Введение в генетические методы

Автор: Конушин Антон ktosh@zmail.ru

Целый класс методов глобального поиска был построен по образу и подобию, данному нам самой Природой. Природа давно экспериментирует со своими игрушками - живыми существами, все время пытаясь найти наиболее подходящих для себя. Ее эксперимент был назван Эволюцией. Поэтому и методы, так или иначе копирующие Эволюцию получили название эволюционных. Одним из самых распространенных классов этих методов является семейство генетических методов, очень точно копирующих созданное Природой

Цели

Целью данной работы является описание базовых основ генетических методов глобальной оптимизации и автоматического программирования. Приводится краткое описание процесса естественной эволюции, классификация эволюционных методов, основные алгоритмы. Даются начальные сведения по применению генетических методов для систем автоматического программирования.

Вступление

Постоянно в нашей жизни мы добиваемся самого лучшего. Ну, если не самого, то нам подходящего. Это касается всего, чтобы мы не делали, пишем ли мы программу, выбираем книгу и думаем об ужине.

Но если задуматься, то это простое желание - найти получше, уже ставит перед нами множество вопросов. Что значит лучше, как мы это оцениваем, как мы это ищем и как стремимся этого добиться?

Для простых математических задачек люди давно придумали иногда простые, иногда сложные, но формулы вычисления. Подставив значения в формулы, мы сразу получаем наше подходящее значение. Таких задачек много, но других, не решенных намного больше.

Особенно в тяжелое положение мы попадаем, когда совершенно не представляем себе как искать этот лучший объект. Мы можем сравнить его с другим, оценить его. Но сказать каким должен быть самый лучший? Мы просто не знаем ответа на этот вопрос. Мы думаем, что такой существует. Если мы берем предмет, ищем другой, получше, но похожий, потом еще один такой же, то такой поиск называется локальным. А найденный нами предмет, будет ли он лучшим среди все ему подобных? Нам это тоже неизвестно, но чаще всего нет, не будет. Задача поиска наилучшего среди всех предметов вообще в некотором классе называется задачей глобального поиска (или оптимизации, но это очень похожие понятия).

Целый класс таких методов глобального поиска был построен по образу и подобию, данному нам самой Природой. Природа давно экспериментирует со своими игрушками - живыми существами, все время пытаясь найти наиболее подходящих для себя. Ее эксперимент был назван Эволюцией. Поэтому и методы, так или иначе копирующие Эволюцию получили название эволюционных. Одним из самых распространенных классов этих методов является семейство генетических методов, очень точно копирующих созданное Природой.

Понятие оптимизации.

Пусть у нас есть некоторый определенный тип или класс объектов. (т.е. множество объектов, удовлетворяющих некоторому набору условий). И пусть нам необходимо найти в этом классе некоторый объект удовлетворяющий другому некоторому условию. Такой процесс можно обобщенно назвать поиском или решением (solving). Класс, среди объектов которого производится поиск, назовем областью поиска или пространством поиска. Искомый объект можно назвать целью или целевым объектом поиска, а условие, которому он должен удовлетворять - целевым условием. Для определения условия обычно задается некоторая функция на пространстве поиска. Достижение функцией определенного значения и является целевым условием. Такая функция называется целевой функцией. Таким образом, поиск заключается в просмотре по определенным правилам пространства поиска всех объектов, пока не будет обнаружен целевой. Функции выбора нового кандидата для проверки называются операторами поиска. Оператор поиска

определяет своего рода прыжок или шаг по пространству поиска.

Примером задачи оптимизации может служить поиск минимума функции z = |x+sin(32*x)|. Областью поиска является все пространство вещественных чисел, целевым условием - минимум функции.

Понятие Машинного обучения

Точного определения машинного обучения пока нет. Можно просто сказать, что это есть процесс получения программой новых знаний. Достаточно хорошим определением может являться следующее, данное Митчеллом в 1996 году: "Машинное обучение это наука, изучающая компьютерные алгоритмы, автоматически улучшающиеся во время работы". Примером подобного рода алгоритмов являются нейросети.

Понятие об автоматическом программировании

Пусть перед нами стоит задача создания некоторой программы. Тогда эта задача тоже является задачей оптимизации, где пространство поиска - множество программ, а целевое условие - то, что программа должна делать (плюс некоторые ограничения на время работы программы, ее размер и т.д.) Термин оптимизация в данном случае обычно не употребляется, хотя с формальной точки зрения все правильно.

Если процесс создания такой программы выполняется человеком, то мы его называем просто программированием. А если одна программа сама создает другие? Тогда такой процесс можно назвать автоматическим программированием. А такую систему - системой автоматического программирования (САП)

Как должна быть устроена будущая программа, САП изначально не знает. Таким образом, САП каким-то образом обучается, т.е. получает откуда-то новые знания, во время создания новой программы, а значит полностью подходит под описание системы машинного обучения.

Стратегия поиска

Правила выбора для применения в данный момент того или иного оператора поиска называются стратегией поиска. Существует множество стратегий поиска, в данном случае рассмотрим три главных:

- поиск вслепую (blind search)
- подъем по холму (hill-climbing)
- поиск по лучу (beam search)

При поиске в слепую не учитываются ни результаты предыдущих шагов, ни какая-либо информация о задаче. Поиск происходит только в соответствии со структурой представления пространства поиска.

Поиск по стратегии hill-climbing начинается из некоторой точки. К начальному объекту применяются операторы поиска, все новые решения оценивается, и лучшее из них становится новым начальным в следующей итерации поиска.

Если два предыдущих метода являются точечными (т.е. в каждый конкретный момент рассматривается только один текущий объект), то поиск по лучу оценивает целый набор, или популяцию точек в пространстве поиска. Вначале по какому-нибудь методу выбирается начальный набор из наиболее многообещающих точек для последующей обработки. Такой набор и называется лучом (beam). Область поиска, таким образом, сжимается до множества всевозможных результатов применений операторов поиска к членам луча.

Естественная эволюция

У Матушки-Природы заранее припасено по примеру почти на каждую из идей, когда-либо пришедших в голову человека. Это же мысль можно выразить и по другому: почти все, придуманное человеком уже когда-то было изобретено природой. У Природы есть и свой метод создания лучших организмов. Дарвин назвал его Эволюцией вследствие Естественного отбора. Эволюция подразумевает под собой последовательное развитие организмов - непрерывную последовательность

родителей и их детей, когда дети многое наследуют от своих родителей, но кое в чем от них отличаются. Естественный отбор - от непрерывное сражение за жизнь между всеми. "Выживает сильнейший" - вот жизненное кредо Природы, если награждать титулом "сильный" самого подходящего, самого приспособленного для жизни.

