

Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий

Реферат

Современные подходы использования генетических алгоритмов в области ИИ и ИС

(тема работы)

Интеллектуальные системы

(наименование дисциплины)

Работу выполнили:

Ернязов Т.Е.

подпись

Ф.И.О.

Преподаватель:

Сазанов А.М.

подпись

Ф.И.О.

Санкт-Петербург
2018

Предисловие

Среди множества проблем, которые возникают перед исследователями как в области теории, так и в многочисленных практических приложениях значительную долю составляют так называемые оптимизационные проблемы. С оптимизационной проблемой мы сталкиваемся каждый раз, когда возникает необходимость выбора из некоторого множества возможных решений наилучшего по определенным критериям и, как правило, удовлетворяющего заданным условиям и ограничениям.

Существуют классы оптимизационных задач, решение которых удается находить с помощью достаточно эффективных методов, вполне приемлемых по трудоемкости. Вместе с тем имеются и такие классы оптимизационных задач (так называемые NP-полные задачи), решение которых невозможно найти без полного перебора вариантов. В частности, к числу последних относятся многие разновидности задач многокритериальной оптимизации. Известно, что при большой размерности этих задач реализация перебора вариантов практически невозможна из-за чрезвычайно больших временных затрат.

В этой ситуации альтернативным походом к решению упомянутых задач является применение методов, базирующихся на методологии эволюционных вычислений. Эффективность различных методов в рамках эволюционного подхода подтверждается многочисленными данными, касающимися достигаемым реальным эффектом. При этом хотя объем вычислений может оказаться большим, но скорость, с которой он растет при увеличении размерности задачи, обычно меньше, чем у остальных известных методов. Отметим, что после того как компьютерные системы стали достаточно быстродействующими и недорогими, эволюционные методы превратились в важный инструмент поиска близких к оптимальным решений задач, которые до этого считались неразрешимыми.

Введение

В настоящее время оформилось и успешно развивается новое направление в теории и практике искусственного интеллекта – эволюционные вычисления. Этот термин обычно используется для общего описания алгоритмов поиска, оптимизации или обучения, основанных на некоторых формализованных принципах естественного эволюционного отбора. Особенности идей эволюции и самоорганизации заключаются в том, что они являются плодотворными и полезность их применения не только для биологических систем перманентно подтверждается. Эти идеи в настоящее время с успехом используются при разработке многих технических и, в особенности, программных систем.

Генетический алгоритм – метод решения как ограниченных, так и неограниченных задач оптимизации, основанный на естественном отборе. Генетический алгоритм неоднократно модифицирует отдельные решения. На протяжении последующих поколений объект «эволюционирует» в направлении оптимального решения. Генетический алгоритм применяется для решения различных задач оптимизации, которые не подходят для стандартных алгоритмов оптимизации, включая задачи, в которых объективная функция является прерывистой, недифференцируемой, стохастической или сильно нелинейной.

Генетический алгоритм обычно используется для решения задач смешанного целочисленного программирования. Генетический алгоритм отличается от классического алгоритма двумя основными характеристиками:

Классический алгоритм	Генетический алгоритм
Формирует одну точку на каждой итерации. Последовательность точек приближается к оптимальному решению	Создает популяцию точек на каждой итерации. Наилучшая точка подходит к оптимальному решению
Выбирает следующую точку в последовательности детерминированным вычислением	Выбор следующей популяции производится генератором случайных чисел

При реализации генетического алгоритма вводится вектор и целевая функция (функция приспособленности). Задача состоит в том, чтобы подобрать такой вектор, который бы соответствовал максимальному значению функции приспособленности. Суть проблемы состоит в том, что не всегда данную задачу можно эффективно решить традиционными методами оптимизации, или простым перебором возможных значений. Разрешение данных задач является главной областью применения генетических алгоритмов.

Простой генетический алгоритм

Основы теории генетических алгоритмов сформулированы Дж. Г.Холландом в основополагающей работе и в дальнейшем были развиты рядом других исследователей. Наиболее известной и часто цитируемой в настоящее время является монография Д.Голдберга, где систематически изложены основные результаты и области практического применения ГА.

ГА используют принципы и терминологию, заимствованные у биологической науки – генетики. В ГА каждая особь представляет потенциальное решение некоторой проблемы. В классическом ГА особь кодируется строкой двоичных символов – хромосомой, каждый бит которой называется геном. Множество особей – потенциальных решений составляет популяцию. Поиск оптимального или субоптимального решения проблемы выполняется в процессе эволюции популяции, т.е. последовательного преобразования одного конечного множества решений в другое с помощью генетических операторов репродукции, кроссинговера и мутации. ЭВ используют механизмы естественной эволюции, основанные на следующих принципах:

1. Первый принцип основан на концепции выживания сильнейших и естественного отбора по Дарвину, который был сформулирован им в 1859 году в книге "Происхождение видов путем естественного отбора". Согласно Дарвину особи, которые лучше способны решать задачи в своей среде, чаще выживают и чаще размножаются (репродуцируют). В генетических алгоритмах каждая особь представляет собой решение некоторой проблемы. По аналогии с этим принципом особи с лучшими значениями целевой (фитнесс) функции имеют большие шансы выжить и репродуцировать. Формализацию этого принципа, как мы увидим далее, реализует оператор репродукции.
2. Второй принцип обусловлен тем фактом, что хромосома потомка состоит из частей, полученных из хромосом родителей. Этот принцип был открыт в 1865 году

Г.Менделем. Его формализация дает основу для оператора скрещивания (кроссинговера).

3. Третий принцип основан на концепции мутации, открытой в 1900 году де Вре. Первоначально этот термин использовался для описания существенных (резких) изменений свойств потомков и приобретение ими свойств, отсутствующих у родителей. По аналогии с этим принципом генетические алгоритмы используют подобный механизм для резкого изменения свойств потомков и, тем самым, повышают разнообразие (изменчивость) особей в популяции (множестве решений).

