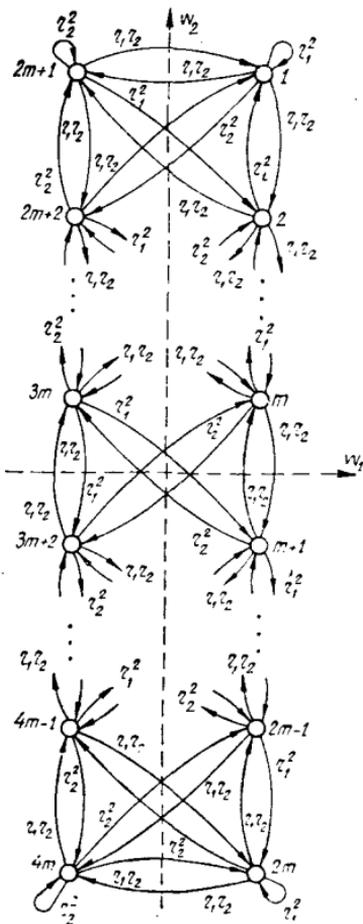
Нетрудно заметить, что в рассмотренном случае увеличение памяти m_1 автомата A_1 приводит к увеличению среднего приращения функции $Q(x_1, x_2)$.

Когда автомат A_1 имеет два состояния, а автомат A_2 — $2m$ состояний, переходы автомата $A = \{A_1, A_2\}$ образуют цепь Маркова (рис. 2.6.7) с матрицей переходных вероятностей

$$A = \begin{vmatrix} T_1(r_1) & T_2(r_2) \\ T_2(r_1) & T_1(r_2) \end{vmatrix} \quad (2.6.25)$$

Рис. 2.6.7. Граф переходов цепи Маркова, описывающий переходы системы автоматов $A = \{A_1, A_2\}$ при $m_1 = 1; m_2 = 2m$.

где блоки $T_i(r_1)$ и $T_i(r_2)$ ($i = 1, 2$) представляют собой матрицы типа Якоби:



$$T_2(r_1) = \begin{pmatrix}
 r_1 r_2 & r_1^2 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\
 r_1 r_2 & 0 & r_1^2 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\
 0 & r_1 r_2 & 0 & r_1^2 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\
 \dots & \dots \\
 0 & 0 & 0 & 0 & 0 & \dots & r_1 r_2 & 0 & r_1^2 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & \dots & 0 & r_1^2 & 0 & r_1 r_2 & 0 & \dots & 0 & 0 & 0 & 0 \\
 \dots & \dots \\
 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & r_1^2 & 0 & r_1 r_2 & 0 \\
 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & r_1^2 & 0 & r_1 r_2 \\
 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & r_1^2 & r_1 r_2
 \end{pmatrix}$$

$\underbrace{\hspace{15em}}_m$
 $\underbrace{\hspace{15em}}_m$

(2.6.27)

$T_1(r_2)$ можно получить из $T_1(r_1)$ и $T_2(r_2)$ — из $T_2(r_1)$, если в этих матрицах поменять местами r_1 и r_2 .

Предельное распределение вероятностей цепи Маркова следующее:

$$p_i = p_{2m+1-i} = \frac{1}{2} r_1 \frac{(a-b) a^{m-i} b^{i-1}}{a^m - b^m};$$

$$p_{2m+i} = p_{4m+1-i} = \frac{1}{2} r_2 \frac{(a-b) a^{m-i} b^{i-1}}{a^m - b^m}$$

(2.6.28)

($i = 1, 2, \dots, m$),

где

$$a = r_1^2 + r_2^2; \quad b = 2r_1 r_2.$$

Среднее приращение функции

$$M[\Delta Q] = \frac{1}{\sqrt{2}} (r_1 - r_2) \alpha_1.$$

(2.6.29)

Полученная формула совпадает с формулой (2.6.17). Следовательно, увеличение памяти автомата A_2 в рассмотренном случае не влияет на среднее приращение функции $Q(x_1, x_2)$ при ее оптимизации.

Из изложенного в этом параграфе видно, что относительно среднего приращения функции $Q(\mathbf{X})$ на одном шаге поиска оба алгоритма эквивалентны, т. е. для обоих можно указать такие значения их параметров, при которых средние приращения функции $Q(\mathbf{X})$ совпадают.

§ 2.7. СТАТИСТИЧЕСКИЕ СВОЙСТВА КОЛЛЕКТИВА ВЕРОЯТНОСТНЫХ АВТОМАТОВ ПРИ ОПТИМИЗАЦИИ ФУНКЦИИ n ПЕРЕМЕННЫХ

В предыдущем параграфе были получены формулы для определения предельных вероятностей цепи Маркова и среднего приращения функции $Q(\mathbf{X})$, если эта функция зависит от одного или двух переменных. В этом параграфе получим общие формулы для случая функции $Q(\mathbf{X})$ n переменных при условии, что

каждый автомат коллектива имеет два состояния [16], т. е. память автоматов $m_l=1$ ($l=1, \dots, n$).

За состояние системы, как и раньше, примем элемент множества возможных значений вектора $\mathbf{W}=(\omega_1, \omega_2, \dots, \omega_n)$, которое обозначим через $\{\mathbf{W}\}$. Общее число возможных значений вектора \mathbf{W} конечно и равно

$$m=2^n \prod_{i=1}^n m_i. \quad (2.7.1)$$

Пусть p_{ij} ($i, j \in \{\mathbf{W}\}$) означает вероятность перехода вектора \mathbf{W} из состояния $\mathbf{W}^{(i)}$ в состояние $\mathbf{W}^{(j)}$ за один шаг. Переход вектора \mathbf{W} из одного состояния в другое описывается цепью Маркова с матрицей переходных вероятностей (2.1.25), матрицы A_0 и A_1 которой определяются формулами (2.1.107) и (2.1.108).

Предельные вероятности p_1, \dots, p_{2^n} этой цепи Маркова в общем случае находятся как решение системы алгебраических уравнений (2.1.20). В этом параграфе [16] покажем другой, более простой прием нахождения предельных вероятностей p_1, \dots, p_{2^n} .

Разобьем множество состояний системы $\{\mathbf{W}\}$ на два непересекающихся подмножества $\{\mathbf{W}\}^+$ и $\{\mathbf{W}\}^-$, причем $\mathbf{W} \in \{\mathbf{W}\}^+$, если $\Delta Q > 0$ при смещении $\Delta \mathbf{X}$; в противном случае, т. е. когда $\Delta Q < 0$, $\mathbf{W} \in \{\mathbf{W}\}^-$. Случай, когда $\Delta Q = 0$, который возможен при совпадении линии уровня функции $Q(x_1, \dots, x_n)$ с одним из поисковых движений $\Delta \mathbf{X}^{(j)}$ ($j=1, \dots, 2^n$), не будем рассматривать.

Пусть глубина памяти всех автоматов коллектива равна единице ($m_l=1$; $l=1, \dots, n$). В этом случае вектор состояний системы $\mathbf{W}=(\omega_1, \dots, \omega_n)$ имеет 2^n различных значений. Перенумеруем эти значения следующим образом: номера от 1 до 2^{n-1} присваиваются состояниям системы, входящим в $\{\mathbf{W}\}^-$. Остальным состояниям, входящим в $\{\mathbf{W}\}^+$, номера присваиваются по следующему правилу: если состояние $\mathbf{W}=(\omega_1, \dots, \omega_n)$ имеет номер $i=1, 2, \dots, 2^{n-1}$ (т. е. $\Delta Q^{(i)} < 0$), то номер $i+2^{n-1}$ присваивается такому состоянию из $\{\mathbf{W}\}^+$, для которого

$$\Delta Q^{(i+2^{n-1})} = -\Delta Q^{(i)}.$$

В случае вероятностных автоматов коллектива ($0 < r_k < 1$; $l=1, \dots, n$) множество состояний вектора \mathbf{W} образует эргодический класс, состоящий из одного подкласса.

Отсюда следует, что существуют предельные вероятности p_i того, что система будет находиться в состояниях $\mathbf{W}^{(i)}$ ($i=1, \dots, 2^n$).

Согласно [16] предельные вероятности p_i и p_{i+s} состояний $\mathbf{W}^{(i)}$ вектора \mathbf{W} ($i=1, \dots, s$), где $s=2^{n-1}$, определяются следующими формулами:

$$p_i = \frac{1}{2^{n-1}} \sum_{j=1}^s p_{ji}; \quad p_{s+i} = \frac{1}{2^{n-1}} \sum_{j=1}^s \tilde{p}_{ji}, \quad (2.7.2)$$

где вероятность перехода p_{ji} для вероятностных автоматов, глубина памяти которых равна единице, имеет вид

$$p_{ji} = \prod_{l=1}^n x_l; \quad \tilde{p}_{ji} = \prod_{l=1}^n (1-x_l), \quad (2.7.3)$$

причем $x_l = r_l$ ($l=1, \dots, n$), если l -я координата вектора $\mathbf{W}^{(j)} = (\omega_1, \dots, \omega_n)$, имеющего номер j ($j=1, \dots, 2^n$), совпадает с l -й координатой вектора состояний $\mathbf{W}^{(i)}$, имеющего номер i ($i=1, \dots, 2^n$). В противном случае $x_l = 1 - r_l$. Для состояний $\mathbf{W} = (\omega_1, \dots, \omega_n)$, принадлежащих $\{\mathbf{W}\}^+$, т. е. имеющих номер $j = s+1, \dots, 2s$, наоборот, в случае совпадения указанных координат $x_l = 1 - r_l$, в случае несовпадения $x_l = r_l$ ($l=1, \dots, n$).

Вероятности перехода p_{ji} удовлетворяют следующим свойствам.

Свойство 1:

$$p_{ji} = p_{j+s, i} \quad (2.7.4)$$

$$(j=1, \dots, s; i=1, \dots, 2s).$$

Свойство 2:

$$p_{j, i+s} = \tilde{p}_{ji} \quad (2.7.5)$$

$$(j=1, \dots, 2s; i=1, \dots, s).$$

Свойство 3:

$$p_{ji} \begin{cases} p_{ij}, & \text{если } i, j \in \{\mathbf{W}\}^- \text{ или } i, j \in \{\mathbf{W}\}^+; \\ \tilde{p}_{ij}, & \text{если } i \in \{\mathbf{W}\}^- \text{ и } j \in \{\mathbf{W}\}^+ \text{ или } i \in \{\mathbf{W}\}^+ \\ & \text{и } j \in \{\mathbf{W}\}^-. \end{cases} \quad (2.7.6)$$

Для доказательства равенств (2.7.2) необходимо предварительно доказать, что [16] сумма предельных вероятностей p_i и p_{i+s} состояний $\mathbf{W}^{(i)}$ и $\mathbf{W}^{(i+s)}$ постоянна для всех i ($i=1, 2, \dots, s$) и равна $\frac{1}{2^{n-i}}$, т. е.

$$p_i + p_{i+s} = \frac{1}{2^{n-i}} \quad (2.7.7)$$

$$(i=1, 2, \dots, s).$$

Для этого запишем систему уравнений, которой удовлетворяют предельные вероятности состояний системы

$$p_i = \sum_{j=1}^{2s} p_{ji} p_j \quad (i=1, \dots, s); \quad (2.7.8)$$

$$p_{i+s} = \sum_{j=1}^{2s} p_{j,j+s} p_j.$$

Используя 1-е и 2-е свойства вероятностей p_{ij} , систему уравнений (2.7.8) запишем в виде

$$p_i = \sum_{j=1}^s p_{ji} (p_j + p_{j+s}); \quad (2.7.9)$$

$$p_{i+s} = \sum_{j=1}^s \tilde{p}_{ji} (p_j + p_{j+s})$$

$$(i=1, \dots, s).$$

Сложив уравнения системы (2.7.9), соответствующие состояниям $\mathbf{W}^{(i)}$ и $\mathbf{W}^{(i+s)}$, получим

$$p_i + p_{i+s} = \sum_{j=1}^s (p_{ji} + \tilde{p}_{ji}) (p_j + p_{j+s}) \quad (2.7.10)$$

$(i = 1, \dots, s).$

Смысл формулы (2.7.7) состоит в том, что система (2.7.10) имеет следующее решение:

$$p_i = \frac{1}{2^{n-1}} - p_{i+s} \quad (2.7.11)$$

$(i = 1, \dots, s).$

Действительно, подставив (2.7.11) в систему (2.7.10), получим

$$\sum_{j=1}^s (p_{ji} + \tilde{p}_{ji}) = 1 \quad (2.7.12)$$

$(i = 1, \dots, s).$

Из 1—3-го свойств вероятностей p_{ji} следует

$$\sum_{j=1}^s (p_{ji} + \tilde{p}_{ji}) = \sum_{j=1}^{2s} p_{ij} = 1, \quad (2.7.13)$$

что и завершает доказательство формулы (2.7.7).

Выражения (2.7.2) получаем, подставляя соотношения (2.7.11) в уравнения системы (2.7.9).

Рассмотрим случай с помехой [16]. Предполагается, что на вычисленное значение функции $Q(\mathbf{X})$ накладывается помеха $\epsilon(\sigma)$, представляющая нормально распределенную случайную величину с математическим ожиданием, равным нулю, и дисперсией σ^2 . Тогда поощрение или штрафование автоматов коллектива производится одновременно в зависимости от знака суммы

$$\Delta Q' = \Delta Q + \Delta \epsilon, \quad (2.7.14)$$

где $\Delta \epsilon$ — нормально распределенная случайная величина

с математическим ожиданием, равным нулю, и дисперсией $2\sigma^2$.

Обозначим

$$s_i = \text{Вер} (\Delta Q^{(i)} > 0 / \Delta Q^{(i)} < 0); \tag{2.7.15}$$

$$\tilde{s}_i = 1 - s_i = \text{Вер} (\Delta Q^{(i)} < 0 / \Delta Q^{(i)} < 0).$$

В силу линейности функции $Q(\mathbf{X})$ и симметричности распределения помехи $\varepsilon(\delta)$ относительно ее нулевого математического ожидания для состояний $i \in \{\mathbf{W}\}^+$ ($i = s+1, \dots, 2s$) вероятности штрафа и нештрафа соответственно равны

$$\text{Вер} (\Delta Q^{(i)} > 0 / \Delta Q^{(i)} > 0) = \tilde{s}_i; \tag{2.7.16}$$

$$\text{Вер} (\Delta Q^{(i)} < 0 / \Delta Q^{(i)} > 0) = s_i,$$

т. е.

$$\tilde{s}_i = s_{i+s}; \quad s_i = \tilde{s}_{i+s}. \tag{2.7.17}$$

Докажем, что предельные вероятности p_i и p_{i+s} состояний $\mathbf{W}^{(i)}$ вектора \mathbf{W} ($i = 1, \dots, s$) при наличии помех определяются по формулам [16]

$$p_i = \frac{1}{2^{n-1}} \sum_{j=1}^s (\tilde{s}_j p_{ji} + s_j \tilde{p}_{ji}); \tag{2.7.18}$$

$$p_{i+s} = \frac{1}{2^{n-1}} \sum_{j=1}^s (s_j p_{ji} + \tilde{s}_j \tilde{p}_{ji})$$

$$(i = 1, \dots, s).$$

Предварительно покажем, что в случае действия помех справедливо то же утверждение, что и для случая, когда нет помехи, т. е.

$$p_i + p_{i+s} = \frac{1}{2^{n-1}} \tag{2.7.19}$$

$$(i = 1, \dots, s).$$

Запишем систему уравнений, которой удовлетворяют стационарные вероятности p_i и p_{i+s} состояний $\mathbf{W}^{(i)}$ и $\mathbf{W}^{(i+s)}$ ($i = 1, \dots, s$):

$$p_i = \sum_{j=1}^{2s} (\tilde{s}_j p_{ji} + s_j \tilde{p}_{ji}) p_j; \quad (2.7.20)$$

$$p_{i+s} = \sum_{j=1}^{2s} (s_j p_{ji} + \tilde{s}_j \tilde{p}_{ji}) p_j$$

$$(i=1, \dots, s).$$

Используя свойства вероятностей p_{ji} (2.7.6) и s_i (2.7.15), (2.7.17), запишем систему (2.7.20) в виде

$$p_i = \sum_{j=1}^s (\tilde{s}_j p_{ji} + s_j \tilde{p}_{ji}) (p_j + p_{j+s}); \quad (2.7.21)$$

$$p_{i+s} = \sum_{j=1}^s (s_j p_{ji} + \tilde{s}_j \tilde{p}_{ji}) (p_j + p_{j+s}).$$

Складывая уравнения системы (2.7.21), соответствующие состояниям $\mathbf{W}^{(i)}$ и $\mathbf{W}^{(i+s)}$ ($i=1, \dots, s$), получим

$$p_i + p_{i+s} = \sum_{j=1}^s (p_{ji} + \tilde{p}_{ji}) (p_i + p_{i+s}) \quad (2.7.22)$$

$$(i=1, \dots, s).$$

Подставив (2.7.19) в (2.7.22), докажем высказанное утверждение:

$$\sum_{j=1}^s (p_{ji} + \tilde{p}_{ji}) = \sum_{j=1}^{2s} p_{ij} = 1. \quad (2.7.23)$$

Формулы (2.7.18) для предельных вероятностей состояний получим, подставив (2.7.19) в (2.7.21).

§ 2.8. СРАВНЕНИЕ АЛГОРИТМОВ САМООБУЧЕНИЯ

В параграфе 2.6 было показано, что при оптимизации в двумерном пространстве алгоритм оптимизации коллективом независимых автоматов в отношении

среднего приращения функции $Q(\mathbf{X})$ на одном шаге эквивалентен алгоритму покоординатного самообучения при случайном поиске.

Это означает, что к определенным параметрам одного алгоритма можно подобрать такие параметры второго алгоритма, при которых результаты действия обоих будут одинаковыми. В этом параграфе рассмотрим другие аспекты эквивалентности алгоритмов самообучения.

**ЭКВИВАЛЕНТНОСТЬ АВТОМАТОВ
ПО РАСПРЕДЕЛЕНИЮ
ИХ НАЧАЛЬНЫХ СОСТОЯНИЙ [21]**

В работах [21, 26] показано, как определить вероятность выходного сигнала $\Delta \mathbf{X}^{(j)}$, если на вход автомата был подан входной сигнал c , для случая автомата, вероятности переходов и выходов которого зависят только от его состояния и от его входного сигнала, но не зависят от выходного сигнала на предыдущем такте. Вероятности переходов, описывающих алгоритм самообучения при случайном поиске, в общем случае зависят также от выходного сигнала автомата на предыдущем шаге. Распространим на этот случай методику и формулы, изложенные в работах [21, 26].

По формулам (2.1.8) и (2.1.9) имеем

$$\begin{aligned} \text{Вер} (\mathbf{W}_{N+1}^{(i_2)} / \mathbf{W}_N^{(i_1)}, \Delta \mathbf{X}_N^{(j)}, c) \cdot \text{Вер} (\Delta \mathbf{X}_N^{(j)} / \mathbf{W}_N^{(i_1)}) = \\ = q_{i_1 j} p_{i_1 i_2}^{(c)}. \end{aligned} \quad (2.8.1)$$

Просуммировав это выражение по всем j ($j=1, \dots, v$), находим вероятность перехода автомата из состояния $\mathbf{W}^{(i_1)}$ на N -м шаге в состояние $\mathbf{W}^{(i_2)}$ на $(N+1)$ -м шаге при подаче на его вход сигнала c :

$$\text{Вер} (\mathbf{W}_{N+1}^{(i_2)} / \mathbf{W}_N^{(i_1)}, c) = \sum_{j=1}^v q_{i_1 j} p_{i_1 i_2}^{(c)}, \quad (2.8.2)$$

где v — количество выходных сигналов автомата.

Переписывая это выражение в матричной форме, имеем следующие матрицы перехода A_0 и A_1 для автомата, переходы которого зависят от его выходов:

$$A_0 = \sum_{j=1}^v Q_i(j) A_0(j); \quad (2.8.3)$$

$$A_1 = \sum_{j=1}^v Q_i(j) A_1(j).$$

Далее, аналогично тому, как это сделано в работах [21, 26], введем матрицу

$$T(\Delta \mathbf{X}^{(j)}/c) = A_c Q_i(j), \quad (2.8.4)$$

где A_c — матрица (2.8.3);

$Q_i(j)$ — матрица (2.1.21).

Элемент матрицы (2.8.4) обозначает вероятность перехода автомата из состояния $\mathbf{W}^{(i_1)}$ в состояние $\mathbf{W}^{(i_2)}$ и появления на его выходе сигнала $\Delta \mathbf{X}^{(j)}$, если на вход автомата был подан сигнал c , т. е.

$$\begin{aligned} t_{i_1, i_2}(\Delta \mathbf{X}^{(j)}/c) &= p(\Delta \mathbf{X}^{(j)}, \mathbf{W}^{(i_2)}/\mathbf{W}^{(i_1)}, c) = \\ &= \left(\sum_{j=1}^v q_{i_1, j} p_{i_1, i_2}(j) \right) q_{i_2 j}. \end{aligned} \quad (2.8.5)$$

Тогда, согласно формуле (0.3.20), переходная матрица автомата при входной последовательности $(c_1 c_2 \dots c_N)$ и выходной $(\Delta \mathbf{X}_1 \Delta \mathbf{X}_2 \dots \Delta \mathbf{X}_N)$ определяется формулой

$$\begin{aligned} T(\Delta \mathbf{X}_1 \Delta \mathbf{X}_2 \dots \Delta \mathbf{X}_N / c_1 c_2 \dots c_N) &= T(\Delta \mathbf{X}_1 / c_1) T(\Delta \mathbf{X}_2 / c_2) \dots \\ &\dots T(\Delta \mathbf{X}_N / c_N). \end{aligned} \quad (2.8.6)$$

Введем обозначение

$$\begin{aligned} P_{\mathbf{P}_i^{(0)}}^{T_i}(\Delta \mathbf{X}_1 \Delta \mathbf{X}_2 \dots \Delta \mathbf{X}_N / c_1 c_2 \dots c_N) &= \mathbf{P}_i^{(0)} T_i(\Delta \mathbf{X}_1 \Delta \mathbf{X}_2 \dots \\ &\dots \Delta \mathbf{X}_N / c_1 c_2 \dots c_N) \mathbf{e}_i, \end{aligned} \quad (2.8.7)$$

где \mathbf{e}_i — вектор, элементы которого равны единице;

$\mathbf{P}_i^{(0)}$ — начальное распределение внутренних состояний автомата A_i ;

$$T_i(\Delta \mathbf{X}_1 \Delta \mathbf{X}_2 \dots \Delta \mathbf{X}_N / c_1 c_2 \dots c_N) —$$

переходная матрица входной-выходной последовательности автомата A_i .

Если для автомата A_1 с начальным распределением $\mathbf{P}_1^{(0)}$ и для автомата A_2 с начальным распределением $\mathbf{P}_2^{(0)}$ имеем

$$P_{\mathbf{P}_1^{(0)}}^{T_1} (\Delta \mathbf{X}_1 \Delta \mathbf{X}_2 \dots \Delta \mathbf{X}_N / c_1 c_2 \dots c_N) = P_{\mathbf{P}_2^{(0)}}^{T_2} (\Delta \mathbf{X}_1 \Delta \mathbf{X}_2 \dots \dots \Delta \mathbf{X}_N / c_1 c_2 \dots c_N) \quad (2.8.8)$$

при всех входных последовательностях $c_1 c_2 \dots c_N$ и всех выходных последовательностях $\Delta \mathbf{X}_1 \Delta \mathbf{X}_2 \dots \Delta \mathbf{X}_N$, имеющих длину N , то системы $(\mathbf{P}_1^{(0)}, T_1)$ и $(\mathbf{P}_2^{(0)}, T_2)$ называются N -эквивалентными. Системы $(\mathbf{P}_1^{(0)}, T_1)$ и $(\mathbf{P}_2^{(0)}, T_2)$ называются эквивалентными, если они N -эквивалентны для всех N .

Достаточным условием эквивалентности двух систем является их $(m_1 + m_2 - 1)$ -эквивалентность, где m_1 и m_2 — количество внутренних состояний соответственно автомата A_1 и автомата A_2 .

ВЕРоятности соотношений входов— выхода алгоритмов самообучения

Рассмотрим одномерный случай ($n = 1$). Построим матрицы $T(\Delta \mathbf{X}^{(j)} / c)$ для алгоритма случайного поиска с самообучением. Нумерация состояний автомата и переходы из одного состояния в другое для этого алгоритма при различных значениях c и $\Delta \mathbf{X}^{(j)}$ показаны на рис. 2.8.1.

