

Метод имитации отжига.

[Ссылка на видеокурс от МИАН](#)

В этой статье будет рассмотрен алгоритм популярного метода МО для решения задач оптимизации. Будет приведено пару примеров использования данного метода для конкретных задач.

Постановка задачи

Пусть X — непустое множество, называемое **метрическим пространством**. Для любых двух элементов метрического пространства X определено расстояние (другими словами, **метрика**). Зададим метрику для этого множества как функцию двух аргументов

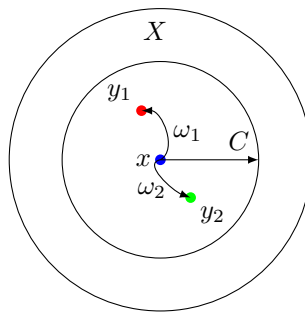
$$\rho : X \times X \rightarrow [0; +\infty).$$

Пусть также на множестве X задана числовая функция

$$f : X \rightarrow \mathbb{R},$$

называемая **целевой функцией**.

Фиксируем **вероятностное пространство** $\langle \Omega, \mathcal{A}, \mathbb{P} \rangle$. Определим функцию двух аргументов G такую, что $\forall x \in X$ и $\forall \omega \in \Omega \exists y \in X : G(x, \omega) = y$. y — случайный элемент из X , так как один из аргументов функции G — ω — случайный элементарный исход некоторого эксперимента Ω . Также пусть метрика $\rho(x, y) < C$, где $C = \text{const}$. Функция G переносит x в y не дальше, чем на расстояние C . Схематично это можно изобразить так:



G переводит $x \in X$ в $y_1 \in X$, если реализовался исход $\omega_1 \in \Omega$, либо в $y_2 \in X$, если реализовался исход $\omega_2 \in \Omega$ (и т.д.), но при этом в любом случае $\rho(x, y_1) < C, \rho(x, y_2) < C$.

Константа C — настраиваемый, постоянно уточняемый параметр алгоритма МО.

Задача оптимизации ставится следующим образом:

$$f(x) \rightarrow \min_{x \in X},$$

иначе говоря, требуется найти $x^* \in X : f(x^*) = \min_{x \in X} f(x)$. Ищется минимум и элемент, на котором он достигается. В случае, когда требуется найти максимум, вместо $f(x)$ используется $-f(x)$.

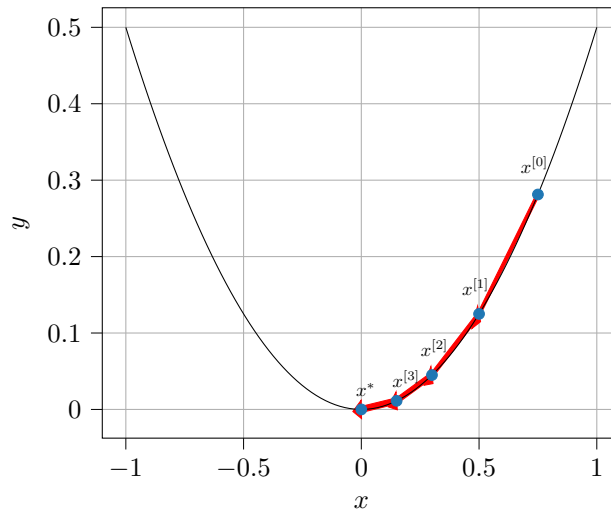
“Математический” способ решения и сопутствующие проблемы

Предположим, что $X \subseteq \mathbb{R}^n$, то есть числовая функция f задана в n -мерном пространстве, а также пусть $f \in C^1(X)$, то есть f является непрерывно-дифференцируемой на X , иными словами, в каждой точке этой области функция имеет непрерывную производную. Выбирается некоторое **начальное приближение** $x^{[0]} \in X$. Основная идея **градиентного метода** заключается в том, чтобы идти в направлении наискорейшего спуска, а это направление задаётся антиградиентом $-\nabla f$:

$$x^{[j+1]} = x^{[j]} - \lambda^{[j]} \nabla f(x^{[j]}),$$

где $\lambda^{[j]}$ задает скорость градиентного спуска.