Если подходить к описанию эволюции более формально, то вначале необходимо отметить что объектом развития (т.е. эволюции) являются не сами организмы, а виды в целом. Вид - это совокупность организмов, сходных по строению и другим признакам. Пользуясь терминологией объектно-ориентированного программирования, вид - это класс, а принадлежащие виду индивиды - объекты это класса. Совокупность индивидов одного вида назовем популяцией. Чтобы эволюция вообще была возможна, организмы должны отвечать 4 важнейшим свойствам:

1. Каждый индивид в популяции способен к размножению
2. Отличия индивидов друг от друга влияют на вероятность их выживания
3. Каждый потомок наследует черты своего родителя (подобное происходит от подобного)
4. Ресурсы для поддержания жизнедеятельности и размножения ограничены, что порождает конкуренцию и борьбу за них

Все процессы в живых организмах работают за счет сложных молекул - белков. Каждый белок представляет собой маленький биологический автомат. Молекула белка состоит из последовательности аминокислот. Совокупность информации и строении всех белков в организме определяет его изначальную структуру (развитие организма происходит также и под действием внешней среды). Вся эта информация называется генетической информацией, или генотипом. Процесс построения, развития организма по информации из генотипа называется онтогенезом. А строение, качества и свойства организма - фенотип. Т.к. внешняя среда воздействует на организм в целом, то можно сказать, что вероятность выживания организма определяется фенотипом.

Генетическая информация в клетке хранится в специальных молекулах - нуклеиновых кислотах. Нуклеиновая кислота представляет собой полимер, т.е. молекулу, представляющую собой последовательность из соединенных между собой небольших молекул - мономеров. Мономерами нуклеиновых кислот являются нуклеотиды. В каждой клетке встречаются два вида кислот - ДНК и РНК. ДНК содержится в ядре клетки и функционально является хранилищем генетической информации. РНК используется для транспортировки отдельных порций информации и построения по ней молекул-белков. Количество входящих нее нуклеотидов назовем длиной ДНК.

Для кодирования информации используется 4 вида нуклеотидов, обозначаемых по названиям входящих в них азотистых оснований A,T,G,C. Таким образом алфавит кодировки состоит из 4 букв.. Последовательность из 3-х нуклеотидов называется кодон. Каждый кодон соответствует одной из аминокислот. Всего имеется 20 базовых аминокислот и 64 кодона, а значит каждой аминокислоте соответствует более одного кодона, поэтому кодировка является избыточной. Участки ДНК, несущие информацию о строении какого-нибудь белка клетки, назовем полезными сегментами. Полезные сегменты, описывающие некоторое свойства индивида (фенотипа), называются геном. А различные значения этого гена называются аллелями.

В ДНК существуют и не-полезные участки, которые и получили название мусорных (junk) сегментов ДНК (мусорной ДНК - junk DNA). Также имеются небольшие последовательности, не содержащие информации о белке, но участвующие в процессе управления копированием ДНК при построении белка. Такие сегменты называются интронами (introns). Учитываю все вышеперечисленное, назовем структурой ДНК последовательность из интронов, мусорных участков, полезных сегментов и соответствие между полезными сегментами и описываемыми ими белками.

При сексуальном размножении потомку передается информации о строении родителей путем передачи ДНК. При этом, для построения ДНК потомка, родительские ДНК меняются своими участками. Это процесс называется перекрест или скрещивание (кросовер - crossover). При этом новый ген представляет собой комбинацию информации из родительских ДНК (рекомбинация наследственной информации). При размножении может произойти мутация ДНК, т.е. случайное изменение небольшой ее части.

ДНК двух совершенно разных организмов могут в значительной степени отличатся друг от друга: и по длине, и по структуре, и по кодируемым ими белкам. В одном организмы могут быть белки, которые вообще не используются в других видах организмов. Если перекрест будет происходить между такими разными ДНК, структура ДНК может быть полностью разрушена - полезные сегменты перемешаются с мусорными участками и т.д. Поэтому новая ДНК скорее всего будет нежизнеспособна - т.е. будет описывать организм, который просто не способен жить. Именно поэтому в

природе происходит скрещивание (а значит и перекрест ДНК) только между особями одного, или очень близких видов. Такое скрещивание (перекрестr) называется гомологичным, т.е. подобным.

Однако даже особи одного вида во многом отличаются друг от друга - некоторыми белками и др. Поэтому чтобы обеспечить жизнеспособность большинства новых ДНК, т.е. достигнуть стабильности в наследовании и сходства между поколениями (heredity), помимо гомологичного скрещивания природа используют еще пару механизмов. Первый - это избыточная кодировка аминокислот. Второй - это устойчивость белковых молекул к небольшим изменениям в их структуре.

Из этого небольшого обзора хорошо видно, что естественный процесс оптимизации является некоторым компромиссом между вариацией потомства (получением новых индивидов), обеспечением достаточного процента жизнеспособности и стремлением получить хорошее потомство, т.е. не хуже, или в большинстве случаев лишь чуть хуже предков.

Эволюционные алгоритмы

Эволюционные алгоритмы - это обобщенное название компьютерных алгоритмов решения (поиска), использующих вычислительные модели (computational) механизмов естественной эволюции в качестве ключевых структурных элементов. Существуют множество разновидностей подобного рода алгоритмов, отличающихся использование или не использованием конкретных механизмов, а также различиями трактовки этих механизмов и представлением индивидов.

В Генетическом Алгоритме (ГА) каждый индивид кодируется сходным с ДНК методом - в виде строки из символов одного типа. Длина строки (ДНК) постоянна. Популяция из индивидов подвергается процессу эволюции с интенсивным использованием перекреста и мутаций.

Генетическое Программирование ставит своей основной задачей автоматическое программирование, т.е. каждый индивид является некоторой программой. Размер программы ограничен, но не постоянен. Также используются помимо строчного (линейного) представления деревья и графы. В общем, во всем остальном ГП очень похоже на ГА.

Методы Эволюционных Стратегий и Эволюционного Программирования уделяют значительно больше внимание самому процессу эволюции. Первым отличием от ГА является отсутствие ограничений на представление. Второе заключается в возможности обобщения процесса эволюции и на сами параметры эволюции. Помимо объекта эволюции выделяются некоторые такие параметры стратегии эволюции как вероятность мутации, сила мутаций и др. Выбирая лучших индивидов мы учитываем и оптимальность этих параметров, таким образом, неявно выделяя параметры, наиболее подходящие для данной задачи.

Процесс работы ЭС и ЭП состоит из следующих фаз:

- Создание популяции объектов, со случайным выбором их параметров
- Порождение каждым объектом новой популяции (клонирование). Каждый объект в новой популяции подвергается мутации
- Все объекты оцениваются, и вся популяции подвергается процедуре естественного отбора, после чего осуществляется переход на 2-ю фазу

Между ЭС и ЭП существуют несколько отличий:

- ЭП работает на уровне видов, а не на уровне индивидов, поэтому скрещивание между объектами невозможно
- В ЭП процедура отбора детерминирована - отбрасываются самые плохие объекты, в ЭС используется турнирный метод

Генетическая оптимизация

Назовем представление каждого индивида геномом. Для каждого вида и каждого представления, для данного целевого

условия задается целевая функция. Значение целевой функции назовем целевым значением. Вектор, состоящий из целевых значений всех индивидов в популяции назовем вектором целевых значений. Тогда если вычислен вектор целевых значений, то можно определить приспособленность (fitness) индивида в популяции, для чего задается специальная функция приспособленности от данного целевого значения и от вектора целевых значений. Аналогично вектору целевых значений введем вектор приспособленности. Мы отделяем приспособленность от целевого значения специально, т.к. приспособленность индивида зависит и от остальных индивидов, и важна для выживаемости индивида, а целевое значение важно в первую очередь для нас. Часто целевое значение называют приспособленностью, а значение приспособленности в смысле вероятность участия в размножении неявно вычисляется во время Отбора.

