

Московский Авиационный Институт  
(национальный исследовательский университет)

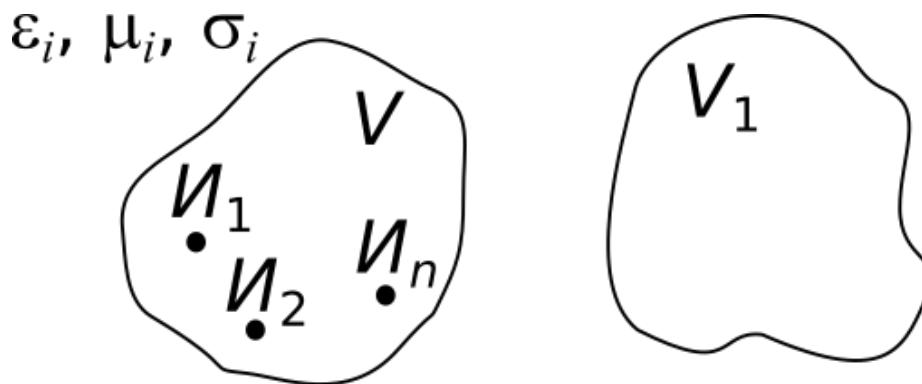
---

Решение обратных задач  
электродинамики.

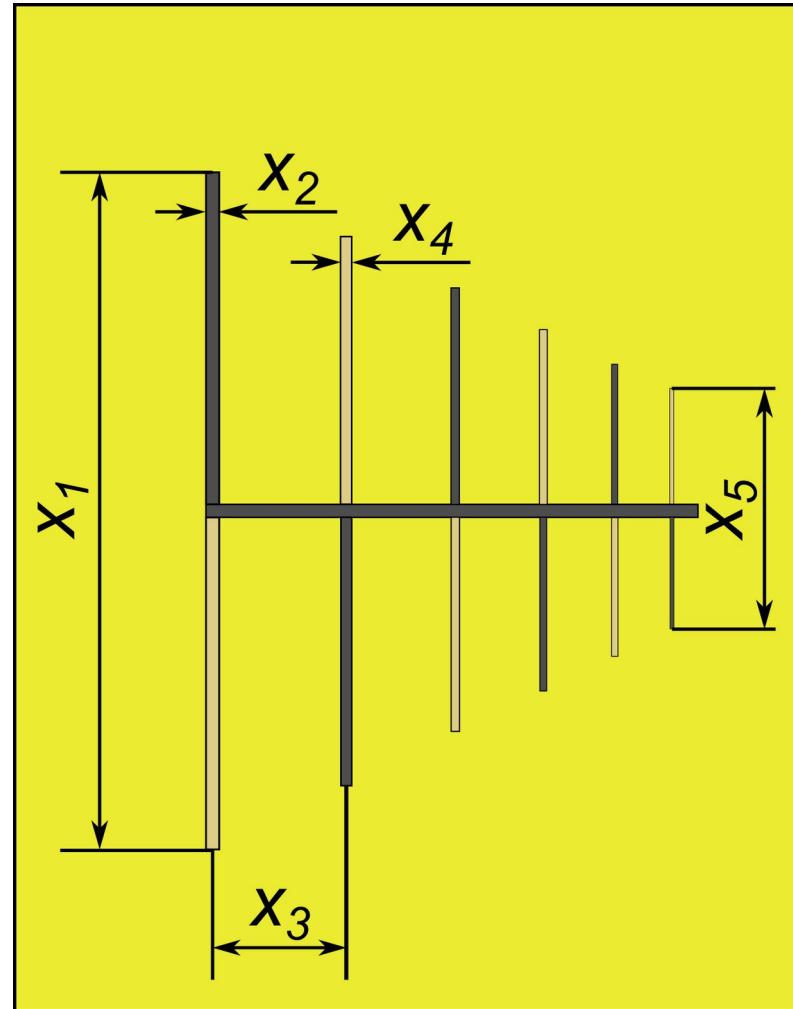
Алгоритмы глобальной  
оптимизации

# Обратная задача электродинамики

**Обратная задача электродинамики (задача синтеза) —**  
определение параметров среды и (или) источников в  
области  $V$  по известному распределению  
электромагнитного поля в некоторой другой области  $V_1$ ,  
которая может не совпадать с  $V$ .



# Параметрическая модель печатной вибраторной антенны



# Целевая функция

$\Phi(\mathbf{X})$  — целевая функция

$\mathbf{X} = \{x_1, x_2, \dots, x_n\}$  — параметры оптимизации

$$\Phi(\mathbf{X}) = \alpha_1 \cdot f_1(\mathbf{X}) + \alpha_2 \cdot f_2(\mathbf{X}) + \dots + \alpha_m \cdot f_m(\mathbf{X})$$

$f_1(\mathbf{X}), f_2(\mathbf{X}), \dots, f_m(\mathbf{X})$  — штрафные функции

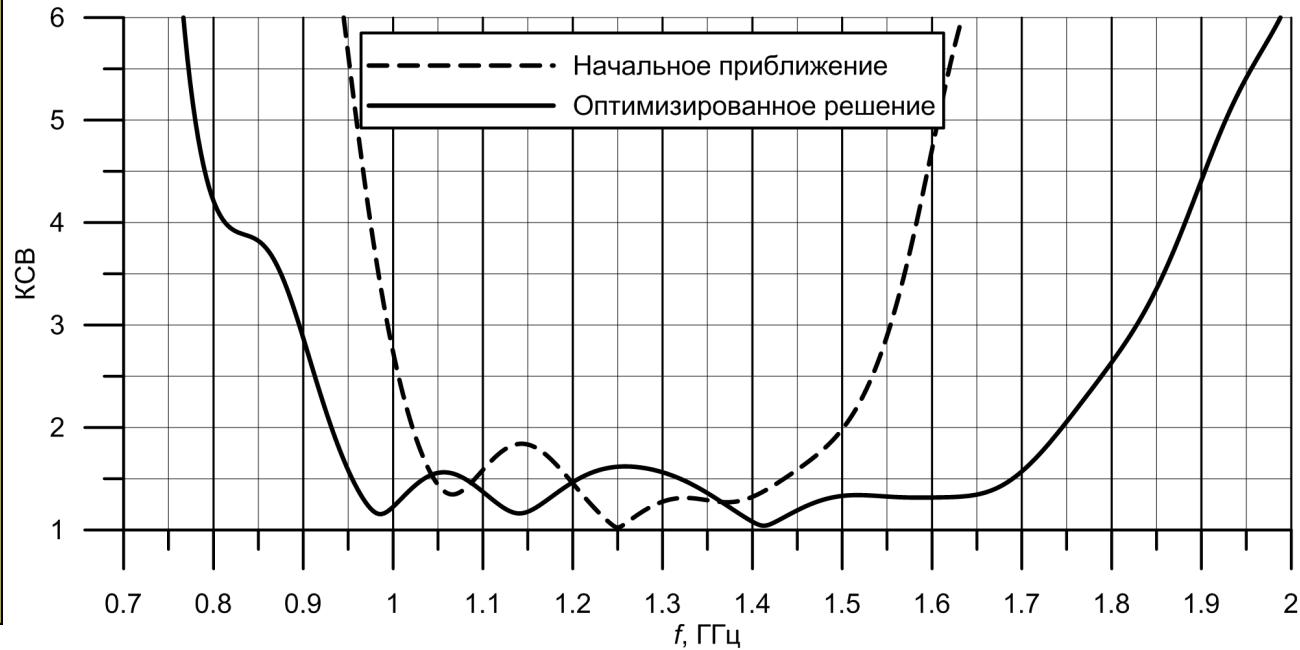
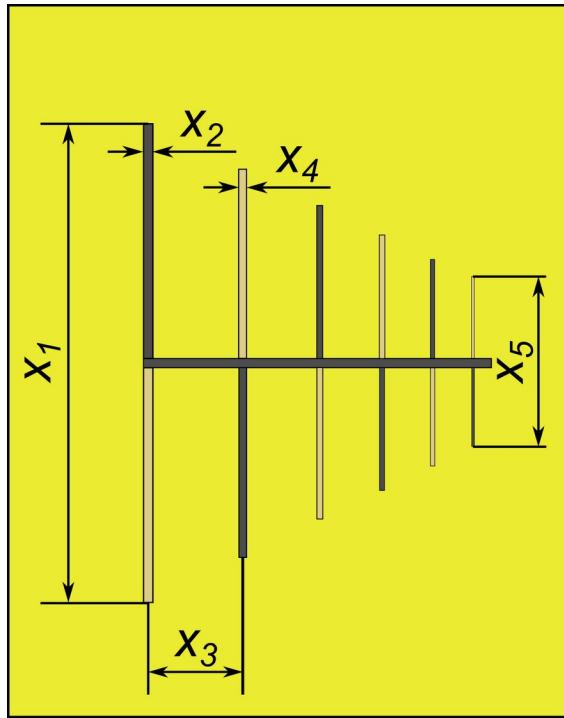
$\alpha_1, \alpha_2, \dots, \alpha_m$  — весовые коэффициенты

# Оптимизация

Оптимизация — (от лат. *optimus* — наилучший) задача нахождения экстремума (максимума или минимума) целевой функции в некоторой области конечномерного векторного пространства, ограниченной набором линейных и/или нелинейных равенств и/или неравенств.

Требуется найти такой  $\tilde{\mathbf{X}}$ , что  $\Phi(\tilde{\mathbf{X}}) \rightarrow \min$

# Оптимизация параметров антенны в с помощью генетического алгоритма

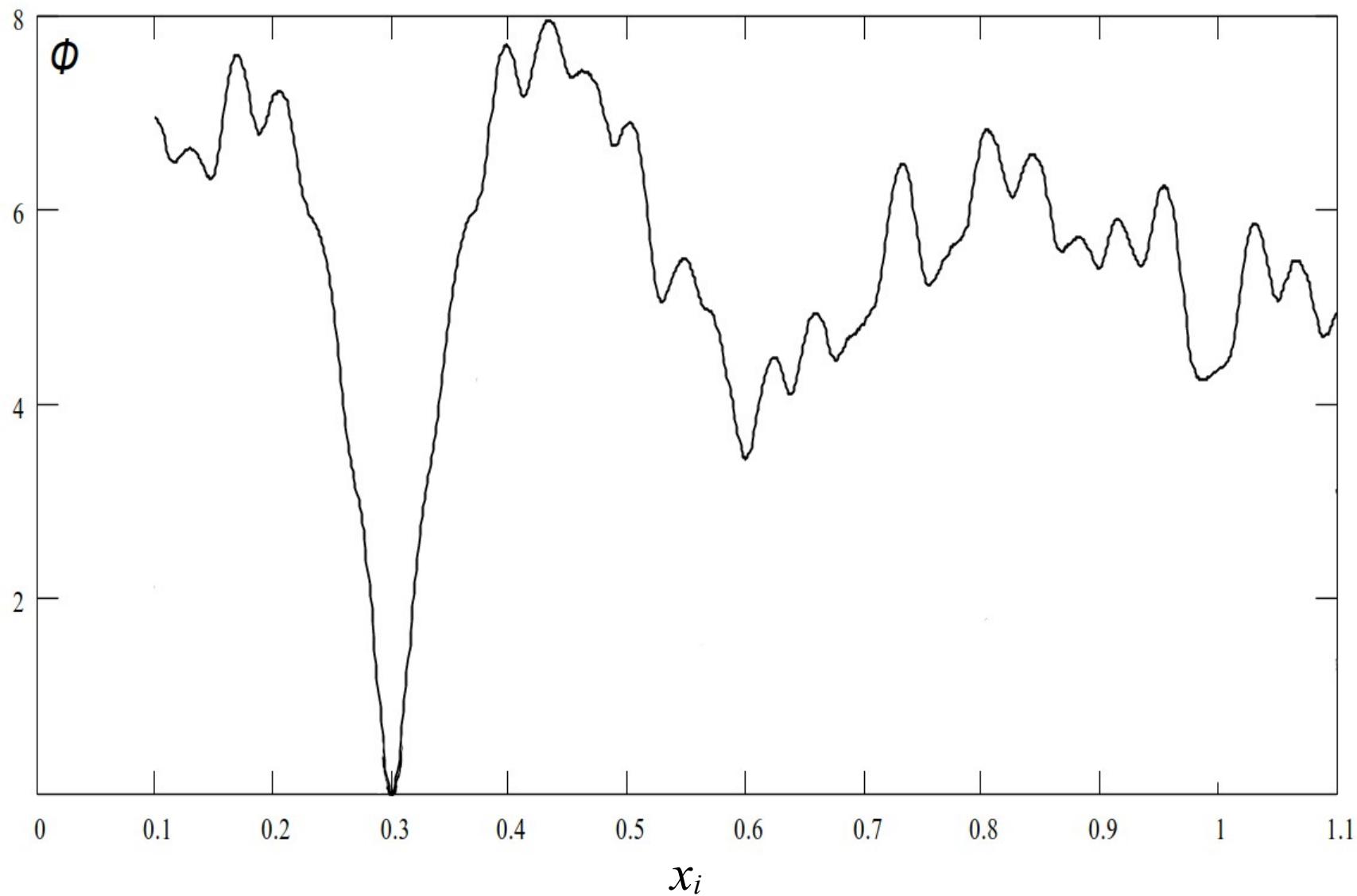


**Параметры алгоритма:**

Минимизация максимального КСВ в диапазоне частот 0.95 – 1.65 ГГц

$$\Phi(\mathbf{X}) = \max(KCB) \Big|_{f \in [f_{min}; f_{max}]}$$

# Пример целевой функции



---

# Метод градиентного спуска

# Метод градиентного спуска

В случае поиска минимума  $\Phi(\mathbf{X})$ :

$$\mathbf{X}_{i+1} = \mathbf{X}_i - \lambda_i \cdot \nabla \Phi(\mathbf{X}_i)$$

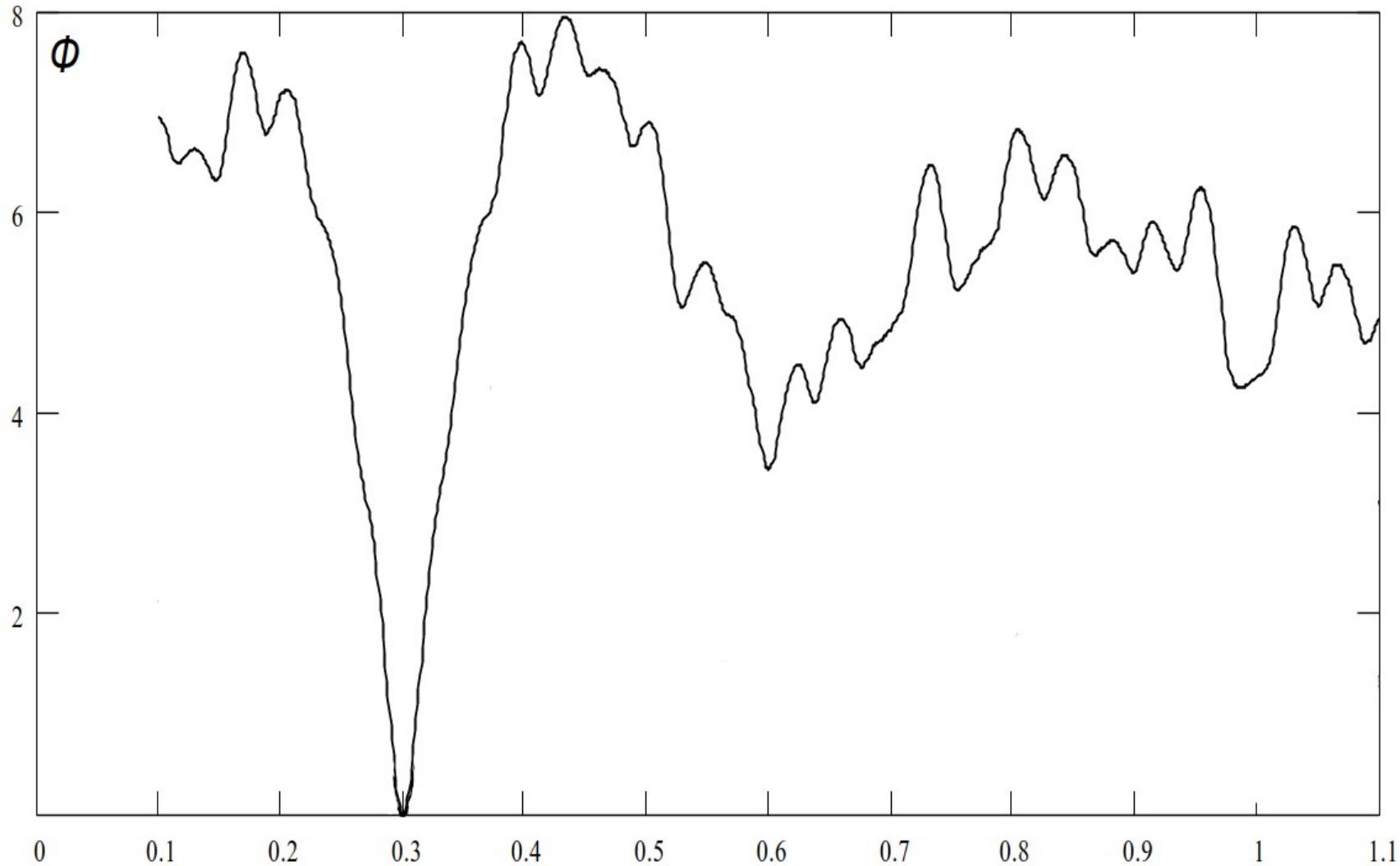
$i$  – номер итерации;