В одномерном случае градиент — это производная, которая направлена в сторону возрастания функции, антиградиент же направлен в сторону убывания функции — то, что нам нужно. Если на каждой итерации мы будем уменьшать скорость спуска, то метод может сойтись, минимум найдется:

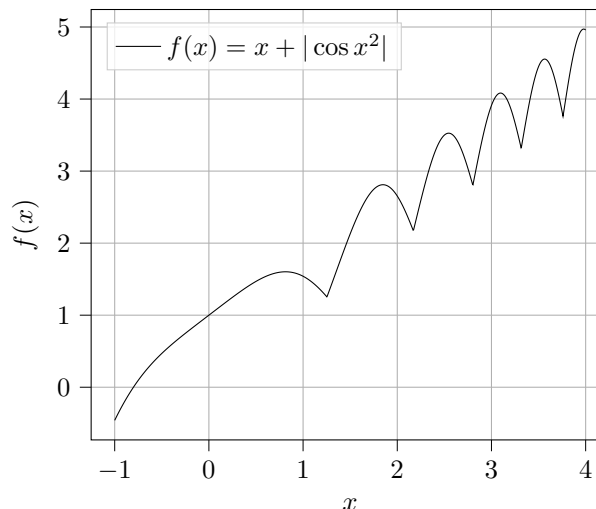


В многомерном случае метод работает аналогично, для каждой переменной x_i нужно посчитать частную производную $\frac{\partial f}{\partial x_i}$ и произвести обновление по формуле $x_i^{[j+1]}$.

Но возникает ряд проблем при выборе такого способа решения.

1. Негладкость целевой функции

Часто в реальных задачах встречаются функции, у которых не в каждой точке соблюдена гладкость. В таких функциях можно наблюдать **углы**.



В отличие от градиентных методов оптимизации для метода отжига не требуется гладкости целевой функции, поскольку нет необходимости вычислять градиент.

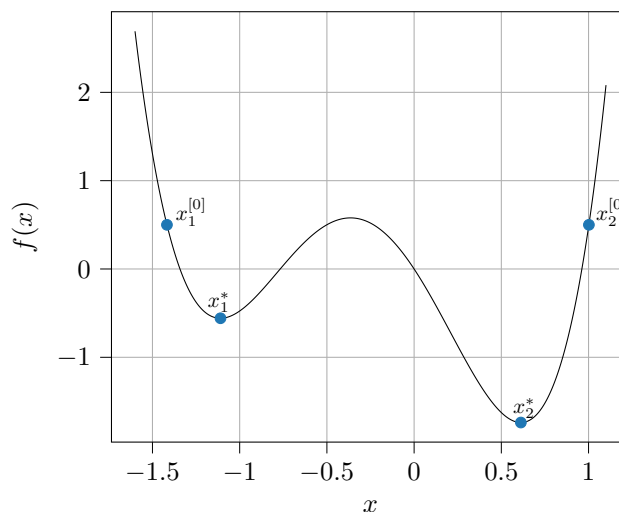
2. Требуется оптимизация на дискретных структурах

Пример, который мы рассмотрим позднее — **задача о расстановке ферзей на шахматной доске**.

Метод отжига можно применять и в таком случае, когда множество X является дискретным. Естественно, градиентный метод неприменим, поскольку невозможно посчитать градиент функции, определенной над дискретным пространством. Хотя для каждой конкретной задачи можно придумать некий аналог градиента и выйти из этого положения.

3. Наиболее остра проблема поиска глобального экстремума

Градиентный метод заточен на поиск лишь **локального экстремума**.



Если в качестве начального приближения мы укажем точку $x_1^{[0]}$, то градиентный спуск приведет нас в локальный минимум x_1^* и остановится, мы не сможем попасть в более оптимальную точку x_2^* . В качестве начальной стоит выбирать $x_2^{[0]}$, тогда мы доберемся до нужного оптимума x_2^* .

В данном случае легко построить график, поэтому прост выбор подходящего начального приближения. Но не всегда эта функция нам в принципе известна, также она может быть сколь угодно сложной, нельзя изобразить ее портрет — градиентный метод перестает работать.

К счастью, метод имитации отжига лишен и этого недостатка, он находит глобальный оптимум при любом начальном приближении, причем функция может иметь сколь угодно много локальных экстремумов.

Прежде чем перейти непосредственно к алгоритму и к примерам, рассмотрим еще один важный момент.