Процесс эволюции останавливается, когда популяция отвечает определенному критерию - критерию завершения (termination criteria).

И ГА и ГП имеют одинаковую принципиальную схему работы и состоит из следующих основных фаз:

1. Создание начальной популяции. Задание генома каждому из индивидов. Расчет вектора целевых значений.
2. Шаг эволюции - построение нового поколения .
3. Проверка критерия завершения, если не выполнено - переход на 2

Шаг эволюции можно разделить на следующие этапы:

- Вычисление вектора приспособленности.
- Отбор кандидатов на скрещивание. (Отбор - Selection)
- Скрещивание, т.е. порождение каждой парой отобранных кандидатов новых индивидов, путем перекреста геномов. (Объединение всех потомков и предков назовем промежуточным поколением)
- Мутация геномов
- Вычисление вектора целевых значений для промежуточного поколения и построение новой популяции (нового поколения) определение выживших/отбракованных геномов. (Отбор 2 - Selection 2)

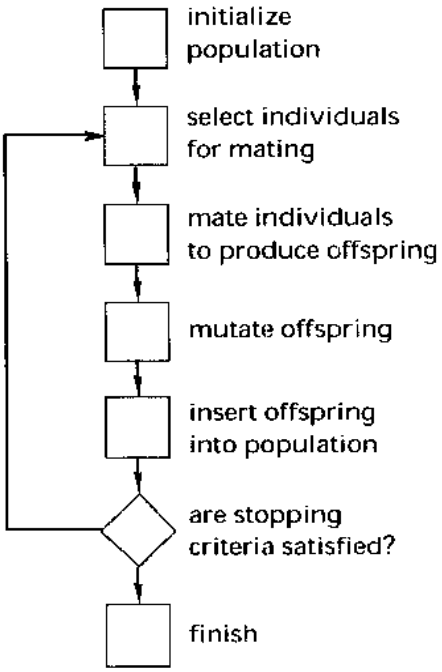

Рис.1 Схема работы ГА и ГП

Назовем порожденные индивиды - потомками, а породивших их - предками.

Как видно, генетический алгоритм - это схема эволюции. Чтобы получить точный вид алгоритма, нам нужно первоначально решить несколько задач:

- Какое представление индивидов выбрать?
- Какую задать целевую функцию и функцию приспособленности?
- Какое подобрать условие завершения?
- Как отбирать кандидатов на скрещивание, как определять отбракованные геномы?
- Могут ли индивиды из текущего поколения переходить в следующее, и будут ли они при этом мутировать?
- Как проводить перекрест генов?
- Как гены будут мутировать?

Почему генетические методы работают?

Генетические методы задают определенную стратегию обхода пространства поиска. Но найдем ли мы в конце концов целевой объект? И если да, то как скоро и будет ли процесс более эффективен по сравнению с другими методами? Ответа на данные вопросы пока нет. Первую попытку дать обоснование генетическим алгоритмам дал сам автор ГА Холланд в 1975 году, в своей теореме схемат.

Схематой называется строчка из символов 0,1 и * (не важно). Схемата - один из методов описания подпространств (указывая свойства - выполнено, не выполнено, не важно). Холланд в своей книге утверждал, что ГА работает за счет подразбиения пространства поиска на подпространства. Теорема схемат говорит, что при использовании отбора, пропорционального целевому значению, вероятность встретить ту или иную схемату меняется со временем таким образом, чтобы приблизиться к глобальному оптимуму.

Можно сказать, что при работе ГА выделяются "хорошие блоки", приближающие нас к оптимуму, которые с помощью перекреста объединяются в еще более успешные геномы. Мутация и перекрест вместе порождают такие "хорошие блоки".

Общие решения для ГА и ГП

Многие идеи применяются с одинаковым успехом и в ГА и ГП. Прежде всего это касается методов Отбора и Отбора 2.

Прежде чем определять метод Отбора, необходимо определить функцию приспособленности и рассчитать вектор приспособленности для популяции.

Существуют несколько наиболее распространенных методов Отбора. Самым первым появился Отбор пропорционально значению приспособленности. По другому он называется методом Рулетки. Вероятность выбора данного индивида рассчитывается как отношение его приспособленности к сумме приспособленности всех в популяции. Вторым часто применяемым методом является Турнирный метод. В Турнире вначале выбирается случайным образом определенное количество индивидов образующих промежуточный набор, внутри которого рассчитывается приспособленности и происходит отбор. Часто реализуемым вариантом такого метода является следующий: по методу рулетки выбираются два претендента, и затем из них выбирается лучший.

Выбирая стратегию Отбора 2, мы отвечаем на следующие вопросы:

- Могут ли индивиды из текущего поколения переходить в следующее?
- Кто и как выбирается из группы предок-потомок для перехода в следующее поколение?

Обычно при каждом скрещивании производятся 2 потомка, которые занимают место своих родителей в новом поколении. Однако мы можем провести вначале отбор внутри группы 2 предка- 2 потомка, и в следующее поколение передут 2 из 4-

x. А можем провести отбор внутри всего промежуточного поколения и выбрать N (N-размер популяции) наилучших индивидов. Такой метод называется методом Стабильного состояния (Steady-state). Можно также указать количество индивидов k, обновляемых в каждом новом поколении. Тогда на каждом шаге эволюции k наихудших индивидов предыдущего поколения заменяются k наилучшими потомками.

Чтобы не потерять наилучших из обнаруженных в процессе эволюции индивидов, можно применить стратегию Элитизм. Элитой называются несколько индивидов, имеющих наилучшее целевое значение. Элита выбирается в каждом поколении и переходит в следующее, не подвергаясь мутации.

Необходимо отметить также двойственность оператора перекреста. С одной стороны - это операция порождения потомков по двум предкам. С другой стороны ее можно рассматривать как операцию обмена частями между двумя особями. В этом нет никаких противоречий, если рассматривать операцию следующим образом. Вначале строятся потомки как копии своих родителей, затем они обмениваются своими сегментами.

Подробнее о ГА

Генетический алгоритм - это интересный метод оптимизации, применяемый для нахождения хороших решений задач оптимизации многомерных функций. Находимое решение скорее всего может и не быть оптимальным, но оно будет хорошим, т.е. близким к наилучшему, и находится за приемлемое время. Многие задачи, такие как известная задача Коммивояжера, решение которых в лоб требует перебора огромного числа вариантов, сейчас эффективно решаются с помощью генетических алгоритмов.

