

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**

(Н И У « Б е л Г У »)

Институт инженерных технологий и естественных наук

Кафедра общей математики

СТРУКТУРА ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ

Магистерская диссертация

Обучающегося по направлению 01.04.01 Математика,
очной формы обучения, группа 07001632

Моргуновой Инны Александровны

Научный руководитель:

доктор физ.-мат. наук,

профессор

Глушак А. В.

Рецензент

к.т.н., доцент

Маматов Е.М.

БЕЛГОРОД 2018

СОДЕРЖАНИЕ

Введение	3
Сущность эволюционных вычислений	5
Определение и понятия генетических алгоритмов	9
Классический генетический алгоритм	16
Иллюстрация выполнения классического генетического алгоритма	23
Кодирование в генетических алгоритмах	30
Генетические операторы	37
Основная теорема о генетических алгоритмах	60
Список использованных источников	76

ВВЕДЕНИЕ

Большинство задач науки и техники относятся к обширному классу проблем поиска оптимальных решений, т.е. к оптимизационным задачам.

Часть задач оптимизации относится к классу комбинаторных и они, в большинстве случаев, имеют не одно, а множество решений различного качества. Существует множество алгоритмов для решения таких задач. Ядром всех комбинаторных алгоритмов являются операции полного или сокращенного перебора в различных алгоритмах реализуется по-разному. Для поиска лучшего решения, как правило, осуществляются направленный, случайным и комбинированный переборы всевозможных значений параметров задачи. В этой связи разрабатывается большое число точных, переборных и эффективного механизма поиска решений на множестве альтернатив, что затрудняло получения качественных результатов за приемлемое время.

В конце 1960-х годов американский исследователь Джон Холланд в качестве принципов комбинаторного перебора вариантов решения оптимизационных задач предложил использовать методы и модели механизма развития органического мира по Земле. Поскольку основные законы эволюции живых организмов были исследованы генетикой, то и предложенный механизм получил название «генетические алгоритмы». Первый ввел в обиход термин «генетический алгоритм» Д.Багли в своей диссертации в 1967 г.

В мире сейчас успешно развиваются три основные научные школы по генетическим алгоритмам. К ним относятся американская, европейская и российская школы. В американской школе отметим таких ученых, как Д. Холланд, Д. Гольдберг, Д. Коза, Л. Чамберс и др. В европейскую школу входят такие ученые, как Р. Клинг, П. Банерджи, Э. Фалькенауер и др. В российской

отметим И. Букатову, Д. Батищева, И. Норенкова, авторов книги и многих др. Интерес к этой области исследований в мире каждый годом возрастает.

Предмет исследования – оптимизационные задачи.

Цель работы заключается в нахождении экстремума функций в случае, когда классические методы нахождения экстремума неприменимы.

СУЩНОСТЬ ЭВОЛЮЦИОННЫХ ВЫЧИСЛЕНИЙ

В последние 30 лет появился интерес к задачам, решение которых основано на принципах эволюции и наследования признаков. Системы подобного рода содержат популяцию потенциальных решений, имеют определенный отбора, использующий критерий пригодности индивидуумов (отдельных решений), применяют некоторые операторы рекомбинации. К таким системам относятся класс эволюционных вычислений (ЭВ).

Под последним понимается термин, используемый для описания алгоритмов поиска, оптимизации или обучения, основанных на некоторых формальных признаках естественного эволюционного отбора. Методы ЭВ часто применяются для описания процессов эволюции программ или функций (генетического программирования), конечных автоматов (эволюционное программирование) и систем, основанных на продукционных правилах (классификационные системы). Иногда эволюционные вычисления вместе с нечетной логикой используются для обучения нейронных сетей, что привело к новому термину «мягкие вычисления», объединившие генетические алгоритмы, нечетную логику и искусственную нейронную сеть.

В самом общем виде эволюционные вычисления можно описать так. Метод эволюционных вычислений представляет собой алгоритм, который содержит популяцию индивидуумов $P(t) = \{x_1^t, x_2^t, \dots, x_n^t\}$ на итерации t . Каждый индивидуум есть потенциальное решение рассматриваемой проблемы, и в любом из методов реализуется как некоторая сложная структура данных S . Каждое решение x_i^t оценивается для получения величины пригодности. После создается новая популяция $t + 1$ путем отбора наиболее пригодных индивидуумов (селекция). Некоторые члены новой популяции подвергаются

преобразованиям (рекомбинация) посредством генетических операторов для формирования новых решений. Среди операторов можно выделить оператор мутации, который создает новый индивидуум путем малых измерений исходного, и оператор скрещивания формирующий новые индивидуумы посредством комбинирования частей нескольких исходных. После ряда таких генераций программа сходится, т.е. находится лучший индивидуум, который представляет собой оптимальное решение.

Понятно, что многие эволюционные программы (синоним эволюционных вычислений) могут быть сформулированы для решения конкретной проблемы. Такие программы могут иметь различную структуру данных для построения отдельного индивидуума, генетические операторы для трансформации индивидуумов, методы для создания начальной популяции, параметры вычислений. Однако не смотря на различия, такие программы имеют общий принцип: популяция индивидуумов подвергается некоторым преобразованиям, и в течение эволюционного процесса индивидуумы сражаются за выживание.

Нужно так же сказать, что методы эволюционных вычислений не гарантируют обнаружения глобального оптимума за приемное время. Практический интерес к ним объясняется тем, что эти методы позволяют найти «хорошие» решения очень трудных задач за меньшее время, чем другими методами.

Необходимо уяснить разницу между эволюционными программами и генетическими алгоритмами: при первом подходе используется любая структура данных (хромосомное представление), подходящая для решения задачи, и любое множество генетических операторов; во второй ситуации применяются бинарные строки фиксированной длины (хромосомы) и два оператора (бинарные мутации и скрещивание).

Естественно, что близкое к эволюционным процессам представление потенциального решения для данной проблемы вместе с совокупностью

подходящих генетических операторов может быть полезным для решения многих задач, и такой подход является весьма перспективным направлением. Многие исследователи в этой области применяют модифицированные реализации ГА или путем использования нестроковой хромосомы при представлении задачи, или посредством разработки специальных генетических операторов для возможности решения конкретной проблемы. Различные нестандартные подходы были предложены для частных задач, так как классические генетические алгоритмы было трудно применить для решения подобных задач.

В связи с вышеуказанными соображениями о модификации генетического алгоритма, естественно, появляется вопрос: являются ли эволюционные стратегии генетическими алгоритмами? Во избежание таких вопросов, связанных прежде всего, с классификацией эволюционных вычислений, будем звать эти методы эволюционными программами (ЭП) и рассмотрим различие между генетическими алгоритмами и эволюционными программами. Несмотря на то что генетические алгоритмы имеют достаточно серьезную теоретическую базу, они не могут обеспечить успешное применение во многих областях. Очевидно, что главный фактор неудач генетического алгоритма является тем же фактором успеха в решении задач - это область независимости. Одним из следствий четкости генетических алгоритмов смысле области независимости является их неспособность к учету нетривиальных ограничений. В большинстве работ по генетическим алгоритмам хромосомы представляют собой битовые строки (состоящие из 1 и 0), поэтому реализация ограничений в такой ситуации является достаточно сложной задачей.

В эволюционных программах проблема ограничений имеет другой характер. Здесь речь идет, скорее, о выборе «лучшего» хромосомного представления искомого решения вместе со значимыми генетическими операторами для удовлетворения всех ограничений, накладываемых данной проблемой. Любой генетический оператор должен проходить через некоторую

характеристическую структуру от «родителей» к «потомкам», поэтому структура представления играет важную роль в определении генетических операторов. Кроме того, различные структуры представлений имеют различные характеристически пригодности для ограничений, что еще более усложняет задачу. Эти два компонента - представление и операторы - влияют друг на друга. Очевидно, что любая проблемы требует тщательного анализа, который позволит выбрать подходящую схему представления и генетические операторы.

Основная концептуальная разница между генетическими алгоритмами и эволюционными программами показана такова:

(Проблема \rightarrow Модифицированная проблема \leftrightarrow
Генетический алгоритм(1.1)) и (Генетический алгоритм \rightarrow
Эволюционная программа \leftrightarrow Проблема(1.2)).

Генетические алгоритмы, которые действуют на бинарных строках, требуют модификации исследуемой проблемы в подходящую для таких алгоритмов форму. Указанная модификация обычно включает в себя отображение между потенциальными решениями и бинарным представлением, изменение алгоритмов и тому подобное.

С другой стороны, эволюционные программы оставляют проблему неизменной, модифицируя хромосомное представление потенциального решения и применяя подходящие генетические операторы.

Подводя итог сопоставления генетического алгоритма и эволюционных программ, скажем так, что для решения нетривиальной проблемы с применением эволюционных программ необходимо или трансформировать задачу в форму, подходящую для генетического алгоритма (1.1), или изменить генетический алгоритм для решения данной проблемы (1.2).

ОПРЕДЕЛЕНИЕ И ПОНЯТИЯ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ

Генетические алгоритмы - это новая область исследований, которая появилась в результате работ Д. Холланда и его коллег. Генетические алгоритмы, описанные Д. Холландом, заимствуют в своей терминологии многое из естественной генетики. Далее будут приведены технические толкования терминов из биологии и генетики, которые используются в теории и практике генетических алгоритмов. Впервые генетические алгоритмы были применены к таким научным проблемам, как распознавание образов и оптимизация. Генетический алгоритм представляет собой адаптивный поисковой метод, основанный на селекции лучших элементов в популяции, подобно эволюционной теории Ч. Дарвина.

Основой для возникновения генетических алгоритмов послужили модель биологической эволюции и методы случайного поиска. Л. Растрингин отмечал, что случайный поиск возник как реализация простейшей модели эволюции, когда случайные мутации моделировались случайными шагами оптимального решения, а отбор - «устранением» неудачных вариантов.

Эволюционный поиск с точки зрения преобразования информации - это последовательное преобразование одного конечного нечеткого множества промежуточных решений в другое. Само преобразование можно назвать алгоритмом поиска, или генетическим алгоритмом. Генетические алгоритмы - это не просто случайный поиск. Они эффективно используют информацию, накопленную в процессе эволюции.

Цель генетических алгоритмов состоит в том, чтобы:

- абстрактно и формально объяснять адаптацию процессов в естественной системе и интеллектуальной исследовательской системе:

- моделировать естественные эволюционные процессы для эффективного решения оптимизационных задач науки и техники.

В настоящее время используется новая парадигма решений оптимизационных задач на основе генетических алгоритмов и их различных модификаций. Генетические алгоритмы осуществляют поиски баланса между эффективностью и качеством решений за счет «выживания сильнейших альтернативных решений» в неопределенных и нечетких условиях.

Генетические алгоритмы отличаются от других оптимизационных и поисковых процедур следующим:

- работают в основном не с параметрами задачи, а с закодированным множеством параметров;
- осуществляют поиск не путем улучшения одного решения, а путем использования сразу нескольких альтернатив на заданном множестве решений;
- используют целевую функцию, а не ее различные приращения для оценки качества принятия решений;
- применяют не детерминированные, а вероятностные правила анализа оптимизационных задач.

Для работы генетических алгоритмов выбирают множество натуральных параметров оптимизационной проблемы и кодируют их в последовательность конечной длины в некотором алфавите. Они работают до тех пор, пока не будет выполнено заданное число генераций или на некоторой генерации будет получено решение определенного качества, или когда найден локальный оптимум, т.е. возникла преждевременная сходимость и алгоритм не может найти выход из этого состояния. В отличие от других методов оптимизации эти алгоритмы, как правило, анализируют различные области пространства решений

одновременно и поэтому они более приспособлены к нахождению новых областей с лучшими значениями целевой функции.

Приведем некоторые понятия и определения из теории генетических алгоритмов. Все генетические алгоритмы работают на основе начальной информации, в качестве которой выступает популяция альтернативных решений P . Популяция $P^t = \{P_1, P_2, \dots, P_i, \dots, P_{Np}\}$ есть множество элементов $P_i, t = 1, 2, \dots$ - номер генерации генетического алгоритма, Np - размер популяции. Каждый элемент этой популяции P_i , как правило, представляет собой одну или несколько хромосом или особей, или индивидуальностей. Хромосомы состоят из генов $P_i = \{g_1, g_2, \dots, g_v\}$, и позиции генов в хромосоме называются лоци или локус для одной позиции в хромосоме, аллель - функциональное значение гена.

Гены могут иметь числовые или функциональные значения. Обычно эти числовые значения берутся из некоторого алфавита. Генетический материал элементов обычно кодируется на основе двоичного алфавита $\{0,1\}$, хотя можно использовать буквенные, а так же десятичные и другие алфавиты. Примером закодированной хромосомы длины девять на основе двоичного алфавита может служить хромосома $P_i = (001001101)$.

Элементы в генетических алгоритмах часто называют родителями. Родители выбирают из популяции на основе заданных правил, а затем смешиваются для производства «детей». Дети и родители в результате генерации, т.е. одного цикла эволюции, создают новую популяцию. Генерация, то есть процесс реализации одной итерации алгоритма, называется поколением.

По аналогии с процессами, происходящими в живой природе и описанными в этом разделе, в технике считают, что эволюция популяции - это чередование поколений, в которых хромосомы изменяют свои значения так, чтобы каждое новое поколение наилучшим способом

приспосабливалось к внешней среде. Тогда общая генетическая упаковка называется генотипом, а организм формируется посредством связи генетической упаковки с окружающей средой и называется фенотипом.

Каждый элемент в популяции имеет определенный уровень качества, который характеризуется значением целевой функции. Эта функция используется в генетических алгоритмах для сравнения альтернативных решений между собой и выбора лучших.

Следовательно, основная задача генетических алгоритмов состоит в оптимизации целевой функции. Другими словами, генетические алгоритмы анализируют популяцию хромосом, представляющих комбинацию элементов из некоторого множества, и оптимизируют целевую функцию, оценивая каждую хромосому. Генетические алгоритмы анализируют и преобразовывают популяции хромосом на основе механизма натуральной эволюции. Каждая популяция обладает наследственной изменчивостью. Это означает

Наличие возможностей случайных отклонений от наиболее вероятного среднего значения целевой функции.

Отклонения описываются нормальным законом распределения случайных величин. При этом наследственные признаки закрепляются, если они имеют приспособительный характер, т.е. обеспечивают популяции лучшие условия существования и размножения.

