1. Символьная модель ГА

1.1. Постановка задачи

Рассмотрим оптимизационную задачу $\max f(x)$ с допустимым множеством

$$D = \{x = (x_1, x_2, \ldots, x_N) | x_i \in [a_i, b_i], i = 1, 2, \ldots, N\},$$

где $f(x)$ — максимизируемая (целевая) скалярная многопараметрическая функция, которая может иметь несколько глобальных экстремумов, прямоугольная область D — область поиска, D — подмножество R^N. Предполагается, что о функции $f(x)$ известно лишь то, что она определена в любой точке области D. Никакая дополнительная информация о характере функции и ее свойствах (дифференцируемость, непрерывность и т.д.) в процессе поиска не учитывается.

Под решением поставленной задачи будем понимать вектор $x = (x_1, x_2, \ldots, x_N)$. Оптимальным решением будем считать вектор $x = x^*$, при котором целевая функция $f(x)$ принимает максимальное значение. Исходя из предположения о возможной многоэкстремальности $f(x)$, оптимальное решение может быть не единственным.

1.2. Символьная модель

Для того, чтобы построить пространство представлений под генетический алгоритм для задачи оптимизации необходимо дискретизировать пространство параметров скалярной функции $f(x)$. Параметры x обычно кодируются бинарной строкой s. Используя целевую функцию $f(x)$, можно построить функцию $\mu(s)$, отобразив, когда это необходимо, $f(x)$ на положительную полуось. Это делается для того, чтобы гарантировать прямое соотношение между значением целевой функции и приспособленностью решения. Впоследствии ГА работает именно с модифицированной целевой функцией $\mu(s)$. Таким образом, каждое возможное решение s, имеющее соответствующую приспособленность $\mu(s)$, представляет решение x. Обычно переход из пространства параметров D в хеммингово пространство бинарных строк осуществляется кодированием переменных $x_1, x_2, \ldots, x_N$ в двоичные целочисленные строки. Длина строк определяется требуемой точностью решения. При этом пространство параметров должно быть дискретизировано таким образом, чтобы расстояние между узлами дискретизации соответствовало требуемой точности. Предположим, по условию задачи с функцией от двух переменных x_1 и x_2, определенной на прямоугольной области $D = \{0 < x_1 < 1; 0 < x_2 < 1\}$, требуется локализовать решение x^* с точностью по каждому из параметров 10^{-6}. Для достижения такой точности пространство параметров дискретизуется равномерной сеткой с $(b_i - a_i)/(10^{-6}) \sim 10^6$ узлами по каждой координате. Закодировать такое количество узлов можно $l = 20$ битами, где l определяется из условия $10^6 < 2^l + 1$. Получается, что общая длина бинарной строки кодировки для двумерной задачи составит $2 \times 20 = 40$ бит. При таком способе кодирования значения варьируемых параметров решений будут располагаться по узлам решетки, дискретизующей D. Соответственно, если кодировки двух решений будут совпадать, то будут совпадать и значения параметров обоих решений.

Во многих случаях такая модель может оказаться неэффективной. Кроме того, что она достаточно громоздка (каким будет хеммингово пространство поиска для задачи с сотней параметров?!). Практика показывает, что длинная кодировка повышает вероятность «преждевременной» сходимости. К тому же применение длинных кодировок вовсе не гарантирует, что найденное решение будет обладать требуемой точностью, поскольку этого, в принципе, не гарантирует сам ГА. Согласно этому, для того, чтобы применять ГА к задаче, сначала выбирается метод кодирования решений в виде строки. Фиксированная длина (l-бит) двоичной кодировки означает, что любая из 2^l возможных бинарных строк представляет возможное решение задачи. По существу, такая кодировка соответствует разбиению пространства параметров на гиперкубы, которым соответствуют уникальные комбинации битов в строке-хромосоме. Идея ГА состоит в том, чтобы, манипулируя имеющейся совокупностью бинарных представлений, с помощью ряда генетических операторов получать новые строки, т.е. перемещаться в новые гиперкубики. Получив бинарную комбинацию для нового решения, формируется вектор (операция декодирования) со значениями из соответствующего гиперкуба. Таким образом, каждое решение генетического алгоритма будет иметь следующую структуру (точка в пространстве параметров фенотип):

$$x = (x_1, x_2, \ldots, x_N) \in D \subset \mathrm{R}^N.$$

Бинарная строка s фиксированной длины, однозначно идентифицирующая гиперкуб разбиения пространства параметров (генотип) $s = (\beta_1, \beta_2, \ldots, \beta_l)$ принадлежит S, где S — пространство представлений бинарных строк длины l.