Пусть $\Delta x^{(1)} = 1$ и $\Delta x^{(2)} = -1$, так как в одномерном случае выходы $\Delta \mathbf{X}^{(j)}$ и состояния $\mathbf{W}^{(i)}$ автомата являются не векторами, а числами. Тогда составленные по рис. 2.8.1 и формуле (2.8.2) матрицы переходов при фиксированном входном сигнале c и при учете вероятностей выходного сигнала $\Delta \mathbf{X}^{(j)}$ на N -м шаге имеют вид (2.8.9) — (2.8.12).

$$Q_i(1)A_0(1) = \left\| \begin{array}{cccccc} q_{11} & 0 & 0 & \dots & 0 & 0 \\ q_{21} & 0 & 0 & \dots & 0 & 0 \\ 0 & q_{31} & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & q_{2m_1,1} & 0 \end{array} \right\| \quad (2.8.9)$$

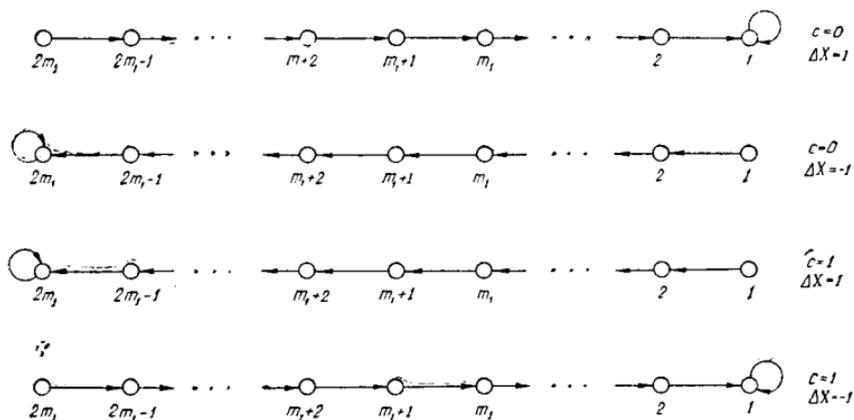


Рис. 2.8.1. Графы переходов автомата с переходной функцией, зависящей от выхода автомата для $n=1$.

$$Q_i(2)A_0(2) = \begin{vmatrix} 0 & q_{12} & 0 & 0 & \dots & 0 \\ 0 & 0 & q_{22} & 0 & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & \dots & q_{2m_i-1,2} \\ 0 & 0 & 0 & 0 & \dots & q_{2m_i,2} \end{vmatrix} \quad (2.8.10)$$

$$Q_i(1)A_1(1) = \begin{vmatrix} 0 & q_{11} & 0 & 0 & \dots & 0 \\ 0 & 0 & q_{21} & 0 & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & \dots & q_{2m_i-1,1} \\ 0 & 0 & 0 & 0 & \dots & q_{2m_i,1} \end{vmatrix} \quad (2.8.11)$$

$$Q_i(2)A_1(2) = \begin{vmatrix} q_{12} & 0 & 0 & \dots & 0 & 0 \\ q_{22} & 0 & 0 & \dots & 0 & 0 \\ 0 & q_{32} & 0 & \dots & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \dots & q_{2m_i,2} & 0 \end{vmatrix} \quad (2.8.12)$$

$$T_1(\Delta X^{(j)}/0) =$$

$$= \begin{vmatrix} q_{11}q_{1j} & q_{12}q_{2j} & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ q_{21}q_{1j} & 0 & q_{22}q_{3j} & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & q_{31}q_{2j} & 0 & q_{32}q_{4j} & 0 & \dots & 0 & 0 & 0 \\ \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots & q_{2m_1-1,1}q_{2m_1-2,j} & 0 & q_{2m_1-1,2}q_{2m_1,j} \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & q_{2m_1,1}q_{2m_1-1,j} & q_{2m_1,2}q_{2m_1,j} \end{vmatrix}$$

(j = 1, 2)

(2.8.13)

$$T_1(\Delta X^{(j)}/1) =$$

$$= \begin{vmatrix} q_{12}q_{1j} & q_{11}q_{2j} & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ q_{22}q_{1j} & 0 & q_{21}q_{3j} & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & q_{32}q_{2j} & 0 & q_{31}q_{4j} & 0 & \dots & 0 & 0 & 0 \\ \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots & q_{2m_1-1,2}q_{2m_1-2,j} & 0 & q_{2m_1-1,1}q_{2m_1,j} \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & q_{2m_1,2}q_{2m_1-1,j} & q_{2m_1,1}q_{2m_1,j} \end{vmatrix}$$

(j = 1, 2)

(2.8.14)

По матрицам (2.8.9) — (2.8.12) и формуле (2.8.4) находим матрицы (2.8.13) и (2.8.14) переходов автомата для фиксированных его входных $c=0, 1$ и выходных $\Delta\mathbf{X}^{(j)}$ ($j=1, 2$) сигналов.

Из формулы (2.8.7) с учетом матриц (2.8.13) и (2.8.14) имеем следующие выражения для определения вероятности появления сигнала $\Delta\mathbf{X}^{(j)}$ ($j=1, 2$) на выходе автомата, если на его вход был подан сигнал c ($c=0, 1$):

$$\begin{aligned} \text{Вер}(\Delta\mathbf{X}^{(j)}/0) &= \mathbf{P}_1^{(0)} T_1(\Delta\mathbf{X}^{(j)}/0) \mathbf{e}_1 = p_1^{(0)} (q_{11}q_{1j} + q_{12}q_{2j}) + \\ &+ \sum_{i=2}^{2m_1-1} p_i^{(0)} (q_{i1}q_{i-1,j} + q_{i2}q_{i+1,j}) + \\ &+ p_{2m_1}^{(0)} (q_{2m_1,1} q_{2m_1-1,j} + q_{2m_1,2} q_{2m_1,j}); \end{aligned} \quad (2.8.15)$$

$$\begin{aligned} \text{Вер}(\Delta\mathbf{X}^{(j)}/1) &= \mathbf{P}_1^{(0)} T_1(\Delta\mathbf{X}^{(j)}/1) \mathbf{e}_1 = p_1^{(0)} (q_{12}q_{1j} + q_{11}q_{2j}) + \\ &+ \sum_{i=2}^{2m_1-1} p_i^{(0)} (q_{i2}q_{i-1,j} + q_{i1}q_{i+1,j}) + \\ &+ p_{2m_1}^{(0)} (q_{2m_1,2}q_{2m_1-1,j} + q_{2m_1,1}q_{2m_1,j}), \end{aligned} \quad (2.8.16)$$

где $\mathbf{P}_1^{(0)}$ — вектор начальных вероятностей состояний автомата A_1 (алгоритма покоординатного самообучения);

$$\mathbf{P}_1^{(0)} = (p_1^{(0)}, \dots, p_{2m_1}^{(0)}); \quad (2.8.17)$$

$T_1(\Delta\mathbf{X}^{(j)}/c)$ определены формулами (2.8.13), (2.8.14); \mathbf{e}_1 — единичный вектор размерности $2m_1$:

$$\mathbf{e}_1 = \left. \begin{array}{|c|} \hline 1 \\ \hline 1 \\ \hline \vdots \\ \hline \vdots \\ \hline 1 \\ \hline \end{array} \right\} 2m_1 \quad (2.8.18)$$

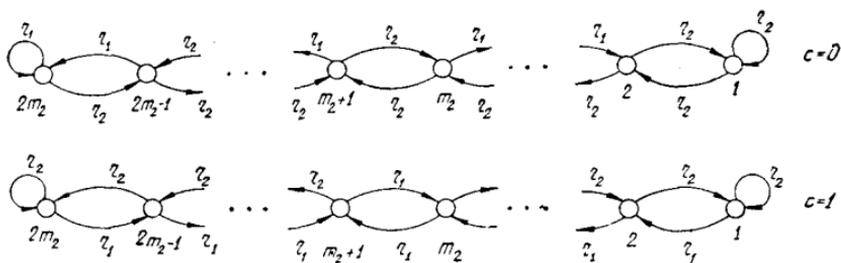


Рис. 2.8.2. Графы переходов автомата с детерминированной функцией выхода для $n=1$.

Найдем аналогичные выражения для вероятностей выходных сигналов $\Delta X^{(j)}$ при оптимизации коллективом независимых вероятностных автоматов. Нумерация состояний автомата и переходы из одного состояния в другое для этого алгоритма показаны на рис. 2.8.2. Предположим, что в состояниях от 1 до m_2 выход автомата равен $+1$, а в состояниях от m_2+1 до $2m_2$ равен -1 , т. е.

$$\Delta X = \begin{cases} +1, & \text{если } W = W^{(i)} \quad (i=1, \dots, m_2); \\ -1, & \text{если } W = W^{(i)} \quad (i=m_2+1, \dots, 2m_2). \end{cases} \quad (2.8.19)$$

Следовательно, матрицы $Q_i(j)$ для этого алгоритма имеют вид

$$Q_i(1) = \left\| \left\| \begin{array}{cc} I & 0 \\ 0 & 0 \end{array} \right\| \right\|_{m_2}^{m_2} \quad Q_i(2) = \left\| \left\| \begin{array}{cc} 0 & 0 \\ 0 & I \end{array} \right\| \right\| \quad (2.8.20)$$

$\underbrace{\hspace{10em}}_{m_2} \quad \underbrace{\hspace{10em}}_{m_2}$

где I — единичная матрица порядка m_2 .

Так как переходы этого автомата не зависят от его выхода, а зависят только от сигнала, поступающего на его вход, то, согласно рис. 2.8.2, автомат будет иметь переходные матрицы вида (2.8.21) и (2.8.22),

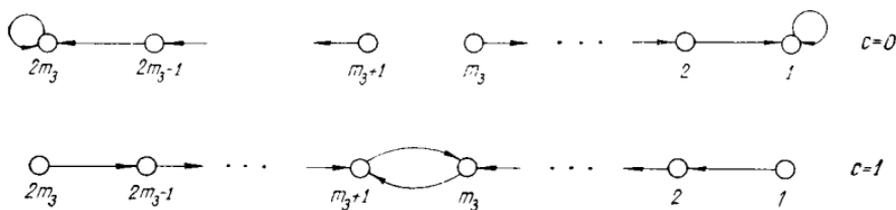


Рис. 2.8.3. Графы переходов автомата с детерминированной функцией перехода для $n=1$.

$$A_0(1) = A_0(2) = R_0 = \left(\begin{array}{cccccccccccc} 1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & 1 & 0 & \dots & 0 & 0 \\ \dots & \dots \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 1 \end{array} \right) \quad (2.8.31)$$

$$A_1(1) = A_1(2) = R_1 = \left(\begin{array}{cccccccccccc} 0 & 1 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 & 0 & \dots & 0 & 0 & 0 \\ \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 1 & 0 \end{array} \right) \quad (2.8.32)$$

У этого автомата вероятности q_{ij} выходов $\Delta X^{(j)}$ определяются матрицей Q_{ij} (2.1.21), элементы которой могут отличаться от нуля и единицы. T_3 ($\Delta X^{(j)}/c$) описываются матрицами (2.8.33) и (2.8.34).

Вероятность появления на выходе автомата сигнала $\Delta X^{(j)}$ при подаче на его вход сигнала c равна

$$\begin{aligned} \text{Вер} (\Delta X^{(j)}/0) = P_3^{(0)} T_3 (\Delta X^{(j)}/0) e_3 = p_1^{(0)} q_{1j} + \sum_{i=2}^{m_3} p_i^{(0)} \times \\ \times q_{i-1,j} + \sum_{i=m_3+1}^{2m_3-1} p_i^{(0)} q_{i+1,j} + p_{2m_3}^{(0)} q_{2m_3,j} \quad (2.8.35) \end{aligned}$$

($j = 1, 2$);

$$\begin{aligned} \text{Вер} (\Delta X^{(j)}/1) = P_3^{(0)} T_3 (\Delta X^{(j)}/1) e_3 = \sum_{i=1}^{m_3} p_i^{(0)} q_{i+1,j} + \\ + \sum_{i=m_3+1}^{2m_3} p_i^{(0)} q_{m_3-1,j} \quad (j = 1, 2). \quad (2.8.36) \end{aligned}$$

СОПОСТАВЛЕНИЕ АЛГОРИТМОВ САМООБУЧЕНИЯ

Сопоставим рассмотренные алгоритмы самообучения, представленные в виде автоматов, и выясним, в каком смысле они эквивалентны. Сначала сравним автоматы с детерминированными переходами и случайными выходами и автоматы со случайными переходами и детерминированным выходом. Предположим, что у первого автомата имеются два состояния, а у второго $2m_2$ состояний. Тогда для первого автомата из формул (2.8.35) и (2.8.36) имеем

$$\text{Вер} (\Delta X^{(j)}/0) = p_1^{(0)} q_{1j} + p_2^{(0)} q_{2j}; \quad (2.8.37)$$

$$\text{Вер} (\Delta X^{(j)}/1) = p_2^{(0)} q_{1j} + p_1^{(0)} q_{2j} \quad (j = 1, 2),$$

где $\mathbf{P}^{(0)} = (p_1^{(0)}, p_2^{(0)})$ — начальное распределение состояний автомата с детерминированными переходами.

Автоматы будут 1-эквивалентными при данных начальных распределениях вероятностей $\mathbf{P}_2^{(0)}$ и $\mathbf{P}_3^{(0)}$, если равны выражения (2.8.37), (2.8.35) и (2.8.36), т. е. если

$$\begin{aligned} p_1^{(0)}q_{11} + p_2^{(0)}q_{21} &= b_1; \\ p_2^{(0)}q_{11} + p_1^{(0)}q_{21} &= b_2, \end{aligned} \quad (2.8.38)$$

где

$$b_1 = A - r_2 a = \sum_{i=1}^{m_2} \tilde{p}_i^{(0)} - r_2 (\tilde{p}_{m_2}^{(0)} - \tilde{p}_{m_2+1}^{(0)}); \quad (2.8.39)$$

$$b_2 = A - r_1 a = \sum_{i=1}^{m_2} \tilde{p}_i^{(0)} - r_1 (\tilde{p}_{m_2}^{(0)} - \tilde{p}_{m_2+1}^{(0)});$$

$\tilde{\mathbf{P}}^{(0)}$ — вектор начальных вероятностей состояний автомата со случайными переходами; $\tilde{\mathbf{P}}^{(0)} = (\tilde{p}_1^{(0)}, \dots, \tilde{p}_{2m_2}^{(0)})$.

Решая систему уравнений (2.8.38) относительно q_{11} и q_{21} , находим:

$$q_{11} = \frac{b_1 p_1^{(0)} - b_2 p_2^{(0)}}{p_1^{(0)} - p_2^{(0)}}; \quad q_{21} = \frac{b_2 p_1^{(0)} - b_1 p_2^{(0)}}{p_1^{(0)} - p_2^{(0)}}. \quad (2.8.40)$$

Можно показать, что вероятности q_{12} и q_{22} определяются формулами

$$q_{12} = 1 - q_{11}; \quad q_{22} = 1 - q_{21}. \quad (2.8.41)$$

Следовательно, для автомата со случайными переходами и детерминированными выходами и произвольным числом состояний $2m_2$ и произвольным начальным распределением состояний можно найти 1-эквивалентный ему автомат с детерминированными переходами и случайными выходами, имеющий два состояния, причем начальное распределение этих состояний произвольное, а вероятности выходов автомата определяются формулами (2.8.40) и (2.8.41).

Найдем требование, при выполнении которого вероятности выходов автомата с детерминированными переходами удовлетворяют условию симметрии

$$q_{11} + q_{21} = 1. \quad (2.8.42)$$

Из формул (2.8.40) имеем

$$q_{11} + q_{21} = \frac{b_1(p_1^{(0)} - p_2^{(0)}) + b_2(p_1^{(0)} - p_2^{(0)})}{p_1^{(0)} - p_2^{(0)}} = b_1 + b_2 = 2A - a. \quad (2.8.43)$$

По формулам (2.8.39) находим:

$$2A - a = 2 \sum_{i=1}^{m_2} \tilde{p}_i^{(0)} - \tilde{p}_{m_2}^{(0)} + \tilde{p}_{m_2+1}^{(0)} = \sum_{i=1}^{m_2-1} \tilde{p}_i^{(0)} - \sum_{i=m_2+2}^{2m_2} \tilde{p}_i^{(0)} + 1. \quad (2.8.44)$$

Отсюда следует, что рассмотренные автоматы 1-эквивалентны при выполнении следующего условия:

$$\sum_{i=1}^{m_2-1} \tilde{p}_i^{(0)} = \sum_{i=m_2+2}^{2m_2} \tilde{p}_i^{(0)}. \quad (2.8.45)$$

Этому требованию удовлетворяет, например, симметричное начальное распределение автомата A_2 .

Следовательно, если начальное распределение автомата со случайными переходами и детерминированными выходами удовлетворяет условию (2.8.45), то он 1-эквивалентен некоторому автомату с детерминированными переходами и случайными выходами, вероятности выходов которого удовлетворяют условию симметрии (2.8.42).

Поскольку в формуле (2.8.45) не содержатся вероятности $p_{m_2}^{(0)}$ и $p_{m_2+1}^{(0)}$, то случай, когда у автомата со случайными переходами имеются только два состояния, необходимо рассмотреть отдельно. Для этого случая по формулам (2.8.39) получаем:

$$b_1 = \tilde{p}_1^{(0)} - r_2(\tilde{p}_1^{(0)} - \tilde{p}_2^{(0)}); \quad (2.8.46)$$

$$b_2 = \tilde{p}_1^{(0)} - r_1(\tilde{p}_1^{(0)} - \tilde{p}_2^{(0)}).$$

Складывая эти выражения, имеем

$$b_1 + b_2 = 2\tilde{p}_1^{(0)} - \tilde{p}_1^{(0)} + \tilde{p}_2^{(0)} = \tilde{p}_1^{(0)} + \tilde{p}_2^{(0)} = 1. \quad (2.8.47)$$

Следовательно, в этом случае оба автомата 1-эквивалентны для произвольных начальных вероятностей состояний автоматов; при этом выполняется условие симметрии (2.8.42).

Теперь сопоставим автомат с детерминированными переходами и случайными выходами, у которого имеются два состояния, и автомат, представляющий алгоритм по координатного самообучения, имеющий $2m_1$ состояний. Обозначим через \bar{b}_1 и \bar{b}_2 правые части формул (2.8.15):

$$\begin{aligned} \bar{b}_1 = & \bar{p}_1^{(0)} (\bar{q}_{11}\bar{q}_{11} + \bar{q}_{12}\bar{q}_{21}) + \sum_{i=2}^{2m_1-1} \bar{p}_i^{(0)} (\bar{q}_{i1}\bar{q}_{i-1,1} + \\ & + \bar{q}_{i2}\bar{q}_{i+1,1}) + \bar{p}_{2m_1}^{(0)} (\bar{q}_{2m_1,1}\bar{q}_{2m_1-1,1} + \bar{q}_{2m_1,2}\bar{q}_{2m_1,1}); \end{aligned} \quad (2.8.48)$$

$$\begin{aligned} \bar{b}_2 = & \bar{p}_1^{(0)} (\bar{q}_{12}q_{11} + \bar{q}_{11}q_{21}) + \sum_{i=2}^{2m_1-1} \bar{p}_i^{(0)} (\bar{q}_{i2}\bar{q}_{i-1,1} + \\ & + \bar{q}_{i1}\bar{q}_{i+1,1}) + \bar{p}_{2m_1}^{(0)} (\bar{q}_{2m_1,2}\bar{q}_{2m_1-1,1} + \bar{q}_{2m_1,1}\bar{q}_{2m_1,1}). \end{aligned} \quad (2.8.49)$$

Приравнивая правые части формул (2.8.37) к правым частям формул (2.8.15), (2.8.16), получаем систему уравнений

$$\begin{aligned} p_1^{(0)}q_{11} + p_2^{(0)}q_{21} &= \bar{b}_1; \\ p_2^{(0)}q_{11} + p_1^{(0)}q_{21} &= \bar{b}_2 \end{aligned} \quad (2.8.50)$$

и находим ее решение:

$$q_{11} = \frac{\bar{b}_1 p_1^{(0)} - \bar{b}_2 p_2^{(0)}}{p_1^{(0)} - p_2^{(0)}}; \quad q_{21} = \frac{\bar{b}_2 p_1^{(0)} - \bar{b}_1 p_2^{(0)}}{p_1^{(0)} - p_2^{(0)}}. \quad (2.8.51)$$

Вероятности q_{12} и q_{22} выхода $\Delta X^{(2)}$ определяются формулами (2.8.41).

Следовательно, для автомата, представляющего собой алгоритм самообучения и имеющего произвольное число $2m_1$ состояний, причем начальные вероятности состояний произвольны, существует 1-эквивалентный ему автомат с детерминированными переходами и случайными выхо-

дами, имеющий два состояния, причем начальные вероятности этих состояний произвольны, а вероятности выходов автомата определяются формулами (2.8.51) и (2.8.41).

Найдем условие, при котором удовлетворяется требование (2.8.42). Из формул (2.8.51), (2.8.48) и (2.8.49) находим

$$\begin{aligned} \bar{b}_1 + \bar{b}_2 = & \bar{p}_1^{(0)} (\tilde{q}_{12}\tilde{q}_{11} + \tilde{q}_{11}\tilde{q}_{21} + \tilde{q}_{11}\tilde{q}_{11} + \tilde{q}_{12}\tilde{q}_{21}) + \\ & + \sum_{i=2}^{2m_1-1} \bar{p}_i^{(0)} (\tilde{q}_{i1}\tilde{q}_{i-1,1} + \tilde{q}_{i2}\tilde{q}_{i+1,1} + \tilde{q}_{i2}\tilde{q}_{i-1,1} + \\ & + \tilde{q}_{i1}\tilde{q}_{i+1,1}) + \bar{p}_{2m_1}^{(0)} (\tilde{q}_{m_1,1}\tilde{q}_{2m_1-1,1} + \tilde{q}_{2m_1,2}\tilde{q}_{2m_1,1} + \\ & + \tilde{q}_{2m_1,2}\tilde{q}_{2m_1-1,1} + \tilde{q}_{2m_1,1}\tilde{q}_{2m_1,2}) = \bar{p}_1^{(0)} (\bar{q}_{11} + \bar{q}_{21}) + \\ & + \sum_{i=2}^{2m_1-1} \bar{p}_i^{(0)} (\tilde{q}_{i-1,1} + \tilde{q}_{i+1,1}) + \bar{p}_{2m_1}^{(0)} (\tilde{q}_{2m_1-1,1} + \\ & + \tilde{q}_{2m_1,1}) = 1. \end{aligned} \quad (2.8.52)$$

Предположим, что функция выхода алгоритма самообучения имеет вид

$$\tilde{q}_{i1} = F(\omega^{(i)}) = \begin{cases} 1, & \text{если } \omega^{(i)} > d; \\ \frac{1}{2}(1 + \omega^{(i)}), & \text{если } |\omega^{(i)}| \leq d; \\ 0, & \text{если } \omega^{(i)} < -d; \end{cases} \quad (2.8.53)$$

$$\begin{aligned} \tilde{q}_{i2} = 1 - \tilde{q}_{i1} & \quad (2.8.54) \\ (d = \text{const}; 0 < d \leq 1) & \end{aligned}$$

и что состояния $\mathbf{W}^{(i)}$ автомата, представляющего собой алгоритм самообучения, определены формулами (2.1.92), т. е. что состояния автомата симметричны относительно нулевой точки

$$\omega^{(2m_1-i+1)} = -\omega^{(i)}, \quad (2.8.55)$$

Тогда вероятности \tilde{q}_{ij} ($i=1, \dots, 2m_1; j=1, 2$) удовлетворяют условию

$$\tilde{q}_{ij} + \tilde{q}_{2m_1-i+1, j} = 1. \quad (2.8.56)$$

Требование (2.8.52) приобретает вид

$$\begin{aligned} b_1 + b_2 = & (\tilde{p}_1^{(0)} - \tilde{p}_{2m_1}^{(0)}) (\tilde{q}_{11} + \tilde{q}_{21}) + 2\tilde{p}_{2m_1}^{(0)} + \sum_{i=2}^{m_1} (\tilde{p}_1^{(0)} - \\ & - \tilde{p}_{2m_1-i+1}^{(0)}) (\tilde{q}_{i-1,1} + \tilde{q}_{i+1,1}) + 2 \sum_{i=m_1+1}^{2m_1-1} \tilde{p}_i^{(0)} = 1. \end{aligned} \quad (2.8.57)$$

Это равенство соблюдается при выполнении условий

$$\tilde{p}_i^{(0)} = \tilde{p}_{2m_1-i+1}^{(0)}. \quad (2.8.58)$$

Следовательно, для автомата, представляющего собой алгоритм покоординатного самообучения с произвольным числом $2m_1$ состояний, начальные вероятности которых симметричны, т. е. удовлетворяют условию (2.8.58), существует 1-эквивалентный автомат с детерминированными переходами и случайными выходами, имеющий два состояния, причем начальные вероятности этих состояний произвольны, а вероятности выходов автомата определяются формулами (2.8.51) и (2.8.41) и удовлетворяют условию симметрии

$$q_{11} + q_{21} = 1.$$

Случай, когда у автомата алгоритма покоординатного самообучения имеются только два состояния, необходимо рассмотреть отдельно. Из формулы (2.8.52) с учетом условия (2.8.56) имеем

$$\begin{aligned} b_1 + b_2 = & \tilde{p}_1^{(0)} (\tilde{q}_{11} + \tilde{q}_{21}) + \tilde{p}_2^{(0)} (\tilde{q}_{11} + \tilde{q}_{21}) = \\ = & (\tilde{p}_1^{(0)} + \tilde{p}_2^{(0)}) (\tilde{q}_{11} + \tilde{q}_{21}) = \tilde{q}_{11} + \tilde{q}_{21} = 1. \end{aligned} \quad (2.8.59)$$

Следовательно, в этом случае автоматы 1-эквивалентны друг другу при произвольных начальных вероятностях их состояний.

§ 2.9. ТРОИЧНЫЙ АЛГОРИТМ ПОИСКА

В предыдущих параграфах были рассмотрены алгоритмы двоичного поиска, при которых по каждой координате x_i приращение Δx_i ($i=1, \dots, n$) может принимать одно из двух значений: -1 или $+1$. Здесь рассмотрим алгоритм троичного поиска, при котором приращение Δx_i принимает одно из трех значений: -1 , 0 или $+1$. Рассмотрим троичный поиск в процессе оптимизации коллективом независимых вероятностных автоматов [20]. Пусть в дискретные моменты времени t_N ($N=1, 2, \dots$) каждый автомат коллектива A_j с вероятностью λ_j ($0 \leq \lambda_j \leq 1$) участвует в процессе поиска, т. е. совершает смену состояний под воздействием входа $\Delta Q(t_N)$ и выдает на выход сигнал $\Delta x_j(t_N)$, или с вероятностью $1 - \lambda_j = \tilde{\lambda}_j$ не участвует в процессе поиска. В момент времени t_N состояния автоматов $\omega(t_N)$ и значения переменной $x_j(t_N)$ определяются по следующим формулам:

$$\omega_j(t_N) = \begin{cases} \Psi(\omega_j(t_{N-1}), \Delta Q(t_N)); \\ \omega_j(t_{N-1}), \text{ если автомат } A_j \text{ не работает;} \end{cases} \quad (2.9.1)$$

$$x_j(t_N) = x_j(t_{N-1}) + \Delta x_j(t_N), \quad (2.9.2)$$

где $\Delta x_j(t_N)$ — выходное действие автомата A_j в момент t_N , которое определяется однозначно состоянием памяти ω_j и участием A_j в процессе поиска;

$$\Delta x_j(t_N) = \begin{cases} -1 & \text{при } 1 \leq \omega_j \leq m; \\ +1 & \text{при } m+1 \leq \omega_j \leq 2m; \\ 0, & \text{если автомат } A_j \text{ не работает.} \end{cases} \quad (2.9.3)$$

Далее для простоты изложения рассмотрим одномерный случай ($n=1$). Пусть автомат имеет $2m$ состояний и переходы автомата из одного состояния в другое, когда он работает, задаются стохастическими матрицами

Введем новый автомат с количеством состояний, равным $4m$. Пусть первую группу состояний нового автомата образуют состояния исходного автомата совместно с событием «Автомат работает», а вторую группу — состояния автомата совместно с событием «Автомат не работает». Нумерация состояний и переходы из одного состояния в другое для нового автомата показаны на рис. 2.9.1.