$\lambda_i$  задает скорость градиентного спуска.

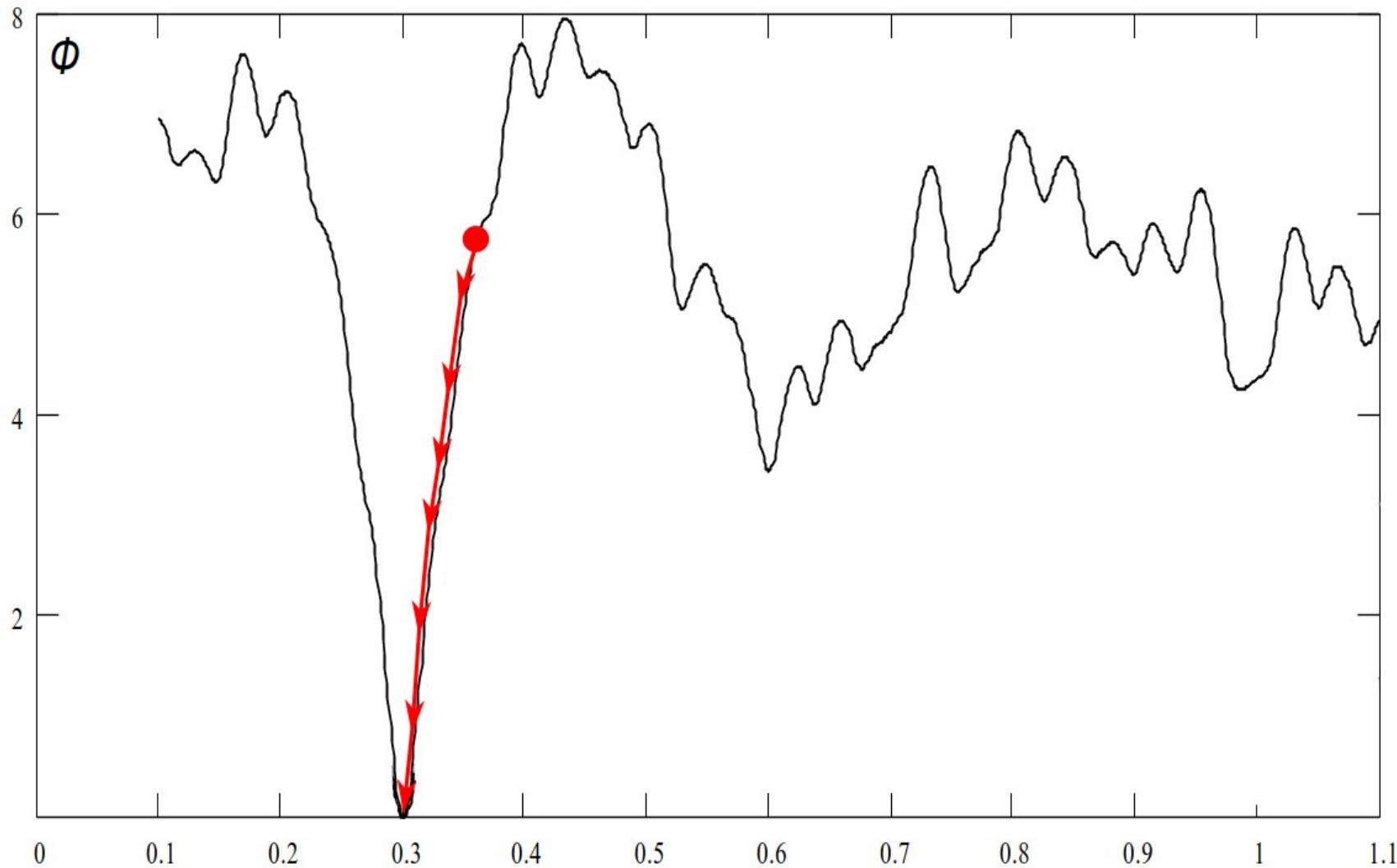
$$\nabla \Phi(\mathbf{X}) = \text{grad } \Phi(\mathbf{X}) = \mathbf{e}_1 \frac{\partial \Phi}{\partial x_1} + \mathbf{e}_2 \frac{\partial \Phi}{\partial x_2} + \dots + \mathbf{e}_n \frac{\partial \Phi}{\partial x_n}$$

$$\nabla \Phi(\mathbf{X}) = \text{grad } \Phi(\mathbf{X}) = \left( \frac{\partial \Phi}{\partial x_1}, \frac{\partial \Phi}{\partial x_2}, \dots, \frac{\partial \Phi}{\partial x_n} \right)$$

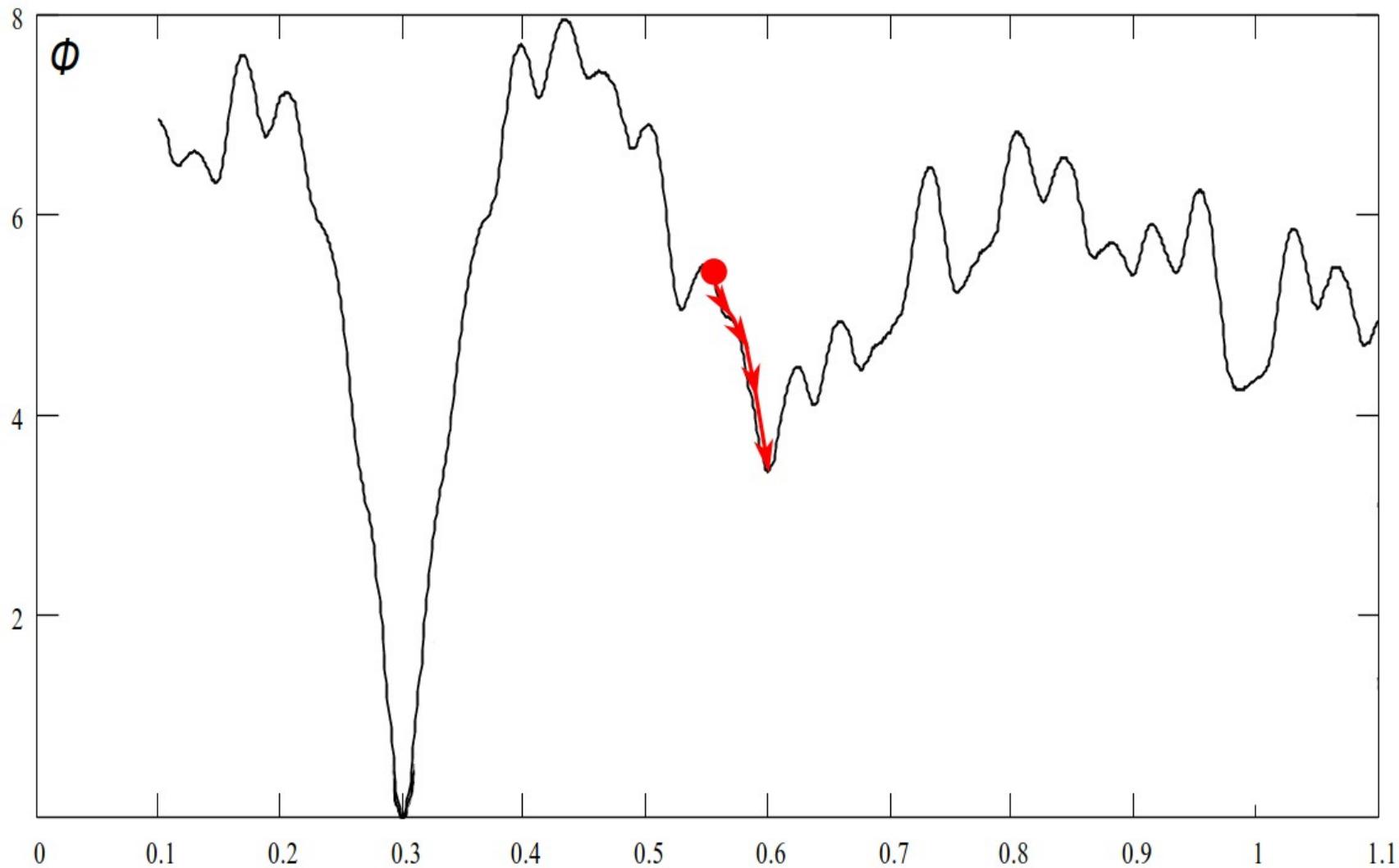
# Пример целевой функции



# Сходимость градиентного метода



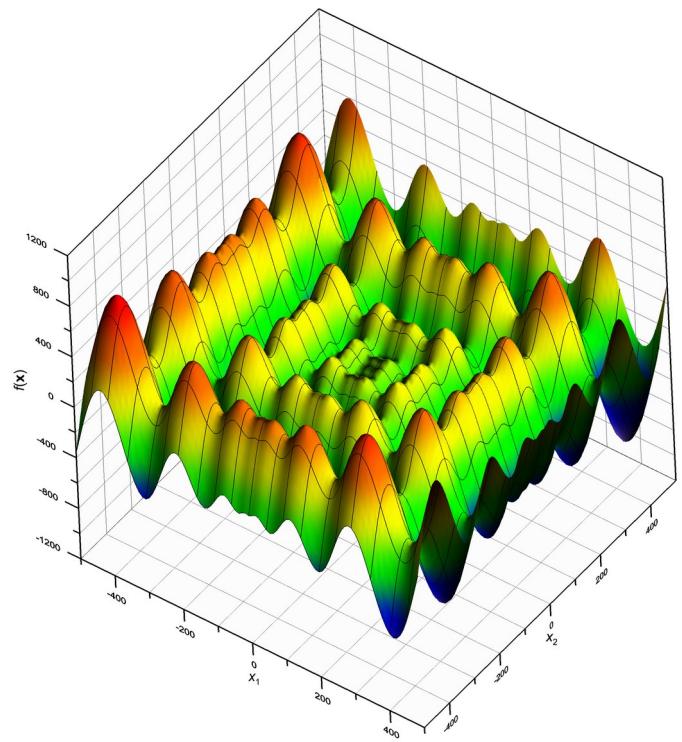
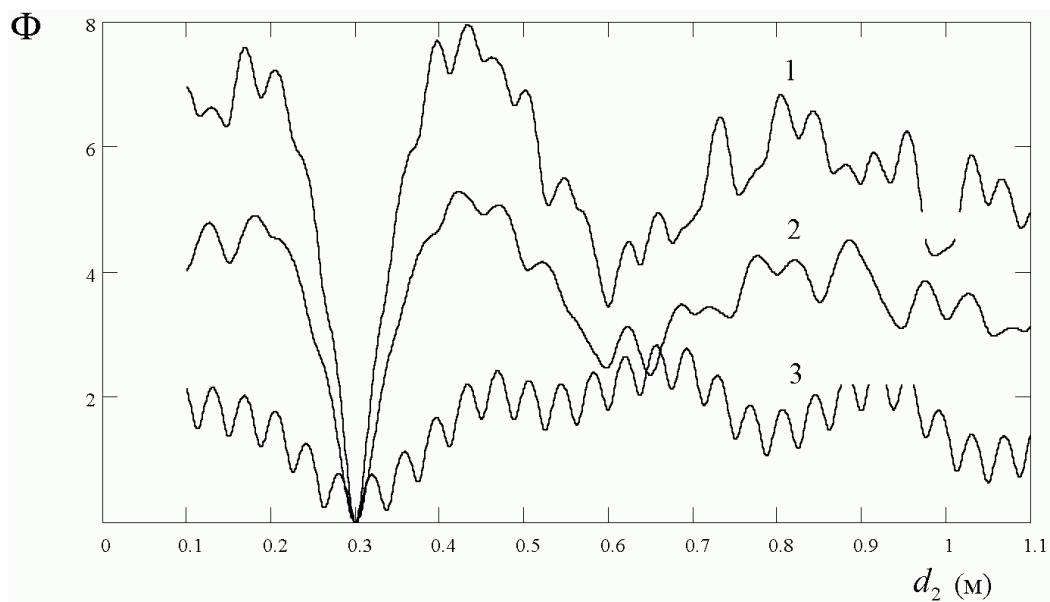
# Сходимость градиентного метода