Скалярная величина μ, соответствующая значению целевой функции в точке x (пригодность): $\mu = f(x)$.

В терминологии, принятой в теории ГА, такую структуру принято называть особью.

Вообще могут существовать особи, обладающие различными фенотипическими признаками, но имеющие одинаковые генотипы (такое явление встречается в природе, например, у однояйцовых близнецов). Это позволяет использовать более крупное разбиение пространства параметров, сужая пространство бинарных строк S и делая при этом длину хромосомного набора короче. Многообразие точек, распределяемых в небольших гиперкубиках, позволяет достигать высокой точности.

1.3. Геометрическая интерпретация символьной модели

В предыдущем разделе было рассмотрено, каким образом будет осуществляться переход из евклидова пространства параметров в пространство представлений (бинарных строк). Рассмотрим эту процедуру на конкретном примере простой одномерной функции $f(x) = 10 + x\sin x$, определенной на отрезке [0, 10]. Пусть кодирование будет осуществляться бинарными строками длины 3 (см. рис. 16), то есть отрезок [0, 10] нужно разбить на $2^3 = 8$ подынтервалов, каждому из которых будет соответствовать уникальная двоичная комбинация, получаемая переводом номера подынтервала, считая слева направо, в двоичную систему. Длина каждого такого интервала будет $h = 10 : 8 = 1{,}25$.

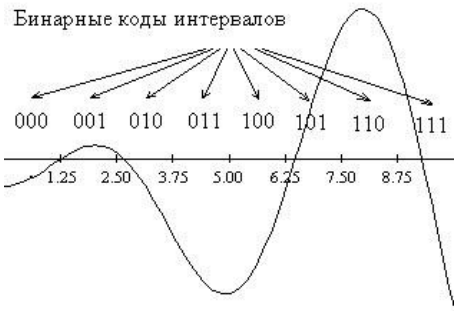

Рис. 16. Построение символьной модели для одномерной задачи с использованием трехбитового представления

Пространством поиска, таким образом, становится множество всех бинарных строк длины 3. Это пространство можно представить в виде трехмерного куба, вершинам которого соответствуют кодовые комбинации, расставленные так, что хэммингово расстояние между смежными вершинами равно 1 (см. рис. 17).

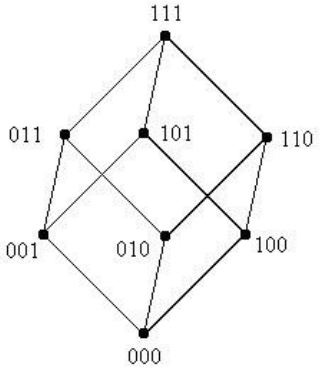

Рис. 17. Пространство поиска для трехбитового представления

Задача алгоритма поиска заключается в том, чтобы, следуя некоторому правилу, перемещаться в новые вершины этого куба, что будет соответствовать исследованию новых подинтервалов в пространстве D.

1.4. Шима

Хотя внешне кажется, что ГА обрабатывает строки, на самом деле при этом неявно происходит обработка шим, которые представляют шаблоны подобия между строками. ГА практически не может заниматься полным перебором всех представлений в пространстве поиска. Однако он может производить выборку значительного числа гиперплоскостей в областях поиска с высокой приспособленностью. Каждая такая гиперплоскость соответствует множеству похожих строк с высокой приспособленностью.

