

---

## Метод роя частиц.

---

[Ссылка на видеокурс от МИАН](#)

### Постановка задачи и идея метода

Как и в прошлый раз, рассматривается задача оптимизации.

Имеем непустое множество  $X$  пробных решений. На множестве  $X$  задана числовая функция  $f : X \rightarrow \mathbb{R}$ , называемая целевой функцией. Требуется найти  $x^* \in X : f(x^*) = \min_{x \in X} f(x) \ (f(x) \rightarrow \min_{x \in X})$ .

Но в отличие от метода имитации отжига, используется принципиально другая идея. Мы запускаем одновременно несколько копий методов, которые будут искать глобальный минимум. Однако обладая полной независимостью, частички метода не смогут добиться поставленной задачи. Мы наделим их **коллективным** свойством оптимизации: между ними будут иметься связи, причем решать они будут одну задачу. Наиболее известными коллективными методами решения оптимизационных задач являются методы **роя частиц** и **генетические алгоритмы**, которые будут рассмотрены в следующий раз. И, вообще говоря, такая идея более эффективна, чем в методе отжига, потому что позволяет с более высокой точностью определять оптимум.

Идея метода отжига подсмотрена из физики, а идея метода роя частиц подсмотрена из биологии. Наблюдая за стаей птиц или роем живности, замечаем, что движение системы происходит цельным образом. Двигаясь каждый по своей траектории, рыбы, птицы или насекомые могут извлекать выгоду из опыта всех членов сообщества, при этом происходит корректировка их собственного движения. Они делятся своими наблюдениями с другими и общим потоком добираются до цели, наиболее благоприятной для всех точки. Это может быть, например, точка с самой вкусной едой.

Одним из главных ограничений метода является то, что на каждой итерации необходимо вычислять функцию, она должна просто вычисляться. А так нам не нужно знать ни градиента этой функции, ни какой либо еще дополнительной информации, что делает этот метод эффективным.

### Подход коллективного решения задач

Система может быть довольно сложной, иметь неопределенную геометрию, при этом она динамическая. Задавая некоторые простые правила поведения для каждого элемента системы, аналитик надеется на то, что в конечном итоге система примет некоторое устойчивое состояние. Произойдет **самоорганизация** системы.

Предполагается также, что в системе присутствуют несколько, вообще говоря различных, **агентов**. Агент — это единичный объект системы, который может воспринимать окружение, не обязательно точно и полно. Он получает некоторую информацию, в зависимости от нее он совершает определенные действия. У агента есть также собственное внутреннее состояние, ему хорошо или плохо. Возможности его действий зависят от ситуации, в которой он сам находится. Агент хоть и получает информацию из внешнего мира, взаимодействует с другими агентами, но действует независимо. Он сам определяет, какое действие ему совершить, он его совершает и оказывается в новом состоянии. При этом он ничего не знает о решаемой задаче, не имеет какой-то своей цели. Отсюда конечное расположение объектов системы ни в коем случае не зависит от начального. Этот подход будет подробнее рассмотрен в обучении с подкреплением.

Наша цель на данном этапе — корректно задать правила поведения каждого из агентов. Мы надеемся на то, что произойдет самоорганизация системы, частицы смогут отыскать глобальный минимум.

### Алгоритм

Пусть  $X \subset \mathbb{R}^n$  многомерная связная область.  $M$  — количество активных частиц, участвующих в оптимизации.  $x^m(t) \in X$  — координаты частицы  $m$  ( $m = 1, \dots, M$ ) в момент времени  $t$ .  $v^m(t) \in \mathbb{R}^n$  — вектор-скорости агента  $m$  в момент времени  $t$ .

У каждой частицы есть память. Она помнит, **в какой из точек ей было лучше всего** на текущий момент. Переводя на математический язык, в момент времени  $\tilde{t}$  для каждого агента рассмотрим точку  $p^m(\tilde{t}) : \min_{t=0,1,\dots,\tilde{t}} f(x^m(t)) = f(p^m(\tilde{t}))$ . Кроме того у частицы есть информация, что на момент времени  $\tilde{t}$  есть точка  $J(\tilde{t}) : \min_{m=1,\dots,M} f(p^m(\tilde{t})) = f(J(\tilde{t}))$ . Соответственно, эта точка является **наилучшей для всего сообщества**.

Итого, в момент времени  $t$  у каждой частицы можно выделить три индивидуальных характеристики —  $x^m(t)$ ,  $v^m(t)$ ,  $p^m(t)$ , и одну общую —  $J(t)$ .

Перейдем непосредственно к алгоритму.