Концепция жадного / нежадного выбора

На каждом шаге градиентного метода делается оптимальный шаг, с каждой итерацией мы приближаемся к минимуму, в конце получаем оптимальное решение. Но не всегда правильной стратегией делать наиболее оптимальный выбор (это и есть **жадная** стратегия), иногда полезно ухудшать текущую ситуацию в надежде на то, что впоследствии это приведет нас к оптимальному исходу. К примеру, в шахматах иногда приходится отдавать свои фигуры противнику, чтобы ближе подобраться к его королю. В игре в карты мы часто не покрываем ход соперника, даже если у нас есть необходимые карты — мы не избавляемся от карт, а наоборот увеличиваем их количество, проигрывая на конкретном шаге. Это и есть концепция **нежадного** выбора. Такие шаги неправильные с точки зрения жадного алгоритма, а с точки зрения оптимизации они полезны, ведь именно они приводят к результату.

Алгоритм метода отжига

Метод имитации отжига (или просто метод отжига, англ. Simulated annealing) представляет собой эффективный алгоритм, основанный на стохастическом поиске, для решения оптимизационных и поисковых задач. Как и любой эвристический метод, он может быстро находить решения экспоненциально сложных задач, но с другой стороны не гарантирует сходимость к решению.

Алгоритм основывается на имитации физического процесса, который происходит при отжиге металлов. Когда металл имеет высокую температуру — он гибкий, при остывании он в конце-концов затвердевает. Активность атомов кристаллической решетки металлов тем больше, чем выше температура, которую постепенно понижают, что приводит к тому, что вероятность переходов отдельных атомов в состояния с большей энергией уменьшается. Также существует вероятность перехода из высокоэнергетического состояния в низкоэнергетическое, которая велика при любой температуре. Если в роли физической системы представить задачу оптимизации, в роли энергии системы — значение целевой функции f , в роли атомов — элементы множества X , то можно решать задачу оптимизации функции f , используя механизмы, которые определяют процесс отвердевания.

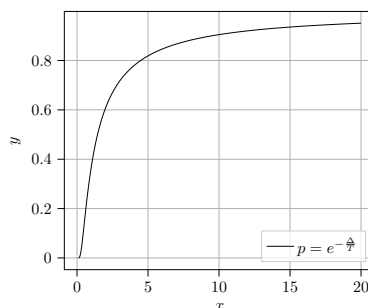
Метода имитации отжига

```

1:  $k \leftarrow 0$ 
2: Зафиксировать значение  $T^{[0]} > 0$  — начальную температуру
3: Выбрать элемент  $x^{[0]} \in X$  в качестве начального приближения
4: for  $k = 0, 1, 2, \dots$  do
5:   Имея некоторое приближение  $x^{[k]}$ , вычислить новое случайное приближение  $\widetilde{x}^{[k]}$ , которое в определенном смысле близко к  $x^{[k]}$  ( $\rho(x^{[k]}, \widetilde{x}^{[k]}) < C$ ) :  $\widetilde{x}^{[k]} \leftarrow G(x^{[k]}, \omega)$ 
6:    $\Delta \leftarrow f(\widetilde{x}^{[k]}) - f(x^{[k]})$  — вычислить разницу между потенциальным приближением и старым приближением
7:   if потенциальное приближение оказалось лучше старого (т.е.  $\Delta < 0$ ) then
8:      $x^{[k+1]} \leftarrow \widetilde{x}^{[k]}$  — в качестве следующего приближения принять потенциальное
9:   else if потенциальное приближение оказалось хуже старого (т.е.  $\Delta \geq 0$ ) then
10:    if с вероятностью  $p = e^{-\frac{\Delta}{T^{[k]}}}$  then
11:       $x^{[k+1]} \leftarrow \widetilde{x}^{[k]}$  — исходя из концепции нежадного выбора, в качестве следующего приближения принять вычисленное, менее оптимальное
12:    else
13:       $x^{[k+1]} \leftarrow x^{[k]}$  — в качестве следующего приближения оставить старое, более оптимальное
14:    end if
15:  end if
16:  Снизить температуру  $T^{[k+1]} \leftarrow \alpha T^{[k]}$ , где коэффициент снижения  $0 < \alpha < 1$ 
17:   $k \leftarrow k + 1$ 
18: end for

```

Рассмотрим вероятность $p = e^{-\frac{\Delta}{T^{[k]}}}$: показатель степени отрицательный, при этом $|\Delta|$ мало. Пока температура высока, вероятность стремится к единице, алгоритм чаще соглашается с менее оптимальными решениями. По мере снижения температуры, вероятность опускается к нулю, алгоритм все реже принимает ухудшающие приближения.