Шима (schema) — это строка длины l (что и длина любой строки популяции), состоящая из знаков алфавита $\{0;1;*\}$, где $\{*\}$ — неопределенный символ. Каждая шима определяет множество всех бинарных строк длины l, имеющих в соответствующих позициях либо 0, либо 1, в зависимости от того, какой бит находится в соответствующей позиции самой шимы. Шима, не содержащая ни одного неопределенного символа, является некоторой строкой. Шима с одним неопределенным символом описывает две бинарные строки, а с двумя — четыре строки. Например, шима, $10**1$, определяет собой множество из четырех пятибитовых строк {10001; 10011; 10101; 10111}. Нетрудно заметить, что шима с r-неопределенными символами описывает 2^r бинарных строк. С другой стороны, каждая стока длины m описывается 2^m шимами. Следовательно, в популяции из n таких строк число возможных шим может достигать $n2^m$! При этом большая часть шим вероятно будет менее приспособленной[1] остальных, что может привести к эпистазу. Поэтому рекомендуется содавать начальную популяцию из шим с высокой приспособленностью.

Все шимы различны между собой. Основными характеристиками шин являются порядок и длина.

Порядок шимы $o(S)$ (order) — это число фиксированных битов (0 или 1) в шиме S.

Определяющая длина $\delta(S)$ (defining length) — это расстояние между первым и последним фиксированными битами в шиме S. Длина шимы определяет концентрацию информации в шиме. Считается, что шима с одной фиксированной позицией имеет нулевую длину. Например, шина $S = (***001*110)$ имеет порядок $o(S) = 6$ и длину $\delta(S) = 10 - 4 = 6$.

Порядок и длина шим используются для определение вероятности мутации и кроссинговера соответственно.

В связи с тем, что более приспособленные особи (хромосомы) описываются шимой с большей приспособленностью, смысл работы ГА заключается в поиске двоичной строки определенного вида из всего множества бинарных строк длины m. Тогда пространство поиска составляет 2^m строк, а его размерность равна m. Шима соответствует некоторой гиперплоскости в этом пространстве. Данное утверждение можно проиллюстрировать следующим образом. Пусть разрядность хромосомы равна 3, тогда всего можно закодировать $2^3 = 8$ строк. Представим куб в трехмерном пространстве. Обозначим вершины этого куба трехразрядными бинарными строками так, чтобы метки соседних вершин отличались ровно на один разряд, причем вершина с меткой "000" находилась бы в начале координат (см. рис. 18).

Если взять шиму вида "$**0$", то она опишет левую грань куба, а шима "$*10$" — верхнее ребро этой грани. Очевидно, что шима "$***$" соответствует всему пространству. Если взять двоичные строки длиной 4 разряда, то разбиение пространства шимами можно изобразить на примере четырехмерного куба с поименованными вершинами (см. рис. 19).

[1] Под приспособленностью шимы понимают среднюю приспособленность строк популяции, которые ей соответствуют.

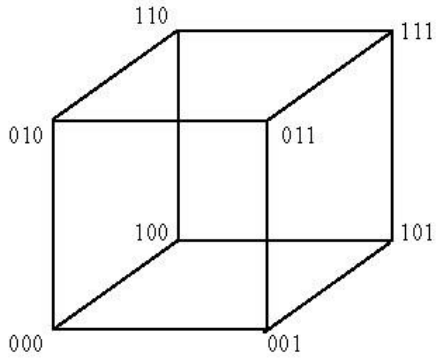

Рис. 18. Трехмерный куб

Здесь шиме "$*1**$" соответствует гиперплоскость, включающая задние грани внешнего и внутреннего куба, а шиме "$**10$" — гиперплоскость с верхними ребрами левых граней обоих кубов. Таким образом термины «гиперплоскость» и «шима» взаимозаменяемы.

Разбиение пространства поиска можно представить и по другому. Представим координатную плоскость, в которой по одной оси мы будем откладывать значения двоичных строк, а по другой — значение целевой функции (см. рис. 20).

Участки пространства, заштрихованные разным стилем, соответствуют разным шимам. Число K в правой части горизонтальной оси соответствует максимальному значению бинарной строки — "$111\ldots111$". Из рисунка видно, что шима "$0***\cdots*$" покрывает всю левую половину отрезка, шима "$**1*\cdots*$" — 4 участка шириной в одну восьмую часть, а шима "$0*10*\cdots*$" — левые половины участков, которые находятся на пересечении первых двух шим. Таким образом в этом случае происходит разбиение пространства.