По этому рисунку получаем следующие переходные матрицы автомата:

$$A_0 = \begin{vmatrix} R_0\lambda & R_0\tilde{\lambda} \\ I\lambda & I\tilde{\lambda} \end{vmatrix} \quad A_1 = \begin{vmatrix} R_1\lambda & R_1\tilde{\lambda} \\ I\lambda & I\tilde{\lambda} \end{vmatrix} \quad (2.9.6)$$

где R_0 и R_1 — матрицы (2.9.4);

I — единичная матрица (2.9.5);

λ — вероятность того, что автомат работает;

$\tilde{\lambda} = 1 - \lambda$ — вероятность того, что автомат не работает.

Объединяя матрицы A_0 и A_1 с учетом вероятностей штрафов действий $\Delta X^{(i)}$ автомата, получаем матрицу A ,

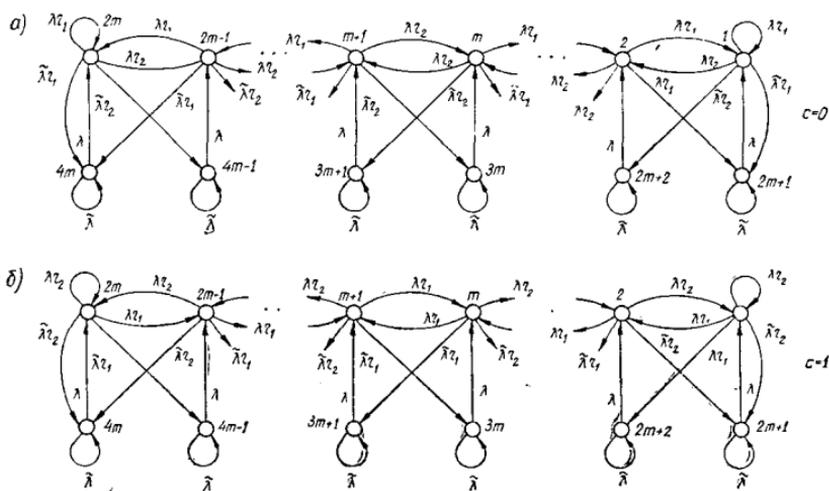


Рис. 2.9.1. Графы переходов автомата, описывающего алгоритм тройного поиска:

а — при нештрафе; б — при штрафе.

которая описывает функционирование автомата в случайной среде:

$$A = \begin{vmatrix} R\lambda & R\tilde{\lambda} \\ I\lambda & I\tilde{\lambda} \end{vmatrix} \quad (2.9.7)$$

Здесь матрица R имеет вид (2.9.8). Ее элементы имеют вид

$$v_i = r_1 s_i + r_2 (1 - s_i) = r_2 + (r_1 - r_2) s_i;$$

$$1 - v_i = r_1 - (r_1 - r_2) s_i;$$

здесь s_i — вероятность штрафа действия $\Delta X^{(i)}$ автомата в i -м состоянии;

$$s_i = \frac{1}{2} \left[1 + \Phi \left(\frac{\Delta Q^{(i)}}{2\sigma} \right) \right]. \quad (2.9.9)$$

Далее предполагается, что функция качества $Q(X)$ линейна. Тогда $\Delta Q^{(i)} = \alpha$ ($i = 1, \dots, m$) и $\Delta Q^{(i)} = -\alpha$ ($i = m + 1, \dots, 2m$). Следовательно,

$$s = s_1 = \dots = s_m = 1 - s_{m+1} = \dots = 1 - s_{2m} = \frac{1}{2} \left[1 + \Phi \left(\frac{\alpha}{2\sigma} \right) \right], \quad (2.9.10)$$

где $\alpha = |\text{grad } Q(X)|$. С учетом формул (2.9.9) и (2.9.10) матрица R принимает вид

$$R = \begin{vmatrix} v_1 & 1-v_1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ v_1 & 0 & 1-v_1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & v_1 & 0 & 1-v_1 & 0 & \dots & 0 & 0 & 0 & 0 \\ \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots & v_1 & 0 & 1-v_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & v_1 & 0 & 1-v_1 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & v_1 & 1-v_1 \end{vmatrix} \quad \left. \vphantom{\begin{vmatrix} \dots \\ \dots \\ \dots \\ \dots \\ \dots \\ \dots \end{vmatrix}} \right\} 2m \quad (2.9.11)$$

$$R = \begin{pmatrix} v_1 & 1-v_1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ v_2 & 0 & 1-v_2 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & v_3 & 0 & 1-v_3 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots & v_m & 0 & 1-v_m & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 1-v_{m+1} & 0 & v_m & 0 & \dots & 0 & 0 & 0 \\ \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 1-v_{2m-1} & 0 & v_{2m-1} \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 1-v_{2m} & v_{2m} \end{pmatrix}$$

(2.9.8)

Из матрицы (2.9.7) получаем систему векторных уравнений для нахождения предельных вероятностей состояний автомата:

$$\begin{aligned} \mathbf{P}_1 &= \mathbf{P}_1 R \lambda + \mathbf{P}_2 \tilde{I} \tilde{\lambda}; \\ \mathbf{P}_2 &= \mathbf{P}_1 R \lambda + \mathbf{P}_2 \tilde{I} \tilde{\lambda}, \end{aligned} \quad (2.9.12)$$

где

$$\begin{aligned} \mathbf{P}_1 &= (p_1, \dots, p_{2m}); \\ \mathbf{P}_2 &= (p_{2m+1}, \dots, p_{4m}). \end{aligned} \quad (2.9.13)$$

Из системы векторных уравнений (2.9.12) находим

$$\mathbf{P}_2 = \frac{\tilde{\lambda}}{\lambda} \mathbf{P}_1. \quad (2.9.14)$$

Подставляя это выражение в первое уравнение системы (2.9.12), получаем

$$\mathbf{P}_1 = \mathbf{P}_1 (\lambda R + \tilde{\lambda} I), \quad (2.9.15)$$

где матрица $\lambda R + \tilde{\lambda} I$ имеет вид (2.9.16).

Решая систему уравнений (2.9.15) и используя формулу (2.9.14), находим

$$p_i = \begin{cases} \left(\frac{1-v_1}{v_1} \right)^{i-1} p_1 & \text{при } i=2, \dots, 2m; \\ \frac{\tilde{\lambda}}{\lambda} \left(\frac{1-v_1}{v_1} \right)^{i-1} p_1 & \text{при } i=2m+1, \dots, 4m. \end{cases} \quad (2.9.17)$$

После нормировки вероятностей p_i ($i=1, \dots, 4m$) получаем

$$\lambda R + \tilde{\lambda} I = \begin{vmatrix} \lambda v_1 + \tilde{\lambda} & \lambda(1-v_1) & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ \lambda v_1 & \tilde{\lambda} & \lambda(1-v_1) & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & \lambda v_1 & \tilde{\lambda} & \lambda(1-v_1) & 0 & \dots & 0 & 0 & 0 & 0 \\ \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots & \lambda v_1 & \tilde{\lambda} & \lambda(1-v_1) & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & \lambda v_1 & \tilde{\lambda} & \lambda(1-v_1) \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & \lambda v_1 & \tilde{\lambda} + \lambda(1-v_1) \end{vmatrix}$$

(2.9.16)

$$p_i = \begin{cases} \lambda \left(\frac{1-v_1}{v_1} \right)^{i-1} \frac{1}{\sum_{j=1}^{2m} \left(\frac{1-v_1}{v_1} \right)^{j-1}} & \text{при } i=1, \dots, 2m; \\ \lambda \left(\frac{1-v_1}{v_1} \right)^{i-1} \frac{1}{\sum_{j=1}^{2m} \left(\frac{1-v_1}{v_1} \right)^{j-1}} & \text{при } i=2m+ \\ & +1, \dots, 4m. \end{cases} \quad (2.9.18)$$

Поскольку $r_1 > r_2$, имеем

$$0 < \frac{1-v_1}{v_1} \leq 1. \quad (2.9.19)$$

По формуле суммы геометрической прогрессии находим

$$\sum_{j=1}^{2m} \left(\frac{1-v_1}{v_1} \right)^{j-1} = \frac{v_1^{2m} - (1-v_1)^{2m}}{v_1^{2m-1}(2v_1-1)}. \quad (2.9.20)$$

В окончательном виде формулу предельных вероятностей p_i можем записать так:

$$p_i = \begin{cases} \lambda \frac{(1-v_1)^{i-1} v_1^{2m-i} (2v_1-1)}{v_1^{2m} - (1-v_1)^{2m}} & \text{при } i=1, \dots, 2m; \\ \lambda \frac{(1-v_1)^{i-1} v_1^{2m-i} (2v_1-1)}{v_1^{2m} - (1-v_1)^{2m}} & \text{при } i=2m+ \\ & +1, \dots, 4m. \end{cases} \quad (2.9.21)$$

Среднее приращение функции качества на одном шаге вычисляется по формуле

$$M[\Delta Q] = \sum_{i=1}^m p_i - \sum_{i=m+1}^{2m} p_i. \quad (2.9.22)$$

Подставляя в эту формулу выражения (2.9.21), находим:

$$\begin{aligned}
 M[\Delta Q] &= \lambda \frac{(2v_1 - 1)}{v_1^{2m} - (1 - v_1)^{2m}} \left[\sum_{i=1}^m v_1^{2m-i} (1 - v_1)^{i-1} - \right. \\
 &\quad \left. - \sum_{i=m+1}^{2m} v_1^{2m-i} (1 - v_1)^{i-1} \right] = \lambda \frac{2v_1 - 1}{v_1^{2m} - (1 - v_1)^{2m}} \times \\
 &\quad \times \frac{v_1^{2m}}{1 - v_1} \left[\sum_{i=1}^m \left(\frac{1 - v_1}{v_1} \right) - \sum_{i=m+1}^{2m} \left(\frac{1 - v_1}{v_1} \right)^i \right] = \\
 &= \lambda \frac{(2v_1 - 1) v_1^{2m}}{[v_1^{2m} - (1 - v_1)^{2m}] (1 - v_1)} \left[1 - \left(\frac{1 - v_1}{v_1} \right)^m \right] \times \\
 &\quad \times \sum_{i=1}^m \left(\frac{1 - v_1}{v_1} \right)^i = \lambda \frac{v_1^m - (1 - v_1)^m}{v_1^m + (1 - v_1)^m} = \\
 &= \lambda \frac{1 - \left(\frac{1 - v_1}{v_1} \right)^m}{1 + \left(\frac{1 - v_1}{v_1} \right)^m}, \tag{2.9.23}
 \end{aligned}$$

где

$$\frac{1 - v_1}{v_1} = \frac{r_1 - (r_1 - r_2) s_1}{r_2 + (r_1 - r_2) s_1}. \tag{2.9.24}$$

Для случая, когда отсутствует помеха, $s_1 = 1$ и

$$\frac{1 - v_1}{v_1} = \frac{r_2}{r_1}, \tag{2.9.25}$$

из формулы (2.9.23) при учете соотношения (2.9.25) имеем

$$M[\Delta Q] = \lambda \frac{1 - \left(\frac{r_2}{r_1} \right)^m}{1 + \left(\frac{r_2}{r_1} \right)^m}. \tag{2.9.26}$$

Полученная формула отличается от выражения (2.6.11) только множителем λ .

Следовательно, поскольку $0 < \lambda < 1$, то для случая линейной функции качества и при размерности $n = 1$ пространства $\{X\}$ алгоритм троичного поиска по сравнению с алгоритмом двоичного поиска имеет, как и ожидалось, более медленную сходимость.

ОПТИМАЛЬНЫЕ АЛГОРИТМЫ

§ 3.1. ПОСТАНОВКА ЗАДАЧИ СИНТЕЗА ОПТИМАЛЬНОГО АЛГОРИТМА В КЛАССЕ АЛГОРИТМОВ ДИСКРЕТНО РАСПРЕДЕЛЕННОГО СЛУЧАЙНОГО ПОИСКА

В главах I и II было показано, что алгоритм случайного поиска в процессе оптимизации можно интерпретировать как вероятностный автомат в случайной среде. Там же приведены конструкции автоматов, описывающих известные алгоритмы детерминированного и случайного поиска. В настоящем параграфе на базе представления алгоритмов поиска в виде вероятностных автоматов будем отыскивать оптимальные алгоритмы для оптимизации широкого класса объектов.

Рассмотрим класс алгоритмов дискретно распределенного поиска. Эти алгоритмы характеризуются тем, что смещения $\Delta \mathbf{X}$ в пространстве параметров $\mathbf{X} = (x_1, \dots, x_n)$ оптимизируемого объекта $Q(\mathbf{X})$ разрешаются по направлениям, число которых m конечно. Пусть дозволенные смещения $\Delta \mathbf{X}$ в пространстве $\{\mathbf{X}\}$ определяются условием, что приращение координат ΔX_i ($i=1, \dots, n$) может принимать только два значения: $+1$ или -1 . Тогда число возможных направлений смещения $m=2^n$. Индексы этих направлений строим следующим образом:

$$\Delta X_{i+\frac{m}{2}} = -\Delta X_i \quad (3.1.1)$$

$$\left(i \leq \frac{m}{2} \right).$$

Алгоритмы поиска задаем в классе автоматов со следующими стохастическими матрицами перехода [53]:

$$\beta_0 = 1; \quad \beta_1 = \beta_2 = \dots = \beta_{\frac{m}{2}} = 0; \quad (3.1.6)$$

$$\gamma_0 = \gamma_1 = \dots = \gamma_{\frac{m}{2}} = \frac{1}{m}$$

имеем алгоритм случайного спуска [1].

Объект оптимизации задаем следующим образом. Пусть объект $Q(\mathbf{X})$ может находиться в одном из состояний, количество которых является конечным числом:

$$Q^{(j)}(\mathbf{X}) = \sum_{i=1}^n \alpha_i^{(j)} x_i, \quad (3.1.7)$$

где

$$\sum_{i=1}^n (\alpha_i^{(j)})^2 = 1 \quad (j=1, 2, \dots, v). \quad (3.1.8)$$

Здесь индекс j определяет состояние объекта $Q^{(j)}(\mathbf{X})$, т. е. направление его вектора градиента. Предположим, что изменение направления вектора градиента объекта в случае, когда не происходит возврата в предыдущее состояние, описывается однородной цепью Маркова со следующей матрицей переходных вероятностей:

$$\kappa_0 = \begin{vmatrix} \kappa_{11} & \kappa_{12} & \dots & \kappa_{1v} \\ \kappa_{21} & \kappa_{22} & \dots & \kappa_{2v} \\ \dots & \dots & \dots & \dots \\ \kappa_{v1} & \kappa_{v2} & \dots & \kappa_{vv} \end{vmatrix} \quad (3.1.9)$$

где

$$0 \leq \kappa_{ij} \leq 1; \quad \sum_{j=1}^v \kappa_{ij} = 1.$$

Когда алгоритм поиска делает возврат в предыдущее состояние, объект свое состояние не изменяет:

$$x_1 = \begin{vmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \end{vmatrix} \quad (3.1.10)$$

Следовательно, в первом случае действие объекта можно интерпретировать как функционирование составной случайной среды, а во втором — как функционирование стационарной случайной среды. Такое задание объекта оптимизации в виде пары стохастических матриц (3.1.9) и (3.1.10) соответствует заданию некоторого вероятностного автомата, j -е состояние которого соответствует j -му состоянию объекта $Q^{(j)}(\mathbf{X})$.

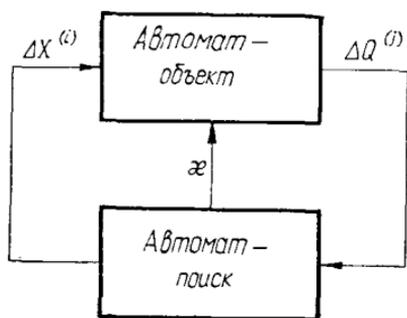


Рис. 3.1.1. Взаимодействие автомата-поиска и автомата-объекта.

На вход автомата, управляющего переходом из одного состояния в другое, может быть принят один из двух сигналов: 1) система поиска сделала случайный шаг;

2) система поиска сделала возврат в предыдущее состояние. Взаимодействие автомата-поиска и автомата-объекта показано на рис. 3.1.1.

Принимаем, что на объект $Q(\mathbf{X})$ действует нормально распределенная помеха $\varepsilon(\sigma)$ с математическим ожиданием, равным нулю, и дисперсией σ^2 , т. е. поиск экстремума осуществляется в соответствии с величиной

$$Q'(\mathbf{X}) = Q(\mathbf{X}) + \varepsilon(\sigma). \quad (3.1.11)$$

Тогда вероятность штрафа i -го действия автомата-поиска при нахождении объекта в j -м состоянии $s_i^{(j)}$ определяется формулой

$$s_i^{(j)} = \frac{1}{2} \left[1 + \Phi \left(-\frac{\Delta Q_i^{(j)}}{2\sigma} \right) \right] \quad (3.1.12)$$

где $\Delta Q_i^{(j)}$ — приращение оптимизируемой функции при нахождении ее градиента в j -м состоянии и

при смещении в пространстве параметров X по i -му направлению;

$\Phi(z)$ — интеграл вероятности.

Построим матрицы переходных вероятностей для цепей Маркова, описывающие функционирование обоих автоматов. Переходная матрица автомата-поиска зависит от состояния, в котором находится автомат-объект. Можно показать, что для j -го состояния объекта она имеет вид

$$A^{(j)} = \begin{vmatrix} a_{10}^{(j)} & a_{11}^{(j)} & \dots & a_{1, \frac{m}{2}-1}^{(j)} & a_{1, \frac{m}{2}}^{(j)} & a_{1, \frac{m}{2}-1}^{(j)} & \dots & a_{11}^{(j)} \\ a_{21}^{(j)} & a_{20}^{(j)} & \dots & a_{2, \frac{m}{2}-2}^{(j)} & a_{2, \frac{m}{2}-1}^{(j)} & a_{2, \frac{m}{2}}^{(j)} & \dots & a_{22}^{(j)} \\ \dots & \dots \\ a_{m1}^{(j)} & a_{m2}^{(j)} & \dots & a_{m, \frac{m}{2}}^{(j)} & a_{m, \frac{m}{2}-1}^{(j)} & a_{m, \frac{m}{2}-2}^{(j)} & \dots & a_{m0}^{(j)} \end{vmatrix} \quad (3.1.13)$$

Здесь

$$a_{ik}^{(j)} = \beta_k (1 - s_i^{(j)}) + \gamma_k s_i^{(j)} \quad (3.1.14)$$

$$\left(j = 1, 2, \dots, v; \quad i = 1, 2, \dots, m; \quad k = 0, 1, \dots, \frac{m}{2} \right),$$

где β_k и γ_k — элементы матриц (3.1.2) и (3.1.3).

Переходная матрица автомата-объекта зависит от состояния, в котором находится автомат-поиск. Для i -го состояния автомата-поиска она равна

$$\Delta^{(i)} = \begin{vmatrix} \delta_{11}^{(i)} & \dots & \delta_{1v}^{(i)} \\ \dots & \dots & \dots \\ \delta_{v1}^{(i)} & \dots & \delta_{vv}^{(i)} \end{vmatrix} \quad (3.1.15)$$

где

$$\delta_{jk}^{(i)} = \left[1 - \frac{\beta_m}{2} + \left(\frac{\beta_m - \gamma_m}{2} \right) s_i^{(j)} \right] \kappa_{jk} + \left[\frac{\beta_m}{2} + \left(\frac{\gamma_m - \beta_m}{2} \right) s_i^{(j)} \right] \eta_{jk}; \quad (3.1.16)$$

$$\eta_{jk} = \begin{cases} 1 & \text{при } j=k; \\ 0 & \text{при } j \neq k. \end{cases} \quad (3.1.17)$$

Далее будем говорить, что система поиск—объект находится в состоянии (i, j) , если автомат-поиск находится в состоянии ΔX_i , а автомат-объект — в состоянии $Q^{(j)}(X)$. Можно показать, что такая система поиск—объект образует более сложный автомат, поведение которого описывается цепью Маркова со следующей матрицей переходных вероятностей:

$$A = \begin{pmatrix} T_{10} & T_{11} & \dots & T_{1, \frac{m}{2}-1} & T_{1, \frac{m}{2}} & T_{1, \frac{m}{2}-1} & \dots & T_{11} \\ T_{21} & T_{20} & \dots & T_{2, \frac{m}{2}-2} & T_{2, \frac{m}{2}-1} & T_{2, \frac{m}{2}} & \dots & T_{22} \\ \dots & \dots \\ T_{m1} & T_{m2} & \dots & T_{m, \frac{m}{2}} & T_{m, \frac{m}{2}-1} & T_{m, \frac{m}{2}-2} & \dots & T_{m0} \end{pmatrix} \quad (3.1.18)$$

где

$$T_{ik} = \begin{pmatrix} a_{ik}^{(1)} \delta_{11}^{(i)} & \dots & a_{ik}^{(1)} \delta_{1v}^{(i)} \\ a_{ik}^{(2)} \delta_{21}^{(i)} & \dots & a_{ik}^{(2)} \delta_{2v}^{(i)} \\ \dots & \dots & \dots \\ a_{ik}^{(v)} \delta_{v1}^{(i)} & \dots & a_{ik}^{(v)} \delta_{vv}^{(i)} \end{pmatrix} \quad (3.1.19)$$

$a_{ik}^{(j)}$ определяются формулой (3.1.14); $\delta_{jk}^{(i)}$ — формулой (3.1.16). Если $s_i^{(j)}$ ($i=1, \dots, m$; $j=1, \dots, v$) постоянны, то цепь Маркова (3.1.18) является однородной, и в случае ее эргодичности вектор предельных вероятностей $\mathbf{P} = (p_1^{(1)}, \dots, p_1^{(v)}, p_2^{(1)}, \dots, p_2^{(v)}, \dots, p_m^{(1)}, \dots, p_m^{(v)})$ находится как решение системы алгебраических уравнений

$$P = PA, \quad (3.1.20)$$

где A — матрица (3.1.18);

$p_i^{(j)}$ — предельная вероятность того, что система поиск—объект находится в состоянии (i, j) .

Средний штраф автомата-поиска определим в виде

$$p_{\text{ср}} = \sum_{j=1}^v \sum_{i=1}^m p_i^{(j)} s_i^{(j)}, \quad (3.1.21)$$

а среднее смещение на одном шаге в пространстве параметров X по направлению градиента — в виде

$$M[\Delta Q] = \sum_{j=1}^v \sum_{i=1}^m \left(\Delta X_i p_i^{(j)}, \text{grad } Q^{(j)}(X) \right), \quad (3.1.22)$$

где скобки обозначают скалярное произведение векторов.

Очевидно, что предельные вероятности $p_i^{(j)}$, а также $p_{\text{ср}}$ и $M[\Delta Q]$ зависят от значений элементов матриц (3.1.2) и (3.1.3). Задача построения оптимального алгоритма поиска заключается в нахождении таких значений переходных вероятностей β_k и γ_k , при которых $p_{\text{ср}}$ или $M[\Delta Q]$ принимает минимальное значение. Отметим, что из общей модели объекта оптимизации (3.1.15) нетрудно получить различные более простые модели, в том числе линейную модель объекта оптимизации.

§ 3.2. ОПТИМАЛЬНЫЕ АЛГОРИТМЫ ПОИСКА ДЛЯ ДВУМЕРНОГО ОБЪЕКТА ОПТИМИЗАЦИИ

ПОСТАНОВКА ЗАДАЧИ ДЛЯ ДВУМЕРНОГО ОБЪЕКТА

В этом параграфе построим оптимальные алгоритмы для случая оптимизации различных двумерных объектов [54]. Пусть $n=2$; $m=4$; $v=2$. Тогда имеем матрицы переходных вероятностей

$$A_0 = \begin{vmatrix} \beta_0 & \beta_1 & \beta_2 & \beta_1 \\ \beta_1 & \beta_0 & \beta_1 & \beta_2 \\ \beta_2 & \beta_1 & \beta_0 & \beta_1 \\ \beta_1 & \beta_2 & \beta_1 & \beta_0 \end{vmatrix} \quad A_1 = \begin{vmatrix} \gamma_0 & \gamma_1 & \gamma_2 & \gamma_1 \\ \gamma_1 & \gamma_0 & \gamma_1 & \gamma_2 \\ \gamma_2 & \gamma_1 & \gamma_0 & \gamma_1 \\ \gamma_1 & \gamma_2 & \gamma_1 & \gamma_0 \end{vmatrix} \quad (3.2.1)$$

где

$$\beta_0 + 2\beta_1 + \beta_2 = 1; \quad \gamma_0 + 2\gamma_1 + \gamma_2 = 1. \quad (3.2.2)$$

Пусть объект оптимизации имеет следующие два состояния:

$$Q^{(1)}(\mathbf{X}) = x_1; \quad Q^{(2)}(\mathbf{X}) = -x_1. \quad (3.2.3)$$

Нетрудно увидеть, что вероятности штрафов $s_i^{(j)}$ для такого объекта равны

$$s_1 = s_1^{(1)} = s_4^{(1)} = s_2^{(2)} = s_3^{(2)} = \frac{1}{2} \left[1 + \Phi \left(\frac{\sqrt{2}}{2\sigma} \right) \right]; \quad (3.2.4)$$

$$s_2 = s_2^{(1)} = s_3^{(1)} = s_1^{(2)} = s_4^{(2)} = \frac{1}{2} \left[1 - \Phi \left(\frac{\sqrt{2}}{2\sigma} \right) \right].$$

Пусть объект изменяет свои состояния в соответствии с матрицами

$$\kappa_1 = \begin{vmatrix} \kappa_{11} & \kappa_{12} \\ \kappa_{21} & \kappa_{22} \end{vmatrix} \quad \kappa_2 = \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix} \quad (3.2.5)$$

Тогда переходная матрица (3.1.15) автомата-объекта имеет вид

$$\Delta^{(i)} = \begin{vmatrix} \delta_{11}^{(i)} & \delta_{12}^{(i)} \\ \delta_{21}^{(i)} & \delta_{22}^{(i)} \end{vmatrix} \quad (3.2.6)$$

где числа $\delta_{jk}^{(i)}$ определяются формулой (3.1.16). Учитывая равенства (3.2.4), из формулы (3.1.14) получаем:

$$a_{1k}^{(1)} = a_{4k}^{(1)} = a_{2k}^{(2)} = a_{3k}^{(2)} = a_{1k} = \beta_k (1 - s_1) + \gamma_k s_1; \quad (3.2.7)$$

$$a_{2k}^{(1)} = a_{3k}^{(1)} = a_{1k}^{(2)} = a_{4k}^{(2)} = a_{2k} = \beta_k (1 - s_2) + \gamma_k s_2$$

($k=0, 1, 2$).