---

# Алгоритмы глобальной оптимизации

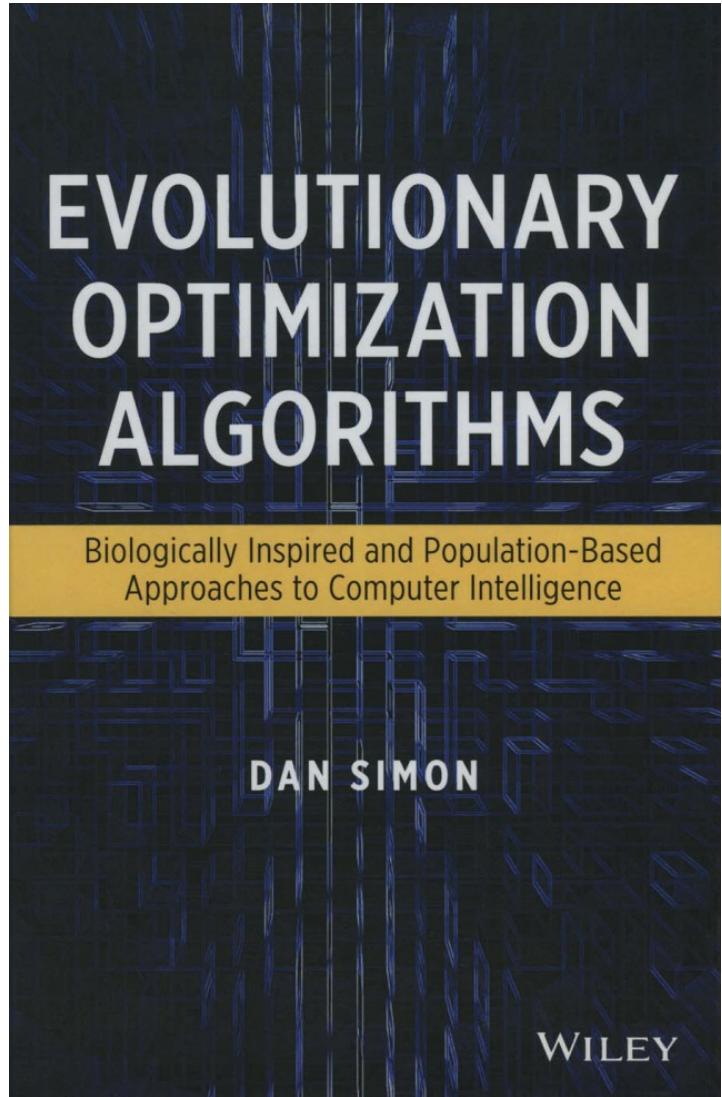
# Задача оптимизации



# Алгоритмы глобальной оптимизации

- Генетический алгоритм
- Метод Нелдера-Мида
- Алгоритм случайного поиска
- Алгоритм имитации отжига
- Алгоритм роя частиц
- Алгоритм пчел
- И многие другие

# Литература



# Генетический алгоритм

J. H. Holland. “Adaptation in natural and artificial systems”.

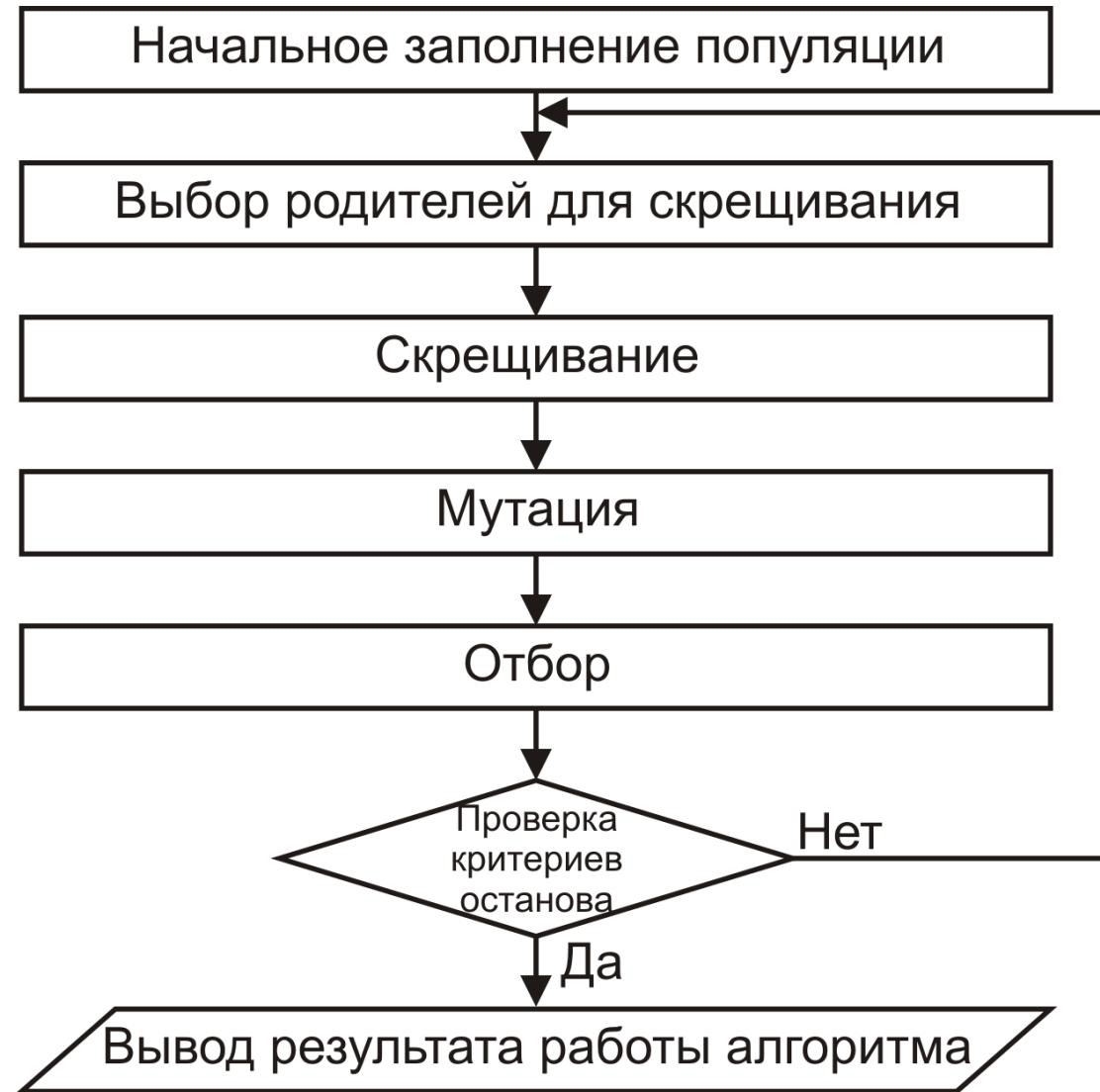
University of Michigan Press, Ann Arbor, 1975.

# Генетический алгоритм.

## Основные понятия

- **Хромосома** – значение одного из искомых параметров.
- **Особь** – одно из возможных решений (набор хромосом).
- **Популяция** – набор решений (особей).
- **Функция приспособленности** – минимизируемая (целевая) функция.

# Генетический алгоритм



## Начальное заполнение популяции

- Случайное распределение особей.
- Равномерное распределение особей.

# Выбор родителей для скрещивания

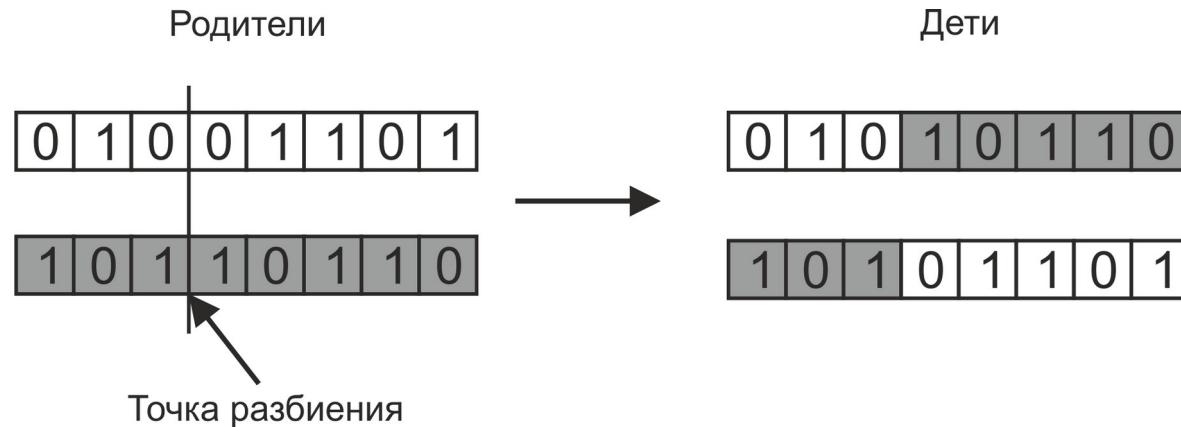
- Панмиксия - оба родителя выбираются случайно, каждая особь популяции имеет равные шансы быть выбранной
- Инбридинг - первый родитель выбирается случайно, а вторым выбирается такой, который **наиболее похож** на первого родителя.
- Аутбридинг - первый родитель выбирается случайно, а вторым выбирается такой, который **наиболее не похож** на первого родителя.
- Метод турнира.
- Метод элиты.

# Операторы скрещивания

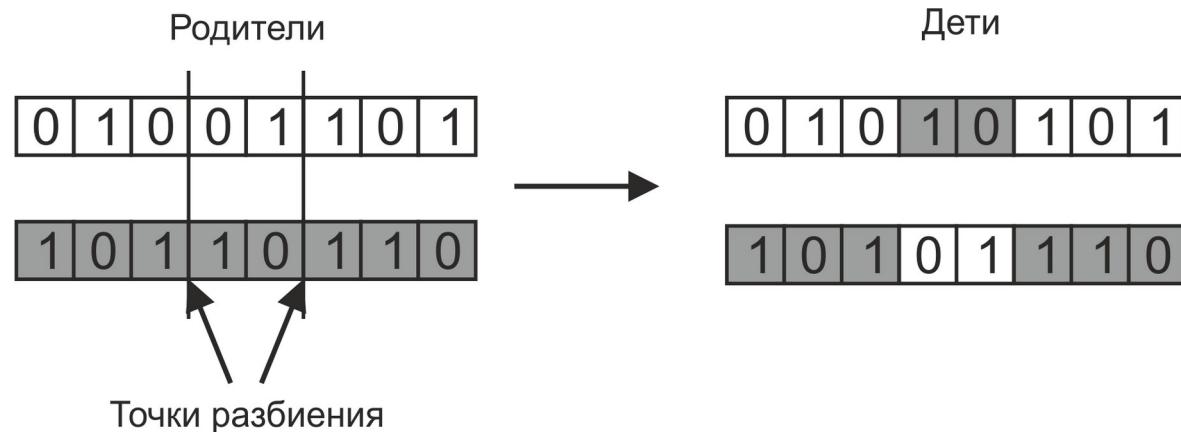
- Расчет среднего арифметического.
- Расчет среднего геометрического.
- Побитовое скрещивание.

# Генетический алгоритм. Операторы скрещивания

## Одноточечное скрещивание



## Многоточечное скрещивание



# Генетический алгоритм. Операторы мутации



## Другие алгоритмы мутации

- Добавление к хромосоме небольшой случайной величины
- Инвертирование всех битов хромосомы
- Замена хромосомы на случайное число

# Операторы отбора

- Метод поддержания постоянного размера популяции
- Метод вероятностного отбора
- Метод турнира
- Метод элиты

## Критерии останова

- Постоянство целевой функции.
- Достижение заданного значения целевой функции.
- Достижение определенного номера поколения.
- Вырождение популяции.

## Недостатки генетического алгоритма

- Не гарантируется нахождение глобального экстремума.
- Использование вероятностных операций.
- Большое количество вычислений целевой функции.
- Большое количество параметров алгоритма.

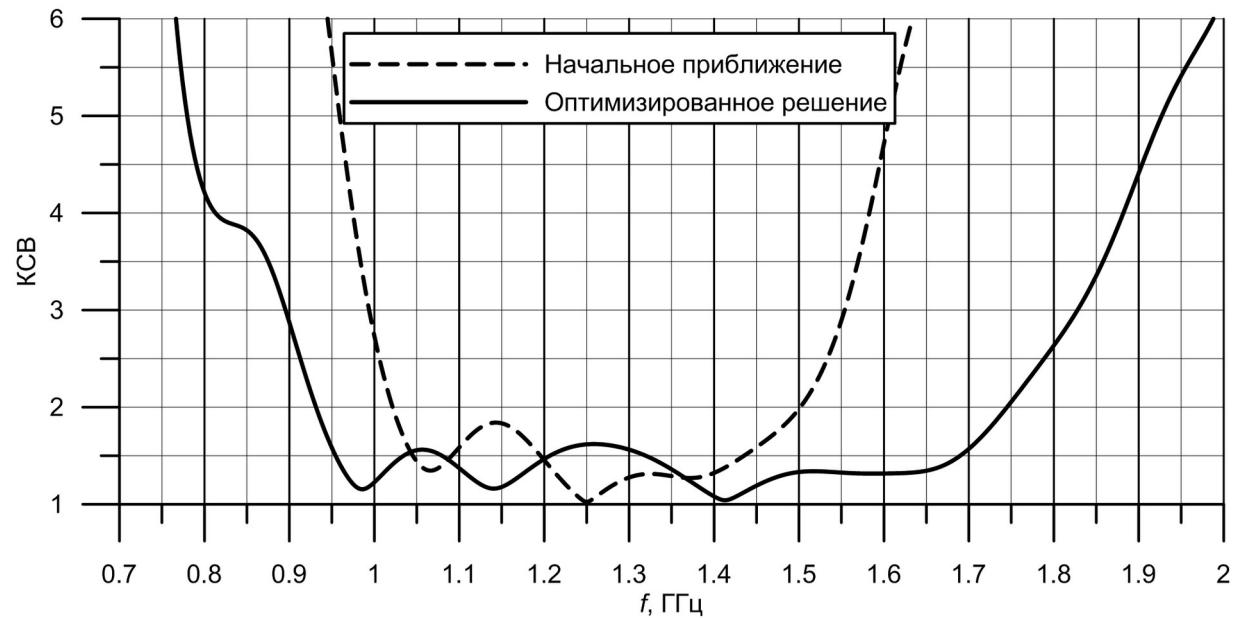
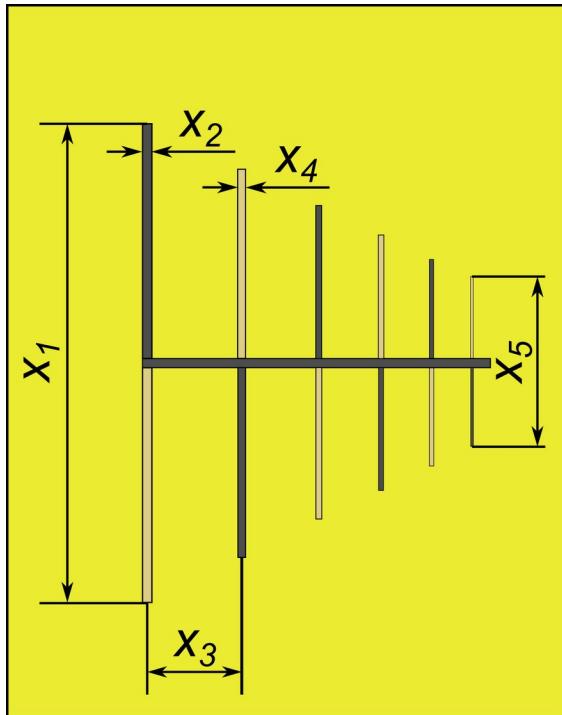
# Модификации генетического алгоритма

- Совместное применение с итерационными алгоритмами.
- Создание нескольких независимых популяций.
- Адаптация параметров алгоритма во время выполнения.
- Использование ГА для определения оптимальных параметров другого ГА.