Эти три принципа составляют ядро ЭВ. Используя их, популяция (множество решений данной проблемы) эволюционирует от поколения к поколению.

Эволюцию искусственной популяции – поиск множества решений некоторой проблемы, формально можно описать в виде алгоритма, который представлен на рис. 1.

ГА получает множество параметров оптимизационной проблемы и кодирует их последовательностями конечной длины в некотором конечном алфавите (в простейшем случае в двоичном алфавите "0" и "1").

Предварительно простой ГА случайным образом генерирует начальную популяцию хромосом (стрингов). Затем алгоритм генерирует следующее поколение (популяцию) с помощью трех следующих основных генетических операторов:

1. оператора репродукции (ОР);
2. оператора скрещивания (кроссинговера, ОК);
3. оператора мутации (ОМ).

Генетические алгоритмы – это не просто случайный поиск, они эффективно используют информацию, накопленную в процессе эволюции.

В процессе поиска решения необходимо соблюдать баланс между "эксплуатацией" полученных на текущий момент лучших решений и расширением пространства поиска. Различные методы поиска решают эту проблему по-разному.

Например, градиентные методы практически основаны только на использовании лучших текущих решений, что повышает скорость сходимости с одной стороны, но порождает проблему локальных экстремумов с другой. В полярном подходе случайные методы поиска используют все пространство поиска, но имеют низкую скорость сходимости. В ГА предпринята попытка объединить преимущества этих двух противоположных подходов. При этом операторы репродукции и кроссинговера делают поиск направленным. Широту поиска обеспечивает то, что процесс ведется на множестве решений – популяции и используется оператор мутации.

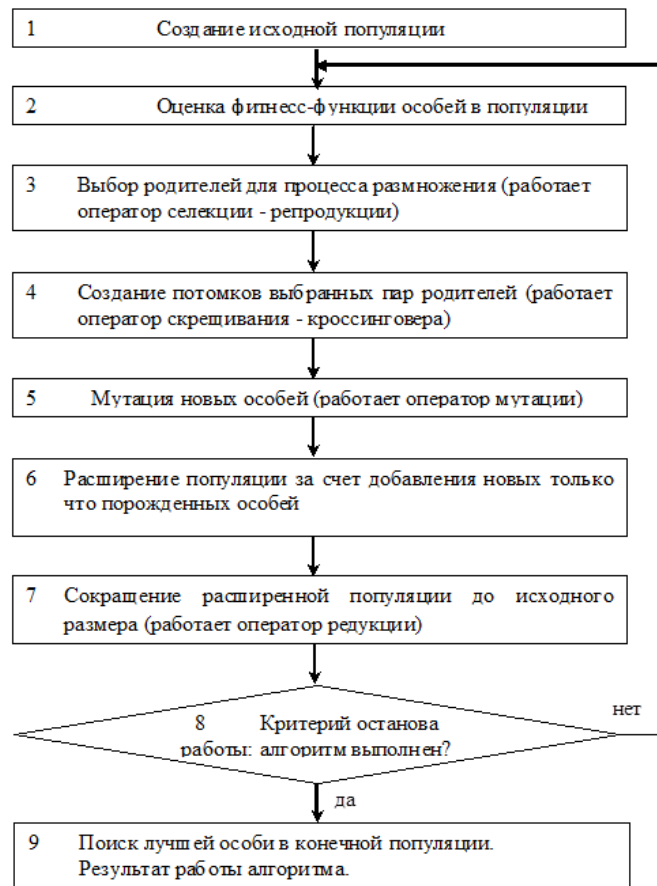


Рис. 1. Простой генетический алгоритм

В отличие от других методов оптимизации ГА оптимизируют различные области пространства решений одновременно и более приспособлены к нахождению новых областей с лучшими значениями целевой функции за счет объединения квазиоптимальных решений из разных популяций.

Упомянутые генетические операторы являются математической формализацией приведенных выше трех основополагающих принципов Ч.Дарвина, Г.Менделя и де Вре естественной эволюции. Каждый из функциональных блоков ГА на рис. 1. может быть реализован различными способами. Но сначала на простом примере мы рассмотрим основные моменты классического ГА.

Кодирование пространства поиска

В ГА часто используют следующие типы кодирования компонент пространства поиска:

- Бинарный, если признак сам по себе является бинарным;
- Численный в двоичной системе. Расширенный вариант бинарного, где используется фиксированное число бит. Самый простой в реализации, но имеет существенный недостаток;
- Кодирование кодом Грея. Избавляет от проблем предыдущего варианта, но добавляет накладные расходы на кодирование/декодирование;
- С небинарными операторами скрещивания и мутаций:

- Числа с плавающей запятой. Используются в том случае, когда масштаб изменения признака заранее не известен;
- Номинальные типы и более абстрактные сущности;
- Типы с автоподстройкой

Начальная популяция

Начальная популяция генерируется обычно случайно. Единственный критерий - достаточное разнообразие особей, чтобы популяция не свалилась в ближайший экстремум.

Оценка приспособленности

Оценка приспособленности часто проводится в две стадии. Первая - собственно

оценка: $F^k = \{f_1^k \dots f_n^k\}$, $f_i^k = W(p_i^k)$. Вторая - дополнительные преобразования. Например, ею может быть нормировка к

виду $F^{k'} = \{f_1^{k'} \dots f_n^{k'}\}$, $f_i^{k'} = (f_i^k - f_0) / (f_1 - f_0)$, где f_1 и f_0 , соответственно, лучший и худший показатели в текущей популяции.

Оператор отбора (селекции)

На этом этапе отбирается оптимальная популяция для дальнейшего размножения. Обычно берут определённое число лучших по приспособленности. Имеет смысл также отбрасывать "клонов", т.е. особей с одинаковым набором генов.

Оператор скрещивания

Размножение в разных алгоритмах определяется по-разному — оно, конечно, зависит от представления данных. Главное требование к размножению — чтобы потомок или потомки имели возможность унаследовать черты обоих родителей, «смешав» их каким-либо способом.