Поэтому матрицы $A^{(j)}$ ($j=1, 2$) имеют вид

$$A^{(1)} = \begin{vmatrix} a_{10} & a_{11} & a_{12} & a_{11} \\ a_{21} & a_{20} & a_{21} & a_{22} \\ a_{22} & a_{21} & a_{20} & a_{21} \\ a_{11} & a_{12} & a_{11} & a_{10} \end{vmatrix} \quad A^{(2)} = \begin{vmatrix} a_{20} & a_{21} & a_{22} & a_{21} \\ a_{11} & a_{10} & a_{11} & a_{12} \\ a_{12} & a_{11} & a_{10} & a_{11} \\ a_{21} & a_{22} & a_{21} & a_{20} \end{vmatrix} \quad (3.2.8)$$

а матрица A — блочная:

$$A = \begin{vmatrix} T_{10} & T_{11} & T_{12} & T_{11} \\ T_{21} & T_{20} & T_{21} & T_{22} \\ T_{22} & T_{21} & T_{20} & T_{21} \\ T_{11} & T_{12} & T_{11} & T_{10} \end{vmatrix} \quad (3.2.9)$$

где

$$T_{ik} = \begin{vmatrix} a_{ik}^{(1)} \delta_{11}^{(i)} & a_{ik}^{(1)} \delta_{12}^{(i)} \\ a_{ik}^{(2)} \delta_{21}^{(i)} & a_{ik}^{(2)} \delta_{22}^{(i)} \end{vmatrix} \quad (3.2.10)$$

Отметим, что из модели (3.2.5), (3.2.6), которая является стохастической и учитывает действия системы поиска, можно получить различные модели объекта оптимизации. Так, например, если элементы матриц (3.2.6) не зависят от индекса i и от элементов матриц (3.2.5), то получим вполне стохастическую модель объекта, поведение которой не зависит от действий системы поиска. Если матрицу κ_1 задать в виде

$$\kappa_1 = \begin{vmatrix} 1 & 0 \\ 1 & 0 \end{vmatrix} \quad (3.2.11)$$

или матрицы $\Delta^{(i)}$ — в виде

$$\Delta^{(i)} = \begin{vmatrix} 1 & 0 \\ 1 & 0 \end{vmatrix} \quad (3.2.12)$$

$$(i=1, \dots, m),$$

то будем иметь линейную модель объекта оптимизации.

**АНАЛИЗ ПРОЦЕССА ОПТИМИЗАЦИИ
СТОХАСТИЧЕСКОЙ МОДЕЛИ, УЧИТЫВАЮЩЕЙ
ДЕЙСТВИЯ СИСТЕМЫ ПОИСКА**

Поведение такой системы поиск—объект описывается цепью Маркова с матрицей перехода (3.2.9). Пусть оба состояния среды равноценны, т. е. $\kappa_{11} = \kappa_{22} = \delta$. Тогда

$$\kappa_1 = \begin{vmatrix} \delta & 1-\delta \\ 1-\delta & \delta \end{vmatrix} \quad (3.2.13)$$

Матрицы $\Delta^{(1)}$ и $\Delta^{(2)}$, если учесть формулы (3.1.17), (3.2.4) и (3.2.13), имеют вид

$$\Delta^{(1)} = \begin{vmatrix} \delta_{11} & \delta_{12} \\ \delta_{21} & \delta_{22} \end{vmatrix} \quad \Delta^{(2)} = \begin{vmatrix} \delta_{22} & \delta_{21} \\ \delta_{12} & \delta_{11} \end{vmatrix} \quad (3.2.14)$$

где

$$\begin{aligned} \delta_{11} &= [1 - \beta_2 + (\beta_2 - \gamma_2) s_1] \delta + \beta_2 + (\gamma_2 - \beta_2) s_1; \\ \delta_{12} &= [1 - \beta_2 + (\beta_2 - \gamma_2) s_1] (1 - \delta); \\ \delta_{21} &= [1 - \beta_2 + (\beta_2 - \gamma_2) s_2] (1 - \delta); \\ \delta_{22} &= [1 - \beta_2 + (\beta_2 - \gamma_2) s_2] \delta + \beta_2 + (\gamma_2 - \beta_2) s_2. \end{aligned} \quad (3.2.15)$$

Систему уравнений (3.1.20) можно переписать в виде системы векторных уравнений

$$\begin{aligned} \mathbf{P}_1 &= \mathbf{P}_1 T_{10} + \mathbf{P}_2 T_{21} + \mathbf{P}_3 T_{22} + \mathbf{P}_4 T_{11}; \\ \mathbf{P}_2 &= \mathbf{P}_1 T_{11} + \mathbf{P}_2 T_{20} + \mathbf{P}_3 T_{21} + \mathbf{P}_4 T_{12}; \\ \mathbf{P}_3 &= \mathbf{P}_1 T_{12} + \mathbf{P}_2 T_{21} + \mathbf{P}_3 T_{20} + \mathbf{P}_4 T_{11}; \\ \mathbf{P}_4 &= \mathbf{P}_1 T_{11} + \mathbf{P}_2 T_{22} + \mathbf{P}_3 T_{21} + \mathbf{P}_4 T_{10}, \end{aligned} \quad (3.2.16)$$

где $\mathbf{P}_i = (p_i^{(1)}, p_i^{(2)})$;
 T_{ik} — матрицы (3.2.10).

Используя свойства этой системы, можно показать, что выполняются равенства

$$\mathbf{P}_1 = \mathbf{P}_4; \quad \mathbf{P}_2 = \mathbf{P}_3. \quad (3.2.17)$$

Поэтому

$$\begin{aligned} \mathbf{P}_1 &= \mathbf{P}_1(T_{10} + T_{11}) + \mathbf{P}_2(T_{21} + T_{22}); \\ \mathbf{P}_2 &= \mathbf{P}_1(T_{11} + T_{12}) + \mathbf{P}_2(T_{20} + T_{21}). \end{aligned} \quad (3.2.18)$$

Вводим обозначения

$$T_{10} + T_{11} = \begin{vmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{vmatrix} \quad T_{11} + T_{12} = \begin{vmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{vmatrix} \quad (3.2.19)$$

Учитывая вид матриц (3.2.14), имеем

$$T_{20} + T_{21} = \begin{vmatrix} g_{22} & g_{21} \\ g_{12} & g_{11} \end{vmatrix} \quad T_{21} + T_{22} = \begin{vmatrix} h_{22} & h_{21} \\ h_{12} & h_{11} \end{vmatrix} \quad (3.2.20)$$

С учетом матриц (3.2.19) и (3.2.20) из системы (3.2.18) получаем

$$p_1^{(1)} = p_2^{(2)} = p_1; \quad p_1^{(2)} = p_2^{(1)} = p_2. \quad (3.2.21)$$

Следовательно, предельные вероятности p_1 и p_2 можно получить из системы линейных однородных уравнений

$$\begin{aligned} p_1 &= (g_{11} + h_{12})p_1 + (g_{21} + h_{22})p_2; \\ p_2 &= (g_{12} + h_{11})p_1 + (g_{22} + h_{21})p_2. \end{aligned} \quad (3.2.22)$$

Из этой системы, с учетом условия $p_1 + p_2 = \frac{1}{4}$, находим:

$$\begin{aligned} p_1 &= \frac{1}{4} \cdot \frac{g_{21} + h_{22}}{g_{12} + h_{11} + g_{21} + h_{22}}; \\ p_2 &= \frac{1}{4} \cdot \frac{g_{12} + h_{11}}{g_{12} + h_{11} + g_{21} + h_{22}}, \end{aligned} \quad (3.2.23)$$

где

$$\begin{aligned} g_{12} &= (a_{10} + a_{11})\delta_{12}; \quad g_{21} = (a_{20} + a_{21})\delta_{21}; \\ h_{11} &= (a_{11} + a_{12})\delta_{11}; \quad h_{22} = (a_{21} + a_{22})\delta_{22}; \end{aligned} \quad (3.2.24)$$

a_{ik} определяются по формуле (3.2.7):

$$\begin{aligned}
 a_{10} &= \beta_0 s_2 + \gamma_0 s_1; & a_{20} &= \beta_0 s_1 + \gamma_0 s_2; \\
 a_{11} &= \beta_1 s_2 + \gamma_1 s_1; & a_{21} &= \beta_1 s_1 + \gamma_1 s_2; \\
 a_{12} &= \beta_2 s_2 + \gamma_2 s_1; & a_{22} &= \beta_2 s_1 + \gamma_2 s_2.
 \end{aligned}
 \tag{3.2.25}$$

Подставляя в формулы (3.2.23) выражения (3.2.24), имеем:

$$\begin{aligned}
 p_1 &= \frac{1}{4} \cdot \frac{a_{20} + a_{21} + (a_{22} - a_{20}) \delta_{22}}{a_{20} + a_{21} + (a_{22} - a_{20}) \delta_{22} + a_{10} + a_{11} + (a_{12} - a_{10}) \delta_{11}}; \\
 p_2 &= \frac{1}{4} \cdot \frac{a_{10} + a_{11} + (a_{12} - a_{10}) \delta_{11}}{a_{20} + a_{21} + (a_{22} - a_{20}) \delta_{22} + a_{10} + a_{11} + (a_{12} - a_{10}) \delta_{11}}.
 \end{aligned}
 \tag{3.2.26}$$

Средний штраф автомата поиска равен

$$p_{\text{ср}} = \sum_{i=1}^8 p_i s_i = 4(p_1 s_1 + p_2 s_2) = s_2 + \frac{s_1 - s_2}{G + 1}, \tag{3.2.27}$$

а предельное среднее приращение функции качества $M[\Delta Q] = 4(p_1 - p_2) =$

$$\begin{aligned}
 &= \frac{a_{20} + a_{21} + (a_{22} - a_{20}) \delta_{22} - (a_{10} + a_{11}) - (a_{12} - a_{10}) \delta_{11}}{a_{20} + a_{21} + (a_{22} - a_{20}) \delta_{22} + (a_{10} + a_{11}) + (a_{12} - a_{10}) \delta_{11}} = \\
 &= - \frac{G - 1}{G + 1}, \tag{3.2.28}
 \end{aligned}$$

где

$$\begin{aligned}
 G &= \frac{a_{10} + a_{11} + (a_{12} - a_{10}) \delta_{11}}{a_{20} + a_{21} + (a_{22} - a_{20}) \delta_{22}} = \\
 &= \frac{[1 - 2(\beta_2 s_2 + \gamma_2 s_1)][1 - (\beta_2 s_2 + \gamma_2 s_1)](1 - \delta) + \beta_2 s_2 + \gamma_2 s_1}{[1 - 2(\beta_1 s_1 + \gamma_1 s_2)][1 - (\beta_1 s_1 + \gamma_1 s_2)](1 - \delta) + \beta_1 s_1 + \gamma_1 s_2},
 \end{aligned}
 \tag{3.2.29}$$

причем

$$\beta = \beta_1 + \beta_2; \quad \gamma = \gamma_1 + \gamma_2; \tag{3.2.30}$$

$$0 \leq \beta_1 \leq 0,5; \quad 0 \leq \beta_2 \leq 1; \tag{3.2.31}$$

$$0 \leq \gamma_1 \leq 0,5; \quad 0 \leq \gamma_2 \leq 1.$$

Из формулы (3.2.28) следует

$$\lim_{G \rightarrow \infty} M[\Delta Q] = -1, \quad (3.2.32)$$

а из (3.2.27) —

$$\lim_{G \rightarrow \infty} p_{cp} = s_2. \quad (3.2.33)$$

Из формул (3.2.32) и (3.2.33) видно, что для построения оптимальных алгоритмов поиска необходимо найти значения параметров

β_1, β_2 и γ_1, γ_2 , при которых выражение G принимает максимальное значение. Проведем теоретический анализ выражения G для случая, когда отсутствует помеха, т. е. когда $\sigma = 0$ и $s_1 = 1, s_2 = 0$. Тогда

$$G = \frac{[1 - 2(\gamma_1 + \gamma_2)](1 - \gamma_2)(1 - \delta) + \gamma_1 + \gamma_2}{[1 - 2(\beta_1 + \beta_2)](1 - \beta_2)(1 - \delta) + \beta_1 + \beta_2} = \frac{f(\gamma_1, \gamma_2)}{f(\beta_1, \beta_2)}. \quad (3.2.34)$$

Найдем максимальное значение функции G . Для этого необходимо найти максимум функции $f(\gamma_1, \gamma_2)$ и положительный минимум функции $f(\beta_1, \beta_2)$. Область определения функции

$$f(x_1, x_2) = 1 - \delta + (2\delta - 1)x_1 + (3\delta - 2)x_2 + 2(1 - \delta)x_1x_2 + 2(1 - \delta)x_2^2 \quad (3.2.35)$$

показана на рис. 3.2.1. Стандартными методами математического анализа покажем, что экстремальные значения функции $f(x_1, x_2)$ находятся на границе области ее определения. Дифференцируя $f(x_1, x_2)$, находим:

$$f'_{x_1}(x_1, x_2) = 2\delta - 1 + 2(1 - \delta)x_2; \quad (3.2.36)$$

$$f'_{x_2}(x_1, x_2) = 3\delta - 2 + 2(1 - \delta)x_1 + 4(1 - \delta)x_2.$$

Отсюда следует, что стационарной является точка с координатами

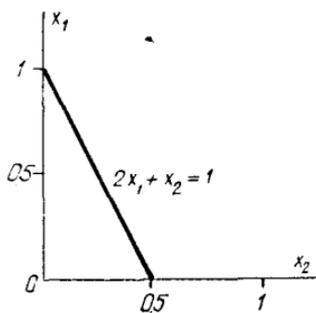


Рис. 3.2.1. Область определения функции $f(x_1, x_2)$.

$$x_0 = \frac{1-2\delta}{2(1-\delta)}; \quad x_1 = \frac{\delta}{2(1-\delta)}; \quad x_2 = \frac{1-2\delta}{2(1-\delta)}. \quad (3.2.37)$$

При $0 < \delta < 0,5$ она находится внутри области определения функции $f(x_1, x_2)$; при $\delta = 0,5$ она помещается на границе этой области и имеет координаты $x_1 = 0,5$; $x_2 = 0$; при $\delta = 0$ — также на границе, с координатами $x_1 = 0$; $x_2 = 0,5$. При $0,5 < \delta \leq 1$ стационарная точка выходит из области определения функции $f(x_1, x_2)$.

Вторые производные функции равны

$$\begin{aligned} D = f''_{x_1 x_1}(x_1, x_2) = 0; \quad C = f''_{x_2 x_2}(x_1, x_2) = 4(1-\delta); \\ B = f''_{x_1 x_2}(x_1, x_2) = f''_{x_2 x_1}(x_1, x_2) = 2(1-\delta). \end{aligned} \quad (3.2.38)$$

Поэтому $B^2 - DC = 4(1-\delta)^2 \geq 0$. Следовательно, точка, определяемая формулами (3.2.37), не является точкой экстремума.

Исследуем функцию $f(x_1, x_2)$ на границах ее определения.

1. На границе $x_1 = 0$; $0 \leq x_2 \leq 1$ функция $f(x_1, x_2)$ имеет вид

$$f(0, x_2) = 1 - \delta + (3\delta - 2)x_2 + 2(1 - \delta)x_2^2, \quad (3.2.39)$$

а экстремальной является точка

$$x_2 = \frac{2-3\delta}{4(1-\delta)}; \quad x_0 = \frac{2-\delta}{4(1-\delta)}. \quad (3.2.40)$$

В этой точке функция $f(0, x_2)$ принимает минимальное значение, равное

$$f\left(0, \frac{2-3\delta}{4(1-\delta)}\right) = \frac{4-4\delta-\delta^2}{8(1-\delta)}. \quad (3.2.41)$$

На концах отрезка $0 \leq x_2 \leq 1$ функция принимает значения

$$f(0, 0) = 1 - \delta; \quad f(0, 1) = 1. \quad (3.2.42)$$

Отсюда видно, что максимальное значение, равное единице, при $0 < \delta < 1$ она принимает на правом конце отрезка $x_2 = 1$, а для $\delta = 0$ — также на конце отрезка $x_2 = 0$. Точка минимума (3.2.40) находится внутри отрезка $0 \leq x_2 \leq 1$ при $0 < \delta < 2/3$. При $2/3 \leq \delta \leq 1$ точкой минимума является левый конец отрезка $x_2 = 0$, а минимальное значение равно

$$f(0, 0) = 1 - \delta. \quad (3.2.43)$$

2. На границе $x_2 = 0$; $0 \leq x_1 \leq 0,5$ функция $f(x_1, x_2)$ является линейной:

$$f(x_1, 0) = 1 - \delta + (2\delta - 1)x_1. \quad (3.2.44)$$

Наибольшее и наименьшее значения она принимает на концах отрезка $0 \leq x_1 \leq 0,5$:

$$f(0, 0) = 1 - \delta; f(0,5; 0) = 0,5. \quad (3.2.45)$$

3. На границе $2x_1 + x_2 = 1$; $0 \leq x_1 \leq 0,5$; $0 \leq x_2 \leq 1$ функция имеет вид

$$f\left(\frac{1-x_2}{2}, x_2\right) = \frac{1}{2} + \left(\delta - \frac{1}{2}\right)x_2 + (1-\delta)x_2^2. \quad (3.2.46)$$

Экстремальной является точка с координатами

$$x_2 = \frac{1-2\delta}{4(1-\delta)}; x_1 = \frac{3-2\delta}{8(1-\delta)}, \quad (3.2.47)$$

в которой функция принимает минимальное значение:

$$f\left(\frac{3-2\delta}{8(1-\delta)}, \frac{1-2\delta}{4(1-\delta)}\right) = \frac{5+4\delta-12\delta^2}{16(1-\delta)}. \quad (3.2.48)$$

На концах границы функция $f\left(\frac{1-x_2}{2}, x_2\right)$ равна

$$f(0,5; 0) = 0,5; f(0, 1) = 1. \quad (3.2.49)$$

Формулы (3.2.47) дают точку минимума функции (3.2.46) для $0 < \delta < 0,5$. При $0,5 \leq \delta < 1$ точкой минимума функции (3.2.46) является точка $x_1 = 0,5$; $x_2 = 0$, а минимальное значение функции равно

$$f(0,5; 0) = 0,5. \quad (3.2.50)$$

Для $\delta=1$ и $\delta=0$ необходимо провести дополнительный анализ. Сравнивая формулы (3.2.41), (3.2.42), (3.2.45), (3.2.48) и (3.2.49), видим, что максимальное значение, равное единице, функция $f(x_1, x_2)$ для случая $0 < \delta < 1$ принимает в точке $x_1=0$; $x_2=1$. Следовательно, оптимальные значения вероятностей $\gamma_0, \gamma_1, \gamma_2$ равны

$$\gamma_0 = \gamma_1 = 0; \gamma_2 = 1, \quad (3.2.51)$$

т. е. при оптимизации стохастической модели объекта в обстановке отсутствия помех при неудачном шаге поиска возврат системы поиска в предыдущее состояние всегда является оптимальным действием.

Далее следует, что для $2/3 \leq \delta < 1$ минимальное значение, равное $1-\delta$, функция $f(x_1, x_2)$ принимает в точке $x_1=x_2=0$, т. е. оптимальными значениями параметров $\beta_0, \beta_1, \beta_2$ являются

$$\beta_0 = 1; \beta_1 = \beta_2 = 0. \quad (3.2.52)$$

Следовательно, для $2/3 \leq \delta < 1$ оптимальным является алгоритм поиска со следующими элементами переходных матриц (3.2.1):

$$\beta_0 = 1; \beta_1 = \beta_2 = 0; \gamma_0 = \gamma_1 = 0; \gamma_2 = 1. \quad (3.2.53)$$

Среднее приращение функции качества, подсчитанное по формуле (3.2.28), для этого алгоритма равно

$$M_1[\Delta Q] = - \frac{\delta}{2-\delta}, \quad (3.2.54)$$

а средний штраф, подсчитанный по формуле (3.2.27),

$$\rho_{\text{ср}}^{(1)} = \frac{1-\delta}{2-\delta}. \quad (3.2.55)$$

При $1/2 \leq \delta < 2/3$ формулы (3.2.47) не имеют физического смысла. Поэтому для этих значений δ функция $f(x_1, x_2)$ принимает минимальное значение в точке, определяемой формулами (3.2.40). Чтобы найти оптимальный алгоритм поиска для $0 < \delta < 0,5$, надо провести срав-

нение выражений (3.2.41) и (3.2.48). Из неравенства

$$\frac{2(4-4\delta-\delta^2)}{5+4\delta-12\delta^2} \geq 1 \quad (3.2.56)$$

следует неравенство

$$10\delta^2 - 12\delta + 3 \geq 0. \quad (3.2.57)$$

Решая это неравенство, получаем точки

$$\delta_1 = \frac{6 + \sqrt{6}}{10}; \quad \delta_2 = \frac{6 - \sqrt{6}}{10}, \quad (3.2.58)$$

определяющие интервал, на котором выражение (3.2.41) принимает значение, меньшее, чем выражение (3.2.48). Следовательно, на интервале $(6 - \sqrt{6})/10 < \delta < 2/3$ оптимальным является алгоритм

$$\beta_0 = \frac{2-\delta}{4(1-\delta)}; \quad \beta_1 = 0; \quad \beta_2 = \frac{2-3\delta}{4(1-\delta)}; \quad (3.2.59)$$

$$\gamma_0 = \gamma_1 = 0; \quad \gamma_2 = 1,$$

а на интервале $0 < \delta < (6 - \sqrt{6})/10$ алгоритм

$$\beta_0 = 0; \quad \beta_1 = \frac{3-2\delta}{8(1-\delta)}; \quad \beta_2 = \frac{1-2\delta}{4(1-\delta)}; \quad (3.2.60)$$

$$\gamma_0 = \gamma_1 = 0; \quad \gamma_2 = 1.$$

Для первого алгоритма среднее приращение функции качества и средний штраф равны

$$M_2[\Delta Q] = -\frac{4-4\delta+\delta^2}{12-12\delta-\delta^2}; \quad (3.2.61)$$

$$p_{\text{ср}}^{(2)} = \frac{4-4\delta-\delta^2}{12-12\delta-\delta^2}, \quad (3.2.62)$$

а для второго алгоритма —

$$M_3[\Delta Q] = -\frac{11-20\delta+12\delta^2}{21-12\delta-12\delta^2}; \quad (3.2.63)$$

$$p_{\text{ср}}^{(3)} = \frac{5 + 4\delta - 12\delta^2}{21 - 12\delta - 12\delta^2}. \quad (3.2.64)$$

Изложенный выше анализ показывает, что для различных интервалов изменения параметра объекта δ имеются различные оптимальные алгоритмы.

1) Для интервала $2/3 \leq \delta < 1$ оптимальным является алгоритм со значениями переходных вероятностей

$$\beta_0 = 1; \beta_1 = \beta_2 = 0; \gamma_0 = \gamma_1 = 0; \gamma_2 = 1. \quad (3.2.65)$$

Среднее значение функции качества для этого алгоритма равно

$$M_1[\Delta Q] = -\frac{\delta}{2 - \delta}, \quad (3.2.66)$$

а средний штраф —

$$p_{\text{ср}}^{(1)} = \frac{1 - \delta}{2 - \delta}. \quad (3.2.67)$$

2) Для интервала $(6 - \sqrt{6})/10 < \delta < 2/3$ имеем следующий оптимальный алгоритм:

$$\beta_0 = \frac{2 - \delta}{4(1 - \delta)}; \beta_1 = 0; \beta_2 = \frac{2 - 3\delta}{4(1 - \delta)}; \quad (3.2.68)$$

$$\gamma_0 = \gamma_1 = 0; \gamma_2 = 1.$$

Среднее приращение функции качества и средний штраф равны

$$M_2[\Delta Q] = -\frac{4 - 4\delta + \delta^2}{12 - 12\delta - \delta^2}; \quad (3.2.69)$$

$$p_{\text{ср}}^{(2)} = \frac{4 - 4\delta - \delta^2}{12 - 12\delta - \delta^2}. \quad (3.2.70)$$

3) На интервале $0 < \delta \leq (6 - \sqrt{6})/10$ оптимальным является алгоритм

$$\beta_0 = 0; \beta_1 = \frac{3 - 2\delta}{8(1 - \delta)}; \beta_2 = \frac{1 - 2\delta}{4(1 - \delta)}; \quad (3.2.71)$$

$$\gamma_0 = \gamma_1 = 0; \gamma_2 = 1.$$

Среднее приращение функции качества и средний штраф равны

$$M_3[\Delta Q] = - \frac{11 - 20\delta + 12\delta^2}{21 - 12\delta - 12\delta^2}; \quad (3.2.72)$$

$$p_{\text{ср}}^{(3)} = \frac{5 + 4\delta - 12\delta^2}{21 - 12\delta - 12\delta^2}. \quad (3.2.73)$$

Для сравнения приводим еще формулы для среднего приращения функции качества на одном шаге поиска для алгоритмов случайного спуска $M_{\text{сп}}[\Delta Q]$ и для алгоритмов случайного поиска с возвратом $M_{\text{в}}[\Delta Q]$. Для случая, когда отсутствует помеха, по формулам (3.2.34) и (3.2.28) имеем:

$$M_{\text{сп}}[\Delta Q] = - \frac{2\delta - 1}{3 - 2\delta}; \quad (3.2.74)$$

$$M_{\text{в}}[\Delta Q] = - \frac{1}{3}. \quad (3.2.75)$$

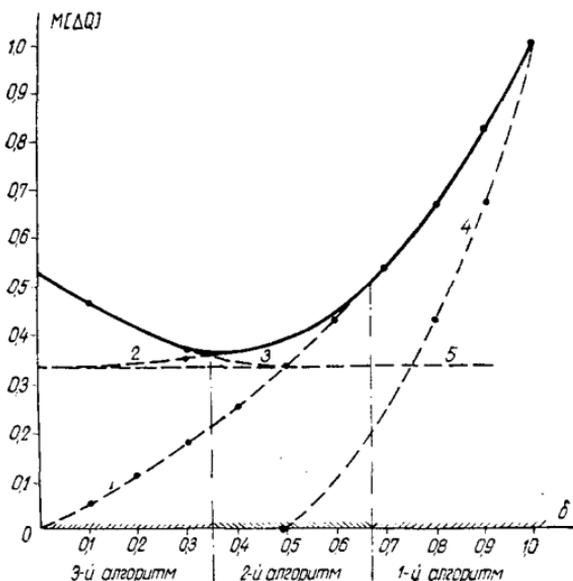


Рис. 3.2.2. Изменение $M[\Delta Q]$ в зависимости от параметра δ для оптимальных алгоритмов поиска.