Так же как процесс эволюции начинается с начальной популяции, так алгоритм начинает свою работу с создания начального множества конкурирующих между собой решений оптимизационной задачи. Зачем эти «родительские» решения создают «потомков» путем случайных и направленных изменений. После этого оценивается эффективность этих решений, и они подвергаются селекции. Аналогично естественным системам здесь действует принцип «выживания сильнейших», наименее приспособленные решения «погибают», а затем процесс повторяется ВНОВЬ И ВНОВЬ.

Традиционные оптимизационные алгоритмы для нахождения лучшего решения используют большое количество допущений при оценке целевой функции. Эволюционный подход не требует таких допущений, что расширяет класс задач, которое можно решать с помощью генетических алгоритмов. Согласно существующим исследованиям можно сказать, что генетические алгоритмы позволяют решать те проблемы, решение которых традиционными алгоритмами затруднительно.

Генетический алгоритм дает преимущества при решении практических задач. Одно из них - это адаптация к изменяющейся окружающей среде. В реальной жизни проблема, которая была поставлена в процессе своего решения. При использовании традиционных методов все вычисления приходится начинать заново, что приводит к большим затратам машинного времени. При эволюционном подходе популяцию можно анализировать, дополнять и видоизменять применительно к изменяющимся условиям, для этого не требуется полный перебор. Другое преимущество генетических алгоритмов для решения задач состоит в способности быстрой генерации достаточно хороших решений.

При решении практических задач с использованием генетических алгоритмов обычно выполняют четыре предварительных этапа:

- выбор способа представления решения;
- разработка операторов случайных изменений;
- определение способов «выживания» решений;
- создание начальной популяции альтернативных решений;

Рассмотрим некоторые особенности выполнения этих этапов.

На первом этапе для представления решения в формальном виде требуется такая структура, которая позволит кодировать любое возможное решение и производить его оценку. Математически доказано, что не

существует идеальной структуры представления, так что для создания хорошей структуры требуется анализ, перебор и эвристические подходы. Возможный вариант представления должен позволять проведение различных перестановок в альтернативных решениях. Для оценки решений необходимо определить способ вычисления целевой функции.

На втором этапе достаточно сложным является выбор случайного оператора для генерации потомков. Существует огромное количество таких операторов. Как отмечалось в начале этой главы, существуют два основных типа размножения: половое и бесполое. При половом размножении два родителя обмениваются генетическим материалом, который используется при создании потомка. Бесполое размножение - это фактически при передаче информации от родителя к потомку. Модели этих типов размножения играют роль в генетических алгоритмах. В общем случае можно применить модели размножения, которые не существуют в природе. Например, использовать материал от трех или более родителей, проводить голосование при выборе родителей т.п. Фактически нет пределов в использовании различных моделей, и поэтому при решении технических задач нет смысла слепо копировать законы природы и ограничиваться только ими.

Успех генетических алгоритмов во многом зависит от того, как взаимодействуют между собой схема представления, методы случайных изменений и способ определения целевой функции. Поэтому для определенного класса задач целесообразно использовать направленные методы.

В качестве примера рассмотрим два способа представления перестановок при решении оптимизационных задач. В первом случае будем использовать одного родителя и получать одного потомка. Во втором случае используем двух родителей, случайно выберем точку перестановки и для образования потомка возьмем первый сегмент первого родителя, а второй сегмент - у второго. Первый метод похож на бесполое размножение, а второй - на половое

размножение. Стоит отмечать, что если первый метод всегда генерирует реальное решение, то второй может генерировать недопустимые решения. При этом требуется «восстанавливать» допустимые решения перед их оценкой.

На третьем из рассматриваемых этапов задают правила выживания решений для создания потомства. Существует множество способов проведения селекции альтернативных решений. Простейшее правило - это «выживание сильнейших», т.е. остаются только лучшие решения с точки зрения заданной целевой функции, а все остальные устраняются. Такое правило часто оказывается малоэффективным при решении сложных технических проблем. Иногда лучшие решения могут происходить от худших, а не только от самых лучших. Тем не менее, логично использовать принцип:

- основное исходное положение какой-либо теории;
- внутренняя убежденность в чем-либо;
- основная особенность работы механизма, устройства и т.п.

На последнем предварительном этапе создается начальная популяция. При неполноте исходных данных о проблеме решения могут случайным образом выбираться из всего множества альтернатив. Это реализуется генерацией случайных внутри хромосомных перестановок, каждая из которых представляет рекомендуется использовать знания о решаемой задаче. Например, эти значения могут быть получены из опыта разработчика, существующих стандартов и библиотек алгоритмов решения задач данного класса.

Эффективность генетического алгоритма - степень реализации запланированных действий алгоритма и достижение требуемых значений целевой функции. Эффективность во многом определяется структурой и составом начальной популяции. При создании начального множества решений происходит формирование популяции на основе четырех основных принципов:

- «одеяло» - генерируется полная популяция, включающая все возможные решения в некоторой заданной области;
- «дробовик» - подразумевает случайный выбор альтернатив из всей области решений данной задачи.
- «фокусировка» - реализует случайный выбор допустимых альтернатив из заданной области решений данной задачи.
- «комбинирование» - состоит в различных совместных реализациях первых трех принципов.

Отметим, что популяция обязательно является конечным множеством.

КЛАССИЧЕСКИЙ ГЕНЕТИЧЕСКИЙ АЛГОРИТМ

Основной (классический) генетический алгоритм состоит из следующих шагов:

- 1) Инициализация, или выбор исходной популяции хромосом;
- 2) Оценка приспособленности хромосом и популяции;
- 3) Проверка условия остановки алгоритма;
- 4) Селекция хромосом;
- 5) Применение генетических операторов;
- 6) Формирование новой популяции;
- 7) Выбор «наилучшей» хромосомы.

Блок-схема основного генетического алгоритма изображена на рис. 1.1. Рассмотрим конкретные этапы этого алгоритма более подробно с использованием дополнительных подробностей, представленных на рис. 1.1.

Инициализация, т.е. формирование исходной популяции, заключается в случайном выборе заданного количества хромосом, представляемых двоичными последовательностями фиксированной длины.

Оценивание приспособленности хромосом в популяции состоит в расчете функции приспособленности для каждой хромосомы этой популяции. Чем больше значение этой функции, тем выше «качество» хромосомы. Форма функции приспособленности зависит от характера решаемой задачи. Предполагается, что функция приспособленности всегда принимает неотрицательные значения и, кроме того, что решения оптимизационной задачи требуется максимизировать эту функцию. Если исходная форма функции приспособленности не удовлетворяет этим условиям, то выполняется соответствующее преобразование.

Проверка условия остановки алгоритма. Определение условия остановки генетического алгоритма зависит от его конкретного применения. В оптимизационных задачах, если известно максимальное значение функции приспособленности, то остановка алгоритма может произойти после достижения ожидаемого оптимального значения, возможно - с заданной точностью. Остановка алгоритма также может произойти в случае, когда его выполнение не приводит к улучшению уже достигнутого значения. Алгоритм может быть остановлен по истечении определенного времени выполнения либо после выполнения заданного количества итераций. Если условие остановки выполнено, то производится переход к завершающему этапу выбора «наилучшей» хромосомы. В противном случае на следующем шаге выполняется селекция.

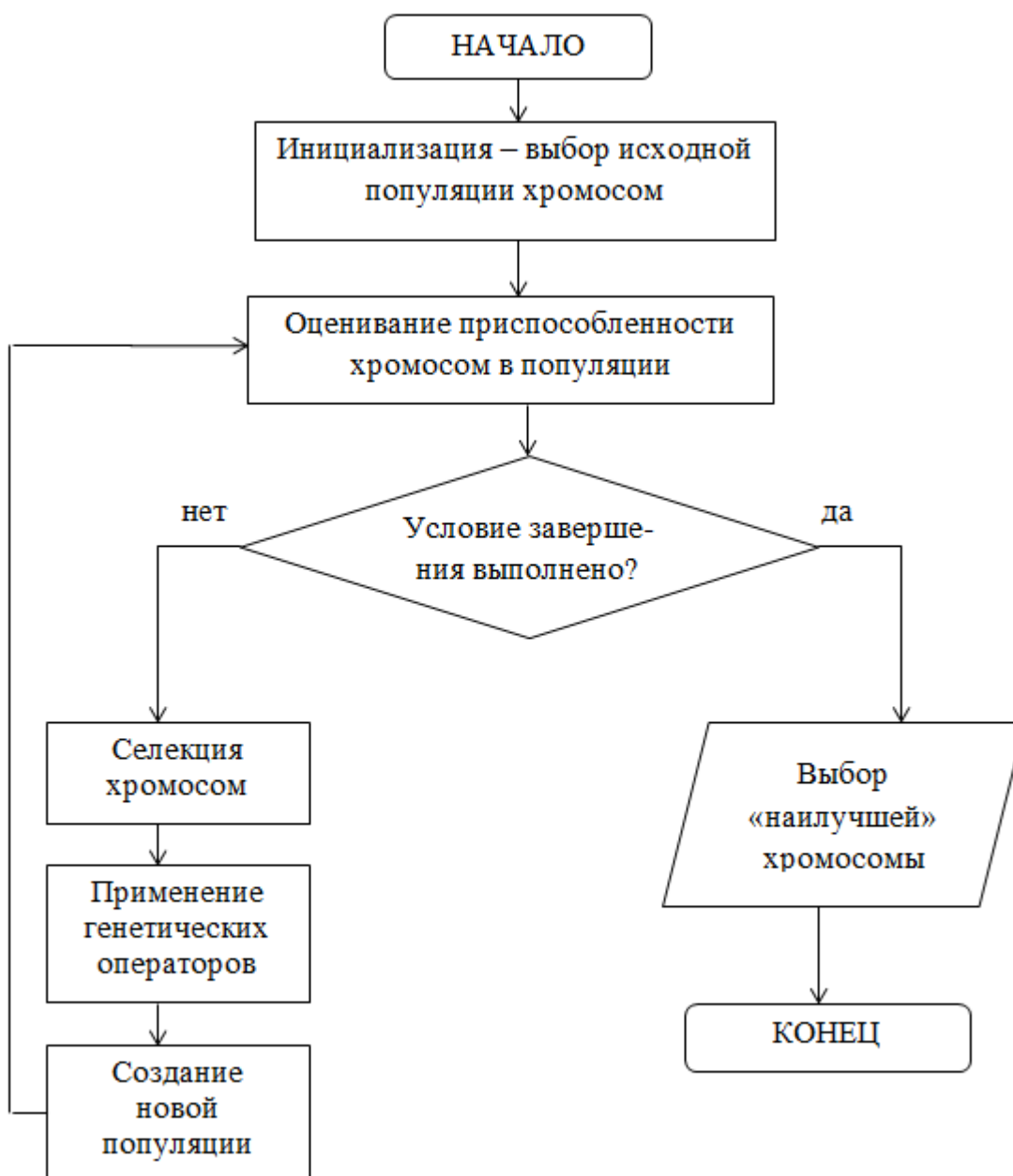


Рис. 1.1. Блок-схема генетического алгоритма

Селекция хромосом заключается в выборе тех хромосом, которые будут участвовать в создании потомков для следующей популяции, т.е. для очередного поколения. Такой выбор производится согласно принципу естественного отбора, по которому наибольшие шансы на участие в создании новых особей имеют хромосомы с наибольшими значениями функции приспособленности. Существуют различные методы селекции. Наиболее популярным считается так называемый метод рулетки, который свое название

получил по аналогии с известной азартной игрой. Каждой хромосоме может быть сопоставлен сектор колеса рулетки, величина которого устанавливается пропорциональной значению функции приспособленности данной хромосомы. Поэтому чем больше значение функции приспособленности, тем больше сектор на колесе рулетки. Все колесо рулетки соответствует сумме значений функции приспособленности всех хромосом рассматриваемой популяции. Каждой хромосоме, обозначаемой ch_i для $i = 1, 2, \dots, N$ (N обозначает численность популяции) соответствует сектор колеса $v(ch_i)$, выраженный в процентах согласно формуле

$$v(ch_i) = p_s(ch_i)100\%, \quad (1.1)$$

$$p_s(ch_i) = \frac{F(ch_i)}{\sum_{i=1}^N F(ch_i)}, \quad (1.2)$$

причем $F(ch_i)$ - значение функции приспособленности хромосомы ch_i , а $p_s(ch_i)$ - вероятность селекции хромосомы ch_i . Селекция хромосомы может быть представлена как результат поворота колеса рулетки, поскольку «выигравшая» (т.е. выбранная) хромосома относится к выпавшему сектору этого колеса. Очевидно, что чем больше сектор, тем больше вероятность «победы» соответствующей хромосомы. Поэтому вероятность выбора данной хромосомы оказывается пропорциональной значению ее функции приспособленности. Если всю окружность колеса рулетки представить в виде цифрового интервала $[0,100]$, то выбор хромосомы можно отождествить с выбором числа из интервала $[a, b]$, где a и b обозначают соответствующего этому сектору колеса; очевидно, что $0 \leq a < b \leq 100$. В этом случае выбор с помощью колеса рулетки сводится к выбору числа из интервала $[0,100]$, которое соответствует конкретной точке на окружности колеса.

В результате процесса селекции создается родительская популяция, также называется родительским пулом с численностью N , равной численности текущей популяции.

Применение генетических операторов к хромосомам, отобранным с помощью селекции, приводит к формированию новой популяции потомков от созданной на предыдущем шаге родительской популяции.

В классическом генетическом алгоритме применяются два основных генетических оператора: оператор скрещивания и оператор мутации. Однако следует отметить, что оператор мутации играет явно второстепенную роль по сравнению с оператором скрещивания. Это означает, что скрещивание в классическом генетическом алгоритме производится практически всегда, тогда как мутация - достаточно редко. Вероятность скрещивания, как правило, достаточно велика, тогда как вероятность мутации устанавливается весьма малой. Это следует из аналогии с миром живых организмов, где мутации происходят чрезвычайно редко.

В генетическом алгоритме мутация хромосом может выполняться на популяции родителей перед скрещиванием либо на популяции потомков, образованных в результате скрещивания.

Оператор скрещивания. На первом этапе скрещивания выбирают пары хромосом из родительской популяции. Это временная популяция, состоящая из хромосом, отобранных в результате селекции и предназначенных для дальнейших преобразований операторами скрещивания и мутации с целью формирования новой популяции потомков. На данном этапе хромосомы из родительской популяции объединяют в пары. Это производится случайным способом в соответствии с вероятностью скрещивания p_c . Далее для каждой пары отобранных таким образом родителей разыгрывается позиция гена в хромосоме, определяющая так называемую точку скрещивания. Если хромосома каждого из родителей состоит из L генов, то очевидно, что точка скрещивания l_k представляет собой натуральное число, меньшее L . Поэтому фиксация точки скрещивания сводится к случайному выбору числа из

интервала $[1, L - 1]$. В результате скрещивания пары родительских хромосом получается следующая пара потомков:

- 1) Потомок, хромосома которого на позициях от 1 до l_k состоит из генов первого родителя, а на позициях от $l_k + 1$ до L - генов второго родителя.
- 2) Потомок, хромосома которого на позициях от 1 до l_k состоит из генов второго родителя, а на позициях от $l_k + 1$ до L - генов первого родителя.