Почему особи для размножения обычно выбираются из всей популяции, а не из выживших на первом шаге элементов (хотя последний вариант тоже имеет право на существование)? Дело в том, что главный недостаток многих генетических алгоритмов — отсутствие разнообразия (diversity) в особях. Достаточно быстро выделяется один-единственный генотип, который представляет собой локальный максимум, а затем все элементы популяции проигрывают ему отбор, и вся популяция «забивается» копиями этой особи. Есть разные способы борьбы с таким нежелательным эффектом; один из них — выбор для размножения не самых приспособленных, но вообще всех особей. Однако такой подход вынуждает хранить всех существовавших ранее особей, что увеличивает вычислительную сложность задачи. Поэтому часто применяют методы отбора особей для скрещивания таким образом, чтобы «размножались» не только самые приспособленные, но и другие особи, обладающие плохой приспособленностью. При таком подходе для разнообразия генотипа возрастает роль мутаций.

Оператор мутаций

Оператор мутаций просто меняет произвольное число элементов в особи на другие произвольные. Фактически он является неким диссипативным элементом, с одной стороны вытягивающим из локальных экстремумов, с другой - приносящим новую информацию в популяцию.

- Инвертирует бит в случае бинарного признака.
- Изменяет на некоторую величину числовой признак. Причём, скорее на ближайший.
- Заменит на другой номинальный признак.

Критерии останова

- нахождение глобального, либо субоптимального решения;
- выходом на «плато»;
- исчерпание числа поколений, отпущенных на эволюцию;
- исчерпание времени, отпущенного на эволюцию;
- исчерпание заданного числа обращений к целевой функции.

Фитнесс-функция

Как видно из основной блок-схемы ГА на рис. 1, каждая особь популяции (потенциальное решение проблемы) оценивается путем вычисления значения фитнес-функции. Следует отметить, что в общем случае целевая функция (ЦФ) и фитнес-функция могут различаться. Целевая функция предназначена для оценки характеристик особи относительно конечной цели (например, экстремумов). Фитнесс-функция предназначена, прежде всего, для отбора особей для дальнейшей репродукции и здесь важны характеристики качества одной особи относительно других особей. После декодирования хромосомы, где выполняется преобразование "генотип \rightarrow фенотип" (например, двоичный код преобразуется в вещественное число), полученное значение далее используется в качестве аргумента для фитнес-функции. Далее для каждой особи популяции вычисляются значения фитнес-функции, которые ранжируют эти особи относительно друг друга в смысле перспективности получения из них хорошего решения.

Определение соответствующей фитнес-функции является решающим для корректной работы ГА. В частности, вид фитнес-функции может зависеть от накладываемых ограничений при решении оптимизационных задач. Отметим, что операторы кроссинговера и мутации не учитывают, попадают ли вновь построенные особи-потомки в область допустимых решений, которая обусловлена накладываемыми ограничениями.

В ГА используются четыре основных метода для учета накладываемых ограничений при решении оптимизационных задач. Вероятно, простейшим способом является метод отклонения (отбрасывания), где недопустимые хромосомы (не удовлетворяющие ограничениям) исключаются из дальнейшей эволюции. Второй метод основан на использовании процедуры восстановления, которая преобразует полученное недопустимое решение в допустимое. Другой альтернативой является применение проблемно-ориентированных генетических операторов, которые порождают только допустимые решения.

Рассмотренные методы не строят недопустимых решений. Но это не всегда дает хорошие результаты. Например, в том случае, когда оптимальные решения лежат на границе

допустимой области, указанные методы могут давать неоптимальные решения. Одним из возможных вариантов преодоления этой проблемы является выполнение процедуры восстановления только для некоторого подмножества решений (например, 10% особей).

Для решения оптимизационных задач со сложными ограничениями иногда позволяют вести поиск решения и в недопустимых областях. Реализуется это подход часто с помощью метода штрафных функций, что позволяет расширить пространство поиска решений. Следует отметить, что часто недопустимая точка, близкая к оптимальному решению, содержит больше полезной информации, чем допустимая точка, далекая от оптимума. С другой стороны, построение штрафных функций является достаточно сложной проблемой, которая сильно зависит от решаемой задачи. Обычно нет априорной информации о расстоянии до оптимальных точек, есть только расстояние до границы области допустимых решений. Поэтому, как правило, штрафные функции используют расстояние до границ допустимой области. Штрафы, основанные на нарушении отдельных ограничений, работают обычно не очень хорошо.

При этом штрафной терм должен изменяться не только в зависимости от степени нарушения ограничения, но и от номера поколения ГА. Наряду с нарушением ограничения, штрафной терм обычно содержит штрафные коэффициенты (по одному для каждого ограничения). На практике большую роль играют значения этих коэффициентов. Маленькие значения коэффициентов могут привести к недопустимым значениям решения, в то время как большие значения полностью отвергают недопустимые подпространства. В среднем абсолютные значения целевой и штрафной функции должны быть соизмеримы. При таком подходе параметры штрафной функции можно включить в параметры ГА, что позволяет разработать адаптивный метод, где значения коэффициентов регулируются в процессе поиска решения.

В целом на выбор (построение) фитнес-функции оказывают влияние следующие факторы:

- тип задачи – максимизация или минимизация;
- содержание шумов окружающей среды в фитнес-функции;
- возможность динамического изменения фитнес-функции в процессе решения задачи;
- объем допустимых вычислительных ресурсов – допускается ли использовать более точные методы и значительные ресурсы, или возможны только приближенные аппроксимации, не требующие больших ресурсов;
- насколько различные значения для особей должна давать фитнес-функция для облегчения отбора родительских особей;
- должна ли она содержать ограничения решаемой задачи;
- может ли она совмещать различные подцели (например, для многокритериальных задач).

В ГА фитнес-функция используется в виде черного ящика: для данной хромосомы она вычисляет значение, определяющее качество данной особи. Внутри она может быть реализована по-разному: в виде математической функции, программы моделирования (в том числе имитационного), нейронной сети, или даже экспертной оценки.