δ	1	0,9	0,8	0,7	0,6
β_0	1	1	1	1	0,87
β_1	1	0	0	0	0
β_2	0	0	0	0	0,13
γ_0	$\gamma_0 \neq 1$	0	0	0	0
γ_1	Произвольное	0	0	0	0
γ_2	Произвольное	1	1	1	1
$M[\Delta Q]$	-1	-0,82	-0,67	-0,54	-0,44
$M_{сп}[\Delta Q]$	-1	-0,67	-0,43	-0,25	0,11
$M_{в}[\Delta Q]$	-0,33	-0,33	-0,33	-0,33	-0,33

Примечания: 1. $s_1=1$; $s_2=0$.
 2. Прочерк в графе означает, что при соответст

Для случая, когда есть помеха, по формулам (3.2.29) и (3.2.28) находим:

$$M_{сп}[\Delta Q] = - \frac{(2\delta - 1)(s_1 - s_2)}{(2 - s_1 s_2)(1 - \delta) + 1}; \quad (3.2.76)$$

$$M_{в}[\Delta Q] = - \frac{s_1 - s_2}{3[1 - s_1 s_2(1 - \delta)]}. \quad (3.2.77)$$

Таблица 3.2.1

0,5	0,4	0,3	0,2	0,1	0
0,75	0,67	0	0	0	0
0	0	0,43	0,41	0,38	0,38
0,25	0,33	0,14	0,18	0,22	0,25
0	0	0	0	0	1 или 0
0	0	0	0	0	0
1	1	1	1	1	1 или 0
-0,39	-0,36	-0,37	-0,41	-0,46	-0,52
—	—	—	—	—	—
-0,33	-0,33	-0,33	-0,33	-0,33	-0,33

вующих значениях параметра δ алгоритм не работает.

Из формул (3.2.74) и (3.2.75) видно, что алгоритм случайного спуска производит оптимизацию только тех объектов, для которых $0,5 < \delta \leq 1$, а объекты, которые характеризуются большей величиной случайности, этот алгоритм не оптимизирует.

На рис. 3.2.2 представлено изменение $M[\Delta Q]$ в зависимости от параметра объекта δ для построенных оптимальных алгоритмов: график 1 — для алгоритма

δ	1	0,9	0,8	0,7	0,6
β_0	1	1	1	1	0,98
β_1	0	0	0	0	0
β_2	0	0	0	0	0,02
γ_0	$\gamma_0 \neq 1$	0	0	0	0
γ_1	Произвольное	0	0	0	0
γ_2	Произвольное	1	1	1	1
$M_B[\Delta Q]$	-0,20	-0,20	-0,19	-0,19	-0,19
$M[\Delta Q]$	-0,6	-0,50	-0,450	-0,38	-0,31
$M_{сп}[\Delta Q]$	-0,6	-0,41	-0,26	-0,15	-0,07

Примечания: 1. $s_1=0,8$; $s_2=0,2$.
2. Прочерк в графе означает, что при соответст

(3.2.65), график 2 — для алгоритма (3.2.68), график 3 — для алгоритма (3.2.71), график 4 — для алгоритма случайного спуска, график 5 — для алгоритма случайного поиска с возвратом. Жирной линией показано $M[\Delta Q]$ для интегрального оптимального алгоритма. По этим графикам видно значительное преимущество оптимального алгоритма перед известными алгоритмами случайного спуска и случайного поиска с возвратом. В таблице 3.2.1

Таблица 3.2.2

0,5	0,4	0,3	0,2	0,1	0
0,83	0,74	0	0	0	0
0	0	0,47	0,45	0,43	0,42
0,17	0,26	0,06	0,10	0,14	0,16
0	0	0	0	0	0
0	0	0	0	0	0
1	1	1	1	1	1
-0,18	-0,18	-0,18	-0,17	-0,17	-0,17
-0,27	-0,25	-0,24	-0,25	-0,26	-0,27
0	—	—	—	—	—

вующих значениях параметра δ алгоритм не работает.

приведены аналитически подсчитанные значения β_0 , β_1 , β_2 , γ_0 , γ_1 , γ_2 для оптимального алгоритма, а также $M_{сп}[\Delta Q]$ и $M_B[\Delta Q]$ для алгоритмов случайного спуска и случайного поиска с возвратом.

Построить оптимальный алгоритм поиска при наличии помех аналитическим путем, используя выражения (3.2.29) и (3.2.28), весьма затруднительно, поэтому он был определен из этих выражений на ЭЦВМ методом

δ	1	0,9	0,8	0,7	0,6
β_0	1	1	1	1	1
β_1	0	0	0	0	0
β_2	0	0	0	0	0
γ_0	$\gamma_0=1$	0	0	0	0
γ_1	Произвольное	0	0	0	0
γ_2	Произвольное	0	1	1	1
$M[\Delta Q]$	-0,20	-0,18	-0,16	-0,14	-0,12
$M_{сп}[\Delta Q]$	-0,20	-0,14	-0,09	-0,05	-0,02
$M_{в}[\Delta Q]$	-0,07	-0,07	-0,06	-0,06	-0,06

Примечания: 1. $s_1=0,6$; $s_2=0,4$.
 2. Проверк в графе означает, что при соответст

случайного поиска. Результаты расчетов оптимального алгоритма для значений s_1 , равных 0,8; 0,6, и значений δ , равных 0,9; 0,8; 0,7; 0,6; 0,5; 0,4; 0,3; 0,2; 0,1; 0, приведены в таблицах 3.2.2 и 3.2.3. Там же приведены значения $M[\Delta Q]$ для построенного оптимального алгоритма и для сравнения значения $M_{сп}[\Delta Q]$ и $M_{в}[\Delta Q]$ для алгоритмов случайного спуска и случайного поиска с возвратом. Значения первого столбца ($\sigma=1$) таблиц найдены ана-

Таблица 3.2.3

0,5	0,4	0,3	0,2	0,1	0
0,89	0,78	0,71	0	0	0
0	0	0	0,48	0,46	0,45
0,11	0,21	0,29	0,04	0,08	0,10
0	0	0	0	0	0
0	0	0	0	0	0
1	1	1	1	1	1
-0,10	-0,09	-0,09	-0,09	-0,09	-0,10
0	—	—	—	—	—
-0,06	-0,06	-0,06	-0,05	-0,05	-0,05

вующих значениях параметра δ алгоритм не работает.

литическим путем по формулам (3.2.29) и (3.2.28). Из таблиц видно, что при увеличении помех, т. е. при $s_1 \rightarrow 0,5$; $s_2 \rightarrow 0,5$ ($\sigma \rightarrow \infty$), уменьшается быстродействие оптимального алгоритма. Это свойство также следует из формул (3.2.29) и (3.2.28), из которых получаем

$$\lim_{\sigma \rightarrow \infty} G = 1; \quad \lim_{\sigma \rightarrow \infty} M[\Delta Q] = 0. \quad (3.2.78)$$

**АНАЛИЗ ПРОЦЕССА ОПТИМИЗАЦИИ
СТОХАСТИЧЕСКОЙ МОДЕЛИ,
НЕ СВЯЗАННОЙ С ПОИСКОМ**

В этом случае матрица переходных вероятностей объекта (3.2.6) не зависит от действия объекта. Предположим, что оба состояния объекта $Q^{(1)}(\mathbf{X})$ и $Q^{(2)}(\mathbf{X})$ равноценны. Тогда матрицы (3.2.6) равны

$$\Delta^{(1)} = \Delta^{(2)} = \begin{vmatrix} \delta & 1-\delta \\ 1-\delta & \delta \end{vmatrix} \quad (3.2.79)$$

Подставляя в формулы (3.2.26) значения $\delta_{11} = \delta_{22} = \delta$ и выражения (3.2.25), имеем:

$$p_1^{(1)} = p_4^{(1)} = p_2^{(2)} = p_3^{(2)} = \frac{1}{4} \cdot \frac{\delta + (1-2\delta)(\beta s_1 + \gamma s_2)}{2\delta + (1-2\delta)(\beta + \gamma)}; \quad (3.2.80)$$

$$p_2^{(1)} = p_3^{(1)} = p_1^{(2)} = p_4^{(2)} = \frac{1}{4} \cdot \frac{\delta + (1-2\delta)(\beta s_2 + \gamma s_1)}{2\delta + (1-2\delta)(\beta + \gamma)},$$

где $\beta = \beta_0 + \beta_1$; $\gamma = \gamma_0 + \gamma_1$. По формулам (3.2.28) и (3.2.27) с учетом (3.2.79), (3.2.25) и (3.2.80) находим:

$$M[\Delta Q] = \frac{(1-2\delta)(\beta - \gamma)(s_1 - s_2)}{2\delta + (1-2\delta)(\beta + \gamma)}; \quad (3.2.81)$$

$$p_{\text{ср}} = \frac{\delta + (1-2\delta)[\beta(s_1^2 + s_2^2) + 2\gamma s_1 s_2]}{2\delta + (1-2\delta)(\beta + \gamma)}. \quad (3.2.82)$$

Из формулы (3.2.81) следует, что если $0,5 < \delta \leq 1$, то $M[\Delta Q]$ достигает минимума при $\gamma = 0$ и $\beta = 1$. Это означает, что оптимальным является алгоритм, значения параметров которого равны: $\gamma_0 = \gamma_1 = 0$; $\gamma_2 = 1$; $\beta_0 + \beta_1 = 1$; $\beta_2 = 0$. Учитывая условие $\beta_0 + 2\beta_1 + \beta_2 = 1$, имеем $\beta_0 = 1$; $\beta_1 = 0$. Получаем алгоритм, в соответствии с которым при неудачном шаге следует делать возврат, а при удачном шаге — продолжать движение по тому же направлению. Среднее приращение функции качества на одном

шаге для такого алгоритма имеет максимальное из возможных значений:

$$M[\Delta Q] = (1 - 2\delta)(s_1 - s_2), \quad (3.2.83)$$

а средний штраф равен

$$\rho_{\text{ср}} = \delta + (1 - 2\delta)(s_1^2 + s_2^2). \quad (3.2.84)$$

Для алгоритма случайного спуска имеем:

$$M_{\text{сп}}[\Delta Q] = \frac{(1 - 2\delta)(s_1 - s_2)}{3 - 2\delta}; \quad (3.2.85)$$

$$\rho_{\text{ср}}^{\text{сп}} = \frac{2[\delta + (1 - 2\delta)(1 - s_1 s_2)]}{3 - 2\delta}, \quad (3.2.86)$$

а для алгоритма случайного поиска с возвратом получаем:

$$M_{\text{в}}[\Delta Q] = \frac{(1 - 2\delta)(s_1 - s_2)}{1 + 2\delta}; \quad (3.2.87)$$

$$\rho_{\text{ср}}^{(\text{в})} = \frac{2\delta + (1 - 2\delta)(s_1^2 + s_2^2)}{1 + 2\delta}. \quad (3.2.88)$$

Из формул (3.2.83), (3.2.85) и (3.2.87) следует, что среднее приращение функции качества при использовании оптимального алгоритма в $(3 - 2\delta)$ раз больше, чем при использовании алгоритма случайного спуска, и в $(1 + 2\delta)$ раз больше, чем при алгоритме случайного поиска с возвратом.

При $0 \leq \delta < 1/2$ алгоритмы случайного спуска и случайного поиска с возвратом не работают, что аналогично результатам, описанным в работе [28]. В этом случае, в соответствии с формулой (3.2.81), отслеживать направление градиента будет алгоритм, для которого $\gamma > \beta$, т. е. алгоритм с неразумным поведением. Оптимальным является алгоритм, для которого $\beta = 0$, $\gamma = 1$, т. е. $\beta_0 = \beta_1 = 0$, $\beta_2 = 1$ и $\gamma_0 = 1$, $\gamma_1 = \gamma_2 = 0$. Это означает, что при нештрафе следует сделать обратный шаг с вероятностью, равной единице, а при штрафе — продолжать движение по тому же направлению. Среднее приращение

функции качества на одном шаге для такого алгоритма имеет максимальное из возможных значений:

$$M[\Delta Q] = - (1 - 2\delta) (s_1 - s_2). \quad (3.2.89)$$

При $\delta = 1/2$ по формулам (3.2.81) и (3.2.82) имеем $M[\Delta Q] = 0$ и $p_{\text{ср}} = 1/2$. Это означает, что в классе рассматриваемых алгоритмов нет алгоритма, отслеживающего поведение такого объекта.

Рассмотрим случай, когда матрица объекта (3.2.79) имеет вид

$$\Delta = \begin{vmatrix} \delta & 1 - \delta \\ \delta & 1 - \delta \end{vmatrix} \quad (3.2.90)$$

т. е. когда объект принимает следующее состояние вполне независимо от предыдущего. В этом случае матрицы (3.2.20) имеют вид

$$T_{20} + T_{21} = \begin{vmatrix} g_{21} & g_{22} \\ g_{11} & g_{12} \end{vmatrix} \quad T_{11} + T_{22} = \begin{vmatrix} h_{21} & h_{22} \\ h_{11} & h_{12} \end{vmatrix} \quad (3.2.91)$$

С учетом матриц (3.2.19) и (3.2.91) из системы (3.2.18) следует:

$$p_1^{(2)} = \frac{1 - \delta}{\delta} p_1^{(1)}; \quad p_2^{(2)} = \frac{1 - \delta}{\delta} p_2^{(1)}. \quad (3.2.92)$$

Для получения предельных вероятностей $p_1^{(1)}$ и $p_2^{(1)}$ имеем систему уравнений

$$p_1^{(1)} = (g_{11} + g_{22}) p_1^{(1)} + (h_{21} + h_{12}) p_2^{(1)}; \quad (3.2.93)$$

$$p_1^{(1)} + p_2^{(1)} = \frac{\delta}{2}.$$

Из этой системы находим:

$$p_1^{(1)} = \frac{\delta}{2} \cdot \frac{h_{21} + h_{12}}{1 - g_{11} - g_{22} + h_{21} + h_{12}}; \quad (3.2.94)$$

$$p_2^{(1)} = \frac{\delta}{2} \cdot \frac{1 - g_{11} - g_{22}}{1 - g_{11} - g_{22} + h_{21} + h_{12}},$$

где

$$\begin{aligned} g_{11} &= (a_{10} + a_{11}) \delta; & g_{22} &= (a_{20} + a_{21}) (1 - \delta); \\ h_{12} &= (a_{11} + a_{12}) \delta; & h_{21} &= (a_{21} + a_{22}) (1 - \delta); \end{aligned} \quad (3.2.95)$$

a_{ik} определяются формулами (3.2.25).

Подставляя выражения (3.2.94) в формулы (3.2.21) и (3.2.22), находим средний штраф и среднее смещение на одном шаге:

$$\begin{aligned} p_{\text{ср}} &= \frac{(1 - g_{11} - g_{22})s_1 + (h_{21} + h_{22})s_2}{1 - g_{11} - g_{22} + h_{21} + h_{12}} + \\ &+ \frac{\delta(g_{11} + g_{22} + h_{21} + h_{12} - 1)(s_1 - s_2)}{1 - g_{11} - g_{22} + h_{21} + h_{12}}; \end{aligned} \quad (3.2.96)$$

$$M[\Delta Q] = (2\delta - 1) \frac{g_{11} + g_{22} + h_{21} + h_{12} - 1}{1 - g_{11} - g_{22} + h_{21} + h_{12}}. \quad (3.2.97)$$

Подставляя в эти выражения равенства (3.2.95) и учитывая формулы (3.2.25), имеем:

$$p_{\text{ср}} = \frac{\beta + 2(\gamma - \beta)s_1s_2}{\beta + \gamma}; \quad (3.2.98)$$

$$M[\Delta Q] = (2\delta - 1)^2(s_1 - s_2) \frac{\beta - \gamma}{\beta + \gamma}, \quad (3.2.99)$$

где

$$\beta = \beta_1 + \beta_2; \quad \gamma = \gamma_1 + \gamma_2.$$

Из последней формулы следует, что $M[\Delta Q]$ принимает минимальное значение, равное $-(2\delta - 1)^2(s_1 - s_2)$, при $\beta = 0$, т. е. при $\beta_0 = 1$; $\beta_1 = \beta_2 = 0$ и $\gamma \neq 0$ ($\gamma_0 \neq 1$). Следовательно, для вполне стохастической модели объекта оптимальным является алгоритм, в соответствии с которым при удачном шаге поиска следует продолжать шаги в том же направлении, а при неудачном — с произвольной вероятностью, отличной от единицы, можно выбрать любой шаг.

**АНАЛИЗ ПРОЦЕССА ОПТИМИЗАЦИИ
ЛИНЕЙНОЙ МОДЕЛИ ОБЪЕКТА**

Линейную модель объекта можно задать при помощи следующей матрицы:

$$\Delta^{(1)} = \Delta^{(2)} = \Delta = \begin{vmatrix} 1 & 0 \\ 1 & 0 \end{vmatrix} \quad (3.2.100)$$

в которой

$$\delta_{11}^{(i)} = \delta_{21}^{(i)} = 1; \quad \delta_{12}^{(i)} = \delta_{22}^{(i)} = 0 \quad (3.2.101)$$

$(i = 1, 2).$

В этом случае матрица (3.2.10) записывается как

$$T_{ik} = \begin{vmatrix} a_{ik}^{(1)} & 0 \\ a_{ik}^{(2)} & 0 \end{vmatrix} \quad (3.2.102)$$

а матрица (3.2.9) имеет вид

$$A = \begin{vmatrix} a_{10}^{(1)} & 0 & a_{11}^{(1)} & 0 & a_{12}^{(1)} & 0 & a_{11}^{(1)} & 0 \\ a_{10}^{(2)} & 0 & a_{11}^{(2)} & 0 & a_{12}^{(2)} & 0 & a_{11}^{(2)} & 0 \\ a_{21}^{(1)} & 0 & a_{20}^{(1)} & 0 & a_{21}^{(1)} & 0 & a_{22}^{(1)} & 0 \\ a_{21}^{(2)} & 0 & a_{20}^{(2)} & 0 & a_{21}^{(2)} & 0 & a_{22}^{(2)} & 0 \\ a_{22}^{(1)} & 0 & a_{21}^{(1)} & 0 & a_{20}^{(1)} & 0 & a_{21}^{(1)} & 0 \\ a_{22}^{(2)} & 0 & a_{21}^{(2)} & 0 & a_{20}^{(2)} & 0 & a_{21}^{(2)} & 0 \\ a_{11}^{(1)} & 0 & a_{12}^{(1)} & 0 & a_{11}^{(1)} & 0 & a_{10}^{(1)} & 0 \\ a_{11}^{(2)} & 0 & a_{12}^{(2)} & 0 & a_{11}^{(2)} & 0 & a_{10}^{(2)} & 0 \end{vmatrix} \quad (3.2.103)$$

Из матрицы видно, что состояния цепи Маркова, соответствующие второму состоянию объекта $Q^{(2)}(X)$, не существенны, а предельные вероятности этих состояний равны нулю:

$$p_1^{(2)} = p_2^{(2)} = p_3^{(2)} = p_4^{(2)} = 0. \quad (3.2.104)$$

Следовательно, процесс оптимизации линейной модели описывается цепью Маркова со следующей матрицей переходных вероятностей:

$$A = \begin{vmatrix} a_{10} & a_{11} & a_{12} & a_{11} \\ a_{21} & a_{20} & a_{21} & a_{22} \\ a_{22} & a_{21} & a_{20} & a_{21} \\ a_{11} & a_{12} & a_{11} & a_{10} \end{vmatrix} \quad (3.2.105)$$

Учитывая равенства (3.2.17), имеем $p_1 = p_4$; $p_2 = p_3$, где введено обозначение $p_i = p_i^{(1)}$ ($i = 1, \dots, 4$). Из системы (3.2.18) получаем уравнения

$$p_1 = (a_{10} + a_{11})p_1 + (a_{21} + a_{22})p_2; \quad (3.2.106)$$

$$p_2 = (a_{11} + a_{12})p_1 + (a_{20} + a_{21})p_2,$$

решая которые с учетом условия $p_1 + p_2 = \frac{1}{4}$, находим:

$$p_1 = \frac{1}{2} \cdot \frac{a_{21} + a_{22}}{a_{21} + a_{22} + a_{12} + a_{11}}; \quad (3.2.107)$$

$$p_2 = \frac{1}{2} \cdot \frac{a_{12} + a_{11}}{a_{21} + a_{22} + a_{12} + a_{11}}.$$

Подставляя в формулы (3.1.21) и (3.1.22) найденные предельные вероятности p_1 и p_2 и учитывая равенства (3.2.104) и выражения (3.2.25), получаем:

$$p_{\text{ср}} = \frac{\beta(s_1^2 + s_2^2) + 2\gamma s_1 s_2}{\beta + \gamma}; \quad (3.2.108)$$

$$M[\Delta Q] = - \frac{(\gamma - \beta)(s_1 - s_2)}{\beta + \gamma}, \quad (3.2.109)$$

где

$$\beta = \beta_1 + \beta_2; \quad \gamma = \gamma_1 + \gamma_2. \quad (3.2.110)$$

Из формулы (3.2.109) видно, что минимальное значение, равное $s_2 - s_1$, $M[\Delta Q]$ принимает при $\beta = 0$; $\gamma \neq 0$, т. е. при $\beta_0 = 1$; $\beta_1 = \beta_2 = 0$; $\gamma_0 \neq 1$; $\gamma_1 + \gamma_2 \neq 0$. Следовательно, при оптимизации линейной модели объекта оптимальным является алгоритм модифицированного случайного спуска с произвольными вероятностями выбора направлений поиска при неудачном шаге и с исключением продолжения

шагов в направлении неудачного шага с вероятностью, равной единице.

Выше были построены оптимальные алгоритмы поиска в классе алгоритмов, определяемом матрицами (3.2.1). Теперь найдем оптимальные алгоритмы в некоторых более узких классах алгоритмов.

1. Класс алгоритмов поиска, задаваемый матрицами

$$A_0 = \begin{vmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{vmatrix} \quad A_1 = \begin{vmatrix} \gamma_0 & \gamma_1 & \gamma_2 & \gamma_1 \\ \gamma_1 & \gamma_0 & \gamma_1 & \gamma_2 \\ \gamma_2 & \gamma_1 & \gamma_0 & \gamma_1 \\ \gamma_1 & \gamma_2 & \gamma_1 & \gamma_0 \end{vmatrix} \quad (3.2.111)$$

По формулам (3.2.110) находим:

$$\beta = \beta_1 + \beta_2 = \frac{1}{2}; \quad \gamma = \gamma_1 + \gamma_2. \quad (3.2.112)$$

Подставляя эти величины в формулы (3.2.108) и (3.2.112), имеем:

$$p_{\text{ср}} = \frac{s_1^2 + s_2^2 + 4\gamma s_1 s_2}{1 + 2\gamma}; \quad (3.2.113)$$

$$M[\Delta Q] = - \frac{(2\gamma - 1)(s_1 - s_2)}{2\gamma + 1}. \quad (3.2.114)$$

Для этого класса алгоритмов $M[\Delta Q]$ принимает минимальное значение $-1/3(s_1 - s_2)$ при $\gamma = 1$, т. е. при $\gamma_0 = 0$ и $\gamma_1 + \gamma_2 = 1$. Следовательно, в оптимальном алгоритме исключается повторение неудачного шага.

2. Класс алгоритмов поиска, задаваемый матрицами

$$A_0 = \begin{vmatrix} \beta_0 & \beta_1 & \beta_2 & \beta_1 \\ \beta_1 & \beta_0 & \beta_1 & \beta_2 \\ \beta_2 & \beta_1 & \beta_0 & \beta_1 \\ \beta_1 & \beta_2 & \beta_1 & \beta_0 \end{vmatrix} \quad A_1 = \begin{vmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{vmatrix} \quad (3.2.115)$$

Для этого класса имеем: $\beta = \beta_1 + \beta_2$; $\gamma = 1/2$. По формулам (3.2.108) и (3.2.109) находим:

$$p_{\text{ср}} = \frac{2\beta(s_1^2 + s_2^2) + 2s_1s_2}{2\beta + 1}; \quad (3.2.116)$$

$$M[\Delta Q] = - \frac{(1 - 2\beta)(s_1 - s_2)}{1 + 2\beta}. \quad (3.2.117)$$

Для этого класса алгоритмов $M[\Delta Q]$ принимает минимальное значение, равное $-(s_1 - s_2)$, при $\beta = 0$, т. е. при $\beta_0 = 1$; $\beta_1 = \beta_2 = 0$. Следовательно, оптимальным является алгоритм случайного спуска.

3. Класс алгоритмов поиска, задаваемый матрицами (алгоритмы поиска с возвратом при неудаче)

$$A_0 = \begin{vmatrix} \beta_0 & \beta_1 & \beta_2 & \beta_1 \\ \beta_1 & \beta_0 & \beta_1 & \beta_2 \\ \beta_2 & \beta_1 & \beta_0 & \beta_1 \\ \beta_1 & \beta_2 & \beta_1 & \beta_0 \end{vmatrix} \quad A_1 = \begin{vmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{vmatrix} \quad (3.2.118)$$

Для этого класса $\beta = \beta_1 + \beta_2$; $\gamma = 1$. Для среднего штрафа и среднего приращения функции качества имеем:

$$p_{\text{ср}} = \frac{\beta(s_1^2 + s_2^2) + s_1s_2}{\beta + 1}; \quad (3.2.119)$$

$$M[\Delta Q] = - \frac{(1-\beta)(s_1-s_2)}{1+\beta}. \quad (3.2.120)$$

$M[\Delta Q]$ принимает минимальное значение, равное $-(s_1 - s_2)$, при $\beta=0$, т. е. при $\beta_0=1$; $\beta_1=\beta_2=0$. Оптимальным является алгоритм, в соответствии с которым при удачном шаге продолжается спуск, а при неудачном происходит возврат в предыдущее состояние.

