Лекция 4

Алгоритмы безусловной нелинейной оптимизации. Стохастические и метаэвристические алгоритмы

Анализ и разработка алгоритмов

УНИВЕРСИТЕТ ИТМО

Содержание

Задача оптимизации

Задача

$f : \mathbb{R}^n \to \mathbb{R}$; $f = f(\mathbf{x})$, где $\mathbf{x} = (x_1, \ldots, x_n)$
Решить задачу оптимизации $f(\mathbf{x}) \to \min_{\mathbf{x} \in Q}$ означает найти точку $\mathbf{x}^* \in Q$, где Q — область допутимых значений, такую, что f достигает минимального значения в точке $\mathbf{x}^*$.
Обозначение: $\mathbf{x}^* = \arg\min_{\mathbf{x} \in Q} f(\mathbf{x})$.

Рассмотренные методы:

- прямые
- первого и второго порядка

(*Вспомните определения и методы*)

Все параметры были четко определены $\to$ детерминированный подход

А если использовать случайность и стохастичность?

Стохастические (Монте-Карло) и метаэвристические методы

Стохастические (Монте-Карло) методы – это широкий класс алгоритмов, которые основаны на повторяющейся *случайной выборке* для решения задачи оптимизации. Эти методы наиболее полезны, когда невозможно или сложно применять другие (напр., нет информации о дифференцируемости оптимизируемой функции или задача дискретная).

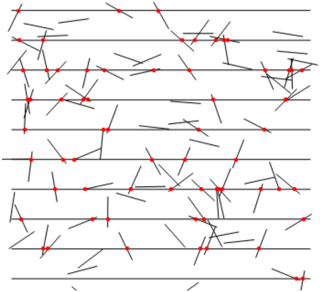
Вероятностная интерпретация: по закону больших чисел среднее значение конечной выборки из фиксированного распределения близко к математическому ожиданию этого распределения.

Демонстрация: интегрирование

Метаэвристические методы — это *вдохновленные природными явлениями* алгоритмы, которые решают задачу оптимизации *методом проб и ошибок*. Метаэвристики не гарантируют, что будет найдено решение задачи оптимизации. Многие метаэвристики используют стохастические методы.

Задача Бюффона о бросании иглы

Будем бросать иглу длиной ℓ на поверхность с начерченными параллельными прямыми, находящимися друг от друга на расстоянии t.
Какова вероятность P того, что брошенная игла пересечет одну из прямых? Предполагаем, что $\ell < t$ (случай "короткой иглы").



Решение:

$$P = \frac{2\ell}{\pi t}.$$

Пусть при бросании n игл m из них пересекли прямые. Тогда P приближается дробью m/n. Отсюда получаем экспериментальную оценку для числа π:

$$\pi \approx \frac{n}{m} \cdot \frac{2\ell}{t}.$$

Демонстрация: Реальный эксперимент, Симуляция

Имитация отжига (алгоритм Метрополиса)

Имитация отжига — это метаэвристический алгоритм, который решает задачу оптимизации подобно процессу отжига в металлургии.

Название и идея: нагрев и контролируемое охлаждение материала для увеличения размера его кристаллов и уменьшения дефектов в металлургии.

Пусть $f : \mathbb{R}^n \to \mathbb{R}$ — *энергия*. Пусть $T = \{T_k\}$ — убывающая неотрицательная последовательность такая, что $T_k = 0$ при $k > N$. Последовательность T называют *графиком охлаждения* или *температурой*.

Алгоритм

Пусть a_0 — начальное приближение.
Пока $T_k \neq 0$ для $k \in \mathbb{N}_0$:

- случайным образом выбираем $a^* \neq a_k$;
- если $f(a^*) \leqslant f(a_k)$, то $a_{k+1} = a^*$; если $f(a^*) > f(a_k)$, то $a_{k+1} = a^*$ с вероятностью
$$\exp\left(-\frac{f(a^*) - f(a_k)}{T_k}\right).$$

Демонстрация: 2D, 3D, Раскраска, Задача коммивояжера

Эволюционные алгоритмы

Эволюционные алгоритмы — это метаэвристические алгоритмы на основе эволюции популяции. Они используют механизмы, вдохновленные биологической эволюцией: отбор, мутация и воспроизводство. Возможные решения задачи оптимизации играют роль отдельных особей в популяции, а целевая функция (определяется средой, в которой находятся популяция) в процессе оптимизации (эволюции) определяет качество решений. Эволюция популяции происходит в результате многократного применения вышеупомянутых механизмов.