---

#### Метода роя частиц

---

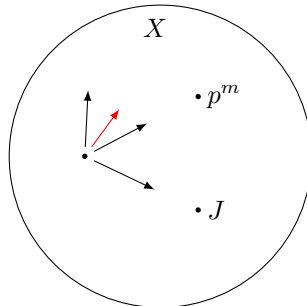
- 1: В начальный момент времени  $t = 0$  выбрать сообщество точек  $\{x^m(0)\}_{m=1}^M \subset X$  случайным образом по равномерному закону распределения
- 2: Для каждой точки задать направление и модуль начальной случайной скорости  $\{v^m(0)\}_{m=1}^M \subset X$ . Скорость должна быть соизмерима с геометрией пространства  $X$
- 3: В качестве  $\{p^m(0)\}_{m=1}^M$  взять начальные приближения  $\{x^m(0)\}_{m=1}^M$ , а в качестве  $J(0)$  взять точку  $\widetilde{x(0)} : \min_{m=1,\dots,M} f(x^m(0)) = f(\widetilde{x(0)}) \Rightarrow \{p^m(0)\}_{m=1}^M \leftarrow \{x^m(0)\}_{m=1}^M, J(0) \leftarrow \widetilde{x(0)}$
- 4:  $t \leftarrow 1$
- 5: **for**  $t = 1, 2, \dots, L$  **do**
- 6:      $m \leftarrow 1$
- 7:     **for**  $m = 1, 2, \dots, M$  **do**
- 8:         Обновить скорость по формуле

$$v_i^m(t) \leftarrow \alpha \cdot v_i^m(t-1) + \beta \cdot \xi^m \cdot (p_i^m(t-1) - x_i^m(t-1)) + \gamma \cdot \eta^m \cdot (J_i(t-1) - x_i^m(t-1)),$$

где  $i$  — компонента скорости ( $i = 1, \dots, n$ ),  $0 < \alpha, \beta, \gamma < 1$  — параметры обучения,  $\xi^m, \eta^m \sim R(0, 1)$  — равномерно распределенные независимые случайные величины

- 9:     Обновить положение  $x_i^m(t) \leftarrow x_i^m(t-1) + v_i^m(t)$
  - 10:    Обновить  $p^m(t)$
  - 11:     $m \leftarrow m + 1$
  - 12:    **end for**
  - 13:    Обновить  $J(t)$
  - 14:     $t \leftarrow t + 1$
  - 15: **end for**
- 

Первая компонента скорости — собственная инерция агента, вторая компонента направлена в сторону точки, в которой ему “лучше всего”, третья компонента направлена в сторону точки, в которой всему коллективу “лучше всего”. Такое задание скорости создает эффект маневренности в движении частицы. Коэффициент  $\beta$  отражает важность “собственного счастья”, а коэффициент  $\gamma$  — важность “коллективного счастья”. Обычно число  $\alpha$  должно быть близким к 1, а числа  $\beta$  и  $\gamma$  должны быть достаточно малыми. При этом компоненты скорости будут содержать случайные компоненты, которые создают сложность динамики. Если бы вместо случайных величин стояли единицы, то движение агентов было бы более предсказуемым и менее маневренным.



Эти вектора-компоненты новой скорости (обозначенные черными стрелочками) складываются по правилу параллелограмма, частица перемещается с результирующей скоростью (вдоль красной стрелочки). И так на каждой итерации.

### Пример. Минимизация функции многих переменных

Пример относится к традиционной задаче нахождения минимума функции.

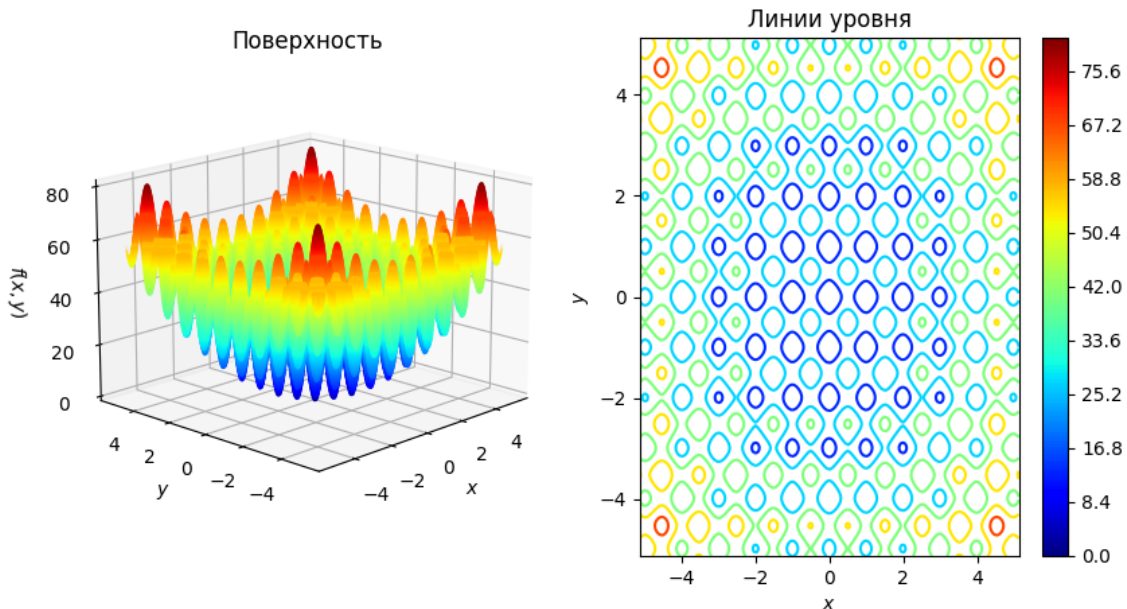
Рассмотрим функцию Растригина

$$f(\mathbf{x}) = 10n + \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)],$$