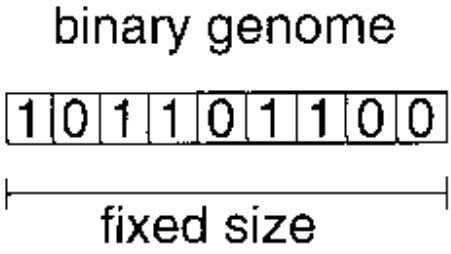

Рис.2 Ген

ГА использует в качестве представления геном - т.е. строчку определенной длины из однотипных элементов - генов. Каждому параметру, по которому происходит поиск (оптимизация), будет соответствовать свой ген. Фактически, геном - это массив фиксированной длины из генов. Элементами массива (генами) могут быть двоичные числа, целочисленные или вещественные переменные. Таким образом, весь геном представляет собой длинную последовательность битов. Такое битовое представление мотивируется теоремой схемат. Теорема утверждает, что алфавит должен быть как можно меньше. Все числа в компьютере представляются в двоичном виде, поэтому и геном часто рассматривается именно как битовая строка.

В качестве оператора мутации чаще всего применяется так называемая побитовая мутация (bit-flip или uniform). При применении такого оператора мутации каждый бит в гене меняет свое значение на противоположенное с некоторой заданной вероятностью. Т.к. и старшие и младшие биты двоичного представления чисел меняются с одинаковой вероятностью, то наряду с небольшими изменениями в значениях могу случаться и очень значительные. Поэтому наряду с побитовой мутацией часто применяется Гауссова мутация. Оператор гауссовой мутации работает только с генами из вещественных переменных. К значению каждой переменной в гене прибавляется некоторое случайное число, полученное с помощью нормального распределения с заданными параметрами. Параметры мутации могут задаваться как таким образом, чтобы мутация была незначительным изменением гена, так и для совершения больших скачков по пространству поиска. Приводятся доводы в пользу обоих вариантов, поэтому обычно для каждой задачи и тип оператора мутации и его конкретные параметры задаются отдельно.

Оператор перекреста обеспечивает обмен отдельными сегментами гена в процессе размножения. Чаще всего применяется одноточечный перекрест. Случайным образом выбирается точка на гене, по которой геном "разрезается".

Ген - потомок получает сегмент гена до точки разреза от одного родителя, а после точки разреза - от другого родителя. Аналогично вводятся двухточечный перекрест по двум точкам разреза и т.д.

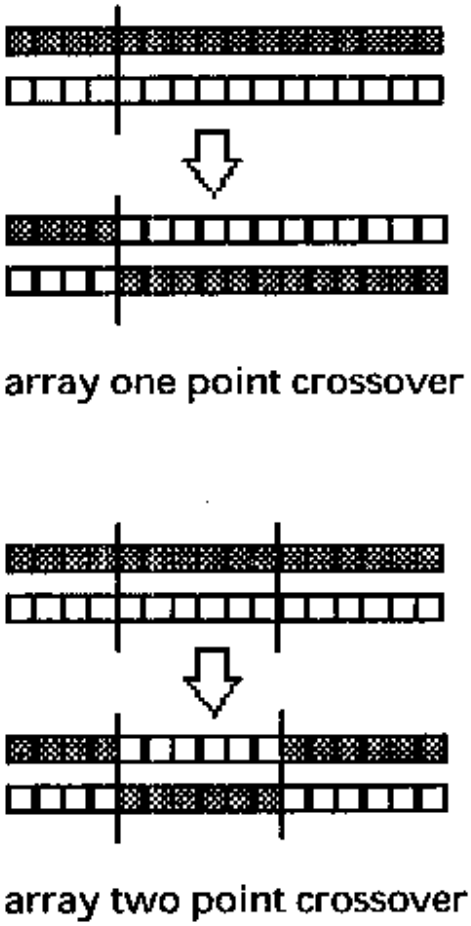

Рис.3 Оператор перекреста

В некоторых случаях, например, когда поиск осуществляется в ограниченной области, на параметры, т.е. на значения переменных в геноме накладываются условия. В таком случае определяется множество аллелей - т.е. возможных значений генов - параметров. Каждый раз при применении операторов мутации и перекреста проверяется, чтобы значение параметра не вышло за пределы множества аллелей.

А в качестве критерия завершения обычно задается ограничение по количеству шагов эволюции.

Пример применения ГА

Пусть мы хотим найти минимум несложной функции в области [-1000 1000]*[-1000 1000]:

$$y = |x1 + \sin(x1)| + |x2 + \sin(x2)|$$

У нас имеются 2 вещественные переменные - x1 и x2. Т.к. поиск осуществляется в области, то создадим множество аллелей - наложим ограничения на их значения. Каждый параметр будет геном. Составим из них новый геном - вещественный массив длины 2. Искомая функция будет целевой функцией, где x1- значение первого гена, а x2 - значение второго гена. Выберем подходящие операторы перекреста и мутации, функцию приспособленности, критерий

завершения. Теперь можно задать эти параметры библиотеке (программе).

Подробнее о ГП

Задачи генетического программирования во многом значительно сложнее задач, стоящих перед ГА. Объект оптимизации, программа на некотором языке, является главной причиной появившихся сложностей. Во-первых, целевая функция на множестве программ обычно намного сложнее, чем в ГА. Чтобы проверить работу программы ее необходимо протестировать, что увеличивает время выполнения одного шага эволюции. Программа в процессе эволюции может менять свою длину в отличие от генома фиксированной длины в ГА.

Представление программ в ГП

Чтобы определить представление объекта программы, необходимо выбрать язык программирования. Поскольку программы обычно небольшие, то стараются ограничиться минимально возможным набором функций, переменных и констант, которые вместе образуют алфавит кодирования программы. Операций +,-,*,/,OR,AND,XOR обычно достаточно для большинства небольших программ.

Для записи программ применяются три вида структур - деревья, списки и графы. Для удобства описания в дальнейшем будем некоторое представление индивида в ГП называть также как в ГА геномом.

Для представлений-деревьев в каждый узел записывается либо переменная, либо константа, либо символ функции. На функции накладывается ограничение по количеству параметров - обычно не более двух. Каждый узел или поддерево - подпрограмма. Если записана переменная или константа, то их значение является значением подпрограммы - узла. Если в узле записан символ функции, то узел должен иметь потомков - подпрограммы, результаты которых будут аргументами этой функции. Тогда при вычислении такого узла вначале вычисляются подпрограммы-потомки, их значения подставляются в функцию как аргументы, и результирующее значение функции становится значением этого узла-подпрограммы. В этом представлении наиболее просто решается вопрос с памятью, с которой работает программа. На каждый узел с функцией необходимы максимум три переменные: по одной на параметр и одну на результат. В других узлах - по одной на результат.