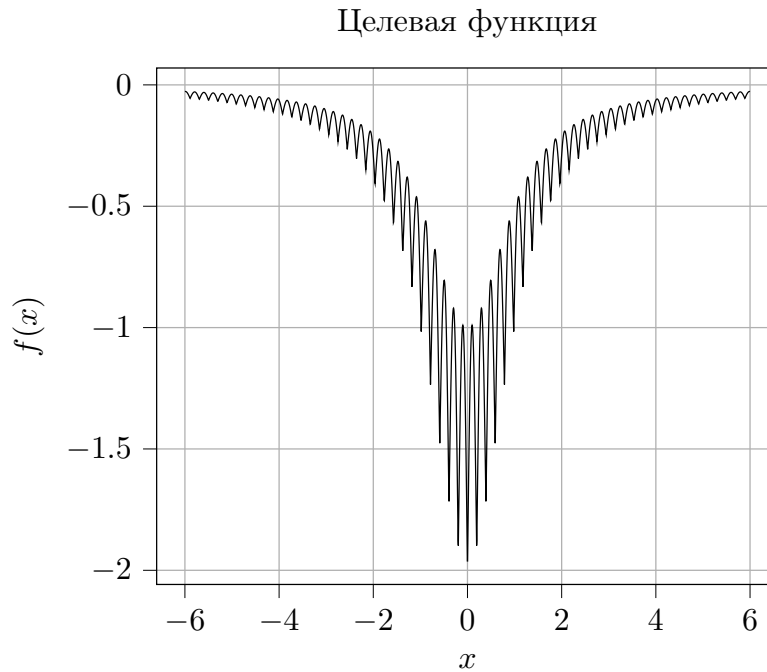


Критерии остановки алгоритма отжига являются эвристическими и определяются постановкой задачи. В частности, используются критерии снижения температуры : как только температура достигнет очень низкого значения, итерироваться дальше не будет иметь смысла, так как решение будет изменяться с очень низкой вероятностью.

Пример 1. Минимизация функции

В первом примере применим метод отжига для нахождения минимума функции

$$f(x) = \frac{-2 + |\sin 16x|}{x^2 + 1}.$$



Эта функция примечательна тем, что, во-первых, имеет периодические локальные минимумы, а, во-вторых, в этих минимумах она не имеет производной. Стандартные градиентные методы для нее не применимы, поскольку это негладкая функция, которая к тому же имеет много локальных минимумов. Её глобальный минимум достигается в точке $(0, -2)$.

Будут использованы следующие параметры обучения:

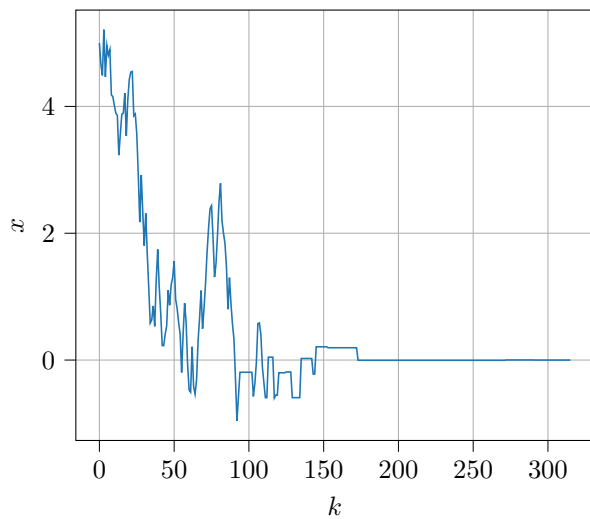
1. Начальная температура: $T^{[0]} = 100$.
2. Коэффициент снижения температуры: $\alpha = 0.95$.
3. Критерий остановки метода: $T^{[k]} < 10^{-5}$.
4. В качестве G возьмем