4. Класс алгоритмов поиска, задаваемый матрицами (алгоритмы спуска)

$$A_0 = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad A_1 = \begin{vmatrix} \gamma_0 & \gamma_1 & \gamma_2 & \gamma_1 \\ \gamma_1 & \gamma_0 & \gamma_1 & \gamma_2 \\ \gamma_2 & \gamma_1 & \gamma_0 & \gamma_1 \\ \gamma_1 & \gamma_2 & \gamma_1 & \gamma_0 \end{vmatrix} \quad (3.2.121)$$

В этом случае $\beta=0$; $\gamma=\gamma_1+\gamma_2$. Отсюда следует:

$$p_{\text{ср}} = 2s_1s_2; \quad M[\Delta Q] = -(s_1 - s_2). \quad (3.2.122)$$

Полученные выражения для $p_{\text{ср}}$ и $M[\Delta Q]$ имеют место, если $\gamma \neq 0$. Поэтому оптимальным является алгоритм, который исключает повторение неудачных шагов поиска.

§ 3.3. СИНТЕЗ ОПТИМАЛЬНОЙ СТРУКТУРЫ АЛГОРИТМОВ САМООБУЧЕНИЯ

В § 3.1 и 3.2 был рассмотрен синтез оптимальных алгоритмов в классе алгоритмов случайного поиска без самообучения. В этом параграфе построим оптимальные алгоритмы в классе алгоритмов случайного поиска с самообучением.

Найдем оптимальные алгоритмы в классе алгоритмов, которые задаются автоматами со случайной переходной функцией $\Psi = \Psi(\mathbf{W}_{N-1}, \Delta Q'_{N-1})$, не зависящей от выхода автомата $\Delta \mathbf{X}_{N-1}$, и со случайной выходной функцией $p_N(\Delta \mathbf{X}) = p(\Delta \mathbf{X}_N / \mathbf{W}_N)$. Пусть переходы автоматов, расположенных на координатах ω_l ($l=1, \dots, n$) вектора \mathbf{W} , из одного состояния в другое $\omega_l^{(i_1)} \rightarrow \omega_l^{(i_2)}$ задаются мат-

рицами переходных вероятностей, одинаковых для всех координат ω_l . Пусть это будут следующие матрицы: при нештрафе ($\Delta Q' < 0$)

$$R(l) = R_0 =$$

$$= \begin{vmatrix} r_1 & r_2 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ r_1 & 0 & r_2 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & r_1 & 0 & r_2 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots & r_1 & 0 & r_2 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & r_2 & 0 & r_1 & 0 & \dots & 0 & 0 & 0 \\ \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & r_2 & 0 & r_1 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & r_2 & r_1 \end{vmatrix} \quad (3.3.1)$$

при штрафе ($\Delta Q' \geq 0$)

$$R_1(l) = R_1 =$$

$$= \begin{vmatrix} r_2 & r_1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ r_2 & 0 & r_1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & r_2 & 0 & r_1 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots & r_2 & 0 & r_1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & r_1 & 0 & r_2 & 0 & \dots & 0 & 0 & 0 \\ \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & r_1 & 0 & r_2 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & r_1 & r_2 \end{vmatrix} \quad (3.3.2)$$

где

$$r_1 + r_2 = 1. \quad (3.3.3)$$

Выходную функцию автомата также задаем по координатно в виде одинаковых выходных функций F_l для всех координат ω_l ($l=1, \dots, n$):

$$F_l = F(\omega_l, q) = \begin{cases} 1 & \text{при } \omega_l > q; \\ \frac{1}{2} \left(1 + \frac{\omega_l}{q} \right) & \text{при } |\omega_l| \leq q; \\ 0 & \text{при } \omega_l < -q, \end{cases} \quad (3.3.4)$$

где

$$F_l = F(\omega_l, q) = \text{Вер} (\Delta x_l = 1/\omega = \omega_l);$$

$$1 - F_l = \text{Вер} (\Delta x_l = -1/\omega = \omega_l).$$

На изменение ω_l накладываем ограничение

$$|\omega_l| \leq d. \quad (3.3.5)$$

Функционирование автомата этого класса описывается цепью Маркова с матрицей переходных вероятностей

$$A = (I - S^*)A_0 + S^*A_1, \quad (3.3.6)$$

где матрицы A_0 и A_1 являются матрицами (2.1.107) и (2.1.108), а матрица S^* — матрицей (2.1.26). Элементы диагональной матрицы S^* равны

$$s^{(i)} = \sum_{j=1}^{2^n} q_{ij}s_j \quad (i=1, \dots, m), \quad (3.3.7)$$

где s_j — вероятность штрафа действия $\Delta \mathbf{X}^{(j)}$;

q_{ij} — вероятность появления выхода $\Delta \mathbf{X}^{(j)}$ при нахождении автомата в состоянии $\omega^{(i)}$; q_{ij} определяется формулой (3.3.4).

Задача синтеза оптимального алгоритма поиска заключается в нахождении таких значений переходных вероятностей r_1 и r_2 матриц (3.3.1) и (3.3.2) и такого значения параметра q в формуле (3.3.4), при которых среднее приращение $M[\Delta Q]$ функции качества максимально. Предположим, что функция качества $Q(\mathbf{X})$ линейна и

что на ее значения накладывается помеха в виде нормально распределенной случайной величины $\varepsilon(\sigma)$ с математическим ожиданием, равным нулю, и дисперсией σ^2 :

$$Q'(X) = Q(X) + \varepsilon(\sigma). \quad (3.3.8)$$

Тогда цепь Маркова (3.3.6) является однородной и задача сводится к нахождению ее предельных вероятностей. С использованием найденных предельных вероятностей составляется выражение для среднего приращения $M[\Delta Q]$ функции качества по направлению ее антиградиента. Это выражение содержит вероятности r_1, r_2 и параметр q .

Построение оптимального алгоритма поиска сводится к нахождению таких значений r_1, r_2, q , при которых $M[\Delta Q]$ принимает минимальное значение. Построены оптимальные алгоритмы для одномерного ($n=1$) и двумерного ($n=2$) случаев.

1. *Одномерный случай* ($n=1$). В этом случае матрицы переходных вероятностей A_0 и A_1 равны матрицам R_0 (3.3.1) и R_1 (3.3.2), т. е.

$$A_0 = R_0; \quad A_1 = R_1. \quad (3.3.9)$$

Для составления матрицы S^* , входящей в формулу (3.3.6), необходимо определить вероятности s_j штрафов действий $\Delta X^{(j)}$ автомата и вероятности q_{ij} выходов $\Delta X^{(j)}$ автомата при его нахождении в состоянии $\omega^{(i)}$. Определим эти вероятности.

Пусть автомат имеет $2k$ состояний, которые расположены на прямой и пронумерованы в следующем порядке:

$$\begin{aligned} \omega_1 = d; \quad \omega_2 = (k-2)\delta + \frac{\delta}{2}; \quad \dots; \quad \omega_{k-1} = \delta + \frac{\delta}{2}; \quad \omega_k = \frac{\delta}{2}; \\ \omega_{k+1} = -\frac{\delta}{2}; \quad \omega_{k+2} = -\left(\delta + \frac{\delta}{2}\right); \quad \dots; \quad \omega_{2k-1} = \\ = -(k-2)\delta - \frac{\delta}{2}; \quad \omega_{2k} = -d, \end{aligned} \quad (3.3.10)$$

где

$$\delta = \frac{2d}{2k-1}.$$

Для одномерного случая автомат имеет два выхода, которые представляют собой следующие числа:

$$\Delta X^{(1)} = \Delta x = 1; \quad \Delta X^{(2)} = -\Delta x = -1, \quad (3.3.11)$$

вероятности появления которых согласно формуле (3.3.4) равны

$$q_{i1} = \text{Вер}(\Delta x = 1/\omega = \omega^{(i)}) = F_i = F(\omega^{(i)}, q); \quad (3.3.12)$$

$$q_{i2} = \text{Вер}(\Delta x = -1/\omega = \omega^{(i)}) = 1 - F_i =$$

$$= \begin{cases} 0, & \text{если } \omega^{(i)} > q; \\ \frac{1}{2} \left(1 - \frac{\omega^{(i)}}{q} \right), & \text{если } |\omega^{(i)}| \leq q; \\ 1, & \text{если } \omega^{(i)} < -q \end{cases} \quad (3.3.13)$$

$$(i = 1, \dots, 2k).$$

Ввиду линейности функция $Q(\mathbf{X})$ обладает свойством

$$\Delta Q^{(1)} = \alpha; \quad \Delta Q^{(2)} = -\alpha, \quad (3.3.14)$$

где $\Delta Q^{(i)}$ — приращение функции $Q(\mathbf{X})$ по направлению $\Delta X^{(i)}$ ($i = 1, 2$). Тогда вероятности штрафов s_j действия $\Delta X^{(j)}$ автоматов равны

$$s_1 = \text{Вер}(\Delta Q' \geq 0/\Delta x > 0) = \frac{1}{2} \left[1 + \Phi \left(\frac{\alpha}{2\sigma} \right) \right]; \quad (3.3.15)$$

$$s_2 = \text{Вер}(\Delta Q' \geq 0/\Delta x < 0) = \frac{1}{2} \left[1 - \Phi \left(\frac{\alpha}{2\sigma} \right) \right].$$

По формуле (3.3.7) находим

$$s^{(i)} = F_i s_1 + (1 - F_i) s_2 = s_2 + F_i (s_1 - s_2) \quad (3.3.16)$$

$$(i = 1, \dots, 2k).$$

Следовательно, имеем диагональную матрицу вероятностей штрафов (3.3.17).

$$S^* = \begin{vmatrix} s_2 + F_1(s_1 - s_2) & 0 & 0 \dots & 0 & 0 \dots & 0 \\ 0 & s_2 + F_2(s_1 - s_2) & 0 \dots & 0 & 0 \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 \dots & s_2 + F_i(s_1 - s_2) & 0 \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 \dots & 0 & 0 \dots & s_2 + F_{2k}(s_1 - s_2) \end{vmatrix} \quad (3.3.17)$$

$$A = \begin{vmatrix} 1 - v_1 & v_1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 1 - v_2 & 0 & v_2 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 - v_3 & 0 & v_3 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots & 1 - v_k & 0 & v_k & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 1 - v_k & 0 & v_k & 0 & \dots & 0 & 0 & 0 \\ \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 1 - v_2 & 0 & v_2 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 1 - v_1 & v_1 \end{vmatrix} \quad (3.3.19)$$

Поскольку состояния автомата, определяемые формулами (3.3.10), симметричны относительно нулевой точки, то, используя формулы (3.3.4) и (3.3.13), мы можем записать

$$F_i = 1 - F_{2k-i+1} \quad (3.3.18)$$

$$(i = 1, \dots, 2k).$$

Учитывая свойство (3.3.18) и используя матрицы (3.3.1), (3.3.2) и (3.3.17), по формуле (3.3.6) находим матрицу (3.3.19) переходных вероятностей цепи Маркова, описывающей переходы автомата из одного состояния в другое, где

$$v_i = \begin{cases} s^{(i)}r_1 + (1-s^{(i)})r_2 & (i = 1, 2, \dots, k); \\ s^{(i)}r_2 + (1-s^{(i)})r_1 & (i = k+1, \dots, 2k); \end{cases} \quad (3.3.20)$$

$s^{(i)}$ определяются формулой (3.3.16).

Сначала рассмотрим простейший случай, когда $k=1$. В этом случае, как нетрудно заметить, цепь Маркова (3.3.6) имеет матрицу переходных вероятностей

$$A = \begin{vmatrix} 1-v_1 & v_1 \\ 1-v_1 & v_1 \end{vmatrix} \quad (3.3.21)$$

где, в соответствии с формулой (3.3.20),

$$v_1 = s^{(1)}r_1 + (1-s^{(1)})r_2 = r_2 + s^{(1)}(r_1 - r_2). \quad (3.3.22)$$

Предельные вероятности этой цепи Маркова равны

$$p_1 = 1 - v_1; \quad p_2 = v_1 \quad (3.3.23)$$

Подставляя вместо v_1 выражение (3.3.22), получаем:

$$p_1 = r_1 - s^{(1)}(r_1 - r_2);$$

$$p_2 = r_2 + s^{(1)}(r_1 - r_2). \quad (3.3.24)$$

Определим среднее смещение на одном шаге по прямой x . При нахождении автомата в состоянии $\omega^{(i)}$ оно вычисляется по формуле

$$\begin{aligned}
 M[\Delta x/w = \omega^{(i)}] &= \text{Вер}(\Delta x = 1/w = \omega^{(i)}) - \\
 &- \text{Вер}(\Delta x = -1/w = \omega^{(i)}) = F_i - (1 - F_i) = 2F_i - 1,
 \end{aligned}
 \tag{3.3.25}$$

а среднее смещение, осредненное по всем состояниям автомата $\omega^{(i)}$ ($i=1, \dots, 2k$), определяется формулой

$$M[\Delta x] = \sum_{i=1}^{2k} p_i M[\Delta x/w = \omega^{(i)}] = \sum_{i=1}^{2k} p_i (2F_i - 1). \tag{3.3.26}$$

Для случая $k=1$ с учетом формул (3.3.24), (3.3.18) и (3.3.16) из формулы (3.3.26) имеем

$$\begin{aligned}
 M[\Delta x] &= \sum_{i=1}^2 p_i (2F_i - 1) = (2F_1 - 1) (1 - 2v_1) = \\
 &= - (1 - 2F_1)^2 (s_1 - s_2) (r_1 - r_2).
 \end{aligned}
 \tag{3.3.27}$$

Видно, что $M[\Delta x]$ будет минимально, если $(1 - 2F_1)^2 (r_1 - r_2)$ будет равен своему максимальному значению, т. е. единице. Здесь возможны два случая:

$$1) \quad d = \omega_1 \leq q; \tag{3.3.28}$$

в этом случае по формуле (3.3.4) находим:

$$F_1 = \frac{1}{2} \left(1 + \frac{d}{q} \right); \quad 1 - 2F_1 = - \frac{d}{q}; \tag{3.3.29}$$

$$M[\Delta x] = - \frac{d^2}{q^2} (s_1 - s_2) (r_1 - r_2); \tag{3.3.30}$$

$$2) \quad q \leq \omega_1 = d; \tag{3.3.31}$$

в этом случае по формуле (3.3.4) имеем:

$$F_1 = 1; \quad 1 - 2F_1 = -1; \tag{3.3.32}$$

$$M[\Delta x] = - (s_1 - s_2) (r_1 - r_2). \tag{3.3.33}$$

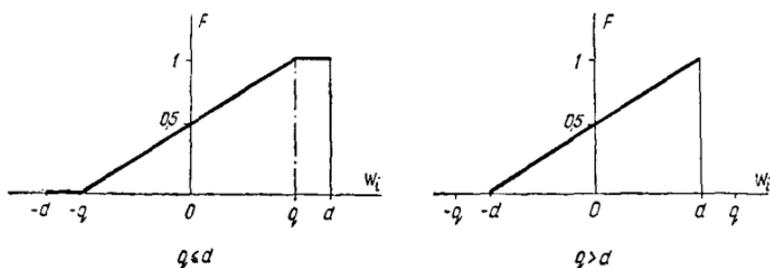


Рис. 3.3.1. Функция выхода автомата в зависимости от параметра q .

На рис. 3.3.1 показаны вид функции (3.3.4) и расположение точек d и q для первого и второго случаев. Из формул (3.3.30) и (3.3.33) видно, что алгоритм поиска будет оптимальным, если

$$r_1 = 1; \quad r_2 = 0; \quad q \leq \omega_1. \quad (3.3.34)$$

Это означает, что в случае линейной функции $Q(\mathbf{X})$ оптимальным алгоритмом поиска по среднему быстрдействию будет алгоритм с детерминированной функцией переходов (3.3.1), (3.3.2) и детерминированной функцией выходов (3.3.4).

Приведем формулы для определения дисперсии условной случайной величины $\Delta x/w$ и безусловной дисперсии величины Δx . Нетрудно показать, что имеет место

$$D[\Delta x/w = \omega^{(i)}] = 4F_i(1 - F_i); \quad (3.3.35)$$

$$D[\Delta x] = (2F_1 - 1)^2 [1 - (s_1 - s_2)(r_1 - r_2)]. \quad (3.3.36)$$

Из этих формул находим, что

$$1) \text{ при } d = \omega_1 \leq q \quad \left(F_i = \frac{1}{2} \left(1 + \frac{d}{q} \right), \quad 1 - 2F_i = -\frac{d}{q} \right)$$

имеем:

$$D[\Delta x/w = \omega^{(i)}] = \left(1 - \frac{d^2}{q^2} \right); \quad (3.3.37)$$

$$D[\Delta x] = \frac{d^2}{q^2} [1 - (s_1 - s_2)(r_1 - r_2)]; \quad (3.3.38)$$

2) при $q \leq \omega = d$ ($F_i = 1$; $1 - 2F_i = -1$)

имеем:

$$D[\Delta x / \omega = \omega^{(i)}] = 0; \quad (3.3.39)$$

$$D[\Delta x] = 1 - (s_1 - s_2)(r_1 - r_2). \quad (3.3.40)$$

Из полученных формул следует, что разброс поиска также будет минимален при тех же значениях параметров, т. е. при

$$r_1 = 1; \quad r_2 = 0; \quad q = \omega, \quad (3.3.41)$$

т. е. для случая линейной функции $Q(X)$ минимальный разброс результата поиска будет при использовании детерминированного алгоритма поиска.

Теперь рассмотрим общий случай, когда $k > 1$. В этом случае переходы цепи Маркова, описывающей переходы автомата из одного состояния в другое, задаются матрицей (3.3.19).

Предельные вероятности этой цепи Маркова находятся как решение следующей системы алгебраических уравнений:

$$v_i p_i = (1 - v_{i+1}) p_{i+1} \quad (i = 1, \dots, k-1);$$

$$v_k p_k = v_{k+1} p_{k+1}; \quad (3.3.42)$$

$$(1 - v_{k+i}) p_{k+i} = v_{k+i+1} p_{k+i+1} \quad (i = 1, \dots, k-1).$$

Решая эту систему, находим:

$$p_2 = \frac{v_1}{1 - v_2} p_1;$$

$$p_i = \frac{\prod_{j=1}^{i-1} v_j}{\prod_{j=2}^i (1 - v_j)} p_1$$

$$(i = 2, \dots, k);$$

$$p_{k+i} = \frac{\prod_{j=1}^{i-1} (1 - v_{k+j})}{\prod_{j=1}^i v_{k+j}} \cdot \frac{\prod_{j=1}^k v_j}{\prod_{j=2}^k (1 - v_j)} p_1; \quad (3.3.43)$$

$$(i=2, \dots, k)$$

$$p_{k+1} = \frac{v_k}{1 - v_{k+1}} \cdot \frac{\prod_{j=1}^k v_j}{\prod_{j=2}^k (1 - v_j)} p_1,$$

где p_1 определяется из условия нормировки:

$$p_1 = \left[1 + \frac{v_1}{1 - v_1} + \sum_{i=2}^k \frac{\prod_{j=1}^{i-1} v_j}{\prod_{j=2}^i (1 - v_j)} + \frac{\prod_{j=1}^k v_j}{\prod_{j=2}^k (1 - v_j)} \times \right. \\ \left. \times \left(\frac{v_k}{1 - v_{k+1}} + \sum_{i=2}^k \frac{\prod_{j=1}^{i-1} (1 - v_{k+j})}{\prod_{j=1}^i v_{k+j}} \right) \right]^{-1}; \quad (3.3.44)$$

v_j определяются формулой (3.3.20). Среднее смещение по оси x определяется формулой (3.3.26).

На рис. 3.3.2 показано изменение $M[\Delta Q]$ в зависимости от r_1 для случаев, когда память автомата равна $k=1, 2, 3, 4, 5$ и $q < \mathbf{W}^{(i)}$ ($i=1, \dots, k$). Из рисунка видно, что с ростом значения k при линейной функции качества быстродействие поиска увеличивается. Графики на рис. 3.3.2 подсчитаны на ЦВМ по формуле (3.4.26) с уче-

гом формул (3.3.43) и (3.3.44) при значении дисперсии помехи $\sigma = 1$.

2. Двумерный случай ($n=2$). Пусть $k=1$. Тогда автомат имеет четыре состояния $W^{(i)}$ и четыре выходных вектора $\Delta X^{(j)}$. Нумерация состояний и переходы из одного состояния в другое показаны на рис. 3.3.3, а. Нетрудно получить переходные матрицы состояний автомата

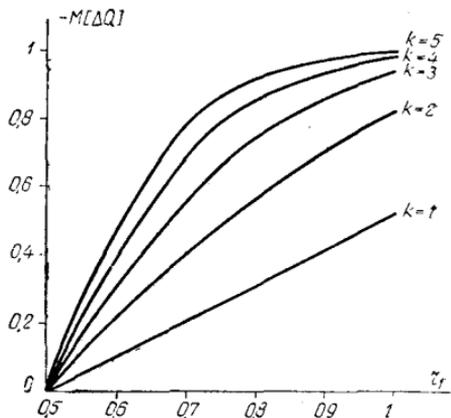


Рис. 3.3.2. Изменение $M[\Delta Q]$ в зависимости от вероятности перехода автомата r_1 и от памяти автомата k .

$$A_0 = \begin{vmatrix} r_1^2 & r_1 r_2 & r_2^2 & r_1 r_2 \\ r_1 r_2 & r_1^2 & r_1 r_2 & r_2^2 \\ \cdot & \cdot & \cdot & \cdot \\ r_2^2 & r_1 r_2 & r_1^2 & r_1 r_2 \\ r_1 r_2 & r_2^2 & r_1 r_2 & r_1^2 \end{vmatrix} \quad (3.3.45)$$

$$A_1 = \begin{vmatrix} r_2^2 & r_1 r_2 & r_1^2 & r_1 r_2 \\ r_1 r_2 & r_2^2 & r_1 r_2 & r_1^2 \\ \cdot & \cdot & \cdot & \cdot \\ r_1^2 & r_1 r_2 & r_2^2 & r_1 r_2 \\ r_1 r_2 & r_1^2 & r_1 r_2 & r_2^2 \end{vmatrix} \quad (3.3.46)$$

Функционирование автомата описывается цепью Маркова со следующей матрицей переходных вероятностей:

$$A = \begin{vmatrix} v_1 & b & 1-v_1 & b \\ b & v_2 & b & 1-v_2 \\ 1-v_3 & b & v_3 & b \\ b & 1-v_4 & b & v_4 \end{vmatrix} \quad (3.3.47)$$

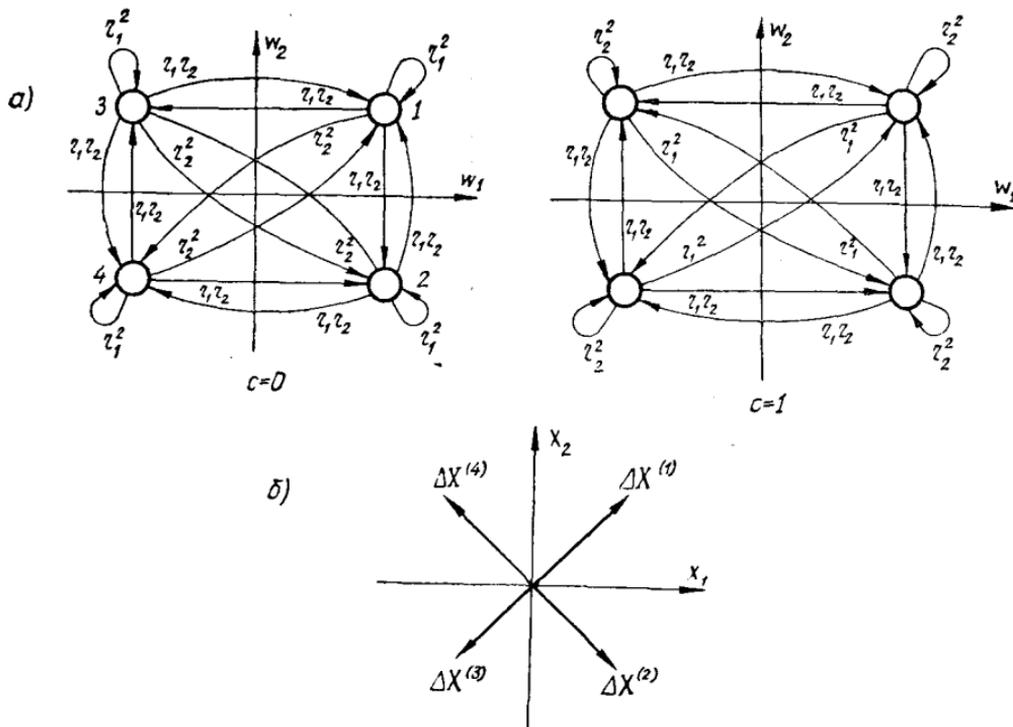


Рис. 3.3.3. Состояния и переходы автомата для $m=4$ ($n=2$) и нумерация выходов ΔX_j автомата.

где

$$v_i = (1 - s^{(i)})r_1^2 + s^{(i)}r_2^2 = r_1^2 - s^{(i)}(r_1^2 - r_2^2), \quad (3.3.48)$$

$$b = r_1 r_2. \quad (3.3.49)$$

Пронумеруем выходы $\Delta X^{(j)}$ автомата так, как это показано на рис. 3.3.3, б. Тогда приращения $\Delta Q^{(j)}$ по направлениям $\Delta X^{(j)}$ равны

$$\begin{aligned} \Delta Q^{(1)} &= \alpha_1 + \alpha_2; & \Delta Q^{(2)} &= \alpha_1 - \alpha_2; \\ \Delta Q^{(3)} &= -(\alpha_1 + \alpha_2); & \Delta Q^{(4)} &= -(\alpha_1 - \alpha_2), \end{aligned} \quad (3.3.50)$$

где α_i ($i=1, 2$) — направляющие косинусы вектора градиента функции $Q(\mathbf{X})$.

Вероятности s_j штрафов действий $\Delta X^{(j)}$ автомата определяются по формуле

$$s_j = \text{Вер} (\Delta Q' \geq 0 / \Delta X = \Delta X^{(j)}) = \frac{1}{2} \left[1 + \Phi \left(\frac{\Delta Q^{(j)}}{2\sigma} \right) \right], \quad (3.3.51)$$

где $\Phi(z)$ — интеграл вероятности. Учитывая формулы (3.3.50), находим:

$$s_1 = \frac{1}{2} \left[1 + \Phi \left(\frac{\alpha_1 + \alpha_2}{2\sigma} \right) \right]; \quad s_2 = \frac{1}{2} \left[1 + \Phi \left(\frac{\alpha_1 - \alpha_2}{2\sigma} \right) \right]; \quad (3.3.52)$$

$$s_3 = \frac{1}{2} \left[1 - \Phi \left(\frac{\alpha_1 - \alpha_2}{2\sigma} \right) \right]; \quad s_4 = \frac{1}{2} \left[1 - \Phi \left(\frac{\alpha_1 + \alpha_2}{2\sigma} \right) \right].$$

Определим вероятности штрафов $s^{(i)}$ ($i=1, \dots, 4$) по отношению к состояниям $\mathbf{W}^{(i)}$ автомата. Для этого сначала определим вероятности выходов q_{ij} :

$$q_{i1} = \text{Вер} (\Delta x_1 = 1 / \omega_1 = \omega_1^{(i)}) \cdot \text{Вер} (\Delta x_2 = 1 / \omega_2 = \omega_2^{(i)}) = F_1^{(i)} F_2^{(i)};$$

$$q_{i2} = F_2^{(i)} (1 - F_2^{(i)});$$

$$q_{i3} = (1 - F_1^{(i)}) (1 - F_2^{(i)});$$

$$q_{i4} = (1 - F_1^{(i)}) F_2^{(i)},$$

(3.3.53)

где $F_l^{(i)}$ заданы формулой (3.3.4) при $\omega_l = \omega_l^{(i)}$.

