Введение

Многие задачи, возникающие в таких фундаментальных науках как физика, химия, молекулярная биология, а также во многих прикладных науках, сводятся к задачам непрерывной глобальной оптимизации.

Отличительной особенностью таких задач часто является нелинейность, недифферинцируемость, многоэкстремальность (мультимодальность), овражность (когда у функции в одном направлении нет изменений, а в другом есть), отсутствие аналитического описания (плохая формализованность), высокая высчислительная сложность, большая размерность пространства поиска, сложная область допустимых значений и т.д.

С самой общей точки зрения указанные особенности объясняют отсутствие универсального алгоритма их решения. Число таких алгоритмов оптимизации увеличивается, растёт количество их параллельных версий, ориентированных на различные классы параллельных вычислительных систем.

Для решения такого класса задач в 80-ых годах XX-го века интенсивно стали развиваться стохастические поисковые алгоритмы (в некоторых публикациях их также называют поведенческими, интеллектуальными, метаэвристическими, вдохновлёнными – или инспирированными – живой природой, роевыми, многоагентными).

Наиболее широко распространено понятие “популяционный алгоритм”. Популяционные алгоритмы предполагают одновременную обработку нескольких вариантов решения задачи оптимизации и представляют собой альтернативу классическим траекторным алгоритмам, в которых в области поиска эволюционирует только один кандидат на решение.

Задачи глобальной оптимизации можно разделить на два класса: детерминированные и стохастические. В первом случае оптимизируемая функция и функции, ограничивающие область решений задачи, являются детерминированными (точно определёнными), во втором случае одна или несколько указанных функций содержат случайные параметры.

Среди задач глобальной оптимизации выделяют статические, т.е. стационарные, и динамические. В статических задачах оптимизируемая функция и область её допустимых значений не меняются во времени, т.е. остаются неизменными положения в области поиска её локальных и глобальных экстремумов. В динамических задачах положения экстремумов могут меняться во времени и задача состоит в отслеживании движении экстремума.

Таким образом предметом рассмотрения являются детерминированные статические задачи глобальной безусловной (без ограничений на варьируемый параметр) и условной (с ограничением на это значение) непрерывной оптимизации.

По способу определения направления движения к экстремуму алгоритмы поисковой оптимизации делят на алгоритмы детерминированного (или регулярного) поиска (алгоритм градиентного спуска) и алгоритмы стохастического или случайного поиска.

Алгоритмы поисковой оптимизации разделяют ещё на два следующих класса: использующие как пробные, так и рабочие шаги алгоритма; алгоритмы, в которых эти шаги совмещены.

Все популяционные алгоритмы являются эвристическими, то есть для них сходимость к глобальному решению не доказано, но экспериментально установлено, что они дают достаточно хороший результат достаточно быстро.

В качестве общего названия для элементов популяции будем использовать термин агент или особь. Также возможны другие вариации (частица, индивид).

Общая схема всех популяционных алгоритмов включает следующую последовательность шагов:

1. Инициализация популяции. В области тем или иным образом создаётся некоторое число начальных приближений к искомому решению задачи, т.е. происходит процесс инициализации популяции агентов.

2. Миграция агентов. С помощью некоторого набора миграционных операторов, специфических для каждого популяционного алгоритма, агенты перемещаются в области поиска, чтобы приблизиться к искомому значению оптимизируемой функции.

3. Завершение поиска. Осуществляется проверка выполнения условий окончания итераций: если они выполнены – завершаем вычисления, принимаем лучшие значения агентов за приближённое решение задачи, иначе – переход к шагу 2.

При инициализации популяции можно использовать детерминированные или случайные алгоритмы. Формирование начальной популяции, агенты которой находятся вблизи глобального экстремума, может существенно сократить время решения задачи. Обычно априорная информация о местоположении экстремума отсутствует, поэтому агентов начальной популяции распределяют равномерно по всей области поиска.

Миграцию агентов, например, в генетическом алгоритме (ГА) реализуют генетические операторы (кроссовер, мутация).

В качестве условий окончания поиска, как правило, используется достижение заданного числа итераций или поколений. Также используется условие стагнации (проверка на изменения значений целевой функции: если изменений значения целевой функции практически нет, то останавливаем алгоритм).

Из представленной общей схемы следует, что популяционные алгоритмы обладают ярко выраженной модульной структурой, что позволяет легко получать большое число вариантов любого из алгоритмов путём варьирования и комбинации правил инициализации, миграции и завершения поиска.