$$G(x, \omega) = x + \xi(\omega),$$

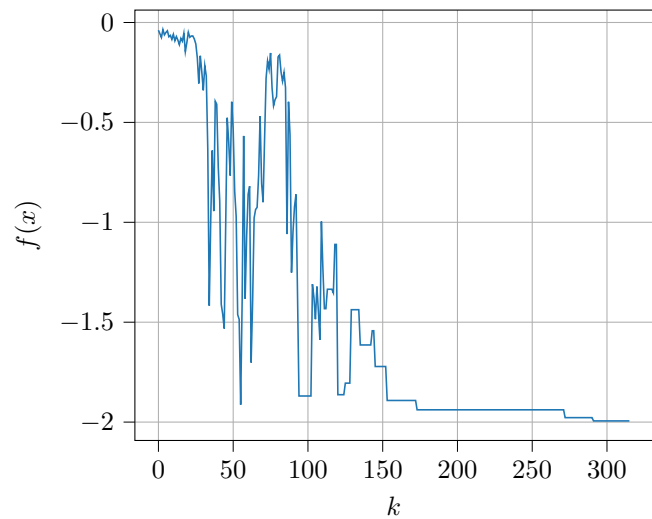
где ξ — случайная равномерно распределенная на интервале $(-0.75, 0.75)$ величина ($\xi \sim R(-0.75, 0.75)$).

В результате применения метода получены результаты, которые представлены на графиках ниже. На графике слева показана динамика решения $x^{[k]}$, на графике справа приведены значения целевой функции на каждой итерации.

Поиск оптимального решения



Значение целевой функции



На результативных графиках мы видим одну общую тенденцию, которую мы обозначили выше: с уменьшением температуры все реже меняется приближение. В самом начале поиск происходит довольно беспорядочно, значение x принимает и положительные, и отрицательные значения, но в конце-концов сплав затвердевает, точность приближения с каждой итерацией все увеличивается. Это можно сравнить с витальностью: когда человек молод, он все чаще принимает рискованные решения, при этом ища свою нишу; к старости он находит себя, поэтому в этом возрасте маловероятны серьезные жизненные колебания, происходит плавное течение к концу.

Нужно отметить, что метод отжига **будет сходиться не всегда**, поэтому иногда приходится запускать алгоритм несколько раз, либо производить дополнительную настройку параметров, это вполне нормальная история для подобного рода методов.

Реализация алгоритма и комментарии к ней, а также наглядная демонстрация работы метода в виде анимации, приведены в приложении.

Пример 2. Задача о расстановке ферзей

В качестве второго примера использования метода имитации отжига мы рассмотрим задачу-головоломку о размещении N ферзей на шахматной доске $N \times N$ таким образом, чтобы ни один из ферзей не атаковал ни одного другого.

Пусть X — это множество всех возможных расположений N ферзей. Тогда $f(x)$ — количество взаимных атак ферзей. Ясно, что $f(x) \geq 0$ и $f(x)$ — четное число, потому что в случае взаимной атаки ударить может как первый, так и второй ферзь. Искомым решением является такое расположение ферзей $x^* \in X$, что

$$f(x^*) = 0.$$

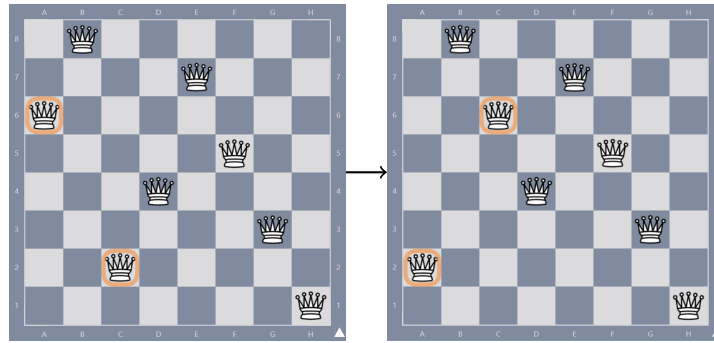
Заметим, что в оптимальной позиции никакие фигуры не могут находиться на одной вертикали, поэтому в множестве X нет таких конфигураций. Тогда каждому элементу $x \in X$ можно сопоставить упорядоченный набор

$$x = (a_1, a_2, \dots, a_N),$$

где $a_i \in \{1, 2, \dots, N\}$ — ферзь на i -ой вертикали находится на a_i -ой горизонтали. Также ферзи не могут быть на одной горизонтали, поэтому на самом деле X можно считать множеством всех возможных перестановок упорядоченного набора

$$(1, 2, \dots, N). \quad (1)$$

Заметим, что мощность множества $|X| = N!$.



$$G((6, 8, 2, 4, 7, 5, 3, 1), \omega) = (2, 8, 6, 4, 7, 5, 3, 1)$$

Будут использованы следующие параметры обучения:

1. Начальная температура: $T^{[0]} = 100$.
2. Коэффициент снижения температуры: $\alpha = 0.98$.
3. Критерий остановки метода: либо $f(x^{[k]}) = 0$, либо $T^{[k]} < 10^{-9}$ (если решение не найдено).
4. Операцию G определим, как перестановку местами двух случайных элементов внутри $x^{[k]}$. Будем считать, что эта операция меняет $x^{[k]}$ на близкий случайный элемент $\widehat{x^{[k]}} = G(x^{[k]}, \omega)$ ($\rho(x^{[k]}, \widehat{x^{[k]}}) < C$).