Для некоторых задач оценку значений фитнес-функции можно выполнять с помощью объектно-ориентированной имитационной модели. Взаимодействие такой модели с ГА показано на рис. 2.

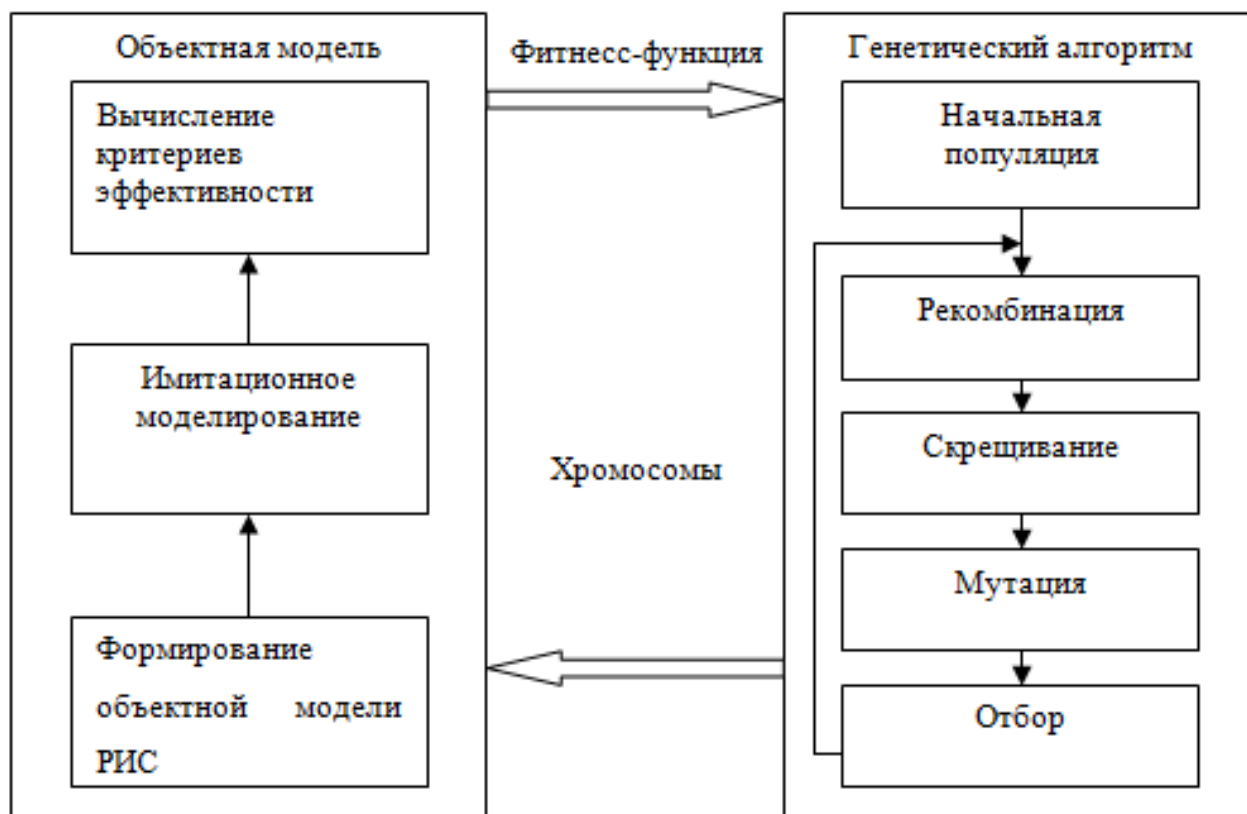


Рис. 2. Схема взаимодействия объектной модели с ГА.

Генетический алгоритм в ИС

В большинстве случаев, ранее, применялись не алгоритмические, а эвристические методы изучения. Одним из основных направлений исследований в этой области является применение ГА. При этом основным объектом исследования являются продукционные системы, то есть системы правил вида: ЕСЛИ <условие> ТО <действие> (которые используются в классических экспертных системах). Одной из самых серьезных проблем (и трудно формализуемых) при разработке экспертных систем является извлечение знаний у экспертов в виде системы правил (продукций). Поэтому была предпринята попытка автоматизировать этот процесс (или упростить и повысить эффективность уже имеющихся знаний в виде продукций). Здесь в качестве потенциального решения, особи популяции, рассматривается продукция (или система продукций).

Проблема состоит в том, чтобы создать систему, которая изучит концепции, то есть, определит решающие правила для всех положительных и отрицательных примеров. Мы можем оценивать и сравнивать потенциальные решения в терминах значений ошибок и сложности построенных правил. Система должна быть способна выполнить классификацию заранее неизвестных примеров, или выполнять (возможно, более чем одну) классификацию частично определенных описаний.

При машинном обучении, основанном на ГА, применяются два основных подхода:

1. Мичиганский
2. Питтсбургский

Исторически Дж. Холландом, основоположником ГА, первым по времени был разработан Мичиганский подход, но в настоящее время на практике более распространен Питтсбургский подход.

Питтсбургский подход

Здесь каждая особь в популяции представляет целый набор правил (а не отдельное правило, как в альтернативном подходе). Как обычно, особи конкурируют между собой, при этом слабые особи умирают, сильные живут и воспроизводятся. Здесь при реализации ГА часто используется пропорциональный отбор родителей, а операторы кроссинговера и мутации определяются над соответствующими структурами данных. Следует отметить, что Питтсбургский подход позволяет избежать тонкой проблемы оценки эффективности отдельного правила, для которой применяются эвристические методы.