где  $x_i \in [-5.12, 5.12]$ . Она задана на  $\mathbb{R}^n$ . Для удобства визуализации будем рассматривать  $n = 2$ .

Функция часто используется для тестирования эффективности алгоритмов оптимизации, потому что имеет большое количество локальных минимумов и большую область поиска. Легко однако видеть, что глобальным минимумом этой функции является точка  $\mathbf{x} = \mathbf{0}$ , в которой  $f(\mathbf{x}) = 0$ .

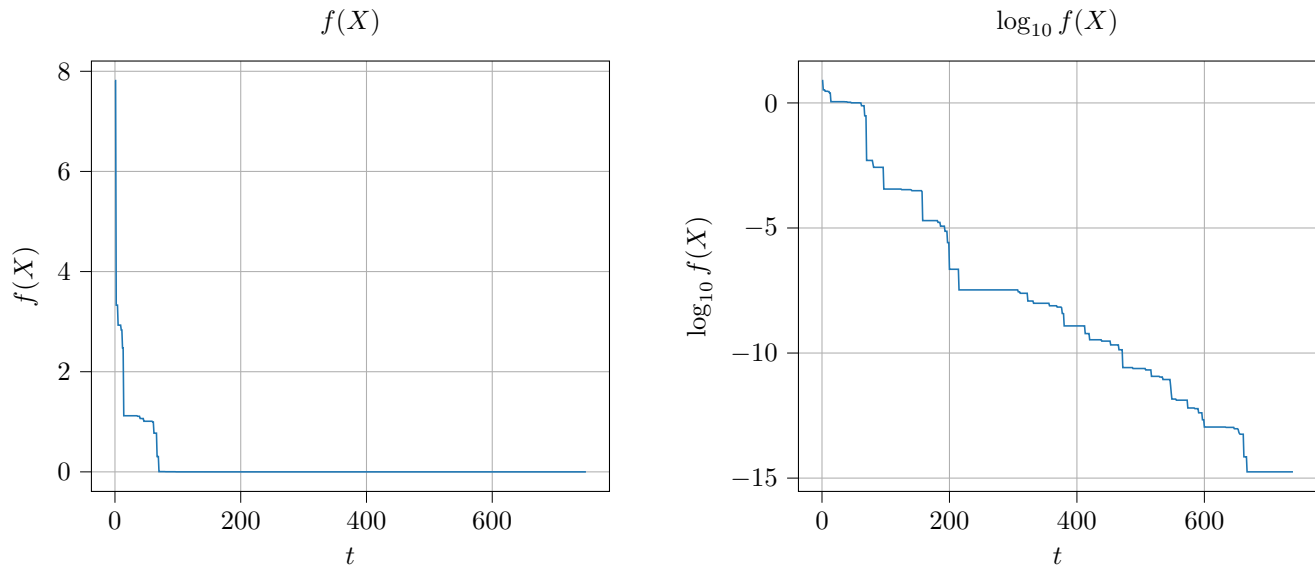
Функция Растригина



В качестве параметров метода роя частиц возьмем следующие значения:

1. Количество частиц  $M = 25$ .
2. Количество итераций  $L = 750$ .
3. Параметры в изменении скорости:  $\alpha = 0.95, \beta = \gamma = 0.2$ .

На графике динамики вычислений целевой функции (отображены лишь значения в общем глобальном минимуме) мы видим, что значение целевой функции быстро убывает. Однако этот график не очень понятен, поскольку на нем неотличимы значения, начиная с  $t = 100$ . Поэтому приведем еще один график, где значения целевой функции отображены в логарифмическом (по основанию 10) масштабе.



Из последнего графика видно, что метод сходится с экспоненциальной скоростью и за 750 итераций достигает значения целевой функции  $10^{-15}$ , при минимуме 0.

### Приложение

В приложении приведена анимированная версия поиска частицами глобального оптимимума на графике линейного уровня. Частицы обозначены точками голубого цвета, направление их скорости обозначено стрелками. На каждой итерации звездочкой обозначено текущее лучшее положение в совокупности, а серыми точками обозначено лучшее положение каждой из точек. Крестиком отмечен искомый минимум. Мы видим, что точки скапливаются в области этого минимума довольно быстро, это и есть самоорганизация.

Также в приложении добавлен код и комментарии к нему.