---

# Примеры использования

# Оптимизация параметров антенны с помощью генетического алгоритма

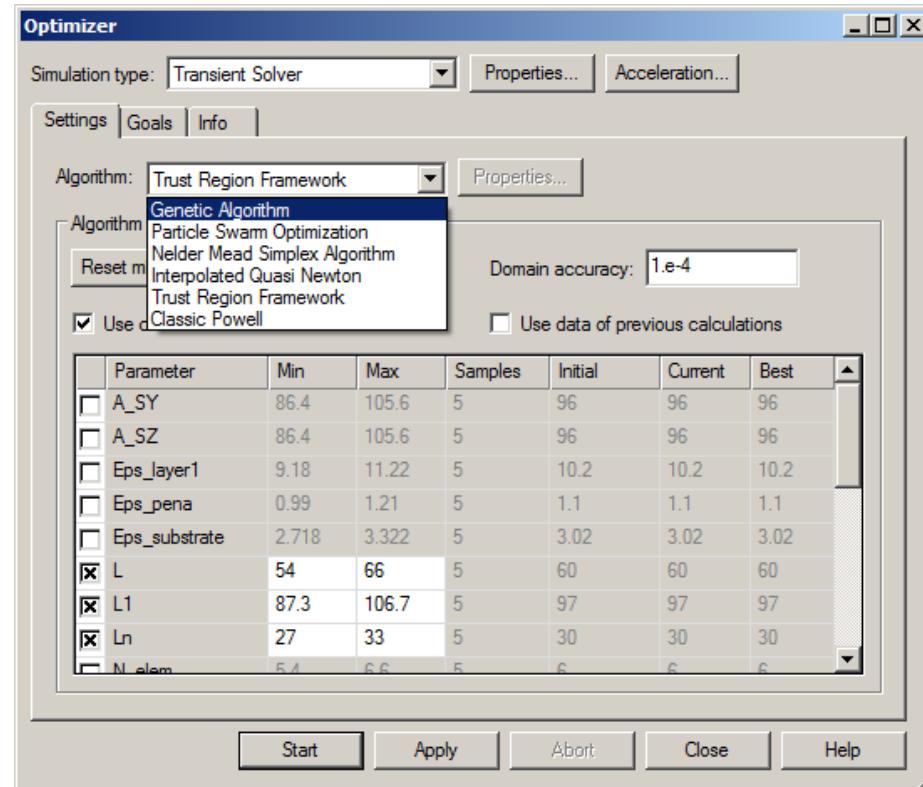
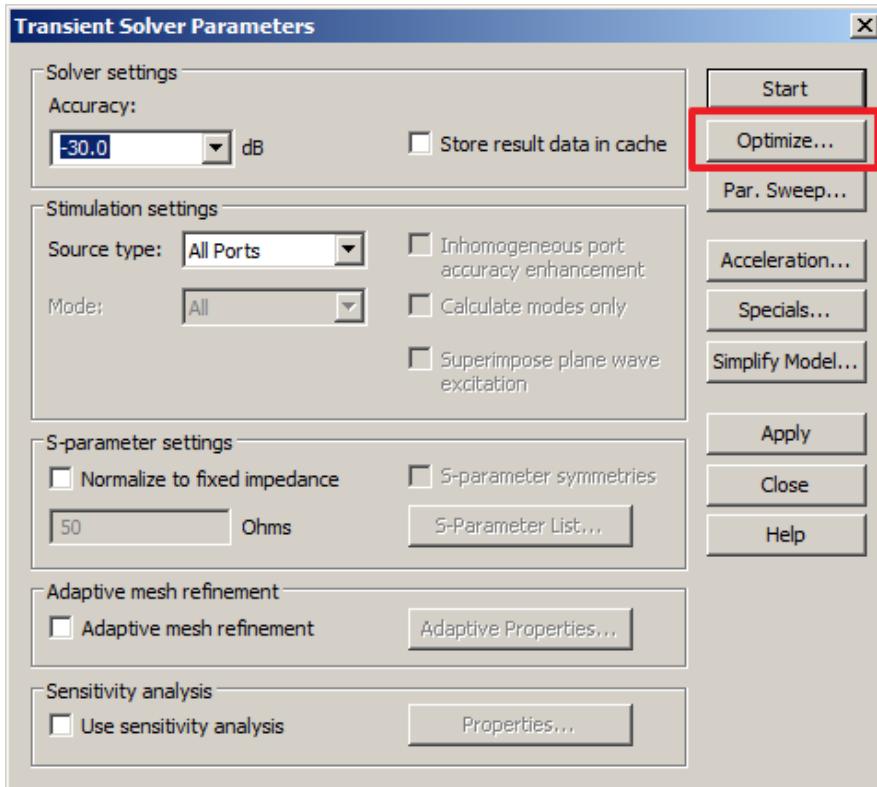


## Параметры алгоритма:

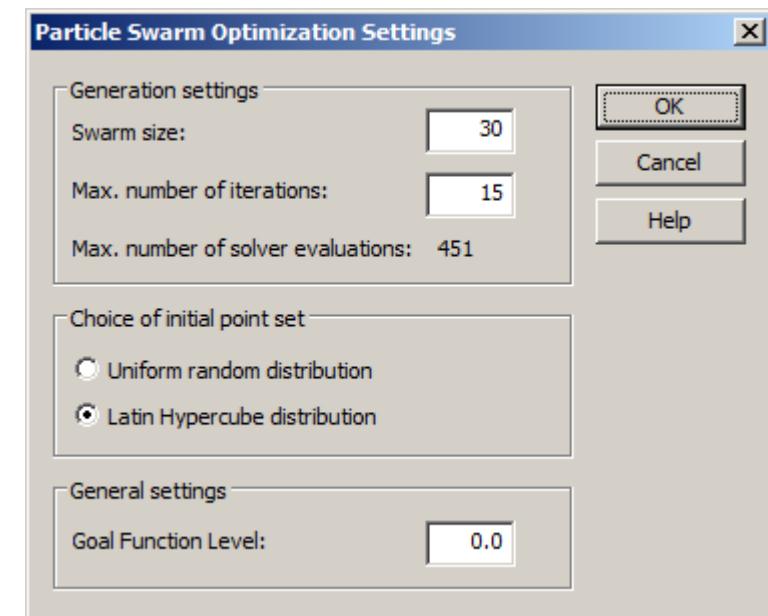
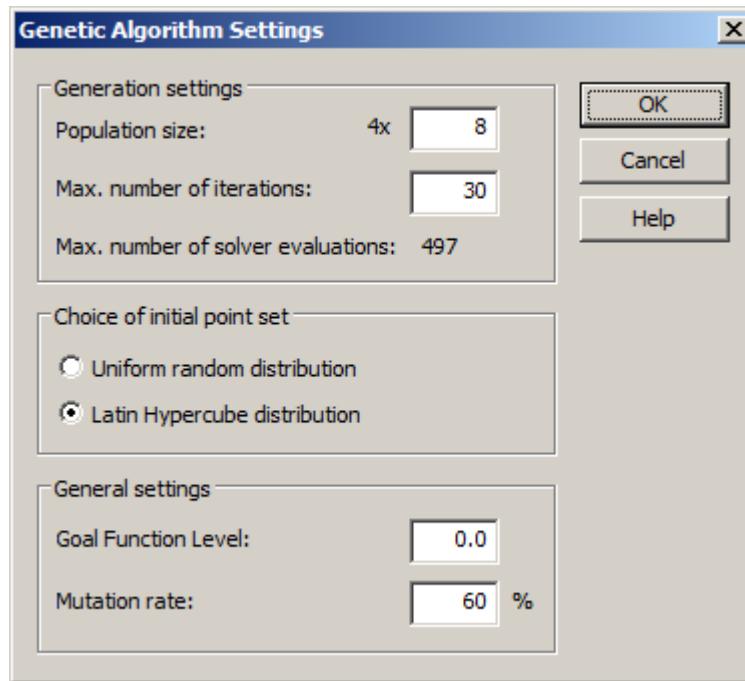
- Минимизация КСВ в заданном диапазоне частот 0.95 – 1.65 ГГц
- Количество оптимизируемых параметров: 3
- Размер популяции: 12 особей
- Вероятность мутации: 60%
- Количество поколений: 30
- Общее количество моделирований: 187
- Время одного расчета: 3 минуты

$$\Phi(\mathbf{X}) = \max (\text{KCB}) \Big|_{f \in [f_{\min}; f_{\max}]}$$

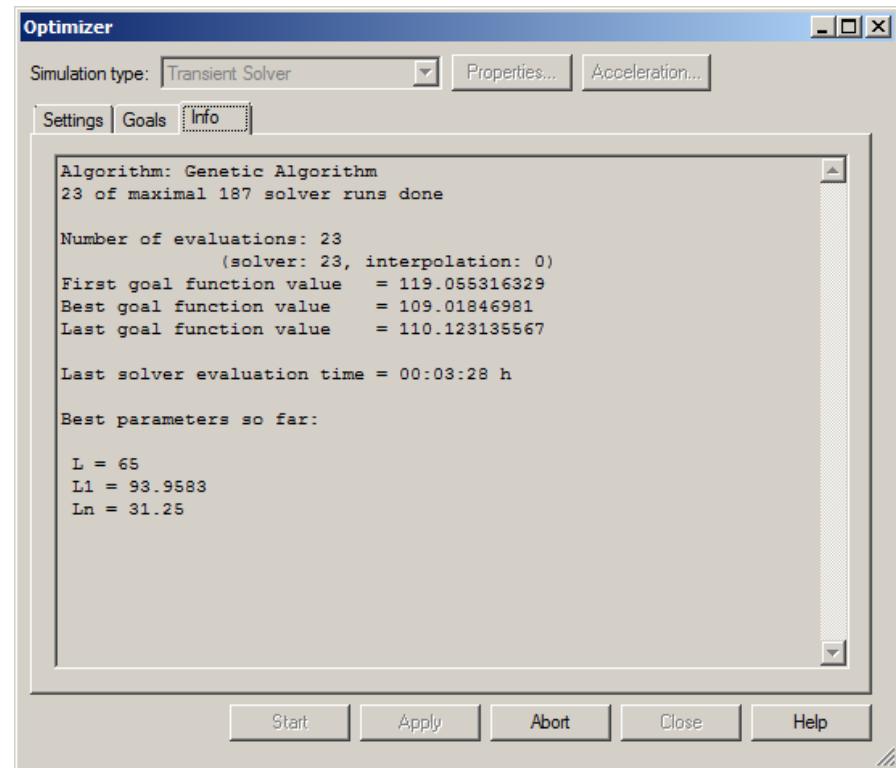
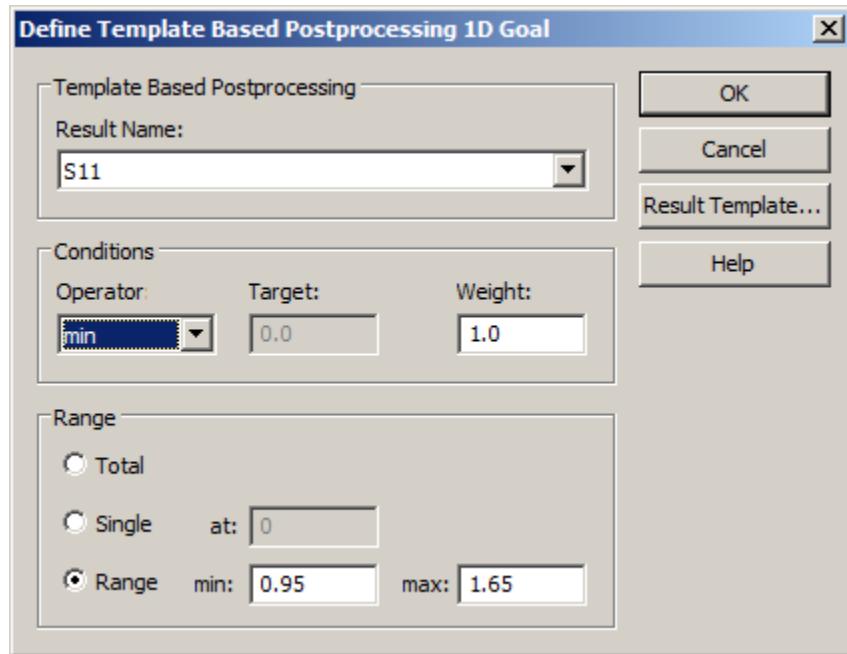
# Алгоритмы глобальной оптимизации в CST Microwave Studio



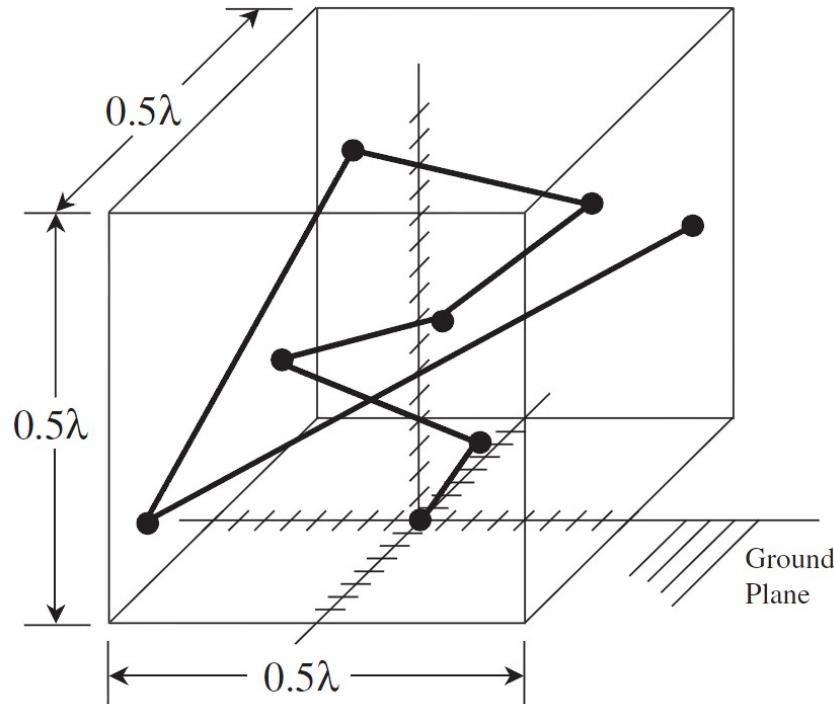
# Алгоритмы глобальной оптимизации в CST Microwave Studio



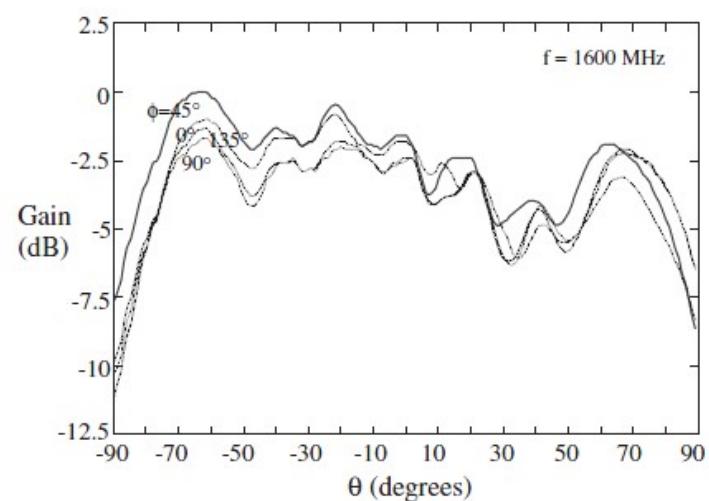
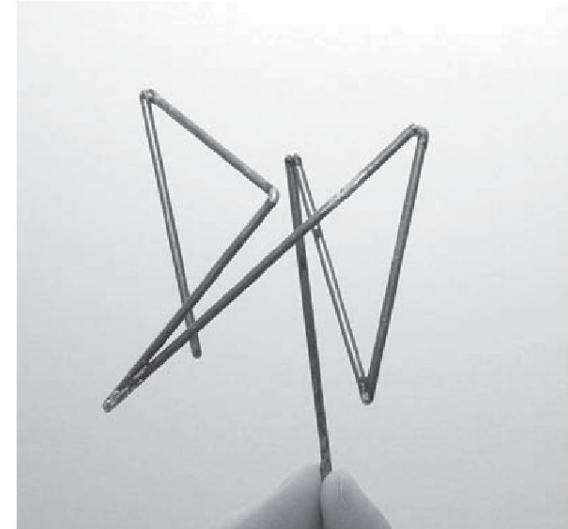
# Алгоритмы глобальной оптимизации в CST Microwave Studio