Оператор мутации с вероятностью p_m изменяет значение гена в хромосоме на противоположное (т.е. с 0 на 1 или обратно). Например, если в хромосоме $[100110101010]$ мутации подвергается ген на позиции 7, то его значение, равное 1, изменяется на 0. Что приводит к образованию хромосомы $[100110001010]$. Как уже упоминалось выше, вероятность мутации обычно очень мала, и именно от нее зависит, будет данный ген мутировать или нет. Вероятность p_m мутации может эмулироваться, например, случайным выбором числа из интервала $[0,1]$ для каждого гена и отбором для выполнения этой операции тех генов, для которых разыгранное число оказывается меньшим или равным значению p_m .

Формирование новой популяции. Хромосомы, полученные в результате применения генетических операторов к хромосомам временной родительской популяции, включаются в состав новой популяции. Она становится так называемой текущей популяцией для данной итерации генетического алгоритма. На каждой очередной итерации рассчитываются значения функции приспособленности для всех хромосом этой популяции, после чего проверяется условие остановки алгоритма и либо фиксируется результат в виде хромосомы с наибольшим значением функции приспособленности, либо осуществляется переход к следующему шагу генетического алгоритма, т.е. к селекции. В классическом генетическом алгоритме вся предшествующая популяция хромосом замещается новой популяцией потомков, имеющей ту же численность.

Выбор наилучшей хромосомы. Если условие остановки алгоритма выполнено, то следует вывести результат работы, т.е. представить искомое решение задачи. Лучшим решением считается хромосома с наибольшим значением функции приспособленности.

В завершение следует признать, что генетические алгоритмы унаследовали свойства естественного эволюционного процесса, состоящие в генетических изменениях популяций организмов с течением времени.

Главный фактор эволюции - это естественный отбор, который приводит к тому, что среди генетически различающихся особей одной и той же популяции выживают и оставляют потомство только наиболее приспособленные к окружающей среде. В генетических алгоритмах также выделяется этап селекции, на котором из текущей популяции выбираются и включаются в родительскую популяцию особи, имеющие наибольшие значения функции приспособленности. На следующем этапе, который иногда называется эволюцией, применяются генетические операторы скрещивания и мутации выполняющие рекомбинацию генов в хромосомах.

Операция скрещивания заключается в обмене фрагментами цепочек между двумя родительскими хромосомами. Пары родителей для скрещивания выбираются из родительского пула случайным образом так, чтобы вероятность выбора конкретной хромосомы для скрещивания была равна вероятности p_c . Например, если в качестве родителей случайным образом выбираются две хромосомы из родительской популяции численностью N , то $p_c = 2/N$. Аналогично, если из родительской популяции численностью N выбирается $2z$ хромосом ($z \leq N/2$), которые образуют z пар родителей, то $p_c = 2z/N$. Обратим внимание, что если все хромосомы текущей популяции объединены в пары до скрещивания, то $p_c = 1$. После операции скрещивания родители в родительской популяции замещаются их потомками.

Операция мутации изменяет значения генов в хромосомах с заданной вероятностью p_m способом, представленным при описании соответствующего оператора. Это приводит к инвертированию значений отобранных генов с 0 на 1 и обратно. Значение p_m , как правило, очень мало, поэтому мутации подвергается лишь небольшое количество генов. Скрещивание - это ключевой оператор генетических алгоритмов, определяющий их возможности и эффективность. Мутация играет более ограниченную роль. Она вводит в популяцию некоторое разнообразие и предупреждает потери, которые могли бы произойти вследствие исключения какого-нибудь значимого гена в результате скрещивания.

Основной (классический) генетический алгоритм известен в литературе в качестве инструмента, в котором выделяются три вида операций: репродукции, скрещивание и мутации. Термины селекция и репродукция в данном контексте используются в качестве синонимов. При этом репродукция в данном случае связывается скорее с созданием копий хромосом родительского пула, тогда как более распространенное содержание этого понятия обозначает процесс формирования новых особей, происходящих от конкретных родителей. Если мы принимаем такое толкование, то операторы скрещивания и мутации могут считаться операторами репродукции, а селекция - отбором особей (хромосом) для репродукции.

ИЛЛЮСТРАЦИЯ ВЫПОЛНЕНИЯ КЛАССИЧЕСКОГО ГЕНЕТИЧЕСКОГО АЛГОРИТМА

Рассмотрим выполнение описанного разделе классического алгоритма на как можно более простом примере. Последим последовательность выполнения его этапов, соответствующих блок-схеме рис.1.1. упрощенный

Пример 1. Рассмотрим сильно упрощенный и довольно искусственный пример, состоящий в нахождении хромосомы с максимальным количеством единиц. Допустим, что хромосомы состоят из 12 генов, а популяция насчитывает 8 хромосом. Понятно, что наилучшей будет хромосома состоящей из 12 единиц. Посмотрим, как протекает процесс решения этой задачи с помощью генетического алгоритма.

Инициализация, или выбор исходной популяции хромосом. Необходимо случайным образом сгенерировать 8 двоичных последовательностей длиной 12 битов. Это можно достигнуть, например, подбрасываем монеты (96 раз, при выпадении «орла» приписывается значение 1, а в случае «решки» - 0). Таким образом можно сформировать исходную популяцию

$$ch_1 = [111001100101]$$

$$ch_2 = [001100111010]$$

$$ch_3 = [011101110011]$$

$$ch_4 = [001000101000]$$

$$ch_5 = [010001100100]$$

$$ch_6 = [010011000101]$$

$$ch_7 = [101011011011]$$

$$ch_8 = [000010111100]$$

Оценка приспособленности хромосом в популяции. В рассматриваемом упрощенном примере решается задача нахождения такой хромосомы, которая содержит наибольшее количество единиц. Поэтому функция

приспособленности определяет количество единиц в хромосоме. Обозначим функцию приспособленности символом F . Тогда ее значения для каждой хромосомы из исходной популяции будут такие:

$$F(ch_1) = 7$$

$$F(ch_2) = 6$$

$$F(ch_3) = 8$$

$$F(ch_4) = 3$$

$$F(ch_5) = 4$$

$$F(ch_6) = 5$$

$$F(ch_7) = 8$$

$$F(ch_8) = 5$$

Хромосомы ch_3 и ch_7 характеризуются наибольшими значениями функции принадлежности. В этой популяции они считаются наилучшими кандидатами на решение задачи. Если в соответствии с блок-схемой генетического алгоритма (рис.1.1) условие остановки алгоритма не выполняется, то на следующем шаге производится селекция хромосом из текущей популяции.

Селекция хромосом. Селекция производится методом рулетки. На основании формул (1) и (2) для каждой из 8 хромосом текущей популяции получаем сектора колеса рулетки, выраженные в процентах (рис.1.2)

$$v(ch_1) = 15,22$$

$$v(ch_2) = 13,04$$

$$v(ch_3) = 17,39$$

$$v(ch_4) = 6,52$$

$$v(ch_5) = 8,70$$

$$v(ch_6) = 10,87$$

$$v(ch_7) = 17,39$$

$$v(ch_8) = 10,87$$

Розыгрыш с помощью колеса рулетки сводится к случайному выбору чисел из интервала $[0,100]$, указывающего на соответствующий сектор на колесе, т.е. на конкретную хромосому. Допустим, что разыграны следующие 8 чисел:

79 44 9 74 44 86 48 23

Это означает выбор хромосом

$ch_7 ch_3 ch_1 ch_7 ch_3 ch_7 ch_4 ch_2$

Как видно, хромосома ch_7 была выбрана трижды, а хромосома ch_3 - дважды. Заметим, что именно эти хромосомы имеют значение функции приспособленности. Однако выбрана и хромосома ch_4 с наименьшим значением функции приспособленности. Все выбранные таким образом хромосомы включаются в так называемый родительский пул.

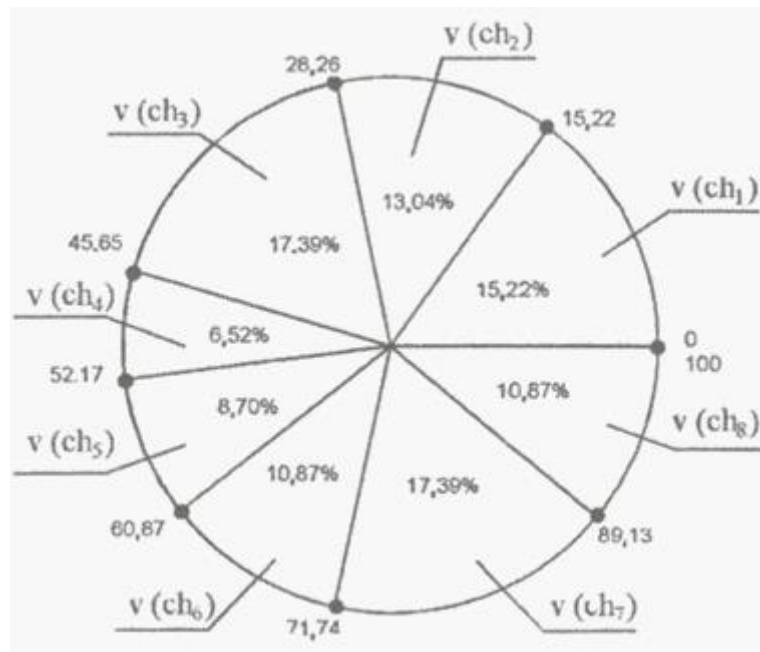


Рис.1.2. Колесо рулетки для селекции в примере 1

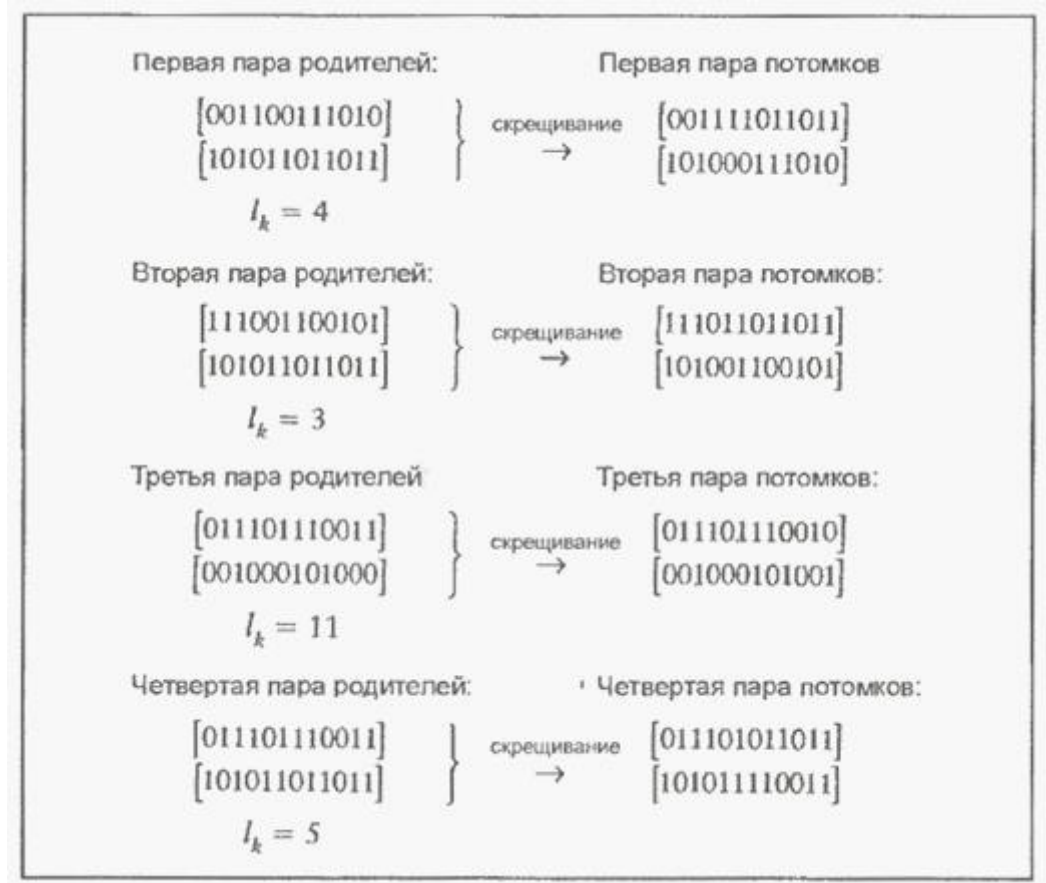
Применение генетических операторов. Допустим, что ни одна из отобранных в процессе селекции хромосом не подвергается мутации, и все они составляют популяцию хромосом, предназначенных для скрещивания. Это означает, что вероятность скрещивания $p_c = 1$, а вероятность $p_m = 0$. Допустим, что из этих хромосом случайным образом сформированы пары родителей

$$ch_2 \text{ и } ch_7 \quad ch_1 \text{ и } ch_7 \quad ch_3 \text{ и } ch_4 \quad ch_3 \text{ и } ch_7$$

Для первой пары случайным образом выбрана точка скрещивания $l_k = 4$, для второй $l_k = 3$, для третьей $l_k = 11$, для четвертой $l_k = 5$. При этом процесс скрещивания протекает так как показано на рис.3. В результате потомков.

Если бы при случайном подборе пар хромосом для скрещивания были объединены, например, ch_3 с ch_3 и ch_4 с ch_7 вместо ch_3 с ch_4 и ch_3 с ch_7 , а другие пары остались без изменения, то скрещивание ch_3 с ch_3 дало бы две такие же хромосомы независимо от разыгранной точки скрещивания. Это означало бы получение двух потомков, идентичных своим родителям.

Заметим, что такая ситуация наиболее вероятна для хромосом с наибольшим значением функции приспособленности, т.е. именно такие хромосомы получают наибольшие шансы на переход в новую популяцию.



Формирование новой популяции. После выполнения операции скрещивания мы получаем следующую популяцию потомков:

$$ch_1 = [001111011011]$$

$$ch_2 = [101000111010]$$

$$ch_3 = [111011011011]$$

$$ch_4 = [101001100101]$$

$$ch_5 = [011101110010]$$

$$ch_6 = [001000101001]$$

$$ch_7 = [011101011011]$$

$$ch_8 = [101011110011]$$

Для отличия от хромосом предыдущей популяции обозначения вновь сформированных хромосом начинаются с заглавной буквы *C*.