Агенты популяции обладают следующим рядом свойств:

1. Автономность. Агенты движутся в пространстве хотя бы частично независимо друг от друга.

2. Стохастичность. Процесс миграции агентов всегда содержит случайную компоненту.

3. Ограниченность представления. Каждый агент обладает информацией только об исследуемой им части области поиска и возможных об окружении некоторых соседних агентов.

4. Децентрализация. Отсутствие агентов, управляющих процессом поиска.

5. Коммуникабельность.

Агенты тем или иным способом могут обмениваться информацией о топологии (ландшафте) оптимизируемой функции.

Даже если стратегия поведения каждого агента достаточно проста, указанные свойства обеспечивают формирование роевого интеллекта, определяющие организацию и сложное поведение популяции в целом.

Одной из особенностей популяционных алгоритмов является то, что в подавляющем большинстве случаев они имеют аналогии в человеческом обществе, живой и неживой природе: алгоритмы эволюции разума, колонии муравьёв, медоносных пчёл, сорной травы, гравитационного взаимодействия и т.д.

Популяционные алгоритмы в сравнении с классическими неоспоримые преимущества: прежде всего в задачах высокой размерности мультимодальных (мультиэкстремальных) и плохо формализуемых задачах. Важно также, что популяционные алгоритмы позволяют эффективнее классических искать субоптимальное решение (приблизительное).

Популяционные алгоритмы практически наверняка проиграют точному при решении задачи небольшой размерности. В случае гладкой мультимодальной функции популяционные алгоритмы менее эффективны, чем, например, градиентный спуск. К недостаткам также можно отнести зависимость от значений свободных параметров.

Важнейшим понятием популяционного алгоритма является fitness-функция (целевая функция, функция пригодности, функция приспособленности). Она используется для оценки качества агентов популяции. Часто, но не всегда fitness-функция совпадает с оптимизируемой функцией.

С использоваем fitness-функции связана суть популяционных алгоритмов, заключающаяся в обеспечении более высокой средней приспособленности популяции агентов данного поколения по сравнению с приспособленностью предыдущего поколения.

Одна из основных проблем конструирования популяционных алгоритмов – обеспечение баланса между интенсивностью (скоростью сходимости) и широтой поиска, т.е. диверсификацией.

Первые поколения популяционного алгоритма занимаются исследованием области поиска, агенты последних поколений заняты уточнением исключительно найденных решений. Баланса между интенсивностью и широтой можно добиться, используя механизмы адаптации и самоадаптации.

Основными критериями эффективности популяционных алгоритмов являются надёжность, т.е. вероятность локализации глобального экстремума; а также скорость сходимости, т.е. оценка матожидания необходимого числа испытаний.

Классификация популяционных алгоритмов:

1. ЭА (эволюционные алгоритмы), в том числе ГА (генетические алгоритмы).

2. Популяционные алгоритмы, вдохновлённые живой природой.

3. Алгоритмы, вдохновлённые неживой природой.

4. Алгоритмы, инспирированные человеческим обществом.

5. Прочие алгоритмы.

Постановка и классификация алгоритмов решения детерминированной задачи поисковой оптимизации

Рассмотрим детерменированную задачу оптимизации. Минимизировать функцию:

min f(x) = f(x*) = f*
x belongs to D lies in $R^{|x|}$

x = (x_1, ... $x_{|X|}$)
|x| — размерность по модулю варьируемых параметров
R — |x|-мерное арифметическое пространство
f(x) — целевая функция (критерий оптимальности)
x*, f* — искомое оптимальное решение и значение целевой функции
D = {x | E(x) = 0, G(x) >= 0} — область поиска (множество с ограничениями в виде равенств или неравенств)

Детерминированность поставленной задачи означает, что целевая и ограничивающая функции E(x), G(x) не содержат случайных параметров.

Целевую функцию f(x), где x belongs to [a; b] называется унимодальной, если в её области определения существует такая x*, что на полуинтервале [a; x*) f(x) убывает, а на (x*; b] f(x) возрастает, при этом x* может располагаться как внутри отрезка [a; b], так и являться одним из его концов.