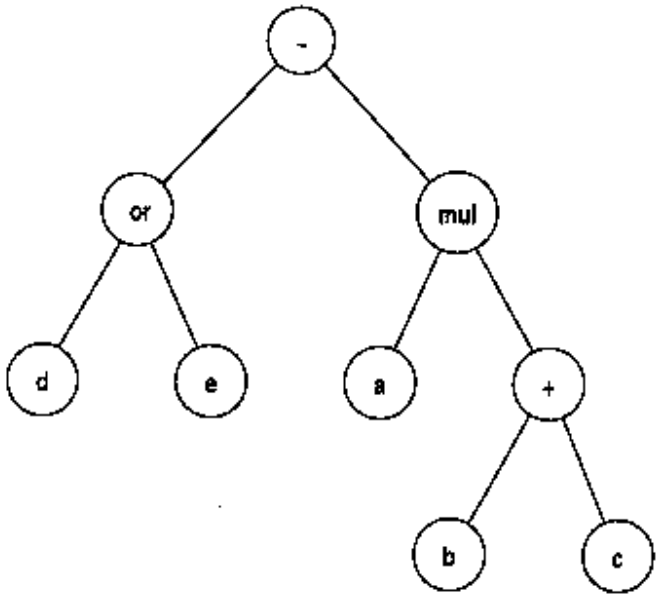

Рис.5 Представление программы в виде дерева

Представление программы списком похоже на стандартную запись программы. Это просто цепочка инструкций, выполняемых последовательно с первой по последнюю. Каждая функция записывается в отдельном элементе списка. Функции производят вычисления над переменными-регистрами, которые задаются отдельно. Начальные значения в

регистрах также задаются отдельно в начале работы программы.

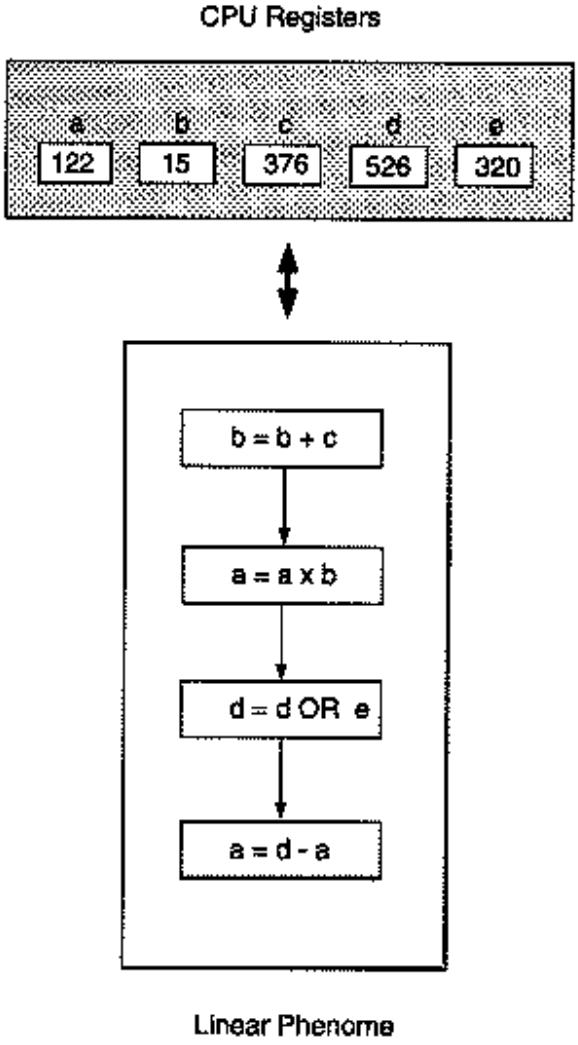

Рис.6 Представление программы списком

Графы для представления программы направленные и могут иметь циклы. Будем обозначать ребра с направлением стрелками (исходящие/входящие). В каждый узел графа записывается либо символ функции, либо символ переменной или константа. Программа граф использует два вида памяти - индексированную и стек. В индексированной памяти хранятся значения переменных, т.е. переменная - это индекс ячейки памяти. Стек используется для передачи данных между инструкциями. Два узла в графе особые - начальный узел и конечный. Переход к следующей инструкции осуществляется по стрелкам. Если у узла более чем одна исходящая стрелка - то выбор между ними происходит особым образом по значению в стеке и/или индексированной памяти. Если передается управление узлу с символом переменной/константой, то происходит запись их значения в стек. Если выполняется узел с функцией, то функция извлекает необходимое число параметров из стека, проводит над ними операцию, и записывает результирующие значение в стек.

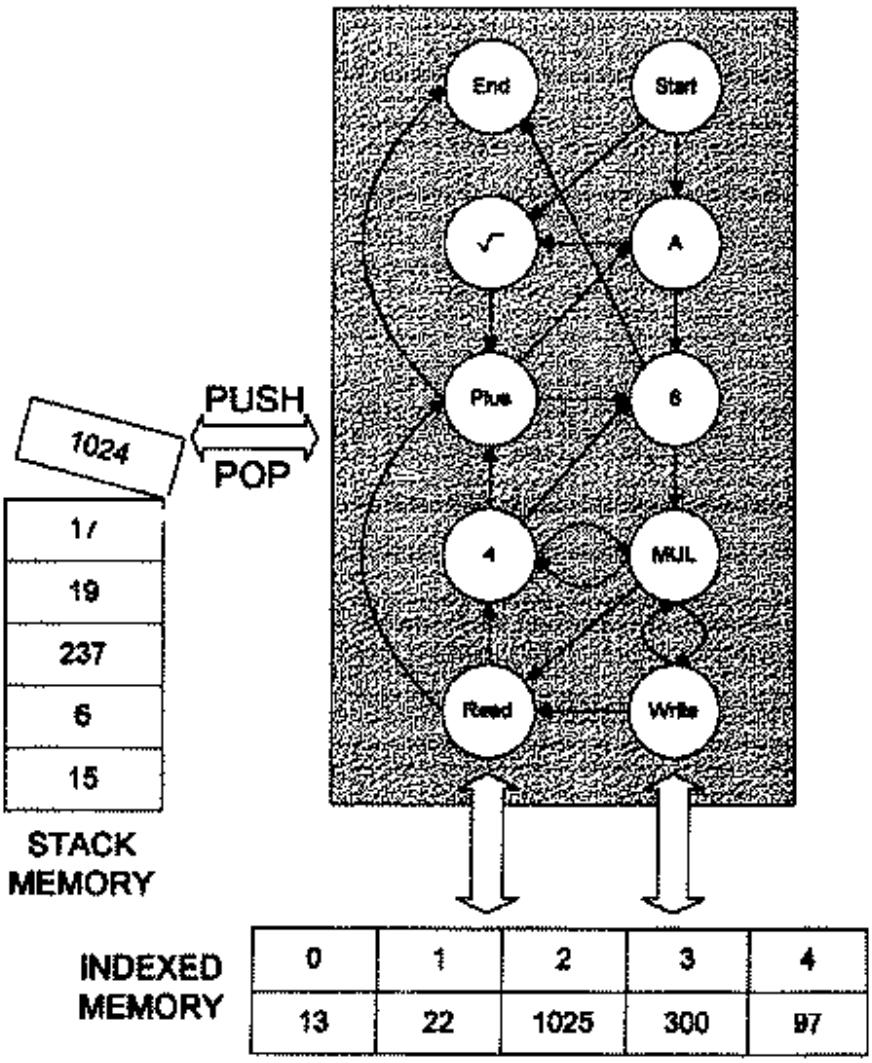

Рис.7 Представление программы графом

Инициализация популяции

В начале работы ГП необходимо как-то инициализировать популяцию программ. Общим для всех представлений является ограничение на размер - глубину дерева, длину списка, количество узлов в графе.