Согласно блок-схеме генетического алгоритма производится возврат ко второму этапу, т.е. к оценке приспособленности хромосом вновь сформированной популяции, которая становится текущей. Значения функций приспособленности хромосом этой популяции составляют

$$F(ch_1) = 8$$

$$F(ch_2) = 6$$

$$F(ch_3) = 6$$

$$F(ch_4) = 6$$

$$F(ch_5) = 7$$

$$F(ch_6) = 4$$

$$F(ch_7) = 8$$

$$F(ch_8) = 8$$

Заметно, что популяция потомков характеризуется гораздо более высоким средним значением функции приспособленности, чем популяция родителей. Обратим внимание, что в результате скрещивания получена хромосома Ch_3 с наибольшим значением функции приспособленности, которым не обладала ни одна хромосома из родительской популяции. Однако могло произойти и обратное, поскольку после скрещивания на первой итерации хромосома, которая в родительской популяции характеризовалась наибольшим значением функции приспособленности,

могла просто «потеряться». Помимо этого средняя приспособленность новой популяции все равно оказалась бы выше предыдущей, а хромосомы с большими значениями функции приспособленности имели бы шансы появиться в следующих поколениях.

КОДИРОВАНИЕ В ГЕНЕТИЧЕСКИХ АЛГОРИТМАХ

Подчеркнем еще раз между фенотипом и генотипом. Из биологии известно, что любой организм может быть представлен своим фенотипом, который фактически определяет, чем является объект в реальном мире (внешняя среда). Генотип содержит всю информацию об объекте на уровне хромосомного набора. Для решения задачи необходимо представить каждый признак объекта в форме, пригодной для использования генетического алгоритма. Кодирование решения задачи в хромосоме является ключевой проблемой применения генетического алгоритма. Проблема исследована с различных сторон, например, по характеру отображения пространства генотипов в пространство фенотипов, когда отдельные особи декодируются с помощью операторов.

В классической работе кодирование выполняется с использованием бинарных строк. Такое кодирование для задач оптимизации имеет несколько недостатков вследствие существования хеммингово расстояние, в то время в фенотипическом пространстве. Например, пара 0111111111 и 1000000000 принадлежит соседним точкам в фенотипическом пространстве (точки с минимальным евклидовым расстоянием), но имеют максимальное хеммингово расстояние в генотипическом пространстве. Для

преодоления хеммингова сдвига все биты должны изменяться одновременно, но вероятность такого события очень мала.

Для многих задач, встречающихся в современных проблемах, достаточно трудно представить их решения с помощью только бинарного кодирования. В течение последних десяти лет были разработаны различные методы кодирования для обеспечения эффективного выполнения генетического алгоритма, которые могут быть разделены на следующие классы:

- бинарное кодирование;
- кодирование действительными числами;
- целочисленное кодирование;
- кодирование общей структуры данных.

Более подробно рассмотрим бинарное кодирование, которое используется в классическом подходе. Пусть вектор допустимого решения $\vec{x} \in D$, где D - область поиска решений. Каждый компонент вектора $\vec{x} = (x_1, x_2, \dots, x_n)$ можно закодировать с помощью целого неотрицательного числа $\beta_i \in [0, K_i]$, $i = 1, n$; (K_i - число возможных дискретных значений i - й переменной).

Введем бинарный алфавит $\beta_2 = \{0; 1\}$. Для представления целочисленного вектора $\beta = (\beta_1, \dots, \beta_n)$ в алфавите β_2 необходимо определить максимальное число двоичных символов θ , которое достаточно для отображения в двоичном коде любого значения β_i из области его допустимых значений $[0, K_i]$. Нетрудно видеть, что параметр θ должен удовлетворять неравенству

$$K < 2^\theta, \quad (1.3)$$

где $K = \max K_i$.

$$1 \leq i \leq n.$$

Тогда для $\beta_i (0 \leq \beta_i \leq 2^\theta)$ можно записать

$$\beta_i = \sum_{i=1}^{\theta} a_i 2^{\theta-1}, \quad (1.4)$$

Где a_i - двоичное число (0 или 1); θ - длина двоичного слова, кодирующего целое число β_i .

Пример. Представим в виде хромосомной строки число $\beta_i = 19$. С учетом формул (1.5) и (1.6) имеем: $\theta = 5$, так как $2^\theta = 2^5 = 32$

По формуле (1.6) получим:

$$19 = 1 \cdot 2^{5-1} + 0 \cdot 2^{5-2} + 0 \cdot 2^{5-3} + 1 \cdot 2^{5-4} + 1 \cdot 2^{5-5} = 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$$

Строка будет иметь следующий вид:

1	0	0	1	1
---	---	---	---	---

В качестве гена, т.е. единицы наследственного материала, ответственного за формирование признаков особи, примем бинарную комбинацию $e_\theta(\beta_i)$, которая определяет фиксированное значение целочисленного кода β_i управляемой переменной x_i в обычном двоичном коде. В этом случае $e_\theta(\beta_i)$ имеет вид:

a_1	a_2	...	a_θ
θ			

Одна особь a_k^t , будет характеризоваться n генами, каждый из которых отвечает за формирование целочисленного кода соответствующей управляемой переменной. Тогда хромосому можно определить в следующем виде $E(X)$:

a_i^1	...	a_θ^1	a_i^2	...	a_θ^2	a_i^n	...	a_θ^n
$e_\theta(\beta_1)$			$e_\theta(\beta_2)$...			$e_\theta(\beta_n)$		

Ген 1	Ген 2	...	Ген n
Локус 1	Локус 2	Локус n
Хромосома			

Местоположение определенного гена в хромосоме называется локус, а альтернативные формы одного и того же гена, расположенные в одних и тех же локусах, называются аллелями.

Хромосома, содержащая в своих локусах конкретные значения аллелей, определяет генотип $E(a_k^t)$, который содержит всю наследственную генетическую информацию об особи a_k^t , получаемую от предков и передаваемую затем потомкам. Конечное множество всех допустимых генотипов образует генофонд.

Таким образом, в реализации генетического алгоритма хромосома представляет собой бинарную строку фиксированной длины. При этом каждому участку строки соответствует ген. Длина гена внутри строки может быть одинаковой или различной. Например, для объекта из пяти признаков, общая длина хромосомы составит $5 \cdot 4 = 20$ битов:

0010 1010 1001 1101.

При использовании N битов для бинарной строки (хромосомы) преобразование от двоичного кода строки к десятичному значению осуществляется по формуле:

$$x_i = a_i + decimal(100 \dots 010_2) * \frac{b_i - a_i}{2^N - 1},$$

Где a_i, b_i - нижняя и верхняя границы изменения i -го признака; $decimal(100 \dots 010_2)$ - десятичное значение бинарной строки.

Кодирование действительными числами является лучшим при решении задач оптимизации функции. Так как топологическая структура пространства генотипов для вида кодирования идентична структуре в пространстве

фенотипов, то легко сформировать эффективные генетические операторы заимствованием полезных приемов у традиционных методов.

Целочисленное кодирование лучше всего подходит для комбинаторных оптимизационных задач.

В соответствии с кодированием общей структуры данных методы кодирования могут быть разделены на два вида:

- одномерный,
- многомерный.

В большинстве случаев используется одномерное кодирование, однако многие проблемы требуют решений в многомерном пространстве, поэтому естественно применять метод многомерного кодирования в таких ситуациях.

Генетические алгоритмы выполняются на двух типах пространств: кодирование и решения, другими словами, пространствах генотипа и фенотипа. Генетические операторы работают в пространстве генотипов, а оценка и отбор происходят в фенотипическом пространстве. Естественный отбор - это связь между хромосомами и поведением декодированных решений.

Отображение из генотипического пространства в фенотипическое оказывает значительное влияние на поведение генетического алгоритма. Одна из главных проблем заключается в том, что некоторые особи соответствуют невозможным решениям данной задачи. При использовании генетического алгоритма необходимо различать две базовые концепции:

- недопустимость,
- незаконность.

Эти термины часто неправильно используются в литературе. Недопустимость относится к ситуации, когда решение, декодированное из

хромосом, лежит вне области допустимых решений данной задачи. Незаконность относится к такому явлению, когда хромосома не описывает конкретной задачи.

Недопустимость хромосом следует из существа задачи оптимизации с ограничениями. Независимо от того, какой метод используется - традиционный или генетический алгоритм - он должен иметь дело с ограничениями. Для многих оптимизационных задач допустимая область может быть представлена как система уравнений или неравенств. В таких случаях при работе с недопустимыми хромосомами могут быть применены штрафные методы. В задачах оптимизации оптимум обычно имеет место на границе между допустимыми и недопустимыми областями. Штрафные методы будут усиливать генетический поиск для приближения к оптимуму с двух сторон (допустимой и недопустимой).

Незаконность хромосом вытекает из природы методов кодирования. Для многих задач комбинаторной оптимизации методы кодирования обычно дают незаконное потомство посредством простой одноточечной операции скрещивания. Так как незаконная хромосома не может быть декодирована в решение, штрафные методы неприменимы в такой ситуации.

При появлении или разработке нового метода кодирования необходимо рассмотреть, можно ли провести эффективный генетический поиск этим методом. Для оценки кодирования предложено использовать следующие принципы:

- недостаточность,
- легальность,
- полноту,
- причинность.

Недостаточность означает, что отображение между кодированием и решением должно иметь вид $1 - k - 1$. В общем случае такое отображение может относиться к одному из следующих типов:

1. $1 - k - 1$.
2. $n - k - 1$.
3. $1 - k - n$.

Самый лучший способ - это отображение $1 - k - 1$. Если имеет место отображение $n - k - 1$, то генетический алгоритм затратит много времени на поиск. Самая плохая ситуация при отображении $1 - k - n$, так как необходимы дополнительные процедуры на фенотипическом пространстве для определения единственного из n возможных решений.

Легальность означает, что любая перестановка кодирования соответствует решению. Это свойство гарантирует, что большинство генетических операторов могут быть легко применены к кодированию.

Полнота определяет, что любое решение имеет соответствующее кодирование и любая точка в пространстве решений доступна для генетического поиска.

Причинность связывает малые вариации в пространстве генотипов, которые появляются вследствие мутации, с малыми вариациями в фенотипическом пространстве. Это свойство было предложено, в первую очередь, для эволюционных стратегий. Оно определяет сохранение соседских структур. Таким образом, для успешного введения новой информации посредством мутации ее операторов должен сохранять структуру соседства в соответствующем фенотипическом пространстве.

ГЕНЕТИЧЕСКИЕ ОПЕРАТОРЫ

1. В каждой генерации генетического алгоритма хромосомы являются результатом применения некоторых генетических операторов.

Оператор - это языковая конструкция, представляющая один шаг из последовательности действий или набора описаний алгоритма.

Генетический алгоритм состоит из набора генетических операторов.

Генетический оператор по аналогии с оператором алгоритма - средство отображения одного множества на другое. Другими словами, это конструкция, представляющая один шаг из последовательности действий генетического алгоритма.

2. Рассмотрим основные операторы генетических алгоритмов.

Оператор репродукции (селекции) - это процесс, посредством которого хромосомы, имеющие более высокое значение целевой функции, получают большую возможность для воспроизводства потомков, чем «худшие» хромосомы. Элементы, выбранные для репродукции, обмениваются генетическим материалом, создавая аналогичных или различных потомков.

Существует большое число видов операторов репродукции. К ним относятся следующие.

- Селекция на основе рулетки - это простой и широко используемый в простом генетическом алгоритме метод. При его реализации каждому элементу в популяции соответствует зона на колесе рулетки, пропорционально соразмерная с величиной целевой функции. Тогда при повороте колеса рулетки каждый элемент имеет некоторую вероятность

выбора для селекции. Причем элемент с большим значением целевой функции имеет большую вероятность для выбора.

- Селекция на основе заданной шкалы. Здесь популяция предварительно сортируется от «лучшей» к «худшей» на основе заданного критерия. Каждому элементу назначается определенное число и тогда селекция выполняется согласно этому числу.
- Элитная селекция. В этом случае выбираются лучшие элементы на основе сравнения значений целевой функции. Далее они вступают в различные преобразования, после которых снова выбираются элитные элементы. Процесс продолжается аналогично до тех пор, пока продолжают появляться элитные элементы.
- Турнирная селекция. При этом некоторое число элементов выбирается случайно или направленно из популяции, и лучшие элементы в этой группе на основе заданного турнира определяются для дальнейшего эволюционного поиска.

Оператор репродукции считается эффективным, если он создает возможность перехода из одной подобласти альтернативных решений области поиска в другую. Это повышает вероятность нахождения глобального оптимума целевой функции. Выделяют два основных типа реализации оператора репродукции:

- Случайный выбор хромосом;
- Выбор хромосом на основе значений целевой функции.

При случайном выборе хромосом частота R образования родительских пар не зависит от значения целевой функции хромосом P_k^t и полностью определяется численностью популяции N :

$$R = \frac{\beta}{N(N-1)}, \quad (1.5)$$

где β - коэффициента селекции, принимающий в зависимости от условий внешней среды значения $1 \div 4$.

Другой способ реализации оператора репродукции связан с использованием значений целевой функции. Существуют две основные стратегии. Стратегия - это оптимальный набор правил и приемов, которые позволяют реализовать общую цель, достигнуть глобальных и локальных целей решаемой задачи. В первой предпочтение отдается хромосомам с близкими и «лучшими» значениями целевой функции. Во второй - хромосомам, со значениями целевой функции, сильно различающимися между собой.

Для реализации первой стратегии с максимизацией функции с вероятностью

$$\Pr(OP) = \frac{\beta}{\text{ЦФ}(P_k^t)} / \sum_{i=1}^N \text{ЦФ}(P_k^t), \quad k = \overline{1, N}, \quad (1.6)$$

Выбирают разные хромосомы. Здесь ЦФ - целевая функция, ОП - это оператор репродукции, моделирующий естественный процесс селекции, $\Pr(OP)$ - вероятность выбора хромосом для репродукции.

Вторая стратегия реализуется так: часть хромосом выбирается случайным образом, а вторая - с вероятностью на основе выражения (1.6).

Если $\text{ЦФ}(P_k^t) < \text{ЦФ}_{\text{ср}}$, где $\text{ЦФ}_{\text{ср}}$ - среднее значение целевой функции в популяции, то оператор репродукции моделирует естественный отбор. Выбор случайных и сильно отличающихся хромосом повышает генетического разнообразия популяции, что повышает скорость сходимости генетического алгоритма на начальном этапе оптимизации и позволяет в некоторых случаях выходить из локальных оптимумов.