Непрерывную в своей области определения (на отрезке [a; b]) одномерную целевую функцию f(x) называют выпуклой вниз, если для любых x_1, x_2 belongs to [a; b] => f(Lx_1 + (1 – L)x_2) <= Lf(x_1) + (1 - L)f(x_2)

Определение выпуклой целевой функции не требует её унимодальности. Здесь x_1 и x_2 не равны, L belongs to [0; 1].

Если заменить все нестрогие неравенства на строгие, то получим определение строго выпуклой функции. Строго выпуклая функция является унимодальной.

Целевую функцию, имеющую в своей области определения несколько локальных минимумов называют многоэкстремальной или мультимодальной.

Целевую функцию называют овражной к своей области определения, если в этой области имеют место слабые изменения первых частных производных функции по одним направлениям и значительные изменения по другим.

Если целевая функция представляет собой сумму нескольких функций, каждая из которых зависит от только от одной компоненты вектора x, то такую функцию называют сепарабельной.

Функцию называют позиномиальной, если она представляет собой сумму произведений $c_i x^a \ldots x^a$, где c_i, $\mathbf{a}$ — любые действительные числа.

Однопараметрические функции:

- унимодальные, мультимодальные;
- выпуклые, строго выпуклые.

Многопараметрические:

- унимодальные, мультимодальные;
- выпуклые, строго выпуклые;
- овражные;
- сепарабельные;
- позиномиальные.

Классификация задач оптимизации

Рассмотрим основные признаки классификации оптимизационных задач:

1. Вид целевой и ограничивающих функций. Если целевая функция линейная, а множество допустимых значений D — выпуклый многогранник, то задачу называют задачей линейного программирования. Если f(x) — отношение двух линейных функций, то задача носит название дробно-линейного программирования. Если область допустимых значений D определяется только ограничениями типа неравенств, а целевая и ограничивающая функции являются сепарабельными, то такую задачу называют задачей сепарабельного программирования. Если целевая и ограничивающая функции (E(x), G(x)) являются позиномами — задача геометрического программирования. Если целевая функция является выпуклой, то и задача называется задачей выпуклого программирования. При квадратичной целевой функции и выпуклом множестве D задача называется задачей квадратичного программирования. Конечное множество D порождает задачу дискретного программирования, а множество D как множество целых чисел — задачу целочисленного программирования. В общем случае задачу минимизации называют задачей нелинейного программирования. Часто задачи выпуклого программирования также относят к задачам нелинейного программирования.

2. Наличие или отсутствие ограничений. Если ограничение на вектор x отсутствует ($D = R^{|x|}$), то задачу называют оптимизацией без ограничений или задачей безусловной оптимизации. При наличии ограничений на вектор x задачу называют задачей с ограничениями или условной оптимизацией (D lies in $R^{|x|}$).
3. Характер ограничений. D = {x | G(x) >= 0}, то и задача носит название задачи с ограничениями типа неравенств. Если D = {x | E(x) = 0}, то задача носит название задачи с ограничениями типа равенств. Если D = {x | E(x) = 0, G(x) >= 0}, то задача носит название задачи с ограничениями общего вида.
4. Размерность вектора x. Если размерность вектора x равна 1, то задача называется задачей однопараметрической оптимизации. Если размерность вектора x больше 1, то задача называется задачей многопараметрической оптимизации.
5. Число точек минимума. Если в области допустимых значений существует только один экстремум, то задачу называют одноэкстремальной. Если более одного — многоэкстремальной.
6. Характер искомого решения. Если отыскивается любой локальный минимум в области D, то задачу называют задачей локальной оптимизации. Если целью является поиск глобального минимума функции, то и задача называется задачей глобальной оптимизации.

Классификация алгоритмов оптимизации

Испытанием называют операцию однократного вычисления значений целевой функции f(x) и ограничивающих функций, представленных в виде G(x) и E(x), а также, быть может, производных указанных функций в некоторой точке x из области допустимых значений. Особенностями задач оптимизации, представляющих практический интерес, является тот факт, что каждое испытание может требовать больших затрат компьютерного времени, поэтому одним из основных требований, предъявляемых к алгоритмам оптимизации, является решение задачи оптимизации при наименьшем числе испытаний. Приближённое решение задачи или кандидата на решение будем называть агентом и обозначать как S_i i belongs to [1; |S|], |S|>=1, где S — используемое множество агентов (популяция агентов). Будем рассматривать итерационные алгоритмы, которые наиболее распространены в вычислительной практике. Рассмотрим общую схему итерационного алгоритма:

1. Инициализируем алгоритм. Задаём начальное значение счётчика числа итераций t=0, начальное положение агента $x_i(0)$, а также значение свободных параметров.
2. Применяем поисковые (миграционные) операторы $x_i(t)=x^t_i=x_i$ и получаем новые положения $x_i(t+1)$.
3. Проверяем выполнение условий завершения итераций. Если они не выполнены, то t+=1 и возврат к шагу 2. В противном случае принимаем лучшее из найденных положений агента за приближённое решение задачи.