Для представления деревьев применяется два метода. Первый - метод Роста деревьев. Дерево начинает строиться с корни, запись в узел случайным образом выбирается из множества, включающего всевозможные функции, переменные и константы. Если выбирается функция, то процесс рекурсивно продолжается для ее потомков. Второй метода - Полный. Строится дерево аналогично методу роста, но записи выбираются только из множества функций, пока дерево не достигнет заданной глубины. После чего записи выбираются только из множества переменных и констант. Чтобы получить набор деревьев всевозможной длины, применяется комбинированный метод. Популяция делится на части, каждой части ставится своя максимальная глубина. Половина деревьев из каждой группы строится по методу роста, другая - по полному методу.

Список инициализируется похоже. Каждому элементу списка ставится в соответствие случайным образом выбранная функция и случайным образом выбранные параметры - переменные. Начальное значение переменных случайным

образом инициализируется.

Построение графа отличается от построения списка тем, что для каждого узла необходимо выбирать запись из функций, констант и переменных, а не только функций, и для каждого узла случайным образом строятся исходящие стрелки.

Оператор перекреста

Оператор перекреста для представления - дерево осуществляет обмен поддеревьями. Случайным образом выбирается узел в одном дереве, в другом дереве, поддеревья отделяются в соответствующих узлах, и ставятся на место соответствующего поддерева в другом дереве.

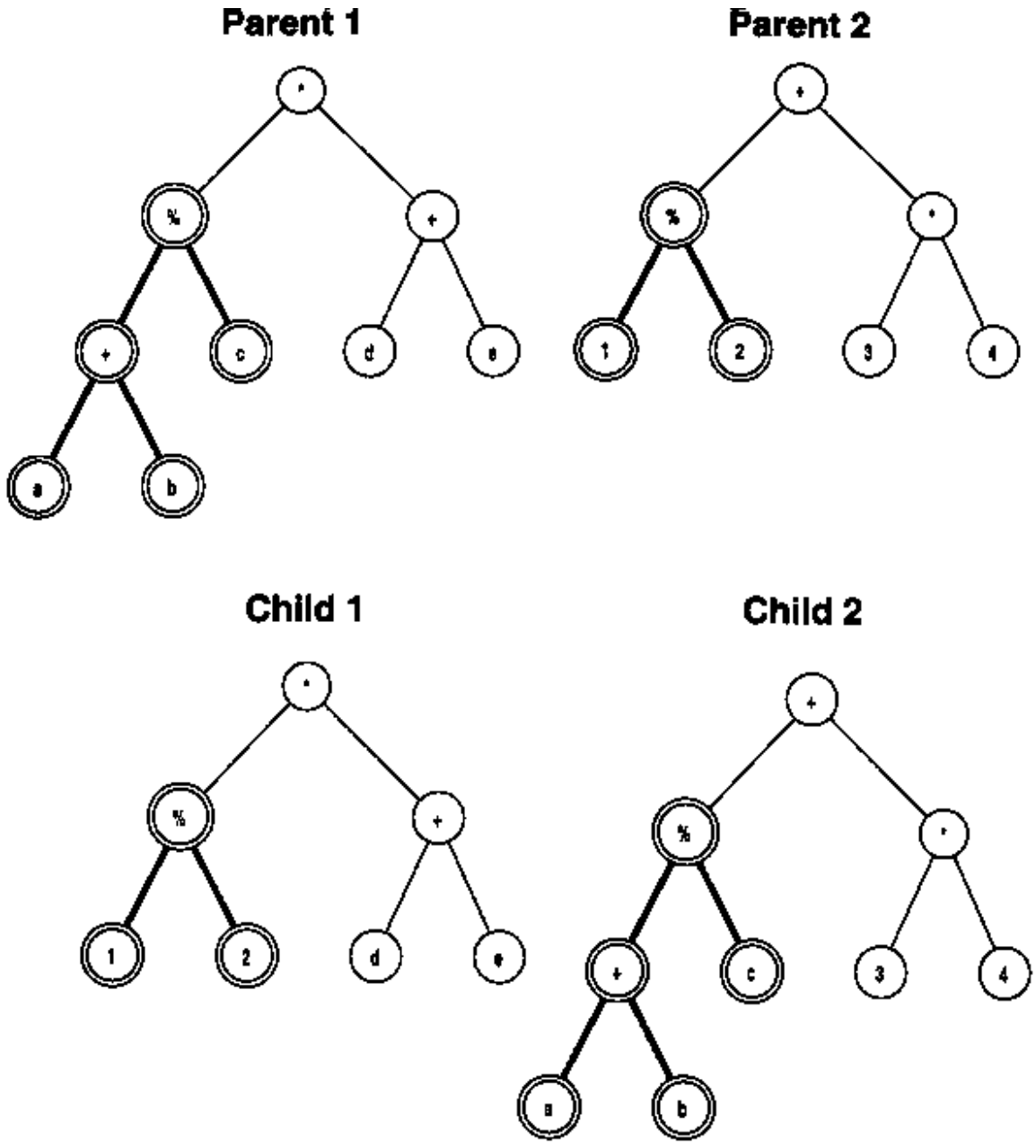

Рис.8 Перекрест для представления - дерева

Перекрест списочного представления аналогичен перекресту в ГА. Случайным образом выбираются точки в одном и другом списках, и списки обмениваются соответствующими сегментами. Однако в отличие от ГА в списке участки выбираются независимо и могут быть разных размеров. (Конечный сегмент одного списка может заместить начальный сегмент второго)

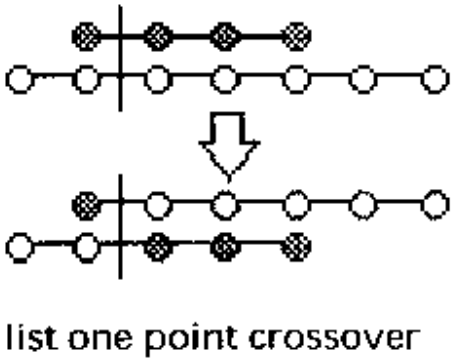

Рис. 9 Перекрест для списочного представления

Самый сложный перекрест - в графе. Вначале необходимо выделить набор узлов для обмена в каждом из графов. Разделить все связанные с ними ребра на два класса - внутренние (соединяющие узлы из набора) и внешние. Все узлы в наборе помечаются следующим образом - входы, если у них есть входящие внешние ребра, и выходы, если у них есть исходящие внешние ребра. Затем индивиды меняются наборами узлов вместе с внутренними ребрами. Все внешние ребра затем подсоединяются случайным образом к входам и выходам нового набора.