При Питтсбургском подходе одна двоичная строка представляет (как и в классическом ГА) потенциальное решение – множество продуктов. В этом случае популяция, как обычно, содержит множество особей - потенциальных решений (систем продуктов). Далее для представленного метода кодирования продуктов необходимо определить генетические операторы кроссинговера и мутации. Поскольку система продуктов кодируется двоичной строкой, то возможно использование стандартных генетических операторов кроссинговера (например, одноточечного, двухточечного, однородного, которые рассмотрены в разделе 4) и классической мутации. Но для повышения эффективности при реализации данного метода в системе GABIL, в которой применяется данный подход, используются следующие модификации генетических операторов.

При Питтсбургском подходе проблема построения фитнес-функции решается гораздо проще, чем в Мичиганском, поскольку здесь оценивается вся система продуктов в целом.

Мичиганский подход

Здесь системы классификации используют структуру, в которой популяция правил закодирована в строки битов и развивается и совершенствуется на основе меняющихся входных данных, поступающих из внешней среды. Система "обучается" на представленных входных данных по методу обучения с учителем, где для каждого набора входных данных известны правильные значения выходов. Правила в системе классификации формируют популяцию из особей, развивающихся во времени. Система классификации, состоит из следующих компонентов:

- датчик и исполнительный элемент;
- система обмена сообщениями (входные, выходные и внутренние списки сообщения);
- система правил (популяция классификаторов);
- система оценки эффективности и отбора правил ("бригадный алгоритм");
- генетический алгоритм (репродукция классификаторов).

Среда (внешнее окружение системы классификации) посылает сообщение, которое принимается датчиками системы классификации и помещается во входной список сообщений. Датчики декодируют сообщение в одно или более

(декодированных) сообщений и размещают его во внутренний список сообщений. Эти сообщения активизируют классификаторы. Наиболее сильные из активизированных классификаторов размещают сообщения в списке внутренних сообщений. Эти новые сообщения могут активизировать другие классификаторы, или послать некоторые сообщения в выходной список сообщений. В последнем случае, исполнительные элементы системы классификации кодируют их в выходные сообщения, которые возвращаются во внешнюю среду. Среда оценивает действие системы посредством обратной связи с помощью "бригадного алгоритма", который модифицирует "силу" классификаторов.

Системы классификации XCS

В настоящее время Мичиганский подход к машинному обучению, который позволяет проводить обучение в on-line режиме, получил наибольшее развитие с 1995г. в XCS системах. Эти системы, взаимодействуют с неопределенной внешней средой и используют обратную связь в виде поощрения при правильном выборе решения. При этом они стремятся научиться точно прогнозировать размер будущих "премий".

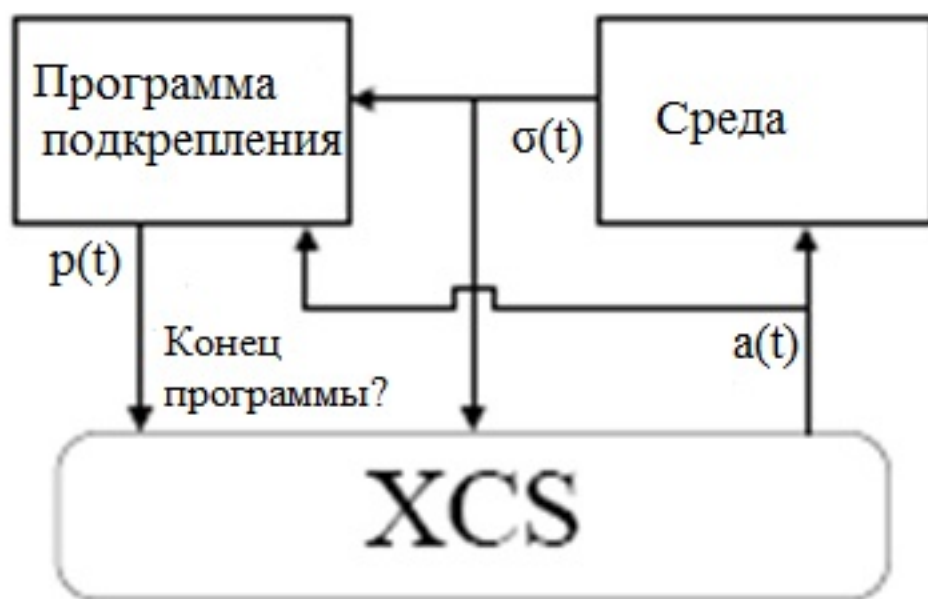


Рис. 3. Взаимодействие XCS систем с окружением и программой обучения

XCS система отличается от традиционных систем машинного обучения (LCS) использованием ГА и другим подходом к определению фитнес-функции. Во-первых, вычисление фитнес-функции для классификатора основано на прогнозировании точности величины поощрения, а не на размере самого поощрения. Во-вторых, ГА используется при выборе во множестве возможных действий классификатора, а не при обработке всей популяции. Наконец, в отличие от традиционных систем, XCS не использует список сообщений и поэтому удобна для обучения в среде, воздействие которой моделируется Марковским процессом.

В XCS системах, как и в других машинных системах обучения (LCS), решение проблемы представляется в виде популяции классификаторов. На каждом шаге XCS система принимает конкретное событие из окружающей среды в виде значений сигналов и основываясь на текущих знаниях предлагает решение для этого случая. В зависимости от проблемной ситуации (значений входов) и предложенного решения система определяет численное значение поощрения (премии), которое характеризует качество рекомендуемого решения. В отличие от обычных систем обучения (LCS), где каждому классификатору соответствовала "сила", в XCS системе она фактически заменяется тремя параметрами: 1) (payoff) production, 2) prediction error, 3) fitness. Роль этих параметров рассмотрена ниже.