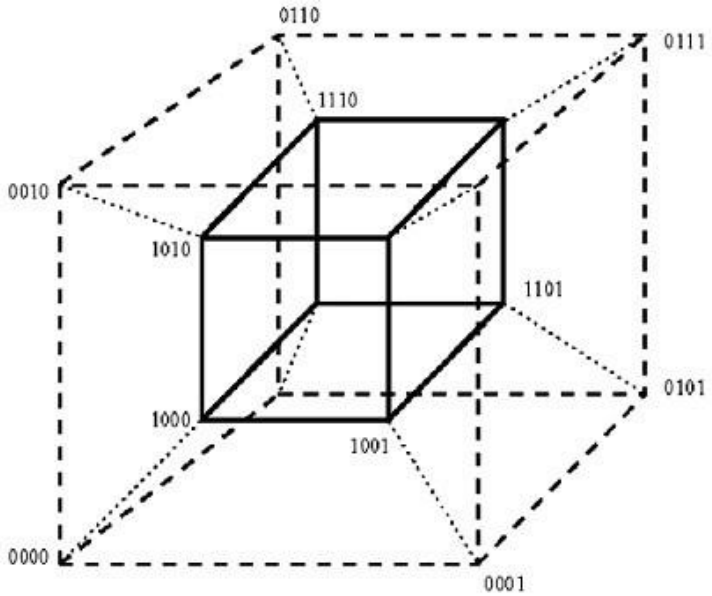

Рис. 19. Четырехмерный куб

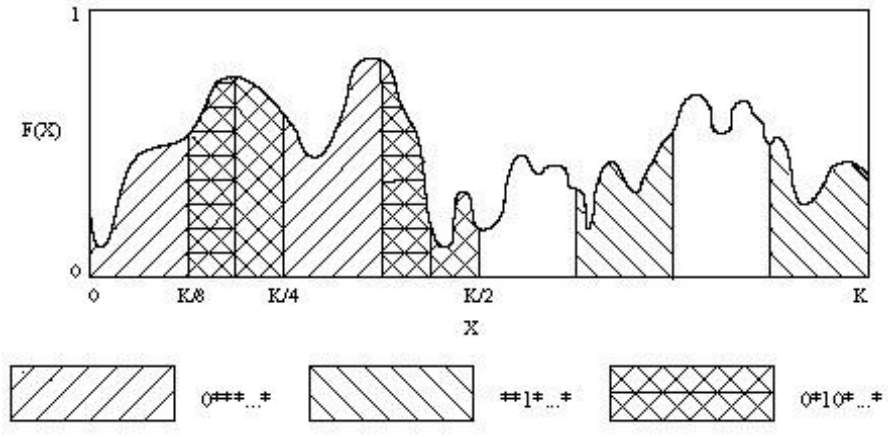

Рис. 20. Разбиение пространства

1.5. Строительные блоки

Строительные блоки — это шимы, обладающие:

- высокой пригодностью;
- низким порядком;
- короткой определенной длиной.

Пригодность шимы определяется как среднее пригодностей строкособей, которые ее содержат. После процедуры отбора остаются только строки с более высокой пригодностью. Следовательно, строки, которые являются примерами шим с высокой пригодностью, выбираются чаще. Кроссинговер реже разрушает шимы с более короткой определенной длиной, а мутация реже разрушает шимы с низким порядком. Поэтому, такие шимы имеют больше шансов переходить из поколения в поколение. Холланд показал, что в то время, как ГА явным образом обрабатывает n строк на каждом поколении, неявно обрабатываются порядка n^3 таких коротких шим низкого порядка и с высокой приспособленностью (полезных шим – useful schemata (см. [8])). Он называл это явление неявным параллелизмом (implicit parallelism). Для решения реальных задач, присутствие неявного параллелизма означает, что большая популяция имеет больше возможностей локализовать решение экспоненциально быстрее популяции с меньшим числом особей.

1.6. Теорема шим

Теорема шим (The schema theorem) показывает, каким образом простой ГА экспоненциально увеличивает число примеров полезных шим или строительных блоков, что приводит к нахождению решения исходной задачи.

Пусть $m(H,t)$ — число примеров шимы H в t-ом поколении. Вычислим ожидаемое число примеров H в следующем поколении или $m(H,t+1)$ в терминах $m(H,t)$. Простой ГА каждой строке при отборе ставит в соответствие вероятность ее «выживания» пропорционально ее приспособленности (например, как в методе рулетки). Ожидается, что шима H может быть выбрана $m(H,t)(f(H)/f_{\text{ср}})$ раз, где $f_{\text{ср}}$ — средняя пригодность популяции, а $f(H)$ — средняя пригодность тех строк в популяции, которые являются примерами H.