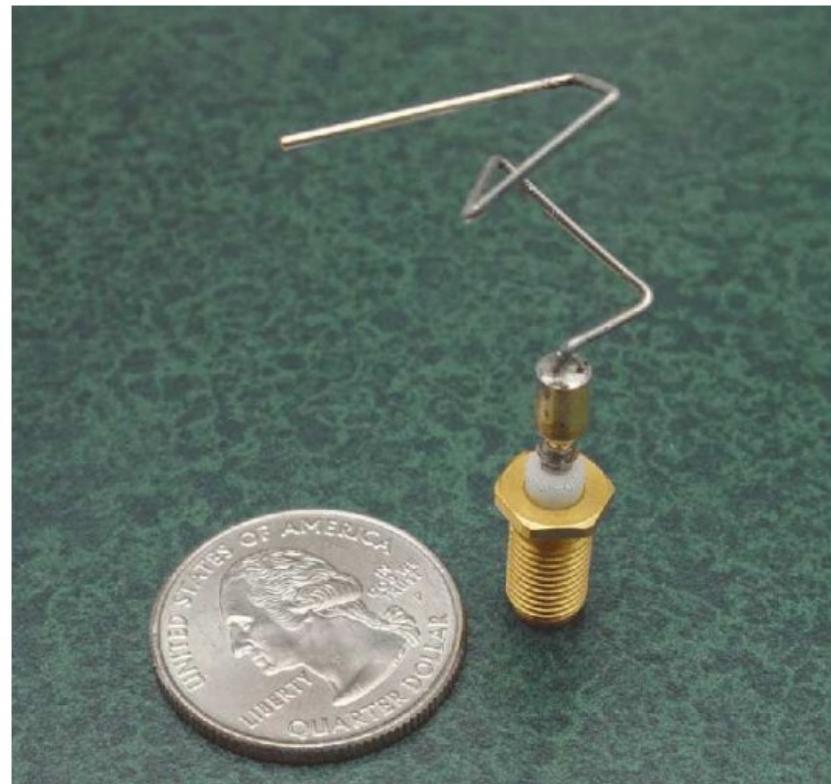
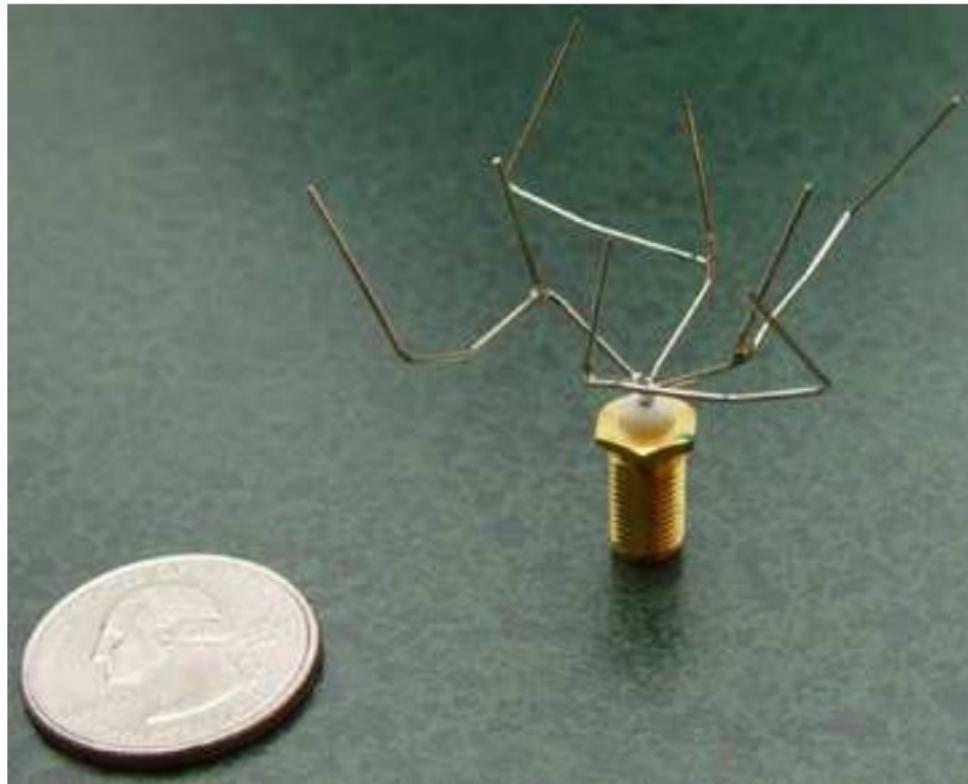
# Пример применения генетического алгоритма



$$cost = \sum_{\text{over all } \theta, \phi} [gain(\theta, \phi) - average\ gain]^2$$

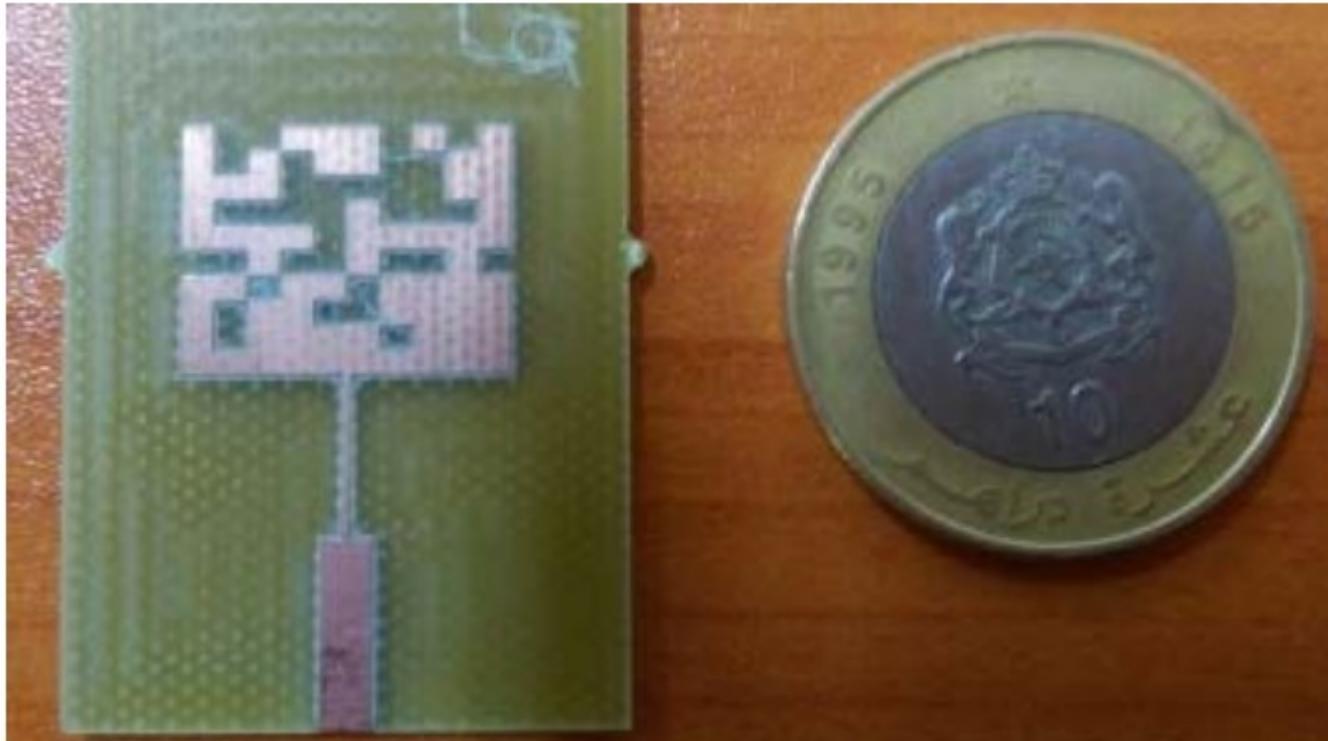


# Пример применения генетического алгоритма



Hornby, Greg & Globus, Al & Linden, Derek & Lohn, Jason. (2006). Automated Antenna Design with Evolutionary Algorithms. Collection of Technical Papers - Space 2006 Conference. 1. 10.2514/6.2006-7242.

# Пример применения генетического алгоритма



Mohammed Lamsalli, Abdelouahab El Hamichi, Mohamed Boussouis, Naima A. Touhami, and Taj-eddin Elhamadi.  
Genetic Algorithm Optimization for Microstrip Patch Antenna Miniaturization. Progress In Electromagnetics  
Research Letters, Vol. 60, 113–120, 2016

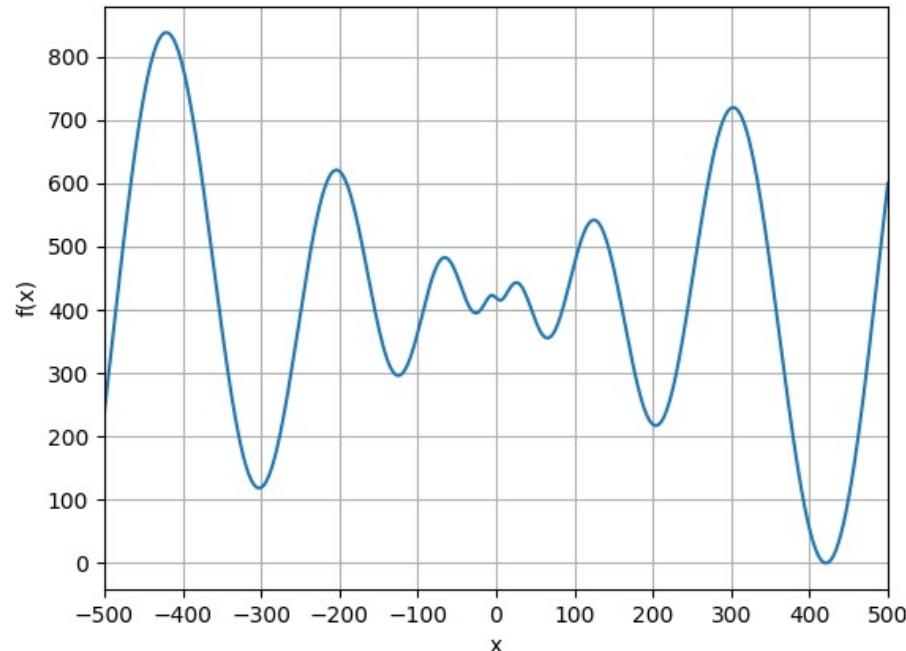
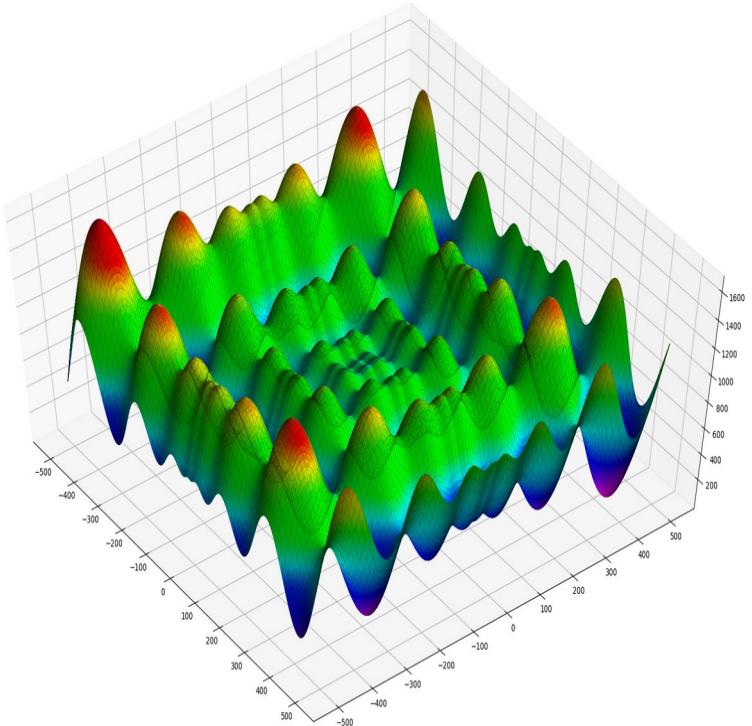
# Реализация генетического алгоритма на C#

<https://jenyay.net/Programming/Genetic>

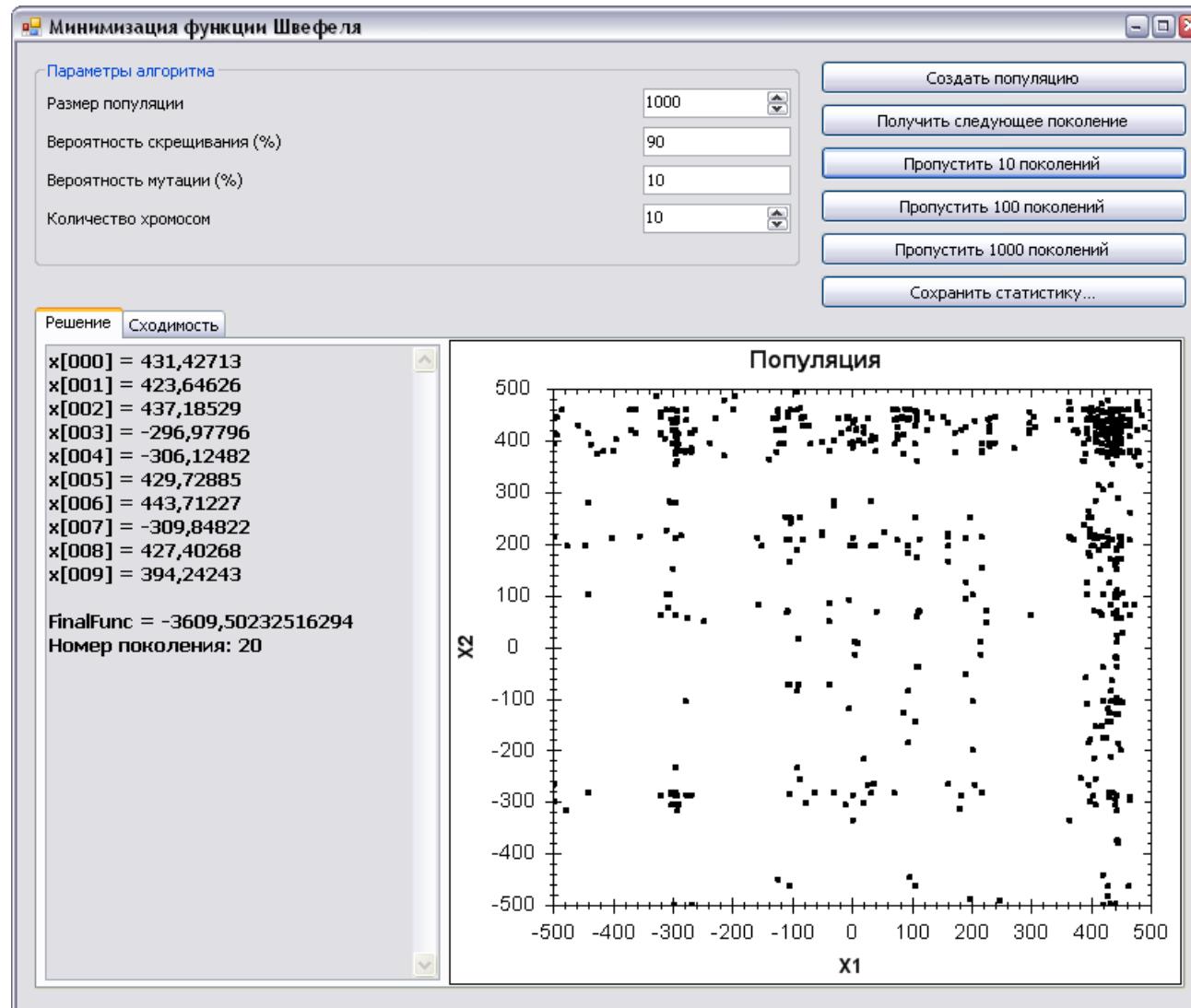
# Тестовые функции. Функция Швефеля (Schwefel function)

$$f(\mathbf{x}) = 418.9829n + \sum_{i=1}^n \left( -x_i \sin \left( \sqrt{|x_i|} \right) \right)$$