Мутация

В деревьях мутация осуществляется за счет замены случайно выбранного поддерева на новое, случайным образом сгенерированное поддерево. Т.е. выбирается узел на дереве, он уничтожается вместе со всеми своими потомками, и на его место ставится новое поддерево.

В списке могут происходить мутации следующих видов:

- Изменение одной переменной на другую
- Изменение значения константы
- Изменение функционального символа, т.е. функции

У графом существует множество возможных мутаций - мутации функций, переменных, констант, перемена направлений ребер и т.д.

Проблемы перекреста

Если измерить целевые значения новых особей после скрещивания, то выясняется, что в большинстве случаев оно либо значительно хуже, чем у родителей, либо примерно такое же. Перекрест часто оказывает разрушительное действие на индивида. Заключается оно в том, что перекрест разрушает хорошие блоки, разделяя их и заменяя одни сегменты на посторонние, совершенно чуждые данному блоку. Таким образом понижается приспособленность потомков. Главной причиной такого эффекта является негомологичность перекреста. Разница в длине между геномами разных индивидов только усугубляет эту проблему.

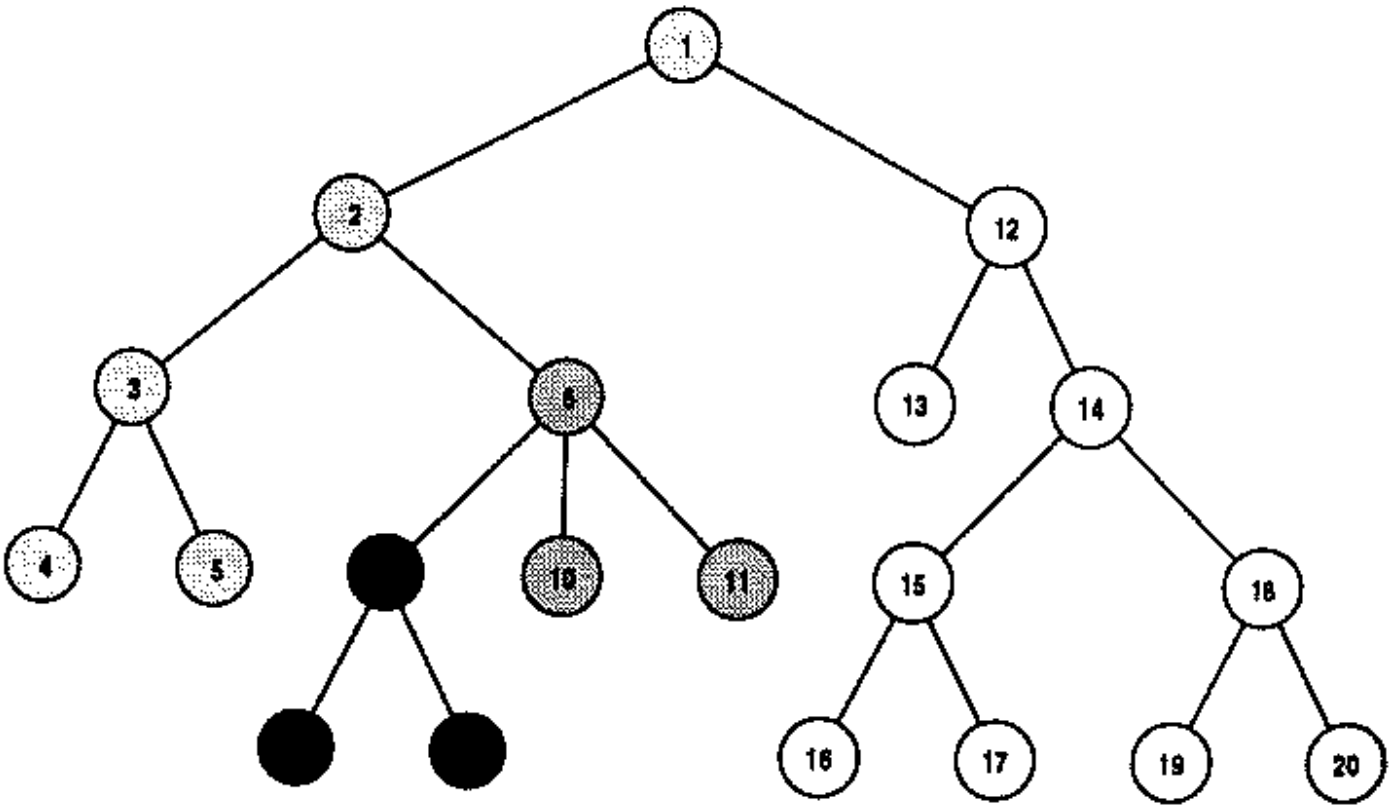

Рис. 10 Разрушительное воздействие перекреста

Проблему разрушительного перекреста пытаются решить с помощью усовершенствованных методов перекреста.

Самым простым из таких методов является метод Выводка (Bread). Этот метод как и сам ГП был позаимствован у природы. Обычно в природе у каждого животного рождаются намного больше детей, чем выживает. Большая часть погибает не достигнув зрелости (т.е. до размножения). Метод Выводка прямо копирует это факт: вместо 2 потомков пары родителей порождается целый выводок, из которого выживают только 2 наилучших индивида. Недостаток этого метода очевиден - процесс перекреста происходит много раз, как и оценка индивидов, и это может значительно замедлить весь алгоритм, особенное если размер выводка задается большим.