С самой общей точки зрения миграционные операторы вычисляют новое положение агента и проверяют на соответствие функции, зависящей от набора значений: x^t_k, $f(x^t_k)$, $E(x^t_k)$, $G(x^t_k)$, k belongs to [1; |S|], t belongs to [0; t’]. Общий смысл состоит в том, что новое положение агента S_i вообще говоря определяется всеми предыдущими положениями всех агентов популяции, а также соответствующими значениями целевой и ограничивающих функций. Здесь t’ есть число итераций. В качестве приближённого решения задачи принимается $f^*=f(x^*)=\min f(x^t_k)$ t belongs to [0; t’]. f* — минимальное значение функции среди всех значений агентов.

В итерационных алгоритмах $F(x^t_k, f(x^t_k), E(x^t_k), G(x^t_k))$ не меняется с течением числа итераций, за исключением, возможно, некоторого числа начальных шагов.

Поисковые алгоритмы оптимизации можно классифицировать в соответствии со следующими признаками:

1. Характер искомого решения. Алгоритмы назвают локальными, если схема поиска нацелена на отыскание локального минимума. Если целью является поиск глобального минимума, то алгоритм будет глобальным.
2. Характер ограничений. Алгоритм называется алгоритмом безусловной оптимизации, если ориентирован на решение соответствующей задачи — задачи безусловной оптимизации. Алгоритм называется алгоритмом условной оптимизации, если ориентирован на решение задачи условной оптимизации.
3. Характер функции F. Если F является детерминированной, то и алгоритм называют детерминированным. Если F содержит один или несколько параметров, которые на различных итерациях принимают случайные значения с заданными законами распределения, то алгоритм называется стохастическим (случайным).
4. Класс алгоритма. Алгоритм носит название пассивного, если все точки x^t_i назначены заранее. Если же данные точки определяются на основе всей или части информации об испытаниях предыдущих точек, то алгоритм называют последовательным.
5. Число предыдущих учитываемых шагов. Если при вычислении новых координат точки учитывается информация об одной итерации (предыдущей), то алгоритм называется одношаговым. Если при вычислении новых координат точки учитывается информация о более чем одном предыдущем испытании, то алгоритм называется многошаговым (с памятью).
6. Порядок используемых производных. Алгоритмы поиска называют прямыми или алгоритмами нулевого порядка, если при вычислении целевой функции и ограничивающих функций не используются производные. Если в тех же условиях используются производные k-ого порядка, то алгоритм называется алгоритмом k-ого порядка.

Выделяют также траекторные и популяционные алгоритмы оптимизации. В траекторных алгоритмах на каждой итерации обновляется положение только одного агента. При этом общее число агентов, как правило, больше одного и на разных итерациях перемещаются разные агенты. Траекторные алгоритмы могут быть одноточечными и многоточечными. В популяционных алгоритмых число агентов всегда больше одного и на каждой итерации перемещаются все агенты.

Важной проблемой при построении поисковых алгоритмов оптимизации является проблема выбора условий или критериев окончания поиска. Простейшими, но широко используемыми в вычислительной практике являются два следующих условия: $||x^{t+1}-x^t||<=e_x$ и $|f(x^{t+1})-f(x^t)|<=e_f$ Данные условия остановки поиска называются стандартными.

Метод поиска с использованием производных.
Алгоритм градиентного спуска с постоянным шагом

Пусть дана функция f(x), ограниченная снизу в n-мерном пространстве поиска и имеющая непрерывные частные производные во всех её точках. Требуется найти локальный минимум функции f(x) на множестве допустимых значений D и найти такую точку x*, чтобы она соответствовала минимуму функции.

Градиентные методы или методы первого порядка используют значения частных производных первого порядка. Стратегия решения задачи состоит в построении последовательности точек, таких, что: $f(x^{k+1})<f(x^k)$. При этом $x^{k+1}= x^k - t_k \text{grad } f(x^k)$.