3. Кроме описанных, существует большое количество других методов селекции, которые можно условно классифицировать на три группы. К первой группе отнесем вероятностные методы. Ко второй -

детерминированные методы. К третьей - различные комбинации методов из первой и второй групп. Построение новых операторов репродукции непрерывно продолжается.

4. Основной трудностью решения инженерных оптимизационных задач с большим количеством локальных оптимумов является предварительная сходимость алгоритмов. Другими словами, попадание решения в один, далеко не самый лучший, локальный оптимум при наличии их большого количества. Различные методы селекции и их модификации как раз и позволяют в некоторых случаях решать проблему предварительной сходимости алгоритмов. Следует отметить, что исследователи генетических алгоритмов все более склоняются к мысли применять комбинированные методы селекции с использованием предварительных знаний о решаемых задачах и предварительных результатах.
5. Опишем теперь операторы кроссинговера (скрещивания). Оператор кроссинговера - это языковая конструкция, позволяющая на основе преобразования (скрещивания) хромосом родителей (или их частей) создавать хромосомы потомков. Существует огромное количество операторов кроссинговера, так как их структура в основном и определяет эффективность генетических алгоритмов. Кратко рассмотрим основные операторы кроссинговера, известные в литературе, и их модификации.
 - 5.1. Простой (одноточечный) оператор кроссинговера. Перед началом работы одноточечного оператора кроссинговера определяется так называемая точка оператора кроссинговера, или разрезающая точка оператора кроссинговера, которая обычно определяется случайно. Эта точка определяет место в двух хромосомах, где они должны быть разрезаны. Например, пусть популяция P состоит из хромосом P_1 и P_2 , которые выступают в качестве родителей, $P = \{P_1, P_2\}$. Пусть первый и второй родители имеют вид $P_1: 11111$, $P_2 = 00000$. Выберем точку оператора кроссинговера между вторым и третьим

генами в P_1, P_2 . Тогда, меняя элементы после точки операторы кроссинговера между двумя родителями, можно создать два новых потомка. В нашем примере получим:

P_1 :	1	1	1	1	1
P_2 :	0	0	0	0	0
P'_1 :	1	1	0	0	0
P'_2 :	0	0	1	1	1

Итак, одноточечный оператор кроссинговера выполняется в три этапа:

1. Две хромосомы $A = a_1, a_2, \dots, a_L$ и $B = a'_1, a'_2, \dots, a'_L$ выбираются случайно из текущей популяции.
2. Число k выбирается из $\{1, 2, \dots, L - 1\}$ так же случайно. Здесь L - длина хромосомы, k - точка оператора кроссинговера (номер, значение или код гена, после которого выполняется разрез хромосомы).
3. Две новые хромосомы формируются из A и B путем перестановок элементов согласно правилу

$$A' = a_1, a_2, \dots, a_k, a'_{k+1}, \dots, a'_L,$$

$$B' = a'_1, a'_2, \dots, a'_k, a_{k+1}, \dots, a_L.$$

После применения оператора кроссинговера имеем две старые хромосомы и всегда получаем две новые хромосомы. Схематически простой оператор кроссинговера показывает преобразование двух хромосом и частичный обмен информацией между ними, использующий точку разрыва, выбранную случайно.

- 5.2. Двухточечный оператор кроссинговера. В каждой хромосоме определяются две точки оператора кроссинговера, и хромосомы

обмениваются участками, расположенными между двумя точками оператора кроссинговера. Например:

P_1 :	1	1	1	0	1	0	0
P_2 :	0	0	0	1	1	1	0
P'_1 :	1	1	1	1	1	0	0
P'_2 :	0	0	0	0	1	1	0

Отметим, что точки оператора кроссинговера в двухточечном операторе кроссинговера также определяются случайно. Существует большое количество модификаций двухточечного оператора кроссинговера. Развитием двухточечного оператора кроссинговера является многоточечный или N - точечный оператор кроссинговера. Многоточечный оператор кроссинговера выполняется аналогично двухточечному, хотя большое число разрезающих точек может привести к потере хороших родительских свойств.

Пример трехточечного оператора кроссинговера:

P_1 :	1	1	1	0	1	0	0
P_2 :	0	0	0	1	0	1	1
P'_1 :	1	0	0	0	1	1	1
P'_2 :	0	1	1	1	0	0	0

Здесь точки оператора кроссинговера делят хромосому на ряд строительных блоков (а данном случае 4). Потомок P'_1 образуется из нечетных блоков родителя P_1 и четных блоков P_2 . Потомок P'_2 образуется соответственно из нечетных блоков родителя P_2 и четных блоков P_1 .

Тогда многоточечный оператор кроссинговера выполняется аналогичным образом.

5.3. Упорядоченный оператор кроссинговера. Здесь разрезающая точка также выбирается случайно. Далее происходит копирование левого сегмента P_1 в P'_1 . Остальные позиции в P'_1 берутся из P_2 в

упорядоченном виде слева направо, исключая элементы, уже попавшие в P'_1 . Например:

P_1 :	A	B	C	D		E	F	G	H
P_2 :	G	A	B	E		C	D	F	H
P'_1 :	A	B	C	D		G	E	F	H

Получение P'_2 может выполняться различными способами. Наиболее распространенный метод копирования левого сегмента из P_2 , а далее анализ P_1 методом, указанным выше. Тогда имеем P'_2 : (G A B E/ C D F H).

5.4. Частично соответствующий оператор кроссинговера. Здесь также случайно выбирается разрезающая точка или точка оператора кроссинговера. Далее анализируются сегмента (части) в обеих хромосомах и устанавливается частичное соответствие между элементами первого и второго родителей с формированием потомков. При этом правый сегмент P_2 переносится в P'_1 , левый сегмент P_1 переносится в P'_1 с заменой повторяющихся генов на отсутствующие гены, находящиеся в частичном соответствии. Например:

P_1 :	A	B	C	D	E	F		G	H	I	J
P_2 :	E	C	I	A	D	H		J	B	F	G
P'_1 :	A	H	C	D	E	I		J	B	F	G

Аналогично можно получить P'_2 :

P_1 :	A	B	C	D	E	F		G	H	I	J
P_2 :	E	C	I	A	D	H		J	B	F	G
P'_2 :	E	C	F	A	D	B		G	H	I	J

5.5. Циклический оператор кроссинговера. Циклический оператор кроссинговера выполняется рекомбинации согласно циклам, которые существуют при установлении соответствия между генами первого и второго родителей. Например, пусть популяция P состоит из двух

хромосом: P_1 и P_2 , $P = \{P_1, P_2\}$. Первый и второй родители и их потомок имеют вид:

P_1 :	1	2	3	4	5	6	7	8	9	10
P_2 :	5	3	9	1	4	8	10	2	6	7
P'_1 :	1	3	9	4	5	8	10	2	6	7

При выполнении циклического оператора кроссинговера P'_1 заполняется, начиная с первой позиции, и копирует элемент с первой позиции P_1 . Элементу 1 в P_1 соответствует элемент 5 в P_2 . Следовательно, имеем первый путь в цикле. Элементу 5 в P_1 соответствует элемент 4 в P_2 , откуда второй путь в первом цикле. Продолжая далее, получим, что элементу 4 в P_1 соответствует элемент 1 в P_2 . Следовательно, сформирован первый цикл. Согласно этому циклу элемент 5 переходит в пятую позицию P'_1 , а элемент 4 - в четвертую позицию.

Сформируем теперь второй цикл. Элемент 2 в P_1 соответствует элементу 3 в P_2 . Продолжая аналогично получим второй цикл и третий циклы.

Следовательно, в P'_1 элемент 3 расположен во втором локусе, т.е. на второй позиции, элемент 9 - в третьем, элемент 6 - в девятом, элемент 8 - в шестом, элемент 2 - в восьмом, элемент 10 - в седьмом и элемент 7 - в десятом локусах. Циклический оператор кроссинговера и его модификации эффективно применяются для решения комбинаторно-логических задач, задач на графах и гиперграфах и других оптимизационных задач.

5.6. Универсальный оператор кроссинговера. В настоящее время он популярен для решения различных задач из теории расписаний. Вместо использования разрезающей точки в универсальный оператор кроссинговера вводят двоичную маску, длина которой равна длине заданных хромосом. Первый потомок получается сложением первого родителя с маской на основе следующих правил: $(0+0=0, 0+1=1, 1+1=0)$. Второй потомок получается аналогичным

образом. Для этого элемента в P_1 и P_2 гены меняются, как показано на следующем примере:

P_1 :	0	1	1	0	0	1	
P_2 :	0	1	0	1	1	1	
	0	1	1	0	1	0	маска
P'_1 :	0	0	0	0	1	1	
P'_2 :	0	0	1	1	0	1	

Маска может быть задана или выбирается случайно с заданной вероятностью или на основе генератора случайных чисел. При этом чередование 0 и 1 в маске происходит с вероятностью $\approx 50\%$. В некоторых случаях используется параметризованный универсальный оператор кроссинговера, где маска может выбираться с вероятностью для 1 или 0 выше, чем 50%. Такой вид маски эффективен, когда хромосомы кодируются в двоичном алфавите.

5.7. Для решения многих оптимизационных задач можно использовать некоторые классы алгоритмов, называемых «жадными». Такой алгоритм делает на каждом шаге локально оптимальный выбор, в надежде, что итоговое решение также окажется оптимальным. Это не всегда так. Но для многих задач такие эвристические алгоритмы дают оптимальный результат. Говорят, что к оптимизационной задаче применим принцип «жадного» выбора, если последовательность локально оптимального значения целевой функции.

Рассмотрим «жадный» оператор кроссинговера. Он может быть реализован на двух и более хромосомах, а в пределе - на всей популяции. Приведем алгоритм «жадного» оператора кроссинговера на примере нахождения пути с минимальной или максимальной стоимостью на графе:

1. Для всех хромосом популяции вычисляется целевая функция. Выбирается заданное число родительских хромосом и случайным образом на одной из хромосом определяется точка «жадного» оператора кроссинговера.
2. В выбранной хромосоме i -го гена, расположенного слева от точки «жадного» оператора кроссинговера, определяется значение частичной целевой функции. Например, это стоимость пути от частичной целевой функции. Например, это стоимость пути от выбранного гена к ближайшему, находящемуся справа гену. Аналогичные действия выполняются по определению стоимости пути от i -го гена во всех остальных хромосомах, выбранных для «жадного» оператора кроссинговера.
3. В хромосому «потомок» выбирают ген, у которого значение целевой функции выше (ниже) при максимизации (минимизации) значений целевой функции.
4. Процесс продолжается до построения хромосомы «потомок». Если в процессе реализации возникает цикл или тупик, то выбираются нерассмотренные гены с лучшим значением целевой функции.

Например, пусть задан граф $G = (X, U)$, $X = \{a, b, c, d, e\}$ в виде матрицы. Построим популяцию P , состоящую из трех родительских хромосом $P = \{P_1, P_2, P_3\}$, где $P_1: abcde$; $P_2: bdeca$; $P_3: ebadc$. Элементы матрицы определяют стоимость пути между любыми двумя вершинами графа, а каждый ген в хромосоме кодируется номером вершины графа:

	a	b	c	d	e
a	-	15	6	7	8
b	15	-	4	3	2
c	6	4	-	1	10
d	7	3	1	-	9

e | | 8 2 10 9 - | |

Согласно алгоритму выберем точку «жадного» оператора кроссинговера между генами b и c в хромосоме P_1 . Теперь выбор $(b - c)$ дает значение целевой функции, равное 4, выбор $(b - d)$ (в хромосоме P_2) определяет целевую функцию со значением 3, а выбор $(b - a)$ (в хромосоме P_3) определяет целевую функцию, равную 15. При минимизации целевой функции выберем путь $(b - d)$. Продолжая, получим путь реализации «жадного» оператора кроссинговера. Итак, хромосома потомка P' : $bdcae$ имеет суммарную целевую функцию, равную 18: $3+1+6+8=18$, а целевая функция родителей для P_1 равна $15+4+1+9=29$, для P_2 равна $3+9+10+6=28$ и для P_3 равна $2+15+7+1=25$.

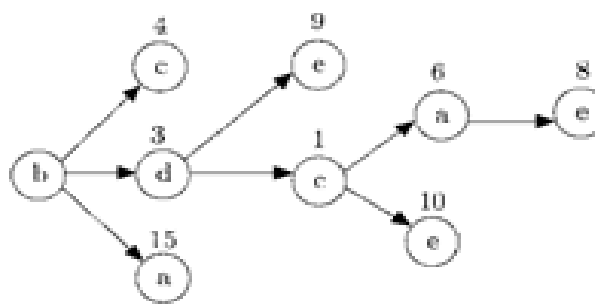


Рис 1.3. Пример реализации «жадного» оператора кроссинговера

Стратегия «жадного» оператора кроссинговера можно выполнять различными способами. Следует отметить, что поиск универсальных и специализированных операторов кроссинговера, ориентированных на решение заданного класса задач, продолжается.

Рассмотрим различные варианты выбора пары хромосом для скрещивания. Вероятность скрещивания $P_{n,x}(OK)$ лучших хромосом с худшими по значению целевой функции, должна уменьшаться при эволюции поколений: OK - оператор кроссинговера.

Вероятность скрещивания лучших хромосом должна увеличиваться на последних этапах оператора кроссинговера для закрепления желаемых признаков в хромосомах.

5.8. Существуют и другие методы выбора пар хромосом для скрещивания. Например, «близкое» родство, «дальнее» родство, выбор на основе кода Грея и т.д.

- для первой хромосомы P_i

$$P_{бр,i}(OK) = \sqrt{P_{бр,o}(OK)} * \left(1 - \exp\left(-\frac{1}{t} a\right)\right), \quad (1.7)$$

Где $P_{бр,o}(OK)$ - вероятность выбора хромосомы на основе близкого родства;

- для второй хромосомы P_j вероятность $P_j(OK)$ задается пользователем.