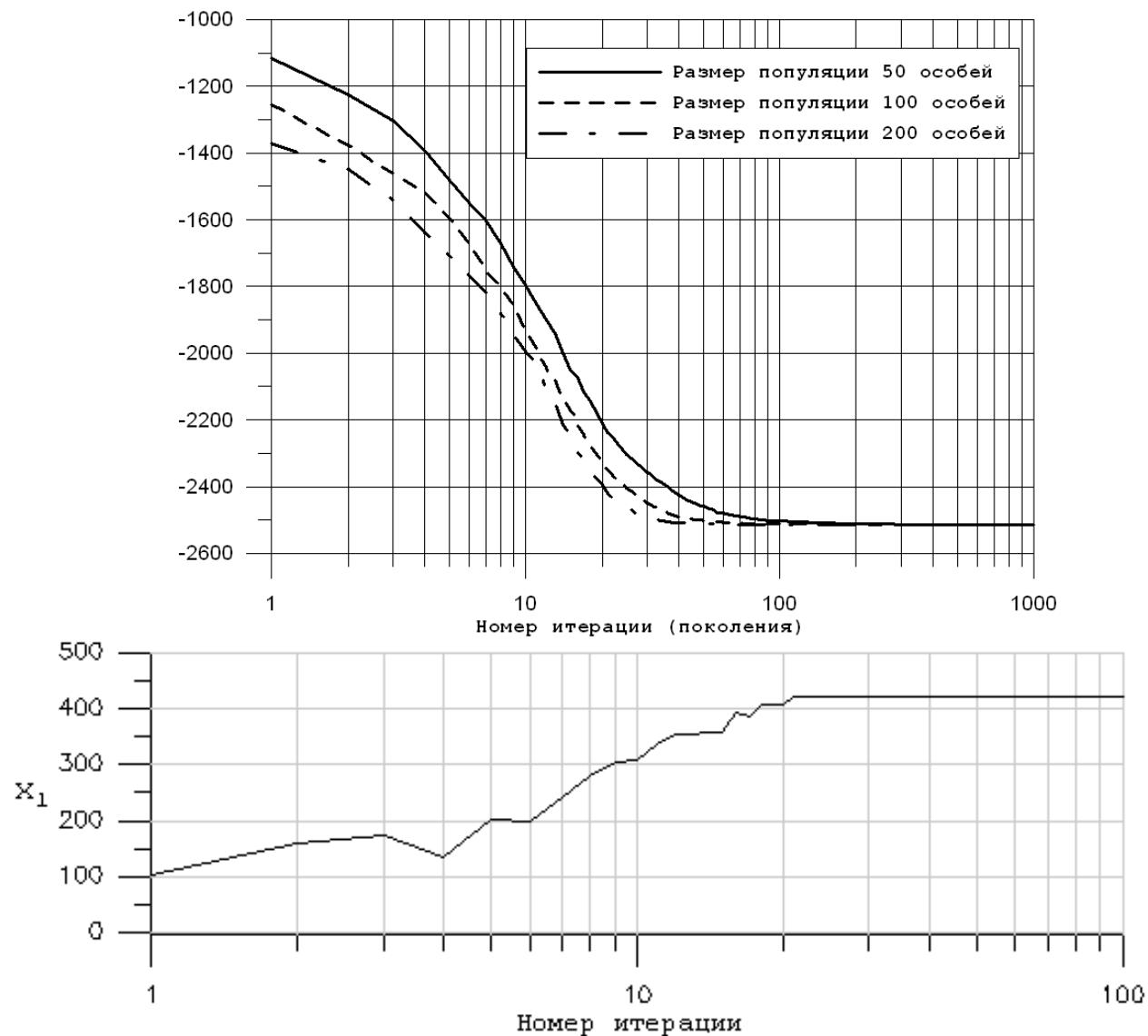
Глобальный минимум:  $f(\mathbf{x}) = 0$  при  
 $x_i = 420.9687, i = 1, \dots, n; -500 \leq x_i \leq 500$



# Демонстрация работы генетического алгоритма



# Сходимость генетического алгоритма



## Ссылки

**Библиотека для реализации генетических алгоритмов на языке C#:**

<https://jenyay.net/Programming/Genetic>

**Реализация генетического алгоритма оптимизации на языке Rust:**

<https://jenyay.net/Programming/OptlibGenetic>

**Выступление в Московском клубе программистов с докладом про генетический алгоритм:**

<https://www.youtube.com/watch?v=89Wk0kNnbJQ>

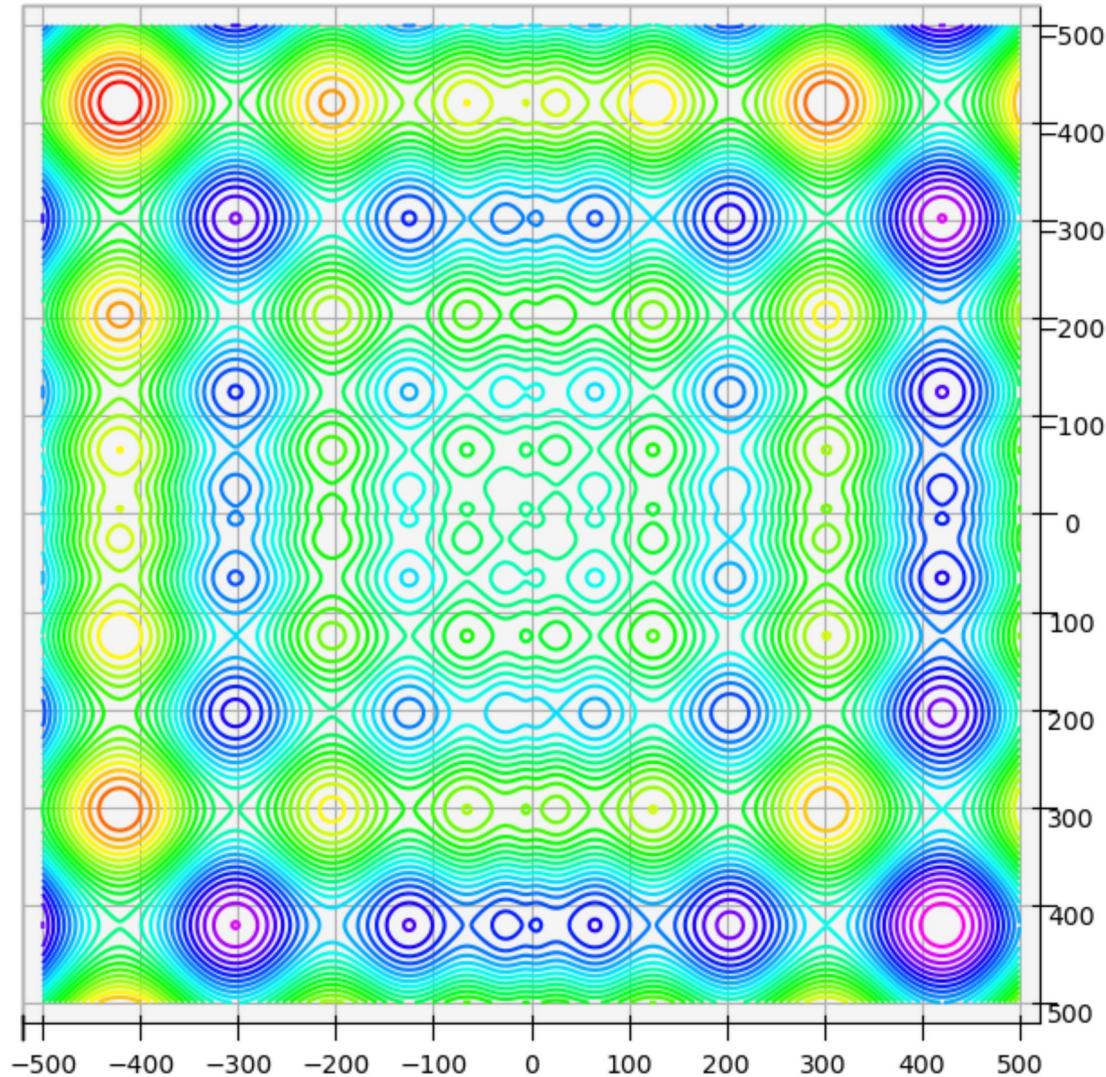
# Алгоритм роя частиц

Kennedy J., Eberhart R. "Particle Swarm Optimization".  
Proceedings of IEEE International Conference  
on Neural Networks. IV. 1995