Предположим, что состояния $\mathbf{W}^{(i)}$ автомата расположены симметрично относительно начала координат:

$$\mathbf{W}^{(1)} = -\mathbf{W}^{(3)}; \quad \mathbf{W}^{(4)} = -\mathbf{W}^{(2)}. \quad (3.3.54)$$

Тогда имеют место следующие равенства:

$$F_1^{(1)} = F_2^{(1)} = F_1^{(2)} = F_2^{(4)} = F_1;$$

$$F_1^{(3)} = F_2^{(3)} = F_1^{(4)} = F_2^{(2)} = F_2.$$

(3.3.55)

При этом можно показать, что

$$F_2 = 1 - F_1.$$

(3.3.56)

По формуле (3.3.7), учитывая выражения (3.3.55) и (3.3.56), находим:

$$\begin{aligned} s^{(1)} &= s_3 + (s_1 - s_3)F_1; \\ s^{(2)} &= s_4 + (s_2 - s_4)F_1; \\ s^{(3)} &= s_1 - (s_1 - s_3)F_1; \\ s^{(4)} &= s_2 - (s_2 - s_4)F_1, \end{aligned} \tag{3.3.57}$$

где

$$F_1 = \begin{cases} 1, & \text{если } \omega_1^{(1)} > q; \\ \frac{1}{2} \left(1 + \frac{\omega_1^{(1)}}{q} \right), & \text{если } |\omega_1^{(1)}| \leq q; \\ 0, & \text{если } q < -\omega_1^{(1)}; \end{cases} \tag{3.3.58}$$

s_j ($j=1, \dots, 4$) определяются формулами (3.3.52).

Из формул (3.3.57) видно, что выполняются равенства

$$s^{(1)} + s^{(3)} = 1; \quad s^{(2)} + s^{(4)} = 1. \tag{3.3.59}$$

Учитывая эти равенства, находим:

$$v_3 = 1 - v_1; \quad v_4 = 1 - v_2. \tag{3.3.60}$$

Следовательно, матрица (3.3.47) имеет вид

$$A = \begin{vmatrix} v_1 & b & 1-v_1 & b \\ b & v_2 & b & 1-v_2 \\ v_1 & b & 1-v_1 & b \\ b & v_2 & b & 1-v_2 \end{vmatrix} \tag{3.3.61}$$

Предельные вероятности этой цепи Маркова находятся как решение следующей системы уравнений:

$$\begin{aligned} p_1 &= v_1(p_1 + p_3) + b(p_2 + p_4); \\ p_2 &= b(p_1 + p_2) + v_2(p_2 + p_4); \\ p_3 &= (1 - v_1)(p_1 + p_3) + b(p_2 + p_4); \\ p_4 &= b(p_1 + p_3) + (1 - v_2)(p_2 + p_4). \end{aligned} \tag{3.3.62}$$

Решая эту систему уравнений и нормируя вероятности p_i , находим:

$$\begin{aligned} p_1 &= \frac{1}{4} [1 + (r_1 - r_2) (1 - 2s^{(1)})]; \\ p_2 &= \frac{1}{4} [1 + (r_1 - r_2) (1 - 2s^{(2)})]; \\ p_3 &= \frac{1}{4} [1 + (r_1 - r_2) (1 - 2s^{(3)})]; \\ p_4 &= \frac{1}{4} [1 + (r_1 - r_2) (1 - 2s^{(4)})]. \end{aligned} \tag{3.3.63}$$

Можно показать, что среднее приращение функции $Q(\mathbf{X})$ при фиксированном $\mathbf{W}^{(i)}$ ($i=1, \dots, 4$) равно

$$M[\Delta Q / \mathbf{W}^{(i)}] = (F_1^{(i)} + F_2^{(i)} - 1) \Delta Q^{(1)} + (F_1^{(i)} - F_2^{(i)}) \Delta Q^{(2)}. \tag{3.3.64}$$

По этой формуле с учетом равенств (3.3.55) находим:

$$\begin{aligned} M[\Delta Q / \mathbf{W}^{(1)}] &= (2F_1 - 1) \Delta Q^{(1)}; \\ M[\Delta Q / \mathbf{W}^{(2)}] &= (2F_1 - 1) \Delta Q^{(2)}; \\ M[\Delta Q / \mathbf{W}^{(3)}] &= (1 - 2F_1) \Delta Q^{(1)}; \\ M[\Delta Q / \mathbf{W}^{(4)}] &= (1 - 2F_1) \Delta Q^{(2)}. \end{aligned} \tag{3.3.65}$$

С учетом этих выражений и формул (3.3.50) приращение функции $Q(\mathbf{X})$, осредненное по всем состояниям $\mathbf{W}^{(i)}$, равно

$$\begin{aligned} M[\Delta Q] &= [(p_1 - p_3) (\alpha_1 + \alpha_2) + (p_2 - p_4) (\alpha_1 - \alpha_2)] (2F_1 - 1) = \\ &= -\frac{1}{4} [(s_1 - s_3) (\alpha_1 + \alpha_2) + (s_2 - s_4) (\alpha_1 - \alpha_2)] \times \\ &\times (r_1 - r_2) (2F_1 - 1). \end{aligned} \tag{3.3.66}$$

Очевидно, что $M[\Delta Q]$ достигнет своего минимального значения тогда, когда выражение

$$(r_1 - r_2) (2F_1 - 1) \tag{3.3.67}$$

примет свое максимальное значение. Это будет иметь место при

$$r_1=1; \quad r_2=0; \quad F_1=1. \quad (3.3.68)$$

$F_1=1$ тогда, когда в формуле (3.3.58) $\omega_1^{(1)} \geq q$. Поскольку $\omega_1^{(1)} = \omega_2^{(1)}$, алгоритм поиска будет оптимальным при следующих значениях его параметров:

$$r_1=1; \quad r_2=0; \quad \omega_1^{(1)} = \omega_2^{(1)} \geq q. \quad (3.3.69)$$

Этот результат согласуется с результатом, полученным для одномерного случая ($n=1$).

§ 3.4. СМЕШАННЫЕ АЛГОРИТМЫ СЛУЧАЙНОГО ПОИСКА

Будем называть смешанным алгоритмом (по аналогии со смешанной стратегией в теории игр) такой алгоритм, в котором некоторые переходы от одного оператора к другому осуществляются вероятностным образом. Простейшим смешанным алгоритмом поиска является алгоритм, стратегия которого образуется вероятностным взвешиванием чистых стратегий:

$$s = \sum_{i=1}^k p_i s_i = [P \ S], \quad (3.4.1)$$

где

$$P = (p_1, \dots, p_k) \quad (3.4.2)$$

— вектор вероятностей весов выбранных стратегий;

$$\sum_{i=1}^k p_i = 1;$$

$$S = (s_1, \dots, s_k) \quad (3.4.3)$$

— вектор имеющихся чистых стратегий поиска (например, s_1 — стратегия метода градиента, s_2 — стратегия

метода покоординатного спуска, s_3 — стратегия случайного поиска с парными пробами и т. д.).

Синтез такого алгоритма сводится к выбору его составляющих из множества алгоритмов (3.4.3) и определению вектора их весов (3.4.2). Очевидно, что при заданном объекте и выбранном критерии эффективности алгоритма вектор (3.4.2) вырождается в единичный. Это означает, что наилучшей смешанной комбинацией алгоритмов является отсутствие комбинации, что и сводится к выбору одного наилучшего алгоритма.

Это обстоятельство заставляет обратиться к другим способам построения смешанного алгоритма.

Пусть в качестве исходных чистых действий $s_1, s_2, \dots, \dots, s_k$ выступают не стратегии поиска, а лишь отдельные операторы. Такими операторами могут быть, например, оператор определения градиента показателя качества, оператор движения в том же направлении, что и ранее (стратегия «Так же»), оператор движения в обратном направлении (стратегия «Иначе») и т. д. Так, известный алгоритм наискорейшего спуска образуется комбинацией трех операторов: градиента, шага в антиградиентном направлении и оператора «Так же». Граф этого алгоритма показан на рис. 3.4.1 (знаком «+» обозначен оператор «Так же»).

Пусть имеется k исходных операторов (а не стратегий) s_1, \dots, s_k . Это означает, что ни одно из s_i не является алгоритмом поиска. Поиск образуется определенной комбинацией этих операторов. Переход от одного оператора к другому имеет вероятностный характер и определяется двумя стохастическими матрицами: матрицей нештрафа при $\Delta Q < 0$

$$A_0 = \|p_{ij}^{(0)}\|, \quad (3.4.4)$$

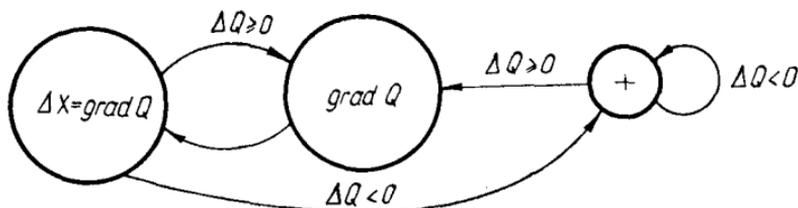


Рис. 3.4.1. Граф алгоритма наискорейшего спуска.

где $p_{ij}^{(0)}$ — элемент матрицы $k \times k$, определяющий вероятность перехода от i -го оператора к j -му при удачном шаге поиска (рассматривается, как обычно, случай минимизации), и матрицей штрафа при $\Delta Q \geq 0$

$$A_1 = \|p_{ij}^{(1)}\|, \quad (3.4.5)$$

которая характеризует вероятности перехода от одного оператора к другому при неудачном шаге поиска, т. е. при $\Delta Q \geq 0$.

Введя параметр штрафа на N -м шаге поиска

$$c_N = \begin{cases} 0 & \text{(нештраф) при } \Delta Q_{N-1} < 0; \\ 1 & \text{(штраф) при } \Delta Q_{N-1} \geq 0, \end{cases} \quad (3.4.6)$$

можно матрицу переходных вероятностей, образующую алгоритм поиска, записать в виде

$$A^N = (1 - c_N)A_0 + c_N A_1. \quad (3.4.7)$$

Таким образом, смешанный алгоритм поиска представляется вероятностным автоматом, входом которого является параметр штрафа c , а выходом — номер j оператора ($j = 1, \dots, k$).

Для анализа работы смешанных алгоритмов необходимо замкнуть обратную связь, т. е. иметь модель объекта и уметь оценивать эффективность работы алгоритма, т. е. располагать критерием качества алгоритма поиска.

Для моделирования объекта выберем стохастическую модель, предложенную в [1]. Эта модель обычно характеризуется двумя вероятностями — вероятностью удачного случайного шага

$$p = \text{Вер} (\Delta Q_{\xi} < 0) \quad (3.4.8)$$

и вероятностью двух непосредственно следующих друг за другом удачных шагов в одном и том же направлении

$$q = \text{Вер} (\Delta Q_N < 0 / \Delta Q_{N-1} < 0). \quad (3.4.9)$$

Однако в общем случае объект следует задавать вероятностью штрафа при использовании i -го оператора

$$l_i = \text{Вер} (c = 1/s = s_i). \quad (3.4.10)$$

Таким образом, вектор

$$\mathbf{L} = (l_1, \dots, l_k) \quad (3.4.11)$$

полностью характеризует объект, где

$$l_i = l_i(p, q). \quad (3.4.12)$$

В качестве критерия оценки эффективности алгоритма естественно выбрать средний штраф, который хорошо характеризует целесообразность автомата поиска (3.4.7):

$$\bar{c} = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N c_i. \quad (3.4.13)$$

Это означает, что наилучшим алгоритмом поиска считается тот, средний штраф которого имеет минимальное значение. Средний штраф определенным образом зависит от матриц A_0 и A_1 и свойств объекта (p, q) :

$$\bar{c} = f(A_0, A_1, p, q). \quad (3.4.14)$$

Задача оптимального синтеза заключается в определении такого алгоритма, т. е. таких стохастических матриц A_0 и A_1 , которые минимизировали бы средний штраф для заданного объекта (p, q) :

$$f(A_0, A_1, p, q) \rightarrow \min_{A_0, A_1}. \quad (3.4.15)$$

Для решения этой задачи сначала нужно выяснить вид зависимости, т. е. определить средний штраф в функции матриц A_0 и A_1 . Определим предельные вероятности:

$$\bar{\mathbf{P}} = (\bar{p}_1, \dots, \bar{p}_k), \quad (3.4.16)$$

где \bar{p}_i — предельная вероятность пребывания автомата поиска в состоянии s_i $\left(\sum_{i=1}^k \bar{p}_i = 1 \right)$. Для этого необходимо иметь матрицу автомата поиск—объект. Эта матрица имеет вид

$$B = \|b_{ij}\| \quad (3.4.17)$$

$$(i, j = 1, \dots, k),$$

где

$$b_{ij} = (1 - l_i) p_{ij}^{(0)} + l_i p_{ij}^{(1)}. \quad (3.4.18)$$

Это означает, что переход из i -го состояния в j -е происходит с вероятностью нештрафа $1 - l_i$ по матрице A_0 и с вероятностью штрафа l_i — по матрице A_1 .

Теперь легко определить вектор предельных вероятностей. Он является решением линейного уравнения

$$B^T \bar{P} = \bar{P}, \quad (3.4.19)$$

где T — знак транспонирования. Средний штраф при этом равен скалярному произведению

$$\bar{c} = [\bar{P}, L]. \quad (3.4.20)$$

Получив это выражение в аналитическом виде, нетрудно решить задачу оптимального синтеза, минимизирующего средний штраф.

Заметим, что средний штраф поиска имеет четкий физический смысл. Это — относительное число неудачных шагов поиска. Естественно, что чем меньше средний штраф, тем эффективнее алгоритм. Поэтому минимизация среднего штрафа дает возможность синтезировать оптимальные алгоритмы поиска. Конкретизируем вышеизложенное на одном классе алгоритмов случайного поиска.

Большое число алгоритмов случайного поиска образуется путем комбинации трех операторов ($k=3$):

- 1) случайный шаг (оператор s_1);
- 2) шаг в том же направлении (оператор s_2);
- 3) шаг в обратном направлении (оператор s_3).

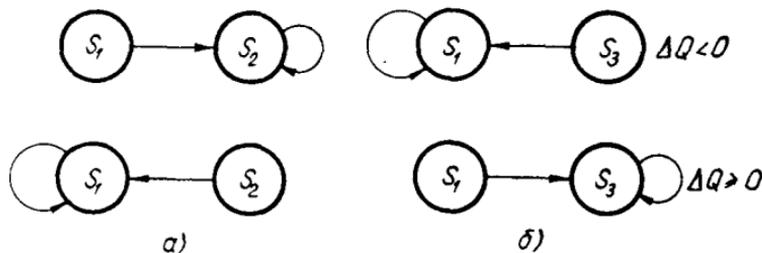


Рис. 3.4.2. Графы методов случайного поиска с линейной (а) и нелинейной (б) тактикой.

На рис. 3.4.2 показаны графы известных алгоритмов случайного поиска этого класса с линейной (а) и нелинейной (б) тактикой для двух случаев удачного ($\Delta Q < 0$) и неудачного ($\Delta Q \geq 0$) шагов.

Рассмотрим следующий смешанный алгоритм с матрицами, где оставлены лишь наиболее существенные и логичные переходы (рис. 3.4.3):

$$A_0 = \begin{vmatrix} 1-p_1 & p_1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{vmatrix} \quad A_1 = \begin{vmatrix} 1-p_2 & 0 & p_2 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{vmatrix} \quad (3.4.21)$$

Здесь для простоты введено обозначение

$$p_1 = p_{12}^{(0)}; \quad p_2 = p_{13}^{(1)}. \quad (3.4.22)$$

Вектор вероятностей штрафов $\mathbf{L} = (l_1, l_2, l_3)$ в этом случае естественно задать в виде

$$\mathbf{L} = (1-p, 1-q, 0), \quad (3.4.23)$$

где p и q определены выше в (3.4.8) и (3.4.9). Матрица автомата поиск—объект принимает вид

$$B = \begin{vmatrix} (1-p_1)p + (1-p)(1-p_2) & pp_1 & (1-p)p_2 \\ 1-q & q & 0 \\ 1 & 0 & 0 \end{vmatrix} \quad (3.4.24)$$

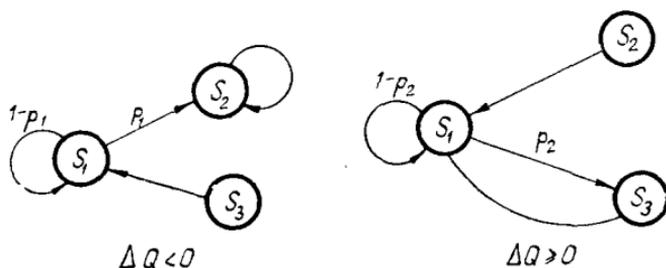


Рис. 3.4.3. Граф смешанного алгоритма случайного поиска.

Система уравнений для предельных вероятностей \bar{p}_1 , \bar{p}_2 и \bar{p}_3 имеет вид

$$\begin{aligned} [p(1-p_1) + (1-p)(1-p_2)]\bar{p}_1 + (1-q)\bar{p}_2 + \bar{p}_3 &= \bar{p}_1; \\ pp_1\bar{p}_1 + q\bar{p}_1 &= \bar{p}_2; \\ (1-p)p_2\bar{p}_1 &= \bar{p}_3; \\ \bar{p}_1 + \bar{p}_2 + \bar{p}_3 &= 1. \end{aligned} \quad (3.4.25)$$

Заметим, что первые три уравнения линейно зависимы, что заставляет расширить систему четвертым уравнением. Решая эту систему, получаем:

$$\begin{aligned} \bar{p}_1 &= \frac{1-q}{[1+(1-p)p_2](1-q) + pp_1}; \\ \bar{p}_2 &= \frac{pp_1}{[1+(1-p)p_2](1-q) + pp_1}; \\ \bar{p}_3 &= \frac{(1-p)(1-q)p_2}{[1+(1-p)p_2](1-q) + pp_1}. \end{aligned} \quad (3.4.26)$$

Следовательно, средний штраф равен

$$\bar{c} = \frac{(1-q)(1-p+pp_1)}{[1+(1-p)p_2](1-q) + pp_1}. \quad (3.4.27)$$

Проанализируем полученные выражения с точки зрения сопоставления алгоритмов поиска. Начнем с анализа известных алгоритмов, которые получаются при нулевых или единичных переходных вероятностях p_1 и p_2 .

1. $p_1 = p_2 = 0$ (алгоритм случайного блуждания). В этом случае средний штраф равен

$$\bar{c}_1 = 1 - p. \quad (3.4.28)$$

2. $p_1 = 0$; $p_2 = 1$ (алгоритм нелинейного поиска «с возвратом»); тогда

$$\bar{c}_2 = \frac{1-p}{2-p}. \quad (3.4.29)$$

3. $p_1=1; p_2=0$ (алгоритм поиска с линейной тактикой); тогда

$$\bar{c}_3 = \frac{1-q}{1-q+p}. \quad (3.4.30)$$

4. $p_1=p_2=1$ (линейная тактика с возвратом при неудачном случайном шаге); тогда

$$\bar{c}_4 = \frac{1-q}{(2-p)(1-q)+p}. \quad (3.4.31)$$

Определим на плоскости параметров объекта (p, q) зону преимущества каждого алгоритма.

Несложный анализ показывает, что

$$\bar{c}_1 \geq \bar{c}_i \quad (i=2, 3, 4), \quad (3.4.32)$$

т. е. случайное блуждание никогда не лучше других рассматриваемых алгоритмов. Это естественно, так как при $p_1=p_2=0$ нет целенаправленности, имеет место случайный перебор.

Далее,

$$\bar{c}_4 < \bar{c}_3, \quad (3.4.33)$$

т. е. линейный алгоритм с возвратом всегда лучше, чем без возврата, что также естественно. Сопоставляя 2-й и 3-й из указанных выше алгоритмов, получаем

$$\bar{c}_2 < \bar{c}_3, \text{ если } q < p^2 - p + 1, \quad (3.4.34)$$

и наоборот (см. рис. 3.4.4, а, где зона преимущества нелинейного алгоритма поиска заштрихована).

Сопоставление 2-го и 4-го алгоритмов приводит к следующему результату:

$$\bar{c}_2 < \bar{c}_4, \text{ если } q < \frac{1}{2-p}, \quad (3.4.35)$$

и наоборот (см. рис. 3.4.4, б, где зона целесообразного применения 2-го алгоритма заштрихована).

Эти результаты определяют оптимальные в данной ситуации алгоритмы и дают возможность построить

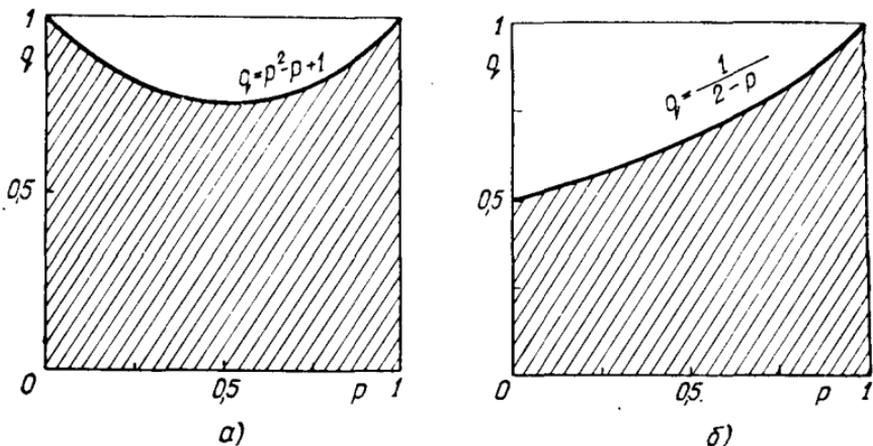


Рис. 3.4.4. Плоскость параметров объекта с зонами целесообразного применения алгоритмов.

алгоритм случайного поиска с адаптивной структурой при $p_2 = 1$:

$$p_1^{(N-1)} = \begin{cases} 0, & \text{если } \hat{q}_N(2 - \hat{p}_N) < 1; \\ 1, & \text{если } \hat{q}_N(2 - \hat{p}_N) > 1, \end{cases} \quad (3.4.36)$$

где \hat{p}_N и \hat{q}_N — текущие оценки параметров p и q объекта на N -м шаге поиска, которые можно производить по следующим формулам:

после случайного шага (s_1):

$$\hat{p}_N = \begin{cases} \frac{1}{N} [\hat{p}_{N-1}(N-1) + 1], & \text{если } \Delta Q_{N-1} < 0; \\ \frac{1}{N} \hat{p}_{N-1}(N-1), & \text{если } \Delta Q_{N-1} \geq 0; \end{cases} \quad (3.4.37)$$

$$\hat{q}_N = \hat{q}_{N-1};$$

после спускового шага (s_2):

$$\hat{p}_N = \hat{p}_{N-1};$$

$$\hat{q}_N = \begin{cases} \frac{1}{N} [\hat{q}_{N-1}(N-1) + 1], & \text{если } \Delta Q_{N-1} < 0; \\ \frac{1}{N} \hat{q}_{N-1}(N-1), & \text{если } \Delta Q_{N-1} \geq 0; \end{cases} \quad (3.4.38)$$

после возврата (s_3):

$$\begin{aligned} \hat{p}_N &= \hat{p}_{N-1}; \\ \hat{q}_N &= \hat{q}_{N-1}. \end{aligned} \tag{3.4.39}$$

Предложенные формулы гарантируют асимптотическую сходимость к истинным значениям параметров p и q объекта, т. е.

$$\begin{aligned} \lim_{N \rightarrow \infty} \hat{p}_N &= p; \\ \lim_{N \rightarrow \infty} \hat{q}_N &= q, \end{aligned} \tag{3.4.40}$$

если p и $q = \text{const}$. Однако для реальных объектов это предположение слишком грубо, и следует считать, что значения p и q изменяются в зависимости от состояния объекта. В соответствии с этим формулы (3.4.37) — (3.4.39) следует записать, например, в следующем виде:

после s_1 :

$$\hat{p}_N = \begin{cases} \hat{p}_{N-1}(1 + \alpha_1) - \alpha_1, & \text{если } \Delta Q < 0; \\ \hat{p}_{N-1}(1 - \alpha_2), & \text{если } \Delta Q \geq 0; \end{cases} \tag{3.4.41}$$

$$\hat{q}_N = \hat{q}_{N-1};$$

после s_2 :

$$\hat{p}_N = \hat{p}_{N-1};$$

$$\hat{q}_N = \begin{cases} \hat{q}_{N-1}(1 + \beta_1) - \beta_1, & \text{если } \Delta Q < 0; \\ \hat{q}_{N-1}(1 - \beta_2), & \text{если } \Delta Q \geq 0; \end{cases} \tag{3.4.42}$$

после s_3 используем (3.4.39).

Параметры α_i, β_i ($i=1, 2$) в этих формулах должны быть назначены исходя из конкретных соображений.

Проведем анализ смешанных алгоритмов. При $p_1, p_2 \neq 0$; 1 получаем смешанные алгоритмы случайного поиска. Решаем для этого случая задачу (3.4.15), а именно

$$\bar{c}(p_1, p_2) = \frac{(1-q)(1-p+pp_1)}{[1-(1-p)p_2](1-q)+pp_1} \rightarrow \min_{p_1, p_2 \in [0,1]} \tag{3.4.43}$$

Как видим, это задача нелинейного программирования. Однако несложный анализ линий равного уровня среднего штрафа

$$\bar{c}(p_1, p_2) = \text{const} \quad (3.4.44)$$

показывает, что они являются прямыми. Это обстоятельство гарантирует решение задачи (3.4.43) на границе области $0 \leq p_1, p_2 \leq 1$, что делает оптимальную стратегию чистой.