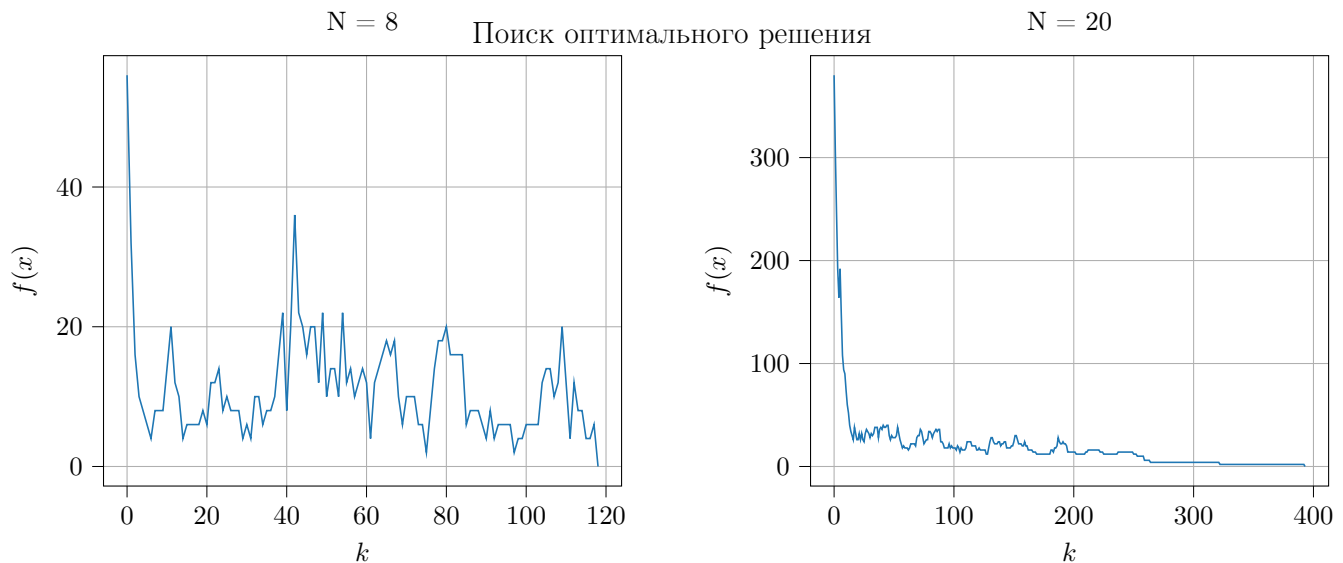
В качестве начального возьмем расположение всех ферзей на главной диагонали, т.е. набор (1).

Программа отработала 119 итераций и нашла решение для стандартной доски 8×8 :

$$(2, 5, 7, 4, 1, 8, 6, 3).$$

За 394 итерации нашлось решение для нестандартной доски 20×20 :

$$(12, 1, 16, 18, 6, 14, 9, 15, 3, 8, 4, 2, 17, 19, 11, 13, 20, 10, 5, 7).$$



Для рассмотренной задачи имеется $8! = 40320$ вариантов, которые на компьютере можно было бы и перебрать, причем довольно быстро. Однако в случае $N = 20$ имеем $20! \cong 2.4 \times 10^{17}$ вариантов, на перебор уйдет довольно много времени. Однако метод отжига справляется с этой задачей примерно за 400 итераций.

Реализация алгоритма и комментарии к ней, а также наглядная демонстрация поиска непосредственно на шахматной доске в виде анимации (будет добавлена скоро), приведены в приложении.

Заключение

Стоит отметить, что реализаций метода имитации отжига довольно много, но идея, которую мы подробно рассмотрели выше, везде одна и та же. Как мы убедились, эта идея позволяет решать ряд задач, которые требуют нестандартного подхода.