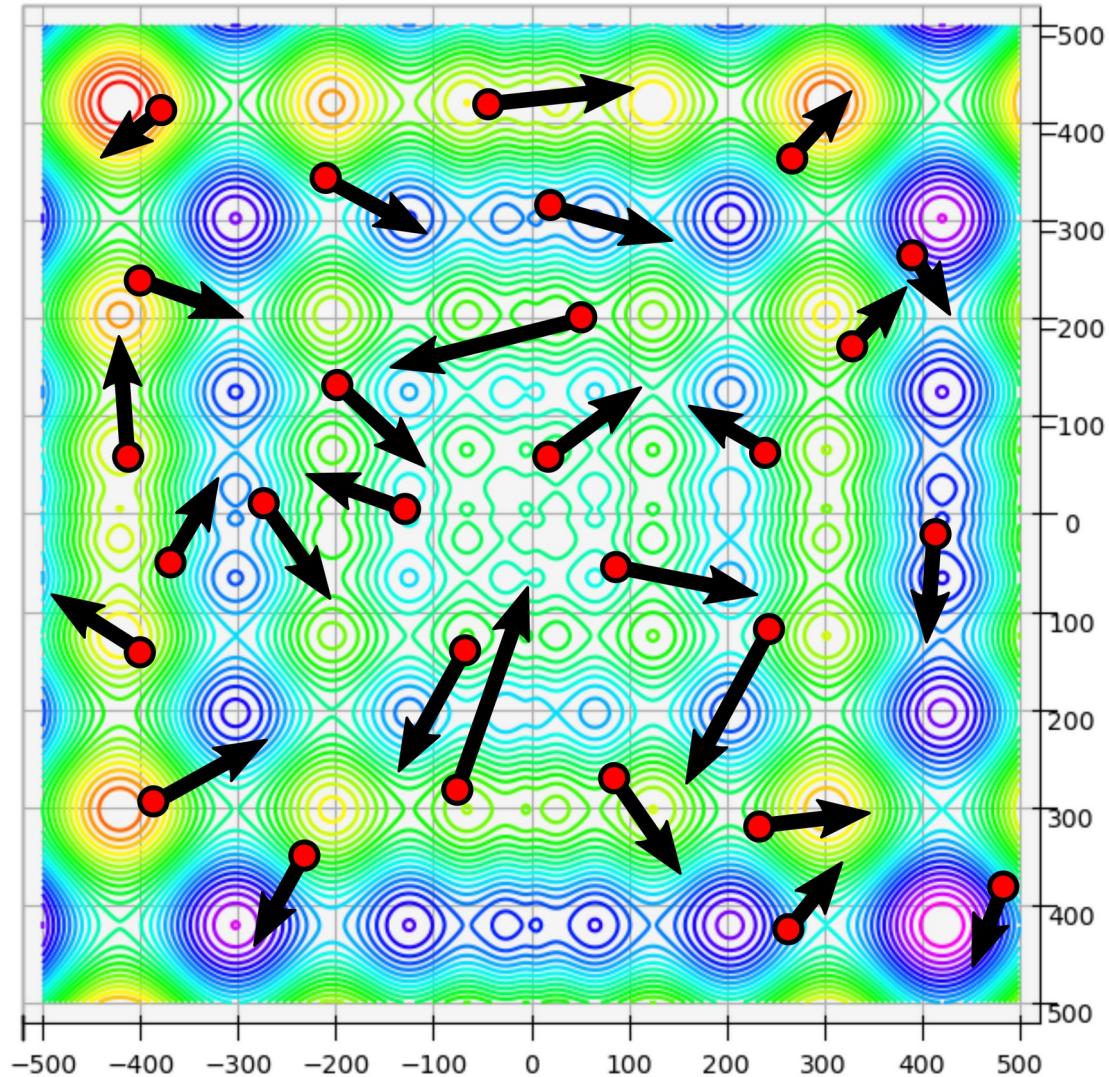
# Идея алгоритма



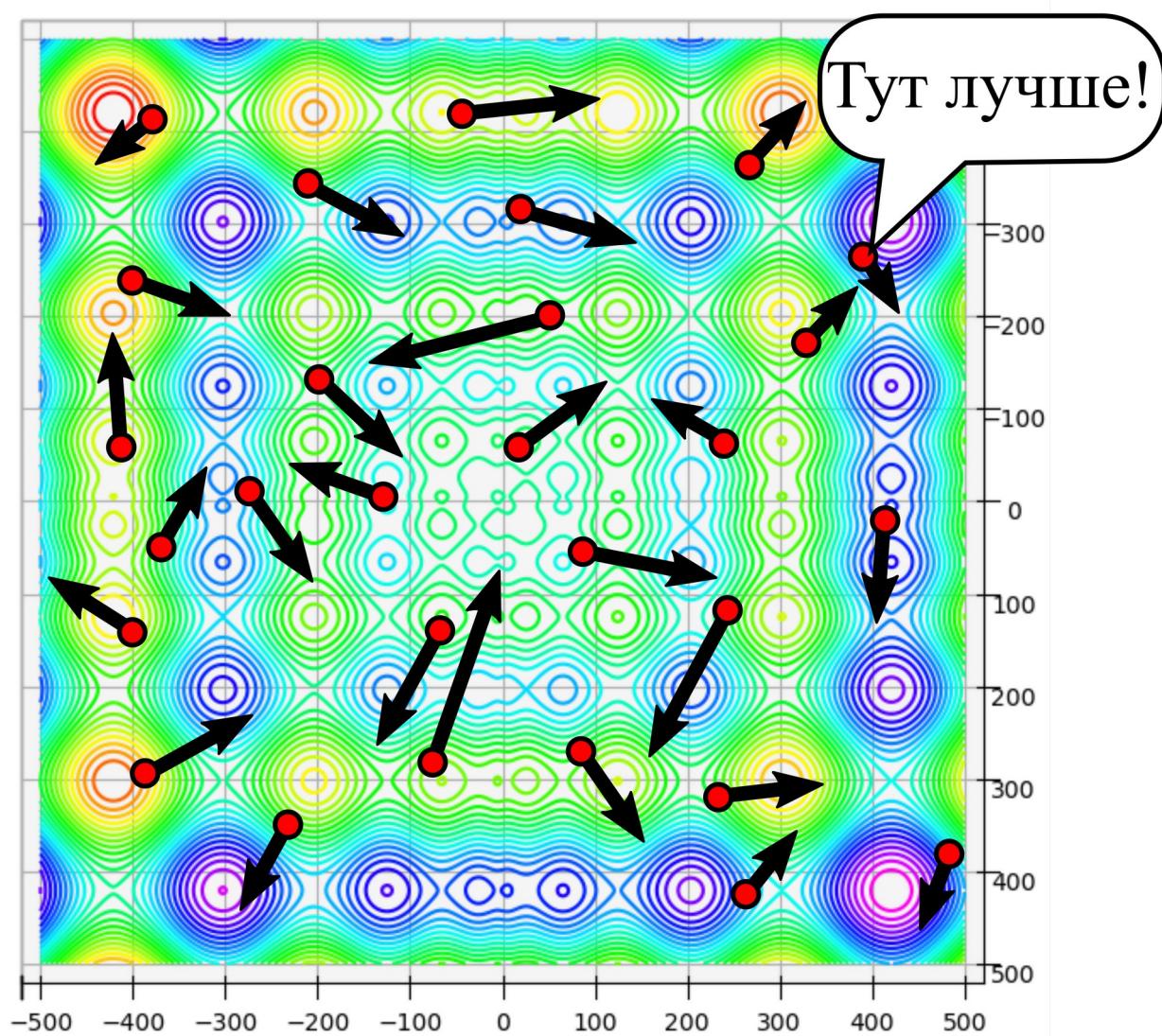
# Алгоритм роя частиц



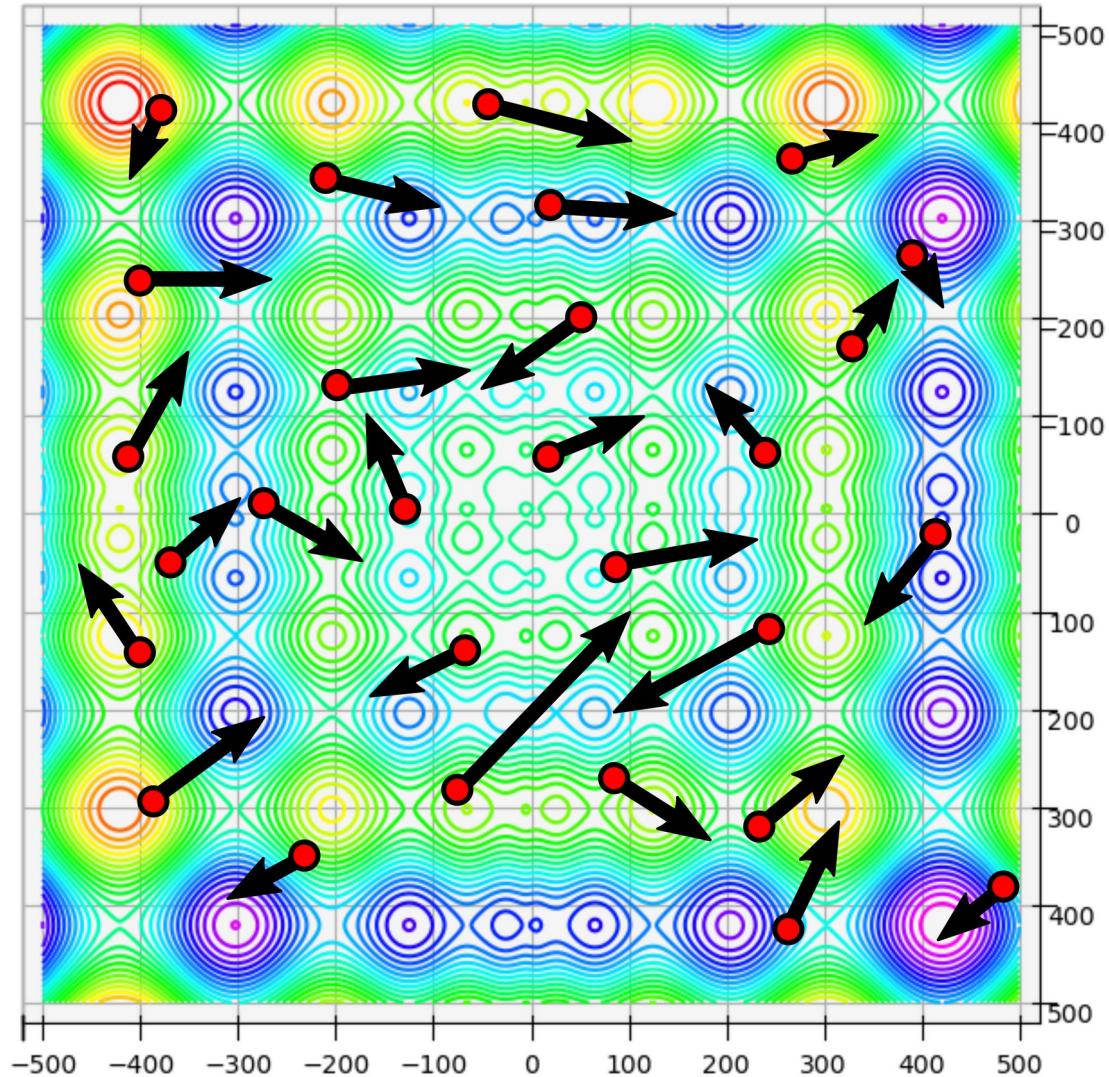
# Алгоритм роя частиц



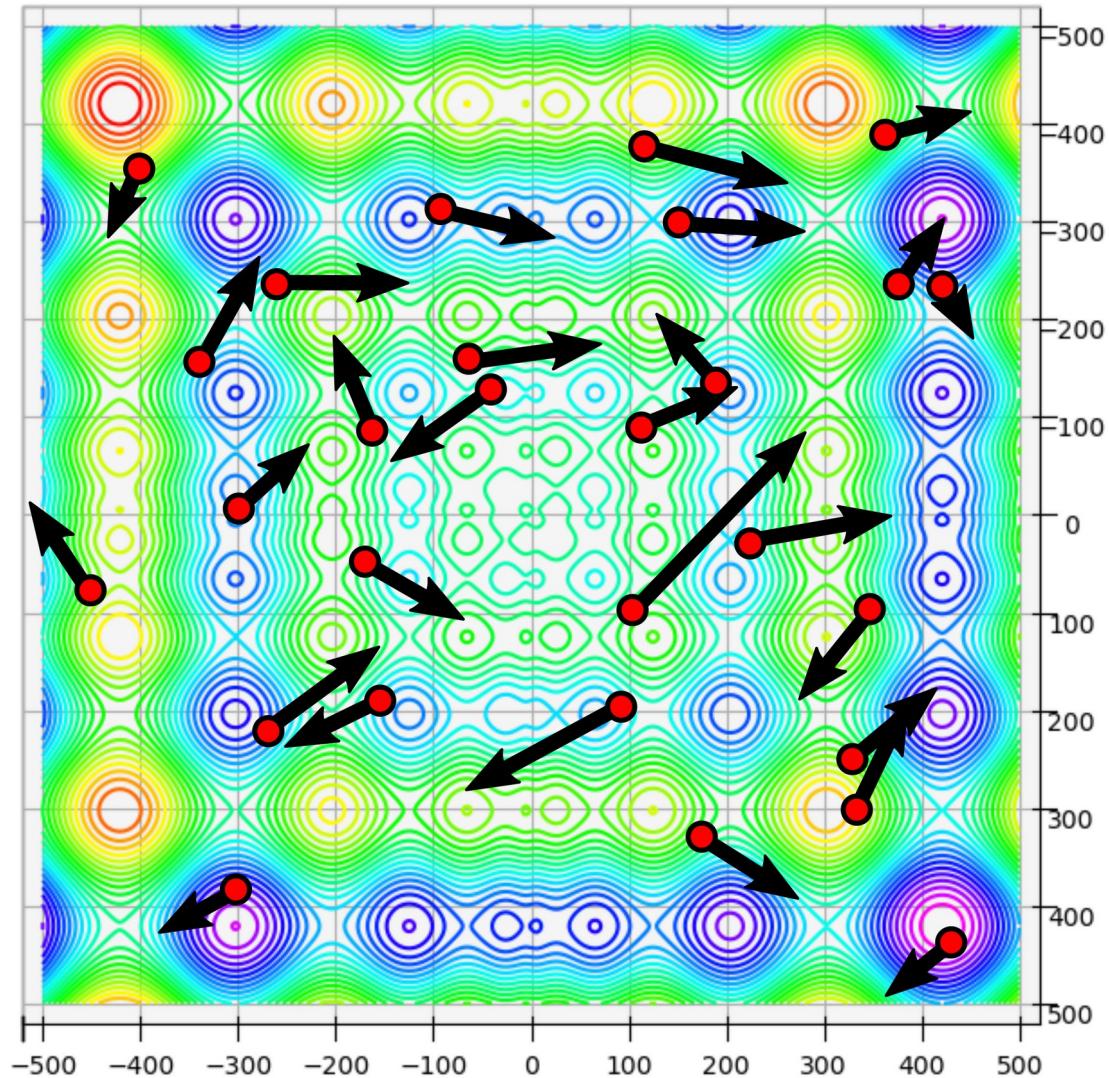
# Алгоритм роя частиц



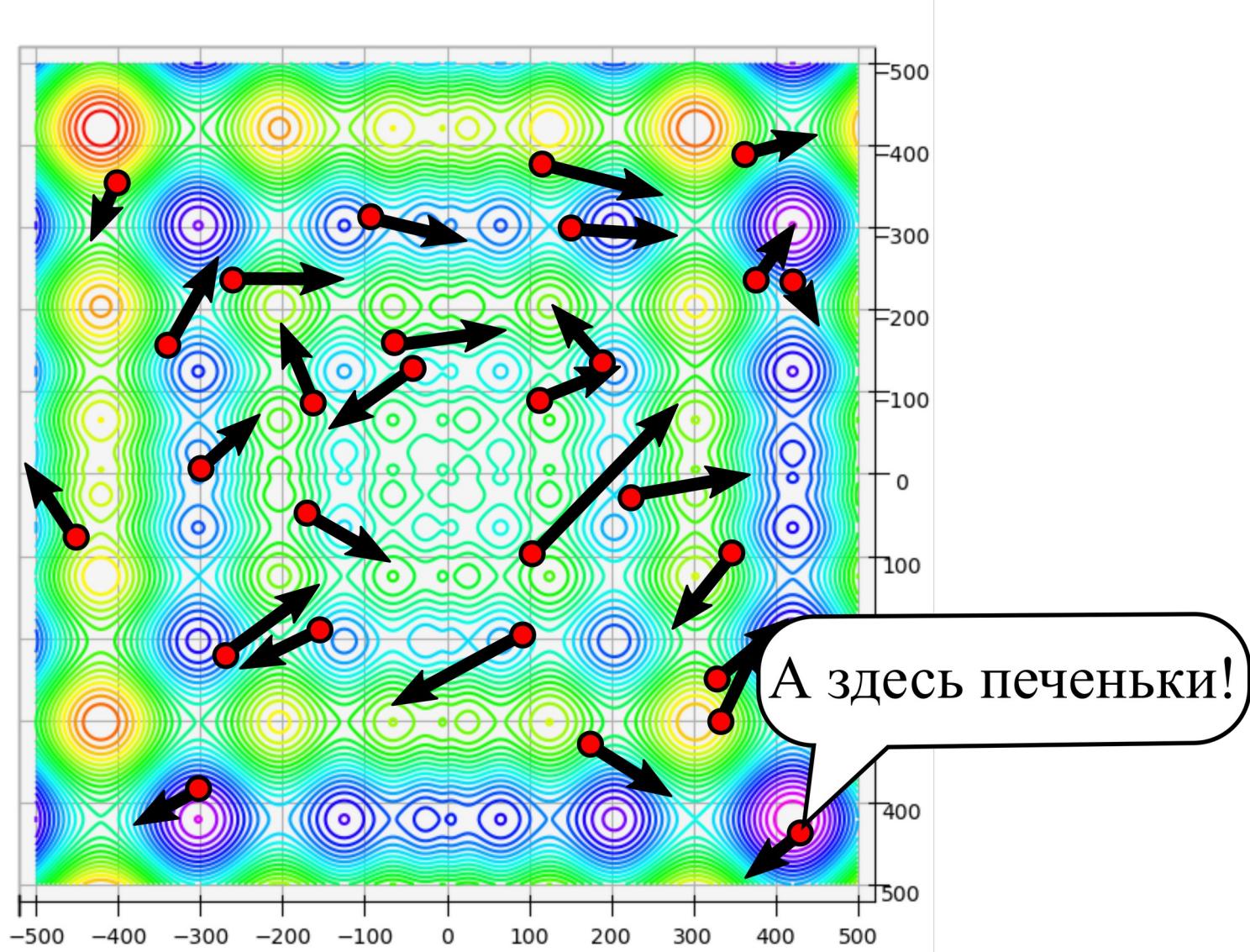
# Алгоритм роя частиц



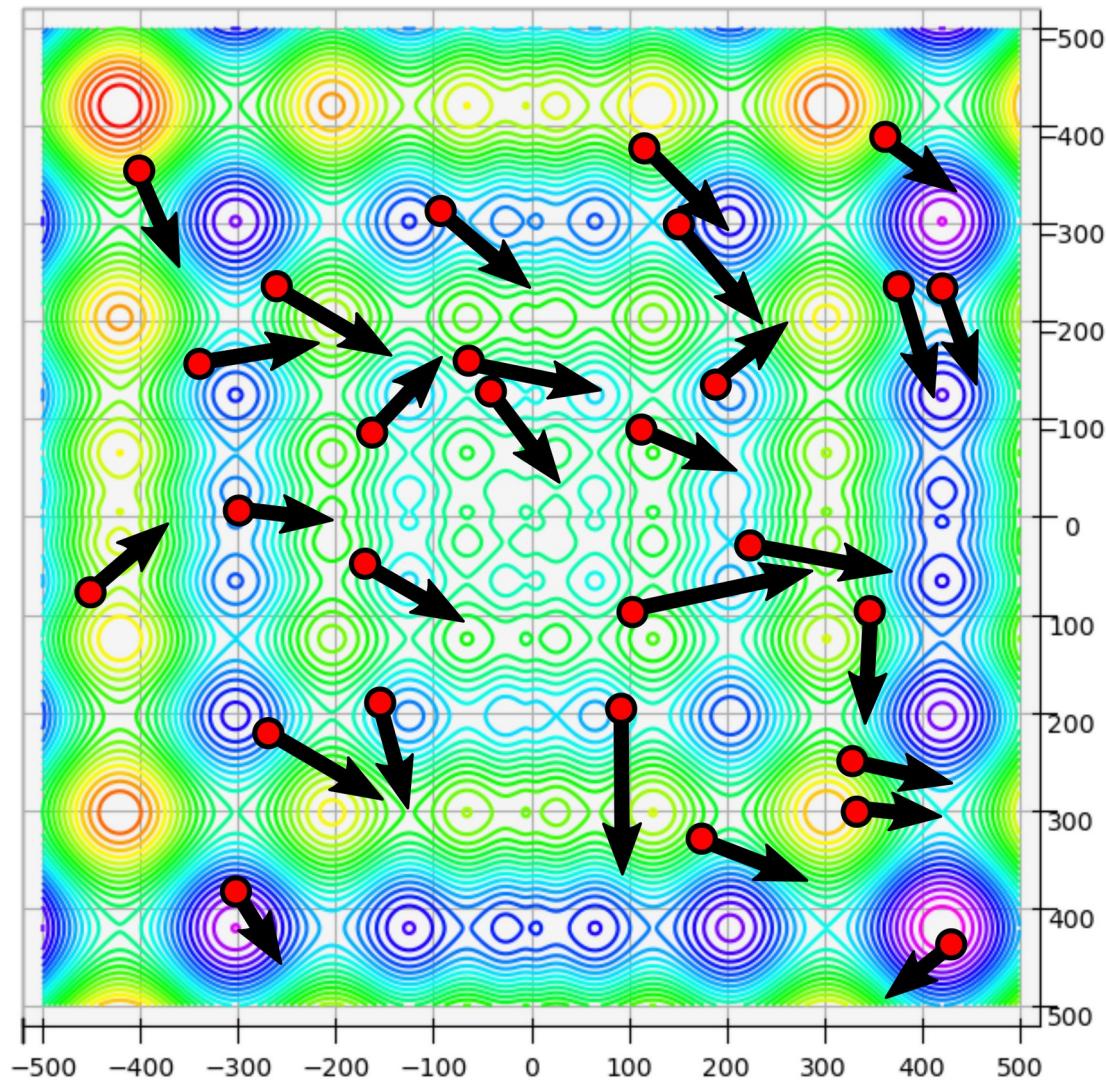
# Алгоритм роя частиц



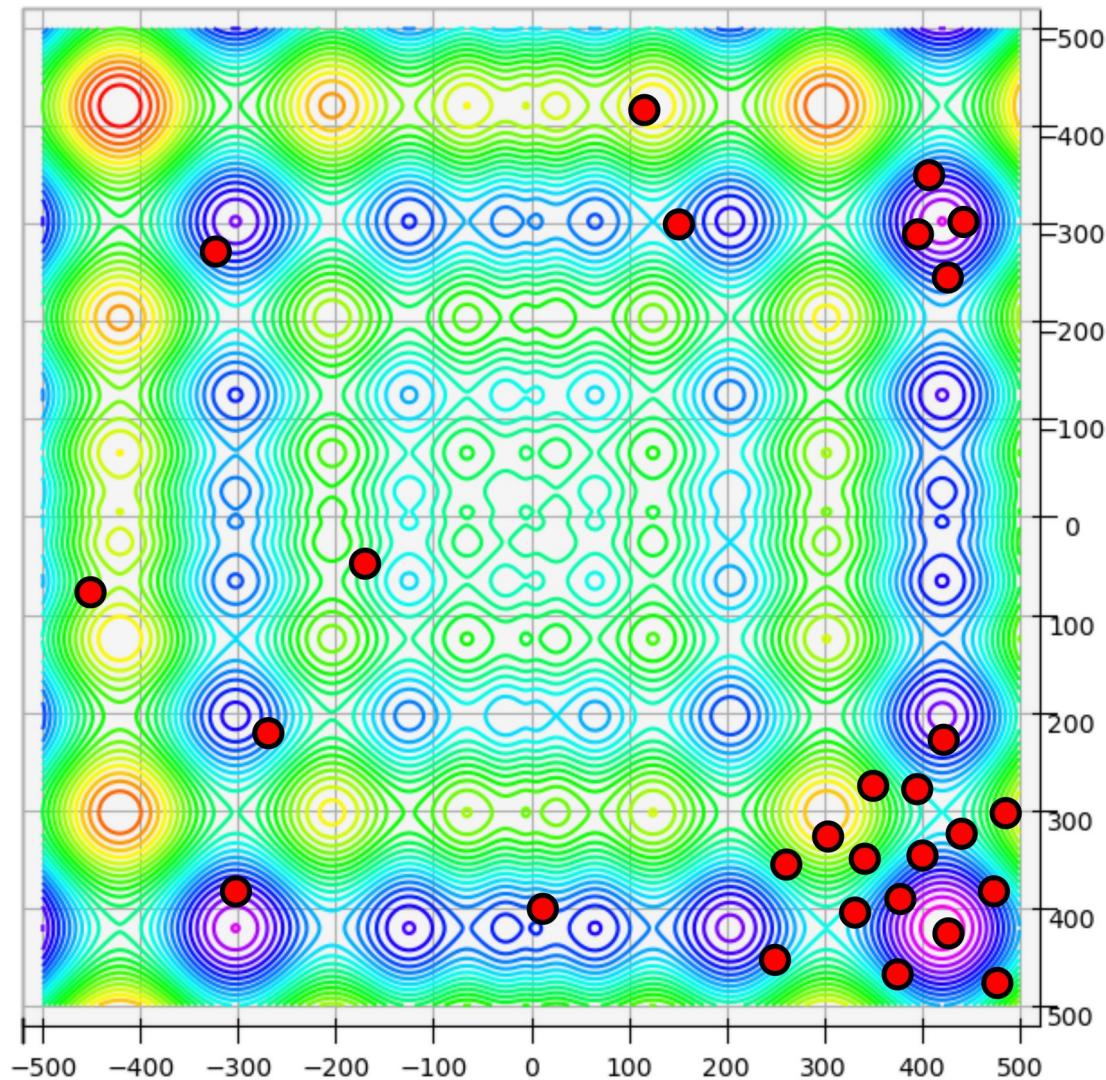
# Алгоритм роя частиц



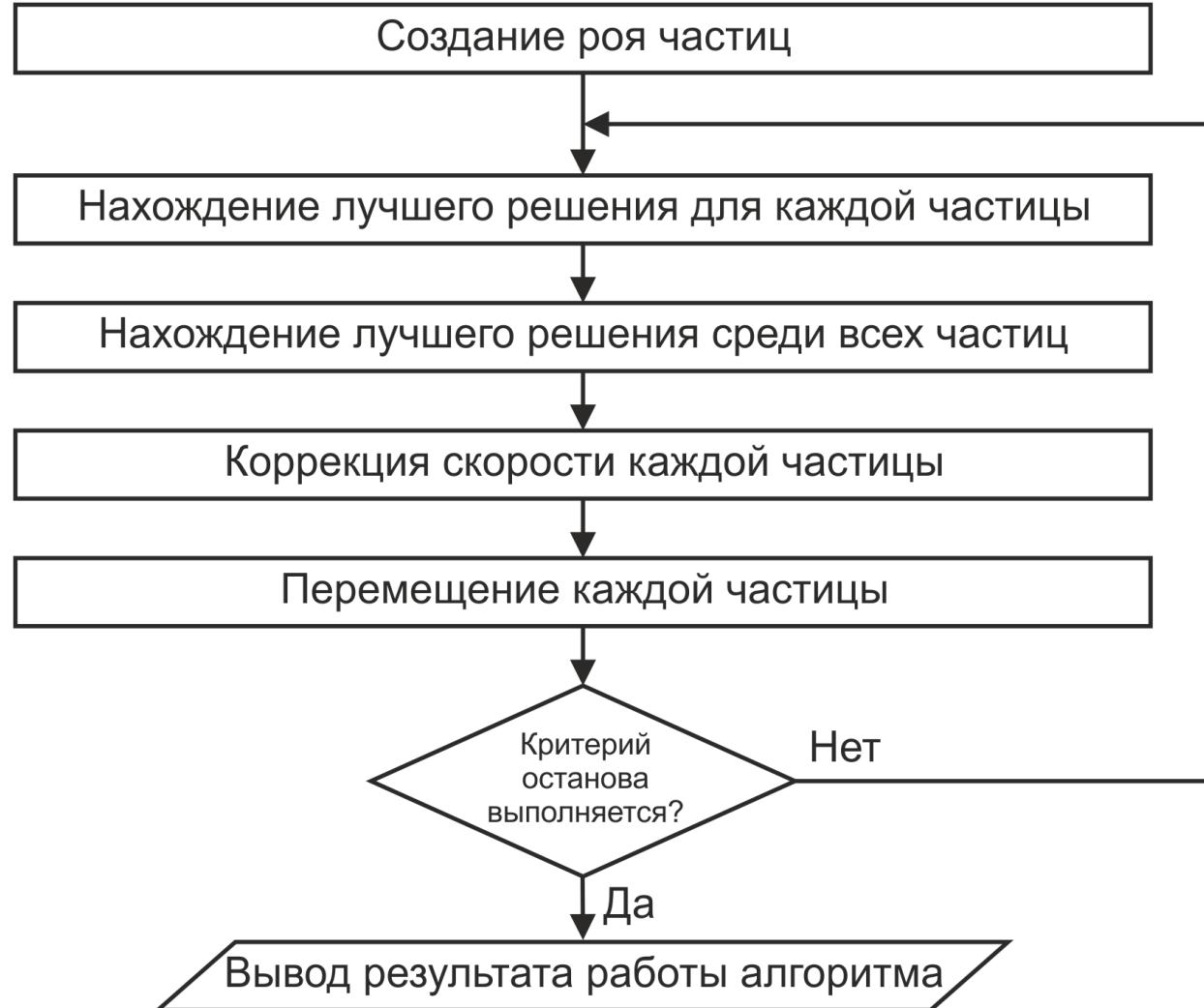
# Алгоритм роя частиц



# Алгоритм роя частиц



# Алгоритм роя частиц



## Критерии останова

- Постоянство целевой функции.
- Достижение заданного значения целевой функции.
- Достижение определенного номера итерации.
- Все частицы сошлись в одну точку.

## Расчет скорости частиц. Классический алгоритм

$$\mathbf{v}_{i,t+1} = \mathbf{v}_{i,t} + \underbrace{\phi_p r_p (\mathbf{p}_{i,t} - \mathbf{x}_{i,t})}_{\Delta v_{\text{собств}}} + \underbrace{\phi_g r_g (\mathbf{g}_t - \mathbf{x}_{i,t})}_{\Delta v_{\text{глоб}}}$$

$\mathbf{v}_{i,t}$  – вектор скорости  $i$ -й частицы при  $t$ -й итерации алгоритма.

$\mathbf{x}_{i,t}$  – вектор координат  $i$ -й частицы при  $t$ -й итерации алгоритма.

$\mathbf{p}_i$  – вектор координат лучшего решения, найденного  $i$ -й частицей.

$\mathbf{g}$  – вектор координат лучшего решения, найденного всеми частицами при  $t$ -й итерации алгоритма.

$r_p, r_g$  – случайные числа в интервале  $(0, 1)$ .

$\phi_p, \phi_g$  – весовые коэффициенты.

# Модификации алгоритма роя частиц. Реализация «LBEST»

$$\mathbf{v}_{i,t+1} = \mathbf{v}_t + \phi_p r_p (\mathbf{p}_{i,t} - \mathbf{x}_{i,t}) + \phi_l r_l (\underline{\mathbf{l}}_{i,t} - \mathbf{x}_{i,t})$$