Вероятность того, что одноточечный кроссинговер разрушит шиму равна вероятности того, что точка разрыва попадет между определенными битами. Вероятность же того, что H «переживает» кроссинговер не меньше $1-p_{\text{c}}(\delta(H)/l-1)$, где p_{c} — вероятность кроссинговера. Эта вероятность — неравенство, поскольку шима сможет выжить, если в кроссинговере также участвовал пример подобной шимы.

Вероятность того, что H переживет точечную мутацию — $(1-p_{\text{m}})^{o(H)}$, где p_{m} — вероятность мутации. Это выражение можно аппроксимировать как $(1-o(H))$ для малых p_{m} и $o(H)$. Произведение ожидаемого число отборов и вероятностей выживания известно как *теорема шим*:

$$\langle m(H,t+1)\rangle \geqslant m(H,t)\,\frac{f(H,t)}{\bar{f}(t)}\left[1-p_{\text{c}}\frac{\delta(H)}{\ell-1}\right](1-p_{\text{m}})^{o(H)}.$$

Теорема шим показывает, что строительные блоки растут по экспоненте, в то время шимы с приспособленностью ниже средней распадаются с той же скоростью. Голдберг в своих исследованиях теоремы шим выдвигает гипотезу строительных блоков, которая состоит в том, что «строительные блоки объединяются, чтобы сформировать лучшие строки» (см. [6]). То есть рекомбинация и экспоненциальный рост строительных блоков ведет к формированию лучших строительных блоков.

В то время как теорема шим предсказывает рост примеров хороших шим, сама теорема весьма упрощенно описывает поведение ГА. Прежде всего, $f(H)$ и $f_{\text{ср}}$ не остаются постоянными от поколения к поколению. Вовторых, теорема шим объясняет потери шим, но не появление новых. Новые шимы часто создаются кроссинговером и мутацией. Кроме того, в результате эволюции члены популяции становятся все более и более похожими друг на друга так, что разрушенные шимы будут сразу же восстановлены. Наконец, доказательство теоремы шим построено на элементах теории вероятности и, следовательно, не учитывает разброс значений. Во многих задачах разброс значений пригодности шимы может быть достаточно велик, делая процесс формирования шим очень сложным.

Существенная разница пригодности шимы может привести к сходимости к неоптимальному решению. Несмотря на простоту, теорема шим описывает несколько важных аспектов поведения ГА. Мутации с большей вероятностью разрушают шимы высокого порядка, в то время как кроссинговер с большей вероятность разрушает шимы с большей определенной длиной. Когда происходит отбор, популяция сходится пропорционально отношению приспособленности лучшей особи, к средней приспособленности в популяции: это отношение — *мера давления отбора* («selection pressure»). Увеличение или p_{c}, или p_{m}, или уменьшение давления отбора ведет к увеличенному осуществлению выборки или исследованию пространства поиска, но не позволяет использовать все хорошие шимы, которыми располагает ГА. Уменьшение или p_{c}, или p_{m}, или увеличение давления выбора ведет к улучшению использования найденных шим, но тормозит исследование пространства в поисках новых хороших шим. Моделирование ГА предполагает сохранение равновесия ГА между тем и другим, что обычно известно как проблема «баланса исследования и использования».

Некоторые исследователи критикуют обычно быструю сходимость ГА, заявляя, что испытание огромных количеств перекрывающихся шим требует большей выборки и более медленной, более управляемой сходимости. Методология управления сходимость простого ГА до сих пор не выработана.

Недостатками теоремы шим является то, что она:

- примененяется только к каноническому ГА;
- не учитывает то обстоятельство, что кроссинговер и мутация могут не только разрушать шиму, но создавать ее из других шим. Поэтому в теореме шим присутствует знак неравенства;
- позволяет рассчитать долю шим в популяции только для следующего поколения, то есть при попытке подсчитать число строк, соответствующих данной шиме, через несколько поколений с использованием теоремы шим к успеху не приведет. Так получается, в частности, изза пропорциональной стратегии отбора.