Примеры:

- Генетические алгоритмы (**в рамках проектов**)
- Дифференциальная эволюция
- Метод роя частиц

Демонстрация: Эволюция синтетического животного

Дифференциальная эволюция

Дифференциальная эволюция — это метаэвристический алгоритм, который решает задачу оптимизации через эволюцию популяции *агентов*, то есть возможных решений, создавая новые поколения агентов путем объединения существующих и дальнейшего отбора лучших.

Выберите *вероятность перехода* $CR \in [0, 1]$, *дифференциальный вес* $F \in [0, 2]$ и *размер популяции* $NP \geq 4$. Пусть $\mathbf{x} \in \mathbb{R}^n$ — агент в популяции.

Алгоритм

Пока критерий остановки не выполнен (напр., макс. число итераций):

- Случайно выберите NP агентов $\mathbf{x}$ (т.е. популяцию).
- Выберите из популяции три различных агента $\mathbf{a}$, $\mathbf{b}$ и $\mathbf{c}$, отличных от $\mathbf{x}$.
- Вычислите *пробный* вектор $\mathbf{y} = (y_1, \ldots, y_n)$: при $i = 1, \ldots, n$ выберите $r_i \in U(0, 1)$ и если $r_i < CR$, то $y_i = a_i + F(b_i - c_i)$, иначе $y_i = x_i$.
- Если $f(\mathbf{y}) \leqslant f(\mathbf{x})$, то замените $\mathbf{x}$ на пробный вектор $\mathbf{y}$, иначе сохраните $\mathbf{x}$.

Выберите лучшего агента в популяции в качестве решения задачи оптимизации.

Демонстрация: 2D, 3D, Геометрия крыла, Генеративный дизайн

Метод роя частиц

Метод роя частиц — метаэвристический алгоритм, который решает задачу оптимизации путем итерационного изменения положения возможных решений (*частиц*) с определенной *скоростью*. На изменение положения каждой частицы влияют ее лучшее известное положение и лучшие известные положения других частиц (**демо**).

Пусть s — число частиц в рое, с положениями x_i и скоростями v_i. Пусть p_i — лучшее известное положение частицы (**ЛИПЧ**) i и g — лучшее известное положения всего роя (**ЛИПР**).

Значения b_l и b_u отвечают за границы пространства поиска. Положительные параметры ω, ϕ_p и ϕ_g выбираются вручную.

Выбор параметров может оказать большое влияние на качество оптимизации и является предметом многочисленных исследований. Как правило, параметры для конкретной задачи подбираются в серии экспериментов.

Метод роя частиц

Алгоритм

Задайте **ЛИПР** g как случайное число
Для каждой частицы $i = 1, \ldots, s$:
  Задайте начальное положение частицы $x_i \sim U(b_l, b_u)$ (равн.распр.с.в.)
  Задайте **ЛИПЧ** через ее начальное положение: $p_i \leftarrow x_i$
  Если $f(p_i) < f(g)$, то обновите **ЛИПР**: $g \leftarrow p_i$
  Задайте скорости частицы $v_i \sim U(-|b_u - b_l|, |b_u - b_l|)$
Пока критерий остановки не выполнен (напр., макс. число итераций):
  Для каждой частицы $i = 1, \ldots, s$:
    Для каждого $d = 1, \ldots, n$:
      Задайте $r_p, r_g \sim U(0, 1)$
      Обновите скорости: $v_{i,d} \leftarrow \omega v_{i,d} + \phi_p r_p (p_{i,d} - x_{i,d}) + \phi_g r_g (g_d - x_{i,d})$
    Обновите позиции: $x_i \leftarrow x_i + v_i$
    Если $f(x_i) < f(p_i)$, то обновите **ЛИПЧ**: $p_i \leftarrow x_i$
      Если $f(p_i) < f(g)$, то обновите **ЛИПР**: $g \leftarrow p_i$

Демонтрация: Пример 1, Пример 2

Спасибо за внимание!