Затем вычисляется Хэммингово расстояние $dist(P_i, P_j)$ между выбранными хромосомами текущей популяции. Рассмотрим, например, две хромосомы, каждая из которых задается конкретными значениями семи переменных: $x_1, x_2, x_3, x_4, x_5, x_6, x_7$; пусть в первом случае ими будут 0,1,0,1,1,0,1, а во втором 1,1,0,1,0,0,0. Запишем данные совокупности значений x_1, x_2, \dots, x_7 одну по другой:

x_1	x_2	x_3	x_4	x_5	x_6	x_7
0	1	0	1	1	0	1
1	1	0	1	0	0	0
-1	0	0	0	1	0	1

Вычислим разность между каждым элементом первой строки и соответствующим элементом второй строки. Запишем результаты под второй строкой. Теперь вычислим сумму «абсолютных значений» полученных результатов. Эта сумма равна 3. Говорят, что расстояние Хэмминга между указанными выше точками $[0,1,0,1,1,0,1]$ и $[1,1,0,1,0,0,0]$ равна 3:

$$|-1| + |1| + |1| = 1 + 1 + 1 = 3.$$

В общем случае рассмотрим две хромосомы $(x'_1, x'_2, \dots, x'_n)$ и (x_1, x_2, \dots, x_n) ; обобщим расстоянием Хэмминга между этими двумя хромосомами называется скаляр

$$\pi = |x'_1 - x_1| + |x'_2 - x_2| + \dots + |x'_n - x_n|.$$

Хромосомы рекомендуется для скрещивания, если хеммингово расстояние между ними $dist(P_i, P_j) > R$, где R - радиус скрещивания, задаваемый лицом,, принимающим решение, или определяется как ближайшее целое от разности значений целевых функций $ЦФ(P_i) - ЦФ(P_j)$, деленное на два.

Вероятность $P_{бр,i}(OK)$ и $P_{бр,j}(OK)$ возрастают на конечных стадиях работы оператора кроссинговера.

«Дальнее» родство определяется условием

$$P_{др}(OK) = \sqrt{P_{бр,o}(OK)} * \exp\left(-\frac{1}{t} a\right),$$

Где $P_{др,i}(OK)$ - вероятность выбора хромосом при «дальнем» родстве на начальных стадиях работы генетических алгоритмов, с учетом особенностей вычисляется для первой и второй хромосомы.

Хромосомы P_i и P_j подлежат скрещиванию, если хеммингово расстояние между ними $dist(P_i, P_j) > R$. Вероятность $P_{др,i}(OK)$ и $P_{др,j}(OK)$ уменьшается на конечных стадиях поиска оптимального решения.

Код Грея - это двоичный код, последовательные значения которого отличаются только одним двоичным разрядом. Код Грея может использоваться для хромосом, предварительных в двоичном виде. Например:

0 – 0000

1 – 0001

2 – 0011

3 – 0010

4 – 0110

5 – 0111 и т.д.

Такое кодирование альтернативных решений позволяет решать вопросы «взбалтывания» популяции; оно эффективно на начальных стадиях генетического алгоритма.

Как следует из биологии, некоторые процессы преобразования популяции происходят толчками. Основой таких процессов являются точковые мутации. В генетическом алгоритме мутация необходима потому, что предотвращают потерю генетического материала. Точковые мутации не изменяют размера и строения хромосом, а изменяют взаимное расположение генов в хромосоме.

5.9. Оператор мутации - языковая конструкция, позволяющая на основе преобразования родительской хромосомы (или её части) создавать хромосому потомка.

Оператор мутации обычно состоит из двух элементов:

1. В хромосоме $A = (a_1, a_2, a_3, \dots, a_{L-2}, a_{L-1}, a_L)$ определяются случайным образом две позиции
2. Гены, соответствующие выбранным позициям, переставляются, и формируется новая хромосома $A' = (a_1, a_{L-1}, a_3, \dots, a_{L-2}, a_2, a_L)$.

Рассмотрим кратко основные операторы мутации. Простейшим оператором мутации является одноточечный. При его реализации случайно выбирается

ген в родительской хромосоме и, обменивая его на рядом расположенный ген, получают хромосому потомка, например:

$$\begin{array}{cccc|ccc}
 P_1: & 0 & 1 & 1 & 0 & 1 & 1 \\
 P'_1: & 0 & 1 & 0 & 1 & 1 & 1
 \end{array}$$

Здесь P_1 - родительская хромосома, а P'_1 - хромосома-потомок после применения точечного оператора мутации.

Развитием двухточечного оператора мутации является многоточечный оператор мутации. В этом случае происходит последовательный обмен генов, расположенных правее точек разреза друг с другом в порядке их расположения. Ген, расположенный правее последней точки разреза, переходит на место первого, например:

$$\begin{array}{cccc|cc|cc}
 P: & A & B & C & D & E & F \\
 P': & A & E & C & D & B & F
 \end{array}$$

Строительные блоки - это тесно связанные между собой гены, которые нежелательно изменять при выполнении генетических операторов. Из строительных блоков можно создавать альтернативные оптимальные или квазиоптимальные решения.

В частном случае, когда строительные блоки, расположенные между точками разреза, одинаковы, многоточечный оператор мутации выполняется следующим образом. При четном числе точек разреза меняются местами гены, расположенные справа и слева от выбранных точек, например:

$$\begin{array}{cccc|cc|cc|cc|cc}
 P: & A & B & C & D & E & F & G & H & I & J \\
 P': & A & C & B & E & D & G & F & I & H & J
 \end{array}$$

При нечетном числе точек потомок получается после обмена участками хромосом, расположенных между четными точками разреза, например:

$P:$	A	B	C		D	E	F		G	H	I		J
$P':$	A	B	C		G	H	I		D	E	F		J

Часто используют операторы мутации, использующие знания о решаемой задаче. Реализация таких операторов заключается в перестановке местами любых выбранных генов в хромосоме, причем точка или точки мутации определяются не случайно, а направленно.

В позиционном операторе мутации две точки разреза выбираются случайно, а затем ген, соответствующий второй точке мутации, размещается в позицию перед геном, соответствующим первой точке, например:

$P:$	A		B	C	D		E	F
$P':$	A		E	B	C		D	F

5.10. Введем понятие оператора инверсии. Оператор инверсии - это языковая конструкция, позволяющая на основе инвертирования родительской хромосомы (или ее части) создавать хромосому потомка. При его реализации случайным образом определяется одна или несколько точек разреза (инверсии), внутри которых элементы инвертируются.

Генетический оператор инверсии состоит из следующих шагов.

1. Хромосома $B = b_1, b_2, \dots, b_L$ выбираются случайным образом из текущей популяции.
2. Два числа y_1' и y_2' выбираются случайным образом из множества $\{0, 1, 2, \dots, L + 1\}$, причем считается, что $y_1 = \min\{y_1', y_2'\}$ и $y_2 = \max\{y_1', y_2'\}$.
3. Новая хромосома формируется из B путем инверсии сегмента, которая лежит справа от позиции y_1 и слева от позиции y_2 в хромосоме B . Тогда после применения оператора инверсии получаем:

$$B_1 = (b_1, \dots, b_{y_1}, b_{y_2-1}, b_{y_2-2}, \dots, b_{y_1+1}, b_{y_2}, \dots, b_L).$$

Для одноточечного оператора инверсии запишем

$$\begin{array}{rcccccccc}
 P_2: & A & | & B & C & D & E & F & G & H \\
 P_2': & A & | & H & G & F & E & D & C & B
 \end{array}$$

Здесь P_2 - родительская хромосома, P_2' - хромосома-потомок после применения оператора инверсии.

Например, для двухточечного оператора инверсии получим

$$\begin{array}{rcccc|cccc}
 P_1: & A & B & C & | & D & E & F & | & G & H \\
 P_1': & A & B & C & | & F & E & D & | & G & H
 \end{array}$$

Здесь P_1 - родительская хромосома, P_1' - хромосома-потомок после применения двухточечного оператора инверсии.

Часто применяется специальный оператор инверсии. В нем точки инверсии определяются с заданной вероятностью для каждой новой создаваемой хромосомы в популяции.

5.11. Рассмотрим оператор транслокации. Оператор транслокации - это языковая конструкция, позволяющая на основе скрещивания и инвертирования из пары родительских хромосом (или ее частей) создавать две хромосомы потомков. Другими словами, он представляет собой комбинацию операторов кроссинговера и инверсии. В процессе его реализации случайным образом производится один разрез в каждой хромосоме. При формировании потомка P_1' берется левая часть до разрыва из родителя P_1 и инверсия правой части до разрыва из P_2 . При создании P_2' берется левая часть P_2 и инверсия правой части P_1 , например:

$$P_1: \quad A \quad B \quad | \quad C \quad D \quad E \quad F$$

$P_2:$	G	K		H	I	J	Q
$P'_1:$	A	B	Q	J	I	H	
$P'_2:$	G	K	F	E	D	C	

Существует большое число других видов оператора транслокации. Отметим, что до последнего времени оператор транслокации не применялся в генетических алгоритмах, а также при разработке интеллектуальной искусственной системы и решении оптимизационных задач.

5.12. Оператор транспозиции - языковая конструкция, позволяющая на основе преобразования и инвертирования выделяемой части родительской хромосомы создавать хромосому потомка. Например,

$P_1:$	A		B	C		D	E	F		G	H
$P'_1:$	A	F	E	D	B	C	G	H			

Здесь три точки разреза. Точки разреза выбираются случайным или направленным образом. В родительской хромосоме P_1 строительный блок D E F инвертируется и вставляется в точку разреза между генами A и B. В результате получаем хромосому-потомок P'_1 . Отметим, что существует большое количество модификаций оператора транспозиции.

5.13. Рассмотрим оператор сегрегации. Это языковая конструкция, позволяющая на основе выбора строительных блоков из хромосом родителей (или ее частей) создавать хромосомы потомков.

Приведем один из примеров его реализации. Отметим, что оператор сегрегации, как правило, реализуется на некотором наборе хромосом. Пусть имеется популяция P , состоящая из четырех родительских хромосом:

$$P = \{P_1, P_2, P_3, P_4\} P_1: (12345678); P_2: (24316587); P_3: (31425768);$$

$$P_4: (87654321).$$

В каждой хромосоме выделим строительные блоки. Выделим по два строительных блока: в P_1 - 23 и 67, в P_2 - 1 и 4, в P_3 - 768 и 425 и в P_4 - 54 и 87. Тогда, например, потомок P'_1 будет представлен последовательностью популяции. Так, вариант P'_1 будет представлен последовательностью (23176854) и является реальным решением, а вариант P'_2 (67442587) является нереальным (недоступным). Очевидно, что оператор сегрегации можно реализовать различными способами в зависимости от выборки строительный блоков или генов из хромосом.

5.14. Опишем оператор удаления. Это языковая конструкция, позволяющая на основе вставки строительных блоков в хромосомы родителей создавать хромосомы потомков.

При его реализации направленным или случайным образом определяется точка или точки разреза. Далее производится пробное удаление генов или их строительных блоков с вычислением изменения значения целевой функции. Элементы, расположенные справа от точки оператора удаления или между двумя точками, удаляются из хромосомы. При этом производится преобразование потомка таким образом, чтобы соответствующее альтернативное решение оставалось реальным.

5.15. Опишем оператор вставки. Это языковая конструкция, позволяющая на основе вставки строительных блоков в хромосомы родителей создавать хромосомы потомков.

При его реализации направленным или случайным образом создается хромосома (донор), состоящая из строительных блоков, которые желательно разместить в другие хромосомы популяции. После этого направленным или случайным образом определяется хромосома для реализации оператора вставки. В ней находится точка или точки разреза. Затем анализируются другие хромосомы в популяции для определения альтернативных вставок. Далее производится пробная вставка строительных блоков с вычислением изменения

значения целевой функции и получением реальных решений. Новые строительные блоки вставляют в хромосому справа от точки оператора вставки или между его двумя точками. Отметим, что оператор удаления и оператор вставки могут изменять размер хромосом. Для сохранения постоянного размера хромосом эти операторы можно применять совместно.

На конечном этапе поиска целесообразно применять выбор близких решений в соответствии с определенным критерием, т.е. искать решение среди лучших P_k^t .

5.16. Оператор редукции - языковая конструкция, позволяющая на основе анализа популяции после одной или нескольких поколений генетического алгоритма уменьшать ее размер до заданной величины. Рассмотрим способы реализации оператора редукции. Он выполняется для устранения неудачных решений. В некоторых генетических алгоритмах, в частности, в простом генетическом алгоритме, этот оператор применяется для сохранения постоянного размера начальной популяции. Основная проблема здесь - нахождение компромисса между разнообразием генетического материала и качеством решений. Сначала формируют репродукционную группу из всех решений, образовавшихся в популяции P^t , затем выполняют отбор решений в следующую популяцию.

Численность новой популяции

$$N^{t+1} = N^t + N_{OK} + N_{OM} + N_{OI} + N_{OT} + N_{OTP} + N_{OC} + N_{OU} + N_{OV},$$

где

N^{t+1} - численность новой популяции,

N^t - численность популяции на предыдущем шаге (поколении) t , N_{OK} , N_{OM} , N_{OI} , N_{OC} , N_{OTR} , N_{OY} , N_{OB} - потомки, полученные в результате применения операторов: кроссинговера, мутации, инверсии, транслокации, транспозиции, сегрегации, удаления, вставки.

Отметим, что оператор редукции может применяться после каждого оператора или после всех в одной генерации генетического алгоритма. Выделяют две основных схемы редукции.

1. Элитная схема редукции. В группу удаления из популяции включаются такие хромосомы, как $P_{k+1}^t \in P^t$ и только те потомки, для которых выполняется условие:

$$(\forall P_{k+1}^t \in P^t)(\exists P_{k(GO)})(\text{ЦФ}(P_{k(GO)}) > \text{ЦФ}(P_k)), \quad k = \overline{1, n},$$

Где $P_{k(GO)}$ - потомки (решения), полученные после применения генетического оператора (ГО).

2. Последовательная схема редукции позволяет варьировать методы выбора хромосом для удаления из популяции:
 - Случайный выбор,
 - Выбор «лучших» и «худших»,
 - «близкое» родство,
 - «дальнее» родство,
 - На основе кода Грея для бинарных хромосом,
 - На основе расстояние Хэмминга,
 - На основе «турнира».

Случайный выбор хромосом позволяет разнообразить генофонд на ранних этапах генетических алгоритмов. Вероятность этого выбора должна снижаться при эволюции поколений.

По аналогии с оператором репродукции известны следующие модификации операторов редукции:

По аналогии с оператором репродукции известны следующие модификации операторов редукции:

1) Равновероятностный отбор с вероятностью

$$P_k(s) = \frac{ЦФ(P_k)}{\sum_{k=1}^n ЦФ(P_k)}.$$

Подведем итоги. С помощью операторов редукции на ранних стадиях работы генетических алгоритмов происходит выбор хромосом без учета значений их $ЦФ(P_k)$, т.е. случайный отбор. На заключительной стадии - определяющий фактор при отборе значения $ЦФ(P_k)$. Чем выше $ЦФ(P_k)$, тем выше вероятность отбора P_k в следующую популяцию. На заключительной стадии проводится уменьшение случайных операций и увеличивается процент направленных.