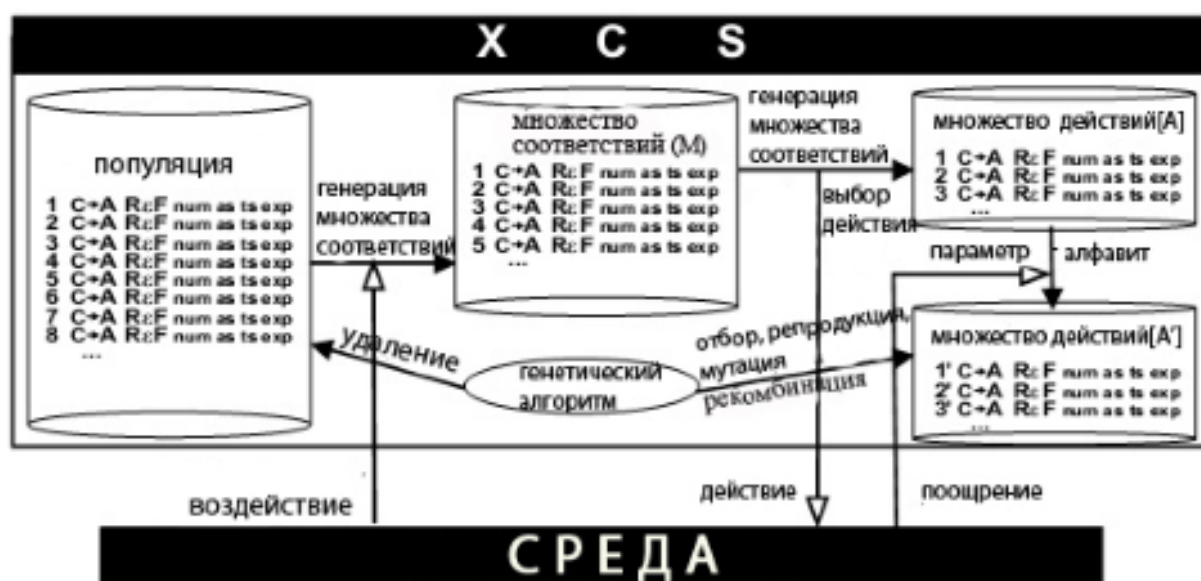


Рис. 4. Общая структура XCS

На рис. 4 представлена общая организация процесса обучения в XCS. Отметим, что в отличие от традиционных систем обучения, система XCS отбирает классификаторы для репродукции из текущего множества действий, определяемого текущей ситуацией S, а не из всей популяции. Кроме этого используются некоторые дополнительные параметры классификаторов и фитнес-функция основана на оценке точности поощрения прогноза. Сам по себе размер поощрения прогноза меньше влияет на процесс обучения. XCS имеет тенденцию в процессе эволюции генерировать наиболее общее решение проблемы, поскольку при репродукции предпочтение отдается классификатором, которые чаще активны. В целом процесс обучения в XCS направлен на достижение глобальной цели: в результате эволюции получить оптимальное решение проблемы, которое:

1. является полным, т.е. описывает все области пространства вход/выход;
2. не имеет пересечений – ни одна часть пространства не охватывается более одного раза;
3. является минимальным, т.е. используется минимальное число непересекающихся правил;
4. является максимально точным.

Преимущества и недостатки генетических алгоритмов

Концептуальная простота

Основным преимуществом ГА является их концептуальная простота. Рассмотрим снова блок-схему ГА. Основными шагами алгоритма являются: инициализация, оценка качества решения с помощью фитнес-функции, итеративное изменение популяции путем отбора особей и применения генетических операторов. Отметим, что здесь не важна высокая точность оценки качества потенциальных решений – необходимо знать прежде всего их ранг (номер позиции по качеству решения). Информация о значениях градиента целевой функции здесь не нужна. Через ряд итераций (поколений) популяция (множество потенциальных решений проблемы) может сойтись асимптотически к оптимальному решению. Эффективность ГА зависит от способа кодирования решения, используемых генетических операторов, включая отбор особей, и начальной инициализации популяции.

Широкая применимость

ГА могут быть использованы при решении любой проблемы, которая может быть сформулирована как задача оптимизации. Они требуют разработки (или выбора) структуры данных для представления потенциального решения, показателя качества для оценки потенциального решения и генетических операторов, порождающих новые решения из старых. Пространство возможных решений может быть разбито на различные области, которые могут включать недостижимые зоны и зоны возможных изменений. Способ представления решения обычно выбирается инженером-разработчиком на основании его интуиции и опыта. В этом смысле процедура выбора кодирования потенциального решения независима в отличие от обычных численных методов, где, как правило, допускаются только непрерывные значения из определенного диапазона. Представление решения должно позволять генетическим операторам при изменении решений сохранять поведенческую связь между родителями и потомками. Небольшие изменения в структуре данных родителя должны вести к небольшим изменениям свойств потомков и наоборот, большие изменения у родителей должны вызывать значительные изменения свойств потомков, что должно способствовать эффективному поиску решения в пространстве поиска. Множество возможных изменений может регулироваться с помощью эффективного размера шага изменения в пространстве поиска, который может регулироваться разработчиком (или даже адаптироваться автоматически). Такой гибкий подход позволяет использовать по сути одну и ту же процедуру при решении задач как численной, так и комбинаторной оптимизации (в частности, целочисленной оптимизации и т.п.).

Менее жесткие требования при решении реальных задач

Реальные задачи оптимизации часто:

1. накладывают нелинейные ограничения;
2. требуют платежной функции, не связанной с наименьшей квадратичной ошибкой;
3. включают наличие шумов или случайных выбросов, которые не позволяют применять классические методы оптимизации.

Целевые функции для реальных проблем часто мультимодальны и градиентные методы сходятся быстро к локальным экстремумам, которые могут давать неудовлетворительные

решения. Для простых задач, где поверхность отклика, например, является строго выпуклой, генетические алгоритмы проигрывают классическим по эффективности. Эксперименты показали, что для мультимодальных функций ГА дают лучшие результаты. В случае нелинейных ограничений классические методы даже при выпуклой поверхности могут давать некорректные результаты. Напротив, эволюционные методы могут непосредственно учитывать произвольные линейные и нелинейные ограничения.