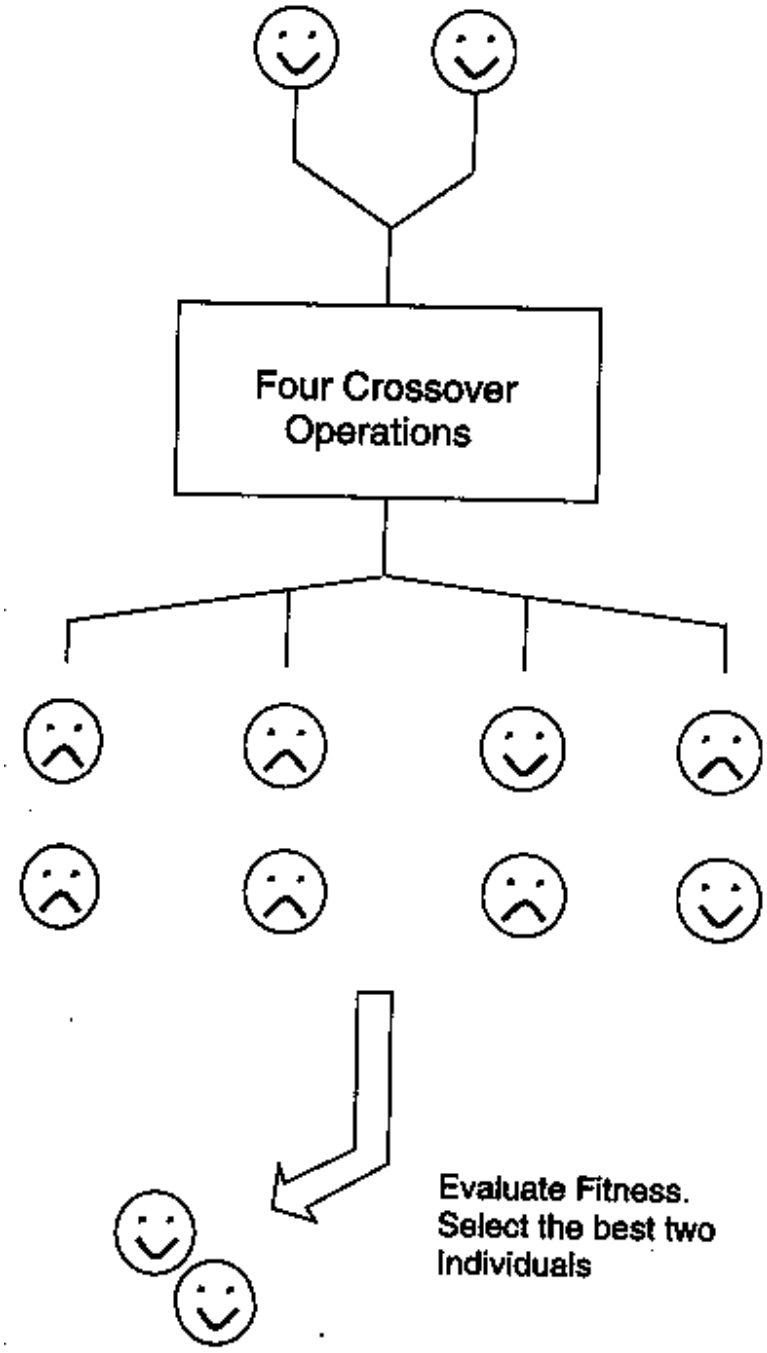

Рис. 11 Перекрест по схеме Bread

Более перспективным путем является Разумный(Intelligent) перекрест. Идея метода заключается в том, чтобы выделять потенциально хорошие блоки в каждом геноме, и не разрушать их. В этом случае требуется определять порядок работы инструкций, вводить всякие эвристики для оценки частей генома и т.д. Это очень сложная задача, которая пока в общем случае не имеет решения. Поэтому Разумный перекрест разрабатывается отдельно для каждого представления, для каждого набора функций и т.д.

Феномены - аналоги естественной эволюции

Большая часть исследователей отмечают, что во время работы ГП, начиная с некоторого момента, размер генома начинает быстро увеличиваться до максимального. При этом целевое значение такого генома практически не изменяется. Т.е. геномы очень разных размеров оцениваются одинаково. Это заставляет предположить, что часть кода этих программ - пустая, т.е. ничего не делающая.

Причины такого резкого роста становятся понятными, если в дополнение к целевому значению ввести понятие эффективного целевого значения. Каждый потомок данного индивида в дальнейшем будет подвергаться отбору. Если его целевое значение будет низким - то будет низкой его приспособленность и шансы на выживание. Поэтому под эффективным целевым значением можно понимать среднее значение целевого значения потомков, а под эффективной приспособленностью - показатель, характеризующий приспособляемость потомков.

Как можно повысить эффективную приспособленность? Очевидны два пути. Первый - это повышение приспособленности индивида. А второй - это уменьшение вероятности разрушающего перекреста. Перекрест будет разрушающим, если будут разделяться хорошие блоки, и будет влиять незначительно, если воздействию будут подвергаться тавтологические сегменты. Т.е. если в программу будут вставлены операторы подобные x=x*1, целевое значение практически не уменьшится (если длина программы не критична), а эффективная приспособленность возрастет. В начале работы ГП целевое значение в основном повышается за счет улучшения программы, а потом целевое значение стабилизируется, и начинают расти эффективные компоненты. Получается, что такие участки не хранят полезной информации, но повышают шансы индивида на выживание и участвуют в управлении перекрестом. По аналоги с биологическим эквивалентом такие участки называли интронами.

Очевидно, что для нас рост эффективных компонент не важен. Все время, которое ГП потратил на увеличение и развитие интронов для нас проходит впустую. С другой стороны, интроны помогают сохранять хорошие блоки для будущих поколений, что повышает шансы на получение еще более эффективных особей. Поэтому были попытки искусственного введения интронов в программу.

Очевидно, что Сверх-Разумный (Super-Intelligent) перекрест избавил нас от возникновения интронов. Но его создание принципиально невозможно, иначе бы можно было по блоку априори определить - будет он полезен в дальнейшей эволюции или нет. Поэтому много усилий сконцентрировано на нахождении комбинированных методов, сочетающих Разумный перекрест и допускающий интроны.

Возможные варианты использования ГП

Особи ГП - программы на некотором языке. ГП пытается строить программы из маленьких кирпичиков - операторов. Сейчас много экспериментируют с логическими и арифметическими конструкциями. Одним из очень перспективных направлений можно считать развитие нейросетей с помощью ГП. Нейросеть - это тоже программа, написанная на особом языке функций - нейронов. Методы обучения нейросети позволяют подбирать параметры целенаправленно, чтобы добиться от данной сети наилучших результатов. Одна из самых больших проблем - это выбор правильной топологии для нейросети. ГП может применяться вместе с обучением для изучения и нахождения хорошей топологии для сети-программы.

Заключение

Генетические методы получают все большее и большее распространение и применяются для самых разнообразных задач. Они проигрывают специализированным методам для многих приложений, но позволяют решать множество задач по одной и той же схеме. Если гипотеза Дарвина верна, то естественная эволюция добилась впечатляющих результатов по созданию приспособленных объектов. Одним из главных ее достижений стало создание самоприспособляющихся организмов, т.е. которые сами в течение своей жизни все больше приспосабливаются к жизни и оставляют знание о методах такой приспособленности потомкам путями, природой не предусмотренными (Наверное...). Поэтому кто знает, может быть однажды какая-нибудь из ГП-программ и породит что-то подобное, что поможет нам разгадать самую большую загадку во вселенной - загадку разума.

Ссылки:

1. "Genetic Programming: An Introduction" Banzhaf, Nordin, Keller, Francone
2. "Hitch-Hiker guide to Evolutionary Computation" Heitkotter
3. "GaLib documentation"
4. "An Introduction to Genetic Algorithms for Numerical Optimization" Charbonneau
5. "Genetic Algorithms" Louis