5.17. Рассмотрим теперь оператор рекомбинации. Оператор рекомбинации - языковая конструкция, которая определяет, как новая генерация хромосом будет построена из родителей и потомков. Другими словами, оператор рекомбинации - это технология анализа и преобразования популяции при переходе из одной генерации в другую. Существует много путей выполнения рекомбинации. Один из них состоит из перемещения родителей в потомки после реализации каждого генетического оператора. Другой путь заключается в перемещении некоторой части популяции после каждой генерации.

Часто в генетических алгоритмах задается параметр $W(P)$, который управляет этим процессом. Так, $N_p(1 - W(P))$ элементов в популяции P , выбранных случайно, могут «выжить» в следующей генерации. Здесь N_p - размер популяции. Величина $W(P) = 0$ означает, что целая предыдущая

популяция перемещается в новую популяцию в каждой генерации. При дальнейшей реализации алгоритма лучшие или отобранные элементы из родителей и потомков будут выбираться для формирования новой популяции.

В инженерных задачах используются различные механизмы и модели этого процесса. Приведем несколько из них:

- М1 - вытеснение. Этот механизм определяет способ и порядок замены родительских хромосом из генерации t хромосомами потомками после генерации $t + 1$. Механизм реализован таким образом, что стремится удалять «похожие» хромосомы из популяции и оставлять отличающиеся.
- М2 - разделение. Этот механизм вводит зависимость значения целевой функции хромосомы от их распределения в популяциях.
- М3 - введение идентификаторов. Специальным хромосомам присваиваются метки. Операторы генетических алгоритмов применяются только к помеченным хромосомам.

Отметим, что оператор редукции является частным случаем оператора рекомбинации.

5.18. Важным понятием при реализации генетических операторов является вероятность, которая определяет, какой генерации. Для оптимизационных задач вероятность оператора кроссинговера обычно принимают в пределах $(0,6 \div 0,99)$; вероятность оператора мутации - $0,6$; инверсии - $(0,1 \div 0,5)$; транслокации - $(0,1 \div 0,5)$; транспозиции - $(0,1 \div 0,5)$; сегрегации - $(0,6 \div 0,99)$; удаления - $(0,6 \div 0,99)$; вставки - $(0,6 \div 0,99)$.

ОСНОВНАЯ ТЕОРЕМА О ГЕНЕТИЧЕСКИХ АЛГОРИТМАХ

Для того чтобы лучше понять функционирование генетического алгоритма, будем использовать понятие схема [7,15,21,33]. Понятие схема было введено для определения множества хромосом, обладающих некоторыми свойствами, т.е. подобных друг другу. Если аллели принимают значение от 0 до 1 (рассматриваются хромосомы с двоичным алфавитом), то схема представляет собой множество хромосом, содержащих нули и единицы на некоторых заранее н позициях. При рассмотрении схем удобно использовать расширенный алфавит $\{0,1,*\}$, в который помимо 0 и 1 введен дополнительный символ *, обозначающий любое допустимое значение, т.е. 0 или 1; символ * в конкретной позиции означает "все равно" Например,

$$10*1=\{1001,1011\}$$

$$*01*10=\{001010, 001110, 101010, 101110\}$$

Считается, что хромосома принадлежит к данной схеме, если для каждой j -й позиции (локус), $j=1,2,\dots,L$, где L -длина хромосомы; символ, занимающий j -ю позицию хромосомы, соответствует символу занимающему j -ю позицию схемы, причем символу * соответствуют как 0, так и 1. То же самое означают утверждения хромосома соответствует схеме и хромосома представляет схему. Отметим, что если в схеме присутствует символ m символов *, то эта схема содержит 2^m хромосом. Кроме того, каждая хромосома (цепочка) длиной L принадлежит к 2^L схемам. В таблицах 1 и 2 представлены схемы к которым принадлежит цепочка длиной 2 и 3 соответственно. Цепочки длиной 2 соответствуют четырем различным схемам, а цепочки длиной 3 - восьми схемам

Генетический алгоритм базируется на принципе трансформации наиболее приспособленных особей (хромосом). Пусть $P(0)$ означает исходную популяцию особей, а $P(k)$ - текущую популяцию. Из каждой

Таблица 1.

Звенья	Схемы			
	1	2	3	4
00	**	*0	0*	00
01	**	*1	0*	01
10	**	*0	1*	10
11	**	*1	1*	11

Таблица 2.

Звенья	Схемы							
	1	2	3	4	5	6	7	8
000	***	**0	*0*	0**	*00	0*0	00*	000
001	***	**1	*0*	0**	*01	0*1	00*	001
010	***	**0	*1*	0**	*10	0*0	01*	010
011	***	**1	*1*	0**	*11	0*1	01*	011
100	***	**0	*0*	1**	*00	1*0	10*	100
101	***	**1	*0*	1**	*01	1*1	10*	101
110	***	**0	*1*	1**	*10	1*0	11*	110
111	***	**1	*1*	1**	*11	1*1	11*	111

популяции $P(k)$, $k=0,1,\dots$ методом селекции выбираются хромосомы с наибольшей приспособленностью, которые включаются в так называемый родительский пул. Далее, в результате объединения особей из популяции $M(k)$ в родительские пары и выполнения операции скрещивания с вероятностью p_c , а

так же операции мутации с вероятностью p_m формируется новая популяция $P(k+1)$, в которую входят потомки особей из популяции $M(k)$.

Следовательно, для любой схемы, представляющей хорошее решение, было бы желательным, чтобы количество хромосом, соответствующих этой схеме, возросло с увеличением количества итераций k .

На соответствующее преобразование схем в генетическом алгоритме оказывают влияние 3 фактора: селекция хромосом, скрещивание и мутация. Проанализируем воздействие каждого из них с точки зрения увеличения ожидаемого количества представителей от отдельно взятой схемы.

Обозначим рассматриваемую схему символов S , а количество хромосом популяции $P(k)$, соответствующих этой схеме - $c(S, k)$. Следовательно, $c(S, k)$ можно считать количество элементов множества $P(k) \cap S$.

Начнем с исследования влияния селекции. При выполнении этой операции хромосомы из популяции $P(k)$ копируются в родительский пул $M(k)$ с вероятностью, определяемой выражением (таблица 1). Пусть $F(S, k)$ обозначает среднее значение функции приспособленности хромосом из популяции $P(k)$, которые соответствуют схеме S .

$$P(k)S = \{ch_1, \dots, ch_{c(S,k)}\}.$$

то

$$F(S, k) = \frac{\sum_{i=1}^{c(S,k)} F(ch_i)}{c(S,k)}. \quad (1.8)$$

Величина $F(S, k)$ также называется приспособленностью схемы S на k -й итерации.

Пусть $J(k)$ обозначает сумму значений функции приспособленности хромосом из популяции $P(k)$ мощностью N , т.е.

$$J(k) = \sum_{i=1}^N F(ch_i^{(k)}). \quad (1.9)$$

Обозначим через $\bar{F}(k)$ среднее значение функции приспособленности хромосом этой популяции, т.е.

$$\bar{F}(k) = \frac{1}{N} J(k). \quad (1.10)$$

Пусть $ch_r^{(k)}$ обозначает элемент родительского пула $M(k)$. Для каждого $ch_r^{(k)} \in M(k)$ и для каждого $i = 1, \dots, c(S, k)$ вероятность того, что $ch_r^{(k)} = ch_i$, определяется отношением $F(ch_i)/F(k)$. Поэтому ожидаемое количество хромосом в популяции $M(k)$, которые равны ch_i , составит

$$N \frac{F(ch_i)}{J(k)} = \frac{F(ch_i)}{\bar{F}(k)}$$

Таким образом, ожидаемое количество хромосом из множества $P(k) \cap S$, отобранных для включения в родительский пул $M(k)$, будет равно

$$\sum_{i=1}^{c(S,k)} \frac{F(ch_i)}{\bar{F}(k)} = c(S, k) \frac{F(S,k)}{\bar{F}(k)},$$

что следует из выражения (1.11).

Поскольку каждая хромосома из родительского пула $M(k)$ одновременно принадлежит популяции $P(k)$, то хромосомы, составляющие множество $M(k) \cap S$ - это те же самые особи, которые были отобраны из множества $P(k) \cap S$ для включения в популяцию $M(k)$. Если количество хромосом родительского пула $M(k)$, соответствующих схеме S (т.е. количества элементов множества $M(k) \cap S$), обозначить $b(S, k)$, то из приведенных рассуждений можно сделать следующий вывод:

Вывод 1. Ожидаемое значение $b(S, k)$, т.е. ожидаемое значение количества хромосом родительского пула $M(k)$, соответствующих схеме S , определяется выражением

$$E[b(S, k)] = c(S, k) \frac{F(S, k)}{\bar{F}(k)}. \quad (1.11)$$

Из этого следует, что если схема S содержит хромосомы со значением функции приспособленности, превышающим среднее значение, то ожидаемое количество хромосом из родительского пула $M(k)$, соответствующих схеме S , будет больше количества хромосом из популяции $P(k)$, соответствующих схеме S . Поэтому можно утверждать, что селекция вызывает распространение схем с приспособленностью лучше средней и исчезновение схем худшей приспособленности.

Прежде чем приступить к анализу влияния генетических операторов скрещивания и мутации на хромосомы из родительского пула, определим необходимые для дальнейших рассуждений понятия порядка и охвата схемы. Пусть L обозначает длину хромосом, соответствующих схеме S .

Определение 1. Порядок схеме S , иначе называемый счетностью схемы и обозначаемый $o(S)$ - это количество постоянных позиций в схеме, т.е. нулей и единиц в случае алфавита $\{0,1,*\}$. Например,

$$o(10 * 1) = 3o(* 01 * 10) = 4o(** 0 * 1 *) = 2 o(* 101 **) = 3.$$

Порядок схемы $o(S)$ равен длине L за вычетом количества символов $*$, что легко проверить на представленных примерах. Легко заметить, что порядок схемы, состоящей исключительно из символов $*$, равен нулю, т.е. $o(***)=0$, а порядок схемы без единого символа $*$ равен L ; например, $o(10011010) = 8$. Порядок схемы $o(S)$ - это всегда целое число из интервала $[0, L]$.

Определение 2. Охват схемы S , называемый также длиной схемы (это не длина L) и обозначаемый $d(S)$ - это расстояние между первым и последним постоянным символом. Например,

$$d(10 * 1) = 4 - 1 = 3 \qquad d(* 01 * 10) = 6 - 2 = 4$$

$$d(** 0 * 1 *) = 5 - 3 = 2$$

$$d(* 101 *) = 4 - 2 = 2$$

Охват схемы $d(S)$ - это целое число из интервала $[0, L - 1]$. Отметим, что охват схемы с постоянными символами на первой и последней позиции равен $L - 1$. Охват схемы с единственной постоянной позицией равен нулю, в частности, $d(** 1 *) = 0$. Охват характеризует содержательность информации, заключенной в схеме.

Перейдем к рассуждениям о влиянии операции скрещивания на обработку схем в генетическом алгоритме. Прежде всего отметим, что одни схемы оказываются более подверженными уничтожению в процессе скрещивания, чем другие. Например, рассмотрим схемы $S_1 = 1****0*$ и $S_2 = **01***$, а также хромосому $ch = [1001101]$, соответствующую обеим схемам. Видно, что схема S_2 имеет больше шансов «пережить» операцию скрещивания, чем схема S_1 , которая больше подвержена «расщеплению» в точках скрещивания 1,2,3,4 и 5. Схему S_2 можно разделить только при выборе точки скрещивания, равной 3. Обратим внимание на охват обеих схем, который - это очевидно - оказывается существенным в процессе скрещивания.

В ходе анализа влияния операции скрещивания на родительский пул $M(k)$ рассмотрим некоторую хромосому из множества, соответствующую $M(k) \cap S$, т.е. хромосому из родительского пула, соответствующую схеме S . Вероятность того, что эта хромосома будет отобрана для скрещивания, равна p_c . Если ни один из потомков этой хромосомы не будет принадлежать к схеме S , то это означает, что точка скрещивания должна находиться между первым и последним постоянным символом данной схемы. Вероятность этого равна $d(S)/(L - 1)$. Из этого можно сделать следующие выводы:

Вывод 2. Для некоторой хромосомы из $M(k) \cap S$ вероятность того, что она будет отобрана для скрещивания и ни один из ее потомков не будет принадлежать к схеме S , ограничена сверху величиной

$$p_c \frac{d(S)}{(L-1)}$$

Эта величина называется вероятностью уничтожения схемы S .

Вывод 3. Для некоторой хромосомы из $M(k) \cap S$ вероятность того, что она не будет отобрана для скрещивания либо, что хотя бы один из ее потомков после скрещивания будет принадлежать к схеме S , ограничена снизу величиной

$$1 - p_c \frac{d(S)}{(L-1)}$$

Эта величина называется вероятностью выживания схемы S . Легко показать, что если данная хромосома принадлежит к схеме S и отбирается для скрещивания, а вторая родительская хромосома также принадлежит к схеме S , то оба их потомка тоже будут принадлежать к схеме S . Выводы 2 и 3 подтверждают значимость показателя охвата схемы $d(S)$ для оценки вероятности уничтожения или выживания схемы.

Рассмотрим теперь влияние мутации на родительский пул $M(k)$. Оператор мутации с вероятностью p_m случайным образом изменяет значение в конкретной позиции с 0 на 1 и обратно. Очевидно, что схема переживет мутацию только в том случае, когда все ее постоянные позиции останутся после выполнения этой операции неизменными.

Хромосома из родительского пула, принадлежащая к схеме S останется в этой схеме да и только тогда, когда ни один символ этой хромосомы, соответствующий постоянным символам схемы S , не изменится в процессе мутации. Вероятность такого события равна $(1 - p_m)^{o(S)}$. Этот результат можно представить в форме следующего вывода:

Вывод 4. Вероятность того, что некоторая хромосома из $M(k) \cap S$ будет принадлежать к схеме S после операции мутации, определяется выражением

$$(1 - p_m)^{o(S)}.$$

Эта величина называется вероятностью выживания схемы S после мутации.