Потенциальное использование априорных знаний и гибридизация с другими методами

При решении конкретной проблемы всегда целесообразно учесть в алгоритме проблемно-ориентированные априорные знания. Специализированные алгоритмы, учитывающие такую информацию (но имеющие ограниченную область применения), как правило, существенно превосходят по характеристикам неспециализированные методы. Эволюционные алгоритмы по своей структуре легче позволяют учитывать априорные знания. Это может быть выражено, например, в виде специальной структуры данных для представления решений или специальных проблемно-ориентированных генетических операторов. Часто такая информация включается в фитнес-функцию (например, физические или химические свойства вещества), что позволяет сфокусировать поиск решения в пространстве поиска.

Генетические алгоритмы могут комбинироваться с другими более традиционными методами. Известны работы, где на первом этапе оптимизации используются генетические алгоритмы совместно с градиентным методом, который применяется на заключительном этапе, когда уже найдена "зона интереса". Эти алгоритмы могут применяться совместно и параллельно. Отметим, что начальная популяция потенциальных решений может быть получена путем применения, например, жадных алгоритмов, а не эволюционных методов. Генетические алгоритмы часто используются для оптимизации и обучения искусственных нейронных сетей или нечетких продукционных систем. В этом случае часто удается преодолеть ограничения, связанные с традиционными подходами.

Параллелизм

Эволюция является высоко параллельным процессом, поскольку популяция состоит из множества особей, которые развиваются параллельно. Это позволяет расширить возможности применения эволюционных вычислений для решения все более сложных задач. Отметим, что основные вычислительные ресурсы в генетических алгоритмах используются при оценке значений фитнес-функций, которые для различных особей могут выполняться параллельно, а последовательной является только процедура отбора. Поэтому эволюционные методы естественно реализуются в многопроцессорных распределенных компьютерных системах, которые в настоящее время все шире применяются в различных областях науки и техники и даже на бытовом уровне в многоядерных процессорах. В действительности при решении некоторых задач на распределенных вычислительных системах время решения в идеале может быть обратно пропорционально числу используемых задач. Это создает благоприятные условия для решения сложных задач высокой размерности за разумное время с помощью генетических алгоритмов.

Устойчивость к динамическим изменениям

Традиционные методы оптимизации неустойчивы к динамическим изменениям окружающей среды и часто требуют полного рестарта при таких изменениях для получения адекватного решения. Напротив, эволюционные алгоритмы могут быть использованы для адаптации

потенциальных решений к изменившимся условиям. Полученная на момент изменения популяция дает базис для дальнейшего улучшения решений и в большинстве случаев нет необходимости проводить случайную реинициализацию.

Способность к самоорганизации

Большинство классических методов требуют начальной установки соответствующих параметров алгоритмов. Это также относится и к генетическим алгоритмам, которые зависят от множества параметров, таких как мощность популяции, вероятности кроссинговера и мутации, шаг мутации и т.п. Однако в эволюционных алгоритмах легче ввести самоадаптацию, когда в процессе поиска решения указанные параметры оптимизируются.

Решение проблем, для которых отсутствует опыт решений

Возможно, самым большим преимуществом генетических алгоритмов является их способность исследовать проблемы, для которых нет экспертов и соответствующего опыта решений. Следует отметить, что экспертные оценки достаточно часто используются при решении трудно формализуемых задач, но они иногда дают менее адекватные решения, чем автоматизированные методы. Существуют определенные проблемы с получением знаний у экспертов: они могут не согласиться на это, могут быть неквалифицированными, могут быть несовместимыми и просто ошибаться.

Исследования по искусственному интеллекту в настоящее время дали ряд интересных результатов, каждый из которых позволяет эффективно решать свой класс задач (например, хорошо играть в шахматы, или распознавать изображения символов и т.п.). Но большинство этих узких приложений требуют участия человека. Эти методы могут эффективно решать некоторые сложные проблемы, требующие высокого быстродействия, но они не могут конкурировать с человеческим интеллектом - "Они решают проблемы, но они не решают проблему как решать проблемы". Напротив, эволюционные алгоритмы дают метод решения проблемы, как решать проблемы при отсутствии экспертов (человеческого опыта).

Недостатки ГА

Естественно ГА не свободны от недостатков. К ним можно отнести прежде всего следующие. Конфигурация ГА для решения сложных реальных задач не очевидна. Для решения конкретной задачи необходимо выбрать или разработать представление (кодирование) потенциального решения. Существует также проблема определения фитнес-функции. Есть проблема выбора параметров ГА, таких как мощность популяции, вероятности генетических операторов и т.д. Нет эффективных критериев окончания работы алгоритма. ГА не могут использовать информацию о градиентах, что уменьшает их эффективность для классических задач. ГА не эффективны для гладких унимодальных (с одним экстремумом) функций. ГА не эффективны при поиске локальных экстремумов. ГА требуют достаточно больших вычислительных ресурсов. При решении мультимодальных задач бывают случаи преждевременной сходимости к локальным экстремумам и поэтому в общем случае не гарантируют нахождение глобального экстремума.

No Free Lunch теорема

Возникает естественный вопрос – существует ли некоторый лучший эволюционный алгоритм, который дает всегда лучшие результаты при решении всевозможных проблем? Например, можно ли выбрать генетические операторы и их параметры так, чтобы алгоритм давал лучшие результаты независимо от решаемой проблемы? К сожалению, ответ

отрицательный – такого лучшего эволюционного алгоритма не существует! Это следует из известной No Free Lunch (NFL) теоремы, которая доказана относительно недавно.

Эта теорема вызвала оживленную дискуссию у специалистов по эволюционным вычислениям и некоторое неприятие. Дело в том, что в семидесятых годах были предприняты значительные усилия по поиску лучших значений параметров и генетических операторов ГА. Исследовались различные генетические операторы, значения вероятностей выполнения кроссинговера и мутации, мощности популяции и т.д. Большинство этих исследований апробировалось на сложившихся в каждой проблемной области тестовых задачах. Но из NFL-теоремы следует, что полученные результаты справедливы только на использованных тестовых задачах, а не для произвольных задач. Все усилия найти лучший оператор кроссинговера или мутации, оптимальные значения их параметров при отсутствии ограничений для исследуемого класса задач не имеют смысла!