Сказанное позволяет сделать вывод о том, что двойная рандомизация, рассмотренная в этом параграфе, по-видимому, не может повысить быстродействия поиска.

§ 3.5. ВЫБОР ОПТИМАЛЬНОЙ ПОСЛЕДОВАТЕЛЬНОСТИ АЛГОРИТМОВ ПОИСКА

В этом параграфе, используя подход, изложенный в работе [55], построим оптимальную последовательность алгоритмов поиска. Пусть имеется набор m алгоритмов (стратегий), предназначенных для поиска экстремума функции $Q(X)$. Пусть каждый алгоритм обладает определенным набором операторов, которые примем за состояния автомата. Пусть также заданы вероятности переходов $p_{ij}^{(k)}$ из состояния i в состояние j ($j = 1, \dots, n$) и соответствующие смещения в цели (доходы) $c_{ij}^{(k)}$ для каждого алгоритма ($k = 1, \dots, m$), причем строки (состояния) матрицы $p_{ij}^{(k)}$ удовлетворяют усло-

виям стохастичности $\sum_{j=1}^n p_{ij}^{(k)} = 1$ и $0 \leq p_{ij}^{(k)} \leq 1$. Необходи-

мо для всех состояний i и всех шагов (переходов) N ($N = 1, 2, 3, \dots$) выбрать такую последовательность алгоритмов, которая бы максимизировала среднее смещение к цели (доход).

Для решения поставленной задачи оптимизации применим «принцип оптимальности» динамического про-

граммирования марковского типа, предложенный Р. Беллманом [56] и использованный Р. Ховардом [57] для решения определенных задач.

Приведем некоторые результаты работ [56—57], которые нам понадобятся в дальнейшем. Среднее смещение к цели (доход) для k -го алгоритма на $(N+1)$ -м переходе при условии, что система находится в i -м состоянии, равно

$$s_i^{(k)}(N+1) = \sum_{j=1}^n p_{ij}^{(k)} [c_{ij}^{(k)} + s_j^{(k)}(N)]. \quad (3.5.1)$$

Таким образом, средний доход в i -м состоянии получается путем осреднения случайной величины $c_{ij}^{(k)} + s_j^{(k)}(N)$ по всем состояниям j . Здесь $c_{ij}^{(k)}$ — доход при переходе из состояния i в состояние j — в соответствии с k -м алгоритмом; $s_j(N)$ — средний доход, получаемый в состоянии j за оставшиеся N шагов.

Обозначим через $d_i(N)$ номер алгоритма, выбираемый в состоянии i на N -м шаге. Тогда оптимальной считается такая последовательность $d_i(N)$, которая максимизирует средний доход для всех i и N :

$$s_i^{(k)}(N+1) = \max_{1 \leq k \leq m} \left[g_i^{(k)} + \sum_{j=1}^n p_{ij}^{(k)} s_j(N) \right], \quad (3.5.2)$$

где $g_i^{(k)}$ — непосредственно ожидаемый доход в момент выхода системы из состояния i по алгоритму k ;

$$g_i^{(k)} = \sum_{j=1}^n p_{ij}^{(k)} c_{ij}^{(k)}. \quad (3.5.3)$$

Решая рекуррентное выражение (3.5.2), находим оптимальную последовательность $d_i(N)$.

Для облегчения вычисления вектора $s(N)$, а также для удобства теоретических выкладок и исследования поведения системы в переходный период применяется метод производящих функций (Z -преобразований) [57]. Так, например, после применения Z -преобразования вектор

$$\mathbf{S}(N+1) = \mathbf{g} + \mathbf{P}\mathbf{S}(N) \quad (3.5.4)$$

запишется в виде

$$\mathbf{W}(Z) = \frac{Z}{1-Z} (I-ZP)^{-1} \mathbf{g} + (I-ZP)^{-1} \mathbf{S}(0), \quad (3.5.5)$$

где

$$\mathbf{W}(Z) = \sum_{N=0}^{\infty} \mathbf{S}(N) Z^N. \quad (3.5.6)$$

Обратное преобразование вектора $\mathbf{W}(Z)$ дает искомый вектор $\mathbf{S}(N)$. Для некоторых алгоритмов случайного поиска этот вектор вычислен в работе [58].

При больших N вместо (3.5.1) имеем следующую систему уравнений:

$$t + v_i = g_i^{(k)} + \sum_{j=1}^k p_{ij}^{(k)} v_j \quad (3.5.7)$$

$(i=1, \dots, N).$

Здесь v_i называется относительным весом [57], t — доходом процесса.

Так как система уравнений (3.5.7) состоит из n уравнений с $n+1$ неизвестным (неизвестны t и v_i ($i=1, \dots, n$)), то одно из v_i полагают равным нулю, что, как показано в работе [57], не влияет на определение оптимальных d_i . Вся процедура определения оптимальных d_i при больших N сводится

а) к решению системы уравнений (3.5.7) относительно v_i ;

б) к нахождению для каждого состояния i стратегии d_i , для которой имеет место

$$g_i^{(d_i)} + \sum_{j=1}^n p_{ij}^{(d_i)} v_j = \max_k \left(g_i^{(k)} + \sum_{j=1}^n p_{ij}^{(k)} v_j \right); \quad (3.5.8)$$

в) к замене в формуле (3.5.8) $g_i^{(k)}$ на $g_i^{(d_i)}$, а $p_{ij}^{(k)}$ на $p_{ij}^{(d_i)}$ и к повторному решению системы (3.5.7).

При этом от итерации к итерации доход t увеличивается, т. е. получается улучшение решения. Остановка

процесса производится при двухразовом совпадении вектора d_i , который в дальнейшем принимается за оптимальный. Причем оптимальному вектору d_i соответствует максимальный доход t .

Пример. Рассмотрим линейную модель

$$Q(\mathbf{X}) = A^T \mathbf{X}, \quad (3.5.9)$$

где T — знак транспонирования. Пусть каждое измерение функции $Q(\mathbf{X})$ сопровождается нормально распределенной независимой помехой $\varepsilon(\sigma)$ с нулевым математическим ожиданием и дисперсией $\sigma^2/2$, т. е.

$$Q'(\mathbf{X}) = Q(\mathbf{X}) + \varepsilon(\sigma). \quad (3.5.10)$$

В качестве алгоритмов оптимизации рассмотрим следующие алгоритмы случайного поиска с «поощрением» случайностью:

а) алгоритм с возвратом при неудачном шаге, для которого матрица вероятностей переходов P и матрица смещений C записываются в виде [58]

$$P = \begin{vmatrix} \frac{1}{2} & \frac{1}{2} \\ 1 & 0 \end{vmatrix}, \quad C = \begin{vmatrix} c - \Phi\left(\frac{|\nabla Q|}{\sigma}\right) & -c - \Phi\left(\frac{|\nabla Q|}{\sigma}\right) \\ c - \Phi\left(\frac{|\nabla Q|}{\sigma}\right) & 0 \end{vmatrix} \quad (3.5.11)$$

б) тот же самый алгоритм с применением процедуры накопления; тогда в каждой точке \mathbf{X}_s имеем

$$Q'(\mathbf{X}_s) = Q(\mathbf{X}_s) + \frac{1}{T} \sum_{t=1}^T \varepsilon_t \quad (T=2, 3, \dots). \quad (3.5.12)$$

В этом случае матрицы P и C принимают следующий вид:

$$P = \begin{vmatrix} \frac{1}{2} & \frac{1}{2} \\ 1 & 0 \end{vmatrix}$$

$$C = \begin{vmatrix} W(T, r) \left[c - \Phi \left(\frac{|\nabla Q|}{\sigma_*} \right) \right] \\ \frac{1+W(T, r)}{2} \left[c - \Phi \left(\frac{|\nabla Q|}{\sigma_*} \right) \right] \\ - \frac{1+W(T, r)}{2} \left[c + \Phi \left(\frac{|\nabla Q|}{\sigma_*} \right) \right] \\ 0 \end{vmatrix} \quad (3.5.13)$$

Здесь

$$\Phi(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt; \quad \sigma_*^2 = \frac{\sigma^2}{T}; \quad 0 \leq W(T, r) \leq 1;$$

$|\nabla Q|$ — модуль градиента.

Функция $W(T, r)$ учитывает число наблюдений T в (3.5.12) и стоимость r каждого наблюдения Q .

Пусть $W(T, r) = (1/T)W(r)$. Примем, что $W(r)$ монотонно убывает с возрастанием r . Множитель $(1 + W(T, r))/2$ появляется вследствие равенства количества наблюдений над $Q(X)$ при переходе p_{11} и переходах $p_{12} + p_{21}$. Количество наблюдений при переходах p_{12} и p_{21} также принято равным. Положим $|\nabla Q| = 1$; $W(T, r) = 2/3$; $\sigma = 2$; $T = 4$. Тогда $\sigma_*^2 = \sigma^2/T = 1$. Пусть также $c_{11} = 1$. Тогда выражения (3.5.11) и (3.5.12) можно будет переписать в виде

$$P_1 = \begin{vmatrix} \frac{1}{2} & \frac{1}{2} \\ 1 & 0 \end{vmatrix} \quad C_1 = \begin{vmatrix} 0,85 & -1,15 \\ 0,85 & 0 \end{vmatrix} \quad (3.5.14)$$

$$P_2 = \begin{vmatrix} \frac{1}{2} & \frac{1}{2} \\ 1 & 0 \end{vmatrix} \quad C_2 = \begin{vmatrix} \frac{2}{3} \cdot 0,92 & -\frac{5}{6} \cdot 1,08 \\ \frac{5}{6} \cdot 0,92 & 0 \end{vmatrix} \quad (3.5.15)$$

Таблица 3.5.1

Состояние i	Стратегия k	Критерий $g_i^k + \sum_{j=1}^2 p_{ij}^k v_j$
1	1	$-0,15 + 0,5 \cdot 0,65 = 0,17$
	2	$-0,12 + 0,5 \cdot 0,65 = 0,2$
2	1	0,85
	2	0,8

Рассмотрим алгоритм с возвратом и матрицами (3.5.14) и (3.5.15). Будем считать, что число переходов N велико.

Чтобы начать процесс последовательных решений, положим $t=0$; $v_1=v_2=0$. Тогда из (3.5.7) $g_1^{(1)} = -0,15$; $g_1^{(2)} = -0,12$; $g_2^{(1)} = 0,85$; $g_2^{(2)} = 0,80$. Отсюда $\max_k g_1^{(k)} = -0,12$; $\max_k g_2^{(k)} = 0,85$. Следовательно, в первом со-

стоянии необходимо выбрать вторую стратегию, а во втором состоянии — первую. Новая матрица, соответствующая этому решению, должна состоять из первой строки матрицы P_2 (3.5.15) и второй строки матрицы P_1 (3.5.14). Так как матрицы P_1 (3.5.14) и P_2 (3.5.15) одинаковы, то вновь получим матрицу P_1 (3.5.14).

Теперь записываем систему уравнений (3.5.7), в которой использованы элементы матрицы P_1 (3.5.14) и ко-

ординаты вектора $g = \begin{pmatrix} -0,12 \\ 0,85 \end{pmatrix}$, т. е.

$$t + v_1 = -0,12 + 0,5v_1 + 0,5v_2; \quad (3.5.16)$$

$$t + v_2 = 0,85 + v_1.$$

Например, при $v_1=0$ получим $v_2=0,65$ и доход $t=0,2$. Возвращаясь к процедуре (3.5.8) при $v_1=0$, $v_2=0,65$, имеем таблицу 3.5.1. Из данных таблицы можно заключить, что в первом состоянии необходимо выбрать вторую стратегию, а во втором — первую. Эти стратегии совпадают со стратегиями, полученными на предыдущем шаге, значит, процесс сошелся и доход t достиг своего

Таблица 3.5.2

W	Состояние i	Оптимальная стратегия
0—0,60	1	1
	2	1
0,60—0,85	1	2
	2	1
0,85—1,00	1	2
	2	2

максимума, а именно 0,2. Таким образом, придерживаясь при $W=2/3$ второй стратегии в первом состоянии и первой стратегии во втором состоянии, получим максимум среднего смещения к цели. Легко проверить, что остальные три комбинации вектора d обеспечивают меньший доход и меньшее среднее смещение к цели.

Оптимальная зависимость вектора d от параметра W приведена в таблице 3.5.2. Из таблицы вытекает, что при малой стоимости каждого наблюдения ($0,85 < W \leq 1$) следует применять алгоритм с возвратом и с накоплением, при большой стоимости ($0 \leq W \leq 0,6$) — алгоритмы с возвратом и без накопления помехи.

При стоимости наблюдения Q , соответствующей $0,6 < W \leq 0,85$, в первом состоянии оптимальной является вторая стратегия, а во втором — первая. Иными словами, после удачного шага следует применить алгоритм с накоплением, а после неудачного — алгоритм без накопления. Такой выбор стратегии при $0,6 < W \leq 0,85$ можно объяснить следующим образом: накопление (вторая стратегия) вводится в новой ситуации, т. е. после удачного перехода p_{11} , и не вводится (первая стратегия) после неудачного перехода p_{12} , за которым следует переход $p_{21}=1$, в результате чего система возвращается в исходное положение, где $Q'(X_s)$ уже было ранее определено путем накопления (3.5.12).

ЛИТЕРАТУРА

1. *Л. А. Растргин*. Статистические методы поиска. М., «Наука», 1968.
2. *Ю. И. Неймарк, В. П. Григоренко, А. Н. Рапопорт*. Об оптимизации независимыми детерминированными и стохастическими автоматами. — Уч. зап. Научно-исследовательского ин-та прикл. математики и кибернетики и ф-та вычисл. математики и кибернетики Горьковского гос. ун-та им. Н. И. Лобачевского, Прикладная математика и кибернетика. Горький, 1967. [Ротапринт].
3. *Л. А. Растргин*. Случайный поиск в задачах оптимизации многопараметрических систем. Рига, «Зинатне», 1965.
4. *Л. А. Растргин*. Случайный поиск с линейной тактикой. Рига, «Зинатне», 1971.
5. Теория и применение случайного поиска. Под общ. ред. *Л. А. Растргина*. Рига, «Зинатне», 1969.
6. *Л. А. Растргин, К. К. Рипа*. Моделирование обучения при экстремальном регулировании многопараметрических систем методом случайного поиска. — Автоматика (Киев), 1964, 5.
7. *Л. А. Растргин, К. К. Рипа*. Непрерывный алгоритм самообучения при многопараметрической оптимизации методом случайного поиска. — В кн.: Автоматика и вычислительная техника, 10. Рига, «Зинатне», 1965.
8. *Л. А. Растргин, К. К. Рипа, Л. В. Сытенко*. Автоматические оптимизаторы, работающие по методу статистического поиска с самообучением. — В кн.: Самообучающиеся автоматические системы. М., «Наука», 1966.
9. *Л. А. Растргин, К. К. Рипа*. Применение цепей Маркова к описанию и исследованию динамики покоординатного обучения при статистической оптимизации многопараметрических систем. — В кн.: Методы статистической оптимизации. Рига, «Зинатне», 1968.
10. *Л. А. Растргин, К. К. Рипа*. Статистический поиск как вероятностный автомат. — Автоматика и вычислительная техника (Рига), 1971, 1.
11. *Л. А. Растргин, Л. В. Сытенко*. Многоканальные статистические оптимизаторы. М., «Энергия», 1973.

12. *В. П. Григоренко, Ю. И. Неймарк, А. Н. Рапопорт.* Об одной гомеостатической модели оптимизации. — В кн.: Вопросы кибернетики и вычислительной математики. Вып. 28. Ташкент, ФАН, 1969.
13. *В. П. Григоренко, Ю. И. Неймарк, А. Н. Рапопорт.* Оптимизация коллективом независимых автоматов и игры автоматов. — Изв. высш. учебн. завед., Радиофизика, 1968, 7.
14. *К. К. Рипа.* Сравнение свойств оптимизирующих автоматов и случайного поиска. — В кн.: Вопросы кибернетики и вычислительной математики. Вып. 28. Ташкент, ФАН, 1969.
15. *К. К. Рипа.* Некоторые статистические свойства оптимизирующих автоматов и случайного поиска. — Автоматика и вычислительная техника (Рига), 1970, 3.
16. *В. П. Григоренко, А. Н. Рапопорт.* К теории поиска коллективом независимых автоматов. — Изв. высш. учебн. завед., Радиофизика, 1970, 11.
17. *И. Л. Антонов.* Самобалансировка вращающегося ротора методом случайного поиска. — В кн.: Колебания и прочность при переменных напряжениях. М., «Наука», 1965.
18. *И. Л. Антонов.* Переходный процесс в двумерной экстремальной системе при наличии запрещенных областей и случайном методе поиска. — В кн.: Задачи статистической оптимизации. Рига, «Зинатне», 1971.
19. *Г. А. Медведев, А. П. Рыжаков.* О применении алгоритмов случайного поиска в системах автоматической оптимизации. — В кн.: Задачи статистической оптимизации. Рига, «Зинатне», 1971.
20. *В. П. Григоренко, Ю. И. Неймарк, А. Н. Рапопорт.* Исследование одной модели коллективного поведения. — Изв. высш. учебн. завед., Радиофизика, 1970, 8.
21. *Р. Г. Бухараев.* Вероятностные автоматы. Казань, Изд-во Казанского ун-та, 1970.
22. *Р. Г. Бухараев.* Теория вероятностных автоматов. — Кибернетика, 1968, 2.
23. *Е. У. Карлайл.* Приведенные формы для стохастических последовательностных машин. — В кн.: Кибернетический сборник. Новая серия. Вып. 3. М., «Мир», 1966.
24. *Р. Н. Starke.* Abstrakte Automaten. Berlin, VEB Deutscher Verlag der Wissenschaften, 1969.
25. *Ю. А. Шрейдер.* Модели обучения и управляющие системы. — В кн.: *Р. Буш, Ф. Мостеллер.* Стохастические модели обучаемости. М., Физматгиз, 1962.
26. *Н. Я. Паршенков, В. М. Ченцов.* О теории стохастических автоматов. — В кн.: Дискретные автоматы и сети связи. М., «Наука», 1970.
27. *А. А. Лоренц.* Вопросы сводимости конечных вероятностных автоматов. — Автоматика и вычислительная техника (Рига), 1969, 1.
28. *М. Л. Цетлин.* Исследования по теории автоматов и моделированию биологических систем. М., «Наука», 1969.
29. *Дж. Кемени, Дж. Снелл.* Конечные цепи Маркова. М., «Наука», 1970.
30. *В. И. Романовский.* Дискретные цепи Маркова. М., Гостехиздат, 1949.

31. *Л. А. Растринг, К. К. Рипа.* Представление случайного поиска в качестве стохастического автомата. — В кн.: Задачи стохастической оптимизации. Рига, «Зинатне», 1971.
32. *В. И. Варшавский, И. П. Воронцова.* О поведении стохастических автоматов с переменной структурой. — Автоматика и телемеханика, 1963, 3.
33. *В. И. Варшавский, И. П. Воронцова, М. Л. Цетлин.* «Обучение» стохастических автоматов. — В кн.: Биологические аспекты кибернетики. М., Изд-во АН СССР, 1962.
34. *R. W. Mc Laren.* A Stochastic Automaton Model for the Synthesis of Learning Systems. — JEEE Trans. Syst. Sci. Cybern., 1966, 2.
35. *G. J. McMurtry, K. S. Fu.* A Variable Structure Automaton Used as a Multi-Modal Searching Technique. — Proc. Nat. Electr. Conf., 1965, 21.
36. *B. Chandrasekaran, D. W. C. Shen.* On Expediency and Convergence in Variable-Structure Automata. — JEEE Trans. Syst. Sci. Cybern., 1968, 1.
37. *В. М. Глушков.* Введение в кибернетику. Киев, Изд-во АН УССР, 1964.
38. *Б. В. Гнеденко.* Курс теории вероятностей. М., Физматгиз, 1961.
39. *В. А. Поспелов.* Игры и автоматы. М.—Л., «Энергия», 1966.
40. *У. Р. Эшби.* Конструкция мозга. М., ИЛ, 1962.
41. *Л. А. Растринг.* Случайный поиск как стохастический автомат. — В кн.: Проблемы статистической оптимизации. Рига, «Зинатне», 1968.
42. *К. К. Рипа.* Случайный поиск экстремума многомерного объекта как стохастический автомат. — В кн.: Задачи статистической оптимизации. Рига, «Зинатне», 1971.
43. *Л. А. Растринг, Л. В. Сытенко.* Релейный многоканальный оптимизатор, работающий по алгоритмам случайного поиска. — В кн.: Автоматика и вычислительная техника, 10. Рига, «Зинатне», 1965.
44. *П. Н. Попов, Л. В. Сытенко.* Быстродействие и статистические свойства случайного поиска на гиперкубе. — Автоматика и вычислительная техника (Рига), 1970, 3.
45. *В. В. Налимов, Н. А. Чернова.* Статистические методы планирования экстремальных экспериментов. М., «Наука», 1965.
46. *А. А. Фельдбаум.* Вычислительные устройства в автоматических системах. М., Физматгиз, 1959.
47. *Чжун Кай-лай.* Однородные цепи Маркова. М., «Мир», 1964.
48. *А. П. Мишина, И. В. Проскуряков.* Высшая алгебра. М., Физматгиз, 1962. (Справочная математическая библиотека).
49. *К. К. Рипа.* Динамика покоординатного обучения при статистической оптимизации в обстановке помех. — Автоматика и вычислительная техника (Рига), 1967, 3.
50. *В. Феллер.* Введение в теорию вероятностей и ее приложения. М., «Мир», 1964.
51. *К. К. Рипа.* Динамика покоординатного обучения с малой интенсивностью при статистической оптимизации в обстановке помех. — В кн.: Проблемы статистической оптимизации. Рига, «Зинатне», 1968.
52. *Ф. Р. Гантмахер.* Теория матриц. М., «Наука», 1968.

53. Л. А. Растргин, К. К. Рипа. Синтез оптимальных алгоритмов в классе алгоритмов дискретно распределенного случайного поиска для марковских объектов оптимизации. — Автоматика и вычислительная техника (Рига), 1971, 6.
54. К. К. Рипа. Оптимальные алгоритмы дискретно распределенного случайного поиска. — В кн.: Случайный поиск в многопараметрических задачах. Рига, «Зинатне», 1972.
55. П. А. Мороз, Я. С. Рубинштейн. О выборе оптимальной последовательности алгоритмов. — В кн.: Проблемы случайного поиска. Рига, «Зинатне», 1973.
56. Р. Беллман. Динамическое программирование. М., ИЛ, 1960.
57. Р. А. Ховард. Динамическое программирование и марковские процессы. М., «Советское радио», 1964.
58. В. В. Шкварцов. Сравнительная характеристика некоторых алгоритмов статистического поиска. — В кн.: Методика статистической оптимизации. Рига, «Зинатне», 1968.

ОГЛАВЛЕНИЕ

Предисловие	5
Введение	7
§ 0.1. Задача поисковой оптимизации	7
§ 0.2. Алгоритмы случайного поиска	11
§ 0.3. Некоторые свойства вероятностных автоматов	16
§ 0.4. Автоматы в случайных средах	24
Глава I. Поиск без самообучения	35
§ 1.1. Случайный поиск как вероятностный автомат	35
§ 1.2. Представление случайного поиска в виде вероятностного автомата	42
§ 1.3. Линейный алгоритм случайного поиска	49
§ 1.4. Нелинейный алгоритм случайного поиска	60
§ 1.5. Алгоритм Гаусса-Зейделя и его стохастический вариант	71
§ 1.6. Детерминированный аналог нелинейного алгоритма случайного поиска	80
§ 1.7. Алгоритм последовательного градиента	87
§ 1.8. Алгоритм параллельного градиента	96
§ 1.9. Алгоритм наискорейшего спуска	104
§ 1.10. Сопоставление непрерывного и автоматного случайного поиска	118
§ 1.11. Переходный процесс в двумерной экстремальной системе при наличии запрещенных областей и случайном методе поиска	123
Глава II. Поиск с самообучением	137
§ 2.1. Алгоритмы самообучения при случайном поиске как вероятностные автоматы	137
§ 2.2. Свойства покоординатного самообучения при оптимизации в отсутствие помех	172

§ 2.3. Свойства покоординатного самообучения при статистической оптимизации с помехами	192
§ 2.4. Свойства покоординатного самообучения с малой интенсивностью при статистической оптимизации с помехами	199
§ 2.5. Свойства покоординатного самообучения с детерминированной переходной функцией, не зависящей от выхода автомата	211
§ 2.6. Статистические свойства коллектива оптимизирующих автоматов и их сравнение со свойствами покоординатного самообучения	221
§ 2.7. Статистические свойства коллектива вероятностных автоматов при оптимизации функции n переменных	233
§ 2.8. Сравнение алгоритмов самообучения	239
§ 2.9. Тройичный алгоритм поиска	261
Глава III. Оптимальные алгоритмы	271
§ 3.1. Постановка задачи синтеза оптимального алгоритма в классе алгоритмов дискретного распределенного случайного поиска	271
§ 3.2. Оптимальные алгоритмы поиска для двумерного объекта оптимизации	277
§ 3.3. Синтез оптимальной структуры алгоритмов самообучения	304
§ 3.4. Смешанные алгоритмы случайного поиска	320
§ 3.5. Выбор оптимальной последовательности алгоритмов поиска	330
Литература	337

Леонард Андреевич Растрингин,
Карл Карлович Рипа

АВТОМАТНАЯ ТЕОРИЯ
СЛУЧАЙНОГО ПОИСКА

Обложка *Р. Эренштейна.*

Редактор *Л. Тюрина.*

Художественный редактор *Г. Крутой.*

Технический редактор *В. Калве.*

Корректор *Т. Сосницкая.*

Сдано в набор 30 июня 1972 г. Подписано к печати 15 ноября 1973 г. Типогр. бумага № 1, Формат 84×108¹/₃₂. 10,75 физ. печ. л.; 18,06 усл. печ. л.; 14,47 уч.-изд. л. Тираж 3000 экз. ЯТ 06458. Цена 1 руб. 17 коп. Издательство «Зинатне», г. Рига, ул. Тургенева, 19. Отпечатано в типографии «Советская Латвия» Государственного комитета Совета Министров Латвийской ССР по делам издательств, полиграфии и книжной торговли, г. Рига, ул. Дзирнаву, 57. Заказ № 2014.

114

15925