Вывод 5. Если вероятность мутации p_m мала ($p_m \ll 1$), то можно считать, что вероятность выживания схемы S после мутации, определенная в выводах 4, приближенно равна

$$1 - p_m o(S).$$

Эффект совместного воздействия селекции, скрещивания и мутации (вывод 1-4) с учетом факта, что если хромосома из множества $M(k) \cap S$ дает потомка, соответствующего схеме S , то он будет принадлежать к схеме $P(k + 1) \cap S$, ведет к построению следующей схемы репродукции:

$$E[c(S, k) + 1] \geq c(S, k) \frac{F(S, k)}{\bar{F}(k)} \left(1 - p_c \frac{d(S)}{L-1}\right) (1 - p_m)^{o(S)} \quad (1.12)$$

Зависимость (1.12) показывает, как изменяется от популяции к популяции количество хромосом, соответствующих данной схеме. Это изменение вызывается тремя факторами, представленными в правой части выражения (1.12), в частности: $F(S, k)/\bar{F}(k)$ отражает роль среднего значения функции приспособленности, $1 - p_c d(S)/(L - 1)$ показывает влияние скрещивания и $(1 - p_m)^{o(S)}$ - влияние мутации. Чем больше значение каждого из этих факторов, тем большим оказывается ожидаемое количество соответствий схеме S в следующей популяции. Вывод 5 позволяет представить зависимость (1.12) в виде

$$E[c(S, k) + 1] \geq c(S, k) \frac{F(S, k)}{\bar{F}(k)} \left(1 - p_c \frac{d(S)}{L-1} - p_m o(S)\right). \quad (1.13)$$

Для больших популяций зависимость (1.13) можно аппроксимировать выражением

$$c(S, k + 1) \geq c(S, k) \frac{F(S, k)}{\bar{F}(k)} \left(1 - p_c \frac{d(S)}{L-1} - p_m o(S)\right). \quad (1.14)$$

Из формул (1.13) и (1.14) следует, что ожидаемое количество хромосом, соответствующих схеме S в следующем поколении, можно считать функцией от фактического количества хромосом, принадлежащих этой схеме, относительной приспособленности схемы, а также порядка и охвата схемы. Заметно, что схемы с приспособленностью выше средней и с малым порядком и охватом характеризуется возрастанием количества своих представителей в последующих популяциях. Подобный рост имеет показательный характер, что следует из выражения (1.11). Для больших популяций эту формулу можно заменить рекуррентной зависимостью вида

$$c(S, k + 1) = c(S, k) \frac{F(S, k)}{\bar{F}(k)}. \quad (1.15)$$

Если допустить, что схеме S имеет приспособленность на ε % выше средней, т.е.

$$F(S, k) = \bar{F}(k) + \varepsilon \bar{F}(k), \quad (1.16)$$

То при подстановке выражения (1.16) в неравенство (1.13) в предположении, что ε не изменяется во времени, при старте от $k = 0$ получаем

$$c(S, k) = c(S, 0)(1 + \varepsilon)^k, \\ \varepsilon = (F(S, k) - \bar{F}(k)) / \bar{F}(k), \quad (1.17)$$

Т.е. $\varepsilon < 0$ для схемы с приспособленностью выше средней и $\varepsilon > 0$ - в противном случае.

Равенство (1.17) описывает геометрическую прогрессию. Из этого следует, что в процессе репродукции схемы, оказавшиеся лучше средних, выбираются на очередных итерациях генетического алгоритма в показательно возрастающих количествах. Обратим внимание, что зависимости (1.11)-(1.15) основаны на предположении, что функция приспособленности F принимает только положительные значения. При использовании генетических алгоритмов для решения оптимизационных задач, в которых целевая функция может

принимать и отрицательные значения, необходимы некоторые дополнительные соотношения между оптимизируемой функцией и функцией приспособленности. Конечный результат, получаемый из выражений (1.12)-(1.14), можно сформулировать в форме теоремы. Это основная теорема генетических алгоритмов, иначе называемая теоремой о схемах.

Теорема 1. Схемы малого порядка, с малым охватом и с приспособленностью выше средней формируют показательно возрастающее количество своих представителей в последующих поколениях генетического алгоритма.

В соответствии с приведенной теоремой важным вопросом становится кодирование, которое должно обеспечивать построение схем малого порядка, с малым охватом и приспособленностью выше средней. Косвенным результатом теоремы можно считать следующую гипотезу, называемую гипотезой о кирпичиках.

Гипотеза 1. Генетический алгоритм стремится достичь близкого к оптимальному результату за счет комбинирования хороших схем малого порядка и малого охвата. Такие схемы называются кирпичиками.

Гипотеза о строительных блоках выдвинута на основании теоремы о схемах с учетом того, что генетические алгоритмы исследуют пространство поиска с помощью схем малого порядка и малого охвата, которые впоследствии участвуют в обмене информацией при скрещивании.

Несмотря на то, что для доказательства этой гипотезы предпринимались определенные исследования, однако в большинстве нетривиальных приложений приходится опираться на эмпирические результаты. В течение последних двадцати лет опубликованы многочисленные работы, посвященные применениям генетических алгоритмов, подтверждающим эту гипотезу. Если она считается истинной, то проблема кодирования приобретает критическое

значение для генетического алгоритма; кодирование должно реализовать концепцию малых строительных блоков. Качество, которое обеспечивает генетическим алгоритмам явное преимущество перед другими традиционными методами, несомненно заключается в обработке большого количества разных схем.

Обратимся к примерам $(b_i - a_i) * 10^q \leq 2^{m_i} - 1$ и $x_i = a_i + y_i \frac{b_i - a_i}{2^{m_i - 1}}$ на их основе проанализируем обработку схем генетическим алгоритмом.

Пример 2. Рассмотрим схему

$$S_0 = \text{*****} 11$$

и покажем, как изменяется количество представителей этой схемы и приспособленность в процессе выполнения генетического алгоритма.

Длина $L = 12$, а охват и порядок схемы S_0 составляют соответственно $d(S_0) = 1$ и $o(S_0) = 2$. В исходной популяции из первого примера схеме S_0 соответствуют две следующие хромосомы:

$$ch_3 = [011101110011]$$

$$ch_7 = [101011011011]$$

Из формулы $E[c(S, k) + 1] \geq c(S, k) \frac{F(S, k)}{\bar{F}(k)} \left(1 - p_c \frac{d(S)}{L-1}\right) (1 - p_m)^{o(S)}$ следует, что после селекции и скрещивания количество хромосом, соответствующих схеме S_0 , должно быть больше или равно 2,5. Напомним, что вероятности скрещивания и мутации считаются равными соответственно $p_c = 1$ и $p_m = 0$. Приспособленность схемы S_0 в исходной популяции, обозначаемая $F(S_0, 0)$, равна 8 и превышает среднюю всех хромосом этой популяции $\bar{F} = 5,75$, что легко рассчитывается по (1.8) и (1.10).

В примере гипотезы после селекции и скрещивания в новой популяции получены четыре хромосомы, соответствующие схеме S_0 :

$$ch_1 = [001111011011]$$

$$ch_3 = [111011011011]$$

$$ch_7 = [011101011011]$$

$$ch_8 = [101011110011]$$

Приспособленность схемы S_0 в новой популяции, т.е. $F(S_0, 1)$, составит 8,25, тогда как средняя приспособленность хромосом этой популяции $F(1) = 7$, что также следует из формул (1.8) и (1.10). Новая популяция характеризуется большим средним значением функции приспособленности особей по сравнению с предыдущей популяцией, что уже отмечалось в примере гипотезы. Кроме того, в новой популяции приспособленность схемы S_0 оказывается лучшей, а количество представителей этой схемы - большим по сравнению с предыдущей популяцией.

Пример 3. Рассмотрим схему

$$S_1 = 1 ****$$

И проследим ее обработку при выполнении генетического алгоритма.

В этом случае $L = 5$, а охват и порядок схемы S_1 составляют $d(S_1) = 0$ и $o(S_1) = 1$ соответственно. В исходной популяции из схемы соответствуют три хромосомы

$$ch_1 = [10011]$$

$$ch_4 = [10101]$$

$$ch_6 = [11101]$$

Приспособленность схеме S_1 в исходной популяции $F(S_1, 0) = 1096$ и превышает среднюю приспособленность особей этой популяции $\bar{F}(0) = 589$, что следует из выражений (1.8)-(1.10). На основе формулы (1.11) легко

рассчитать ожидаемое количество хромосом родительского пула, соответствующих схеме S_1 . Оно состоит $3 \cdot 1096 / 589 = 5,58$. По результатам селекции в родительский пул включены 6 таких хромосом: $ch_6, ch_4, ch_6, ch_1, ch_6$. Ожидаемое количество хромосом, соответствующих схеме S_1 . Это все хромосомы данной популяции.

Пример 4. Рассмотрим схему

$$S_1 = 11 ***$$

и проследим ее обработку при выполнении генетического алгоритма.

Длина $L = 5$, а охват и порядок схемы S_2 составляют $d(S_2) = 1$ и $o(S_2) = 2$ соответственно. В исходной популяции из схемы соответствуют одна хромосома

$$ch_6 = [11101].$$

Поэтому приспособленность схемы S_2 в исходной популяции равна функции приспособленности хромосомы ch_6 и составляет 1683. Она превышает среднюю приспособленность особей исходной популяции, равную 589. По формуле (1.11) рассчитываем ожидаемое количество хромосом родительского пула, соответствующих схеме S_2 . Оно составит $1683 / 589 = 2,89$. По результатам селекции в родительский пул включены 3 одинаковых хромосомы $[11101]$, соответствующих схеме S_2 . Ожидаемое количество хромосом в новой популяции, соответствующих схеме S_2 , после скрещивания с вероятностью $p_c = 1$, должно превышать 5,58. В новую популяцию включены 3 хромосомы, соответствующих схеме S_2 . Это

$$ch_4 = ch_5 = ch_6 = [11101].$$

Пример 5. Рассмотрим схему

$$S_3 = *** 11$$

и проследим ее обработку при выполнении генетического алгоритма.

Длина $L = 5$, а охват и порядок схемы S_3 составляют $d(S_3) = 1$ и $o(S_3) = 2$ соответственно. В исходной популяции из схемы соответствуют три хромосомы

$$ch_1 = [10011]$$

$$ch_2 = [00011]$$

$$ch_3 = [00111]$$

В отличие от предыдущих примеров (2) и (3) приспособленность схеме S_3 в исходной популяции оказывается меньше средней приспособленности особей этой популяции $\bar{F}(0) = 589$ и составляет $F(S_3, 0) = 280$. Ожидаемое количество хромосом, соответствующих схеме S_3 и рассчитанное по формуле (1.11), равно $3 \cdot 280 / 589 = 1,426$. На основе формулы (1.12) получаем значение 1,068, определяющее количество представителей схемы S_3 в новой популяции. После скрещивания с вероятностью $p_c = 1$ в новую популяцию была включена одна хромосома, соответствующая схеме S_3 т.е. $ch_2 = [00011]$.

Пример 6. Рассмотрим схему

$$S_4 = * 10 **$$

и проследим ее обработку при выполнении генетического алгоритма.

Длина $L = 5$, а охват и порядок схемы S_4 составляют $d(S_4) = 1$ и $o(S_4) = 2$ соответственно. В исходной популяции из схемы соответствует только одна хромосома

$$ch_5 = [01000],$$

поэтому приспособленность схеме S_4 в исходной равна значению функции приспособленности хромосомы ch_5 и составляет 129. Аналогично предыдущему примеру (4), она меньше средней приспособленности особей

начальной популяции, которая равна 589. Ожидаемое количество представителей схемы S_4 в родительском пуле составляет $129/589=0,22$, что следует из формулы (1.11). Родительский пул не содержит не одной хромосомы, соответствующей схеме S_4 . При расчете ожидаемого количества представителей схемы S_4 в новой популяции по формуле (1.12) для вероятности скрещивания $p_c = 1$ и вероятности мутации $p_m = 0$ получаем значение 0,165. После скрещивания в новую популяцию не была включена ни одна хромосома, соответствующая схеме S_4 .

Из этих примеров посвященных обработке схем, можно сделать следующие выводы. Эти примеры иллюстрируют основную теорему генетических алгоритмов - теорему о схемах. Они затрагивают обработку схем низкого порядка с малым охватом. Некоторые примеры демонстрируют увеличение количества представителей данной схемы в следующем поколении, для случая когда приспособленность схемы превышает среднюю приспособленность всех особей популяции. Другие же примеры показывают ситуацию, когда приспособленность схемы превышает оказывает меньше средней приспособленности особей популяции. Количество представителей таких схем в следующих поколениях не увеличивается, а наоборот - наблюдается уменьшение количества соответствующих им хромосом.

При анализе подобных примеров для схем большего порядка и большего охвата также не регистрируется увеличение количества их представителей в следующем поколении, что согласуется с теоремой о схемах.

Выполнение генетических алгоритмов основано на обработке схем. Схемы малого порядка, с малым охватом и приспособленностью выше средней выбираются, размножаются и комбинируются, в результате чего формируются все лучшие кодовые последовательности. Поэтому оптимальное решение строится путем объединения наилучших из полученных к текущему моменту частичных решений. Простое скрещивание не слишком часто уничтожает

схемы с малым охватом. Однако невзирая на губительность операций скрещивания и мутации для схем высокого порядка и охвата, количество обрабатываемых схем настолько велико, что даже при относительно низком количестве хромосом в популяции достигаются весьма неплохие результаты выполнения генетического алгоритма.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы / Под ред. В.М. Курейчика. - 2-е изд., и доп. - М.:ФИЗМАТЛИТ, 2006. - 320 с.
2. Кричевский М. Л. Интеллектуальный анализ данных в менеджменте: Учеб. Пособие / СПбГУАП. СПб., 2005. 208 с.
3. Д. Рутковская, М. Пилиньский, Л. Рутковский. - 2-е изд., и доп. - М.: Горячая линия - Телеком, 2006. - 452 с.
4. Пантелеев А.В. Матаэвристические алгоритмы поиска глобального экстремума / рец.: А.В. Борисов, А Р. Панков. - М.: МАИ-ПРИНТ, 2009. - 160с.
5. Вольтерра В. Математическая теория борьбы за существование: монография / В. Вольтерра. - М.: Ижевск: Институт компьютерных исследований, 2004. - 288с. - (Биофизика. Математическая биология). - Библиог.: с. 284-286.
6. Панащенко Т.В. Генетические алгоритмы : Учебно-методическое пособие / Под ред. Ю.Ю. Тарасевича. – Астрахань: Издательский дом «Астраханский университет», 2007. – 87 с.
7. Барсегян А.А. Технологии анализа данных: Data Mining, Visual Mining, Text Mining, OLAP / А.А. Барсегян, М.С. Куприянов, В.В. Степаненко, И.И. Хо- лод. – 2-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2007. – 384 с.

Выпускная квалификационная работа выполнена мной совершенно самостоятельно. Все использованные в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

«___» _____ Г.

(подпись)

(Ф.И.О.)