Каждому эволюционному алгоритму присуще некоторое представление, которое позволяет манипулировать с потенциальными решениями. NFL теорема утверждает, что не существует лучшего эволюционного алгоритма для решения всех проблем.

Для того чтобы разрабатываемый алгоритм решал поставленную задачу лучше, чем случайный поиск (который с точки зрения NFL теоремы является просто другим алгоритмом) необходимо в нем использовать (отразить) структуру (априорные знания) этой проблемы. Из этого следует, что такой алгоритм может не соответствовать структуре другой проблемы (и покажет для нее плохие результаты). Следует отметить, что недостаточно просто указать, что проблема имеет некоторую структуру - такая структура должна соответствовать разрабатываемому алгоритму. Более того, структура должна быть определена. Недостаточно, как это иногда бывает, сказать "Мы имеем дело с реальными проблемами, а не с всевозможными, поэтому NFL теорема не применима". Что значит структура реальной проблемы? Очевидно, что формальное описание такой структуры проблематично. Например, реальные проблемы нашего времени и столетней давности могут сильно отличаться. Следует отметить, что простое сужение области возможных проблем без идентификации соответствия между рассматриваемым множеством проблем и алгоритмом недостаточно для получения преимущества данного метода решения этих проблем по сравнению с другими.

NFL-теорема подтверждает, что разные алгоритмы имеют различную эффективность при решении разных задач. Например, классические методы оптимизации, как правило, более эффективны при решении линейных, квадратичных, строго выпуклых, унимодальных, разделяемых и других специальных классов проблем. С другой стороны, генетические алгоритмы часто успешно решают задачи там, где классические методы не работают – там, где целевые функции терпят разрывы, не дифференцируемы, мультимодальны (имеют много экстремумов), зашумлены и т.п. Обычно их эффективность и устойчивость выше там, где целевые функции имеют сложный (не стандартный) вид, что более характерно для решения реальных практических задач. Конечно, лучшим способом подтверждения эффективности алгоритма является доказательство его сходимости и оценки вычислительной сложности. Но, как правило, это возможно только в случае упрощенной постановки задачи. Другой альтернативой является проверка алгоритмов на тестовых задачах (benchmarks) данной проблемной области. К сожалению, в настоящее время не существует согласованного каталога таких задач для оценки старых или новых алгоритмов решения, хотя для многих типовых задач они уже сложились и широко используются.

Выводы

В настоящее время эволюционные вычисления представляют одну из развивающихся областей науки, исследования в которой ведут к постоянному повышению эффективности их использования и к появлению новых подходов к построению конкретных алгоритмов. Развитие теории решения комбинаторных задач оптимизации происходит непосредственно под влиянием запросов практики, в частности, необходимости производить оптимальный выбор решений на множестве вариантов.

Хотя модель эволюционного развития, применяемая в ГА, сильно упрощена по сравнению со своим природным аналогом, тем не менее, ГА является достаточно мощным средством и может с успехом применяться для широкого класса прикладных задач, включая те, которые трудно, а иногда и вовсе невозможно, решить другими методами. Однако, ГА, как и другие методы эволюционных вычислений, не гарантирует обнаружения глобального решения за полиномиальное время. ГА не гарантируют и того, что глобальное решение будет найдено, но они хороши для поиска "достаточно хорошего" решения задачи "достаточно быстро". Там, где задача может быть решена специальными методами, почти всегда такие методы будут эффективнее ГА и в быстродействии и в точности найденных решений. Одним из главных преимуществ ГА является то, что они могут применяться как в сложных задачах, где не существует никаких специальных методов, так и там, где хорошо работают существующие методики, эффективность которых можно повысить за счет сочетания с ГА.

Литература

- Генетические алгоритмы.
- |
 - Гладков Л.А., Курейчик В.В., Курейчик В.М
 - **Генетические алгоритмы.**
 - М.: Физматлит, 2006.
- Генетические алгоритмы.
- |
 - Курейчик В.В., Курейчик В.М., Родзин С.И.
 - **Теория эволюционных вычислений.**
 - М.: Физматлит, 2013.
- **Теория эволюционных вычислений.**
- |
 - Скобцов Ю.А.
 - **Основы эволюционных вычислений.**
 - Донецк: ДонНТУ, 2008.- 326с.
- **Genetic Algorithms + data structures=Evolution Programs.**
- |
 - Пилинский М., Рутковская Д., Рутковский Л.
 - **Нейронные сети, генетические алгоритмы и нечеткие системы.**
 - М: Горячая линия, 2004.- 452с.
- **Genetic Algorithms in Search, Optimization and Machine Learning.**
- |
 - Гладков Л.А., Курейчик В.В., Курейчик В.М.
 - **Генетические алгоритмы.**
 - М.:Физматлит.-2006.-319с.
- Современные модификации и обобщения генетических алгоритмов // Таврический вестник компьютерных наук и математики
- |
 - Скобцов Ю.А.
 - Основы эволюционных вычислений.
 - Донецк: ДонНТУ, 2008.- 326с.
- An introduction to genetic algorithms.
- |
 - Пилинский М., Рутковская Д., Рутковский Л.
 - Нейронные сети, генетические алгоритмы и нечеткие системы.
 - М:Горячая линия, 2004.- 452с
- The selfish gene.
- |
 - Carlos Cotta, Pablo Moscato
 - Hand book of memetic algorithms.
 - Springer- Verlag Heidelberg, 2012.
- Adaptation of genetic algorithm parametres based on fuzzy logic controllers// In F.Herera & J.Verdegay, editors. Genetic algorithms and soft computing, Physica-Verlag.
- |
 - Lin L., Mitsuo G., Runwei C.
 - Network models and optimization.
 - Springer-Verlag London Limited, 2008.-701p.