$\mathbf{l}_t$  – вектор координат лучшего решения, найденного среди соседних частиц.

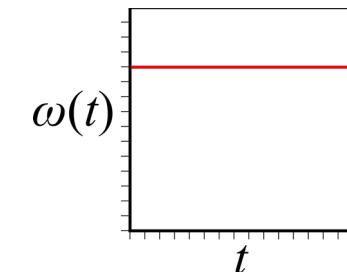
# Модификации алгоритма роя частиц. Реализация с учетом инерции

$$\mathbf{v}_{i,t+1} = \underline{\omega(t)} \mathbf{v}_{i,t} + \phi_p r_p (\mathbf{p}_{i,t} - \mathbf{x}_{i,t}) + \phi_g r_g (\mathbf{g}_t - \mathbf{x}_{i,t})$$

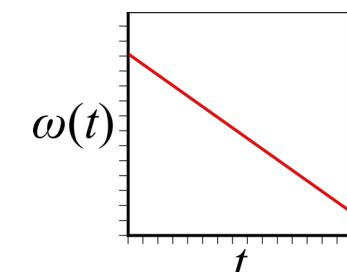
$\omega(t)$  – коэффициент инерции

# Примеры зависимостей коэффициента инерции от номера итерации

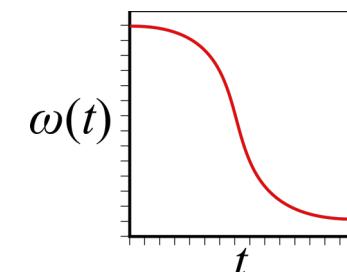
1. Постоянное значение.  $w(t) = c$



2. Линейное убывание.  $w(t) = w_{max} - \frac{w_{max} - w_{min}}{t_{max}} t$



3. Сигмоид.  $w(t) = \frac{w_{start} - w_{end}}{1 + e^{-u(t - n \cdot t_{max})}} + w_{end}$ ,  $u = 10^{\log(t_{max}) - 2}$



Bansal, Singh, Pramod et al. **Inertia Weight Strategies in Particle Swarm Optimization**. Proceedings of the 2011 3rd World Congress on Nature and Biologically Inspired Computing, NaBIC 2011. 633-640.  
10.1109/NaBIC.2011.6089659.

# Алгоритм роя частиц с использованием коэффициента сужения

$$\mathbf{v}_{i,t+1} = \underline{K} \left( \mathbf{v}_{i,t} + \phi_p r_p (\mathbf{p}_{i,t} - \mathbf{x}_{i,t}) + \phi_g r_g (\mathbf{g}_t - \mathbf{x}_{i,t}) \right)$$

$$K = \frac{2\alpha}{\phi_p + \phi_g - 2}$$

$$\phi_p + \phi_g > 4$$

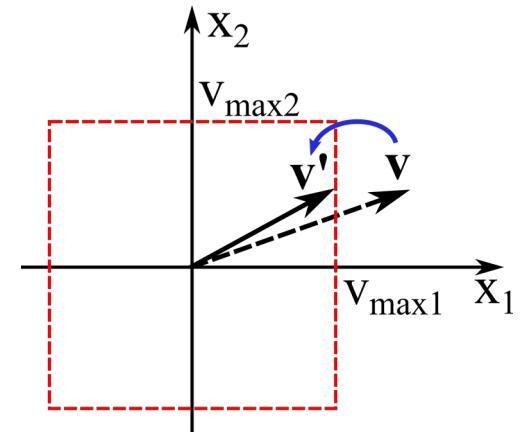
$$\alpha \in (0, 1), \text{ обычно } \alpha \approx 0.9$$

# Способы ограничения скорости частиц

1. Ограничение проекций скорости по каждой координате.

**if**  $v_{ik} > v_{\max k}$  **then**

$$v_{ik} = v_{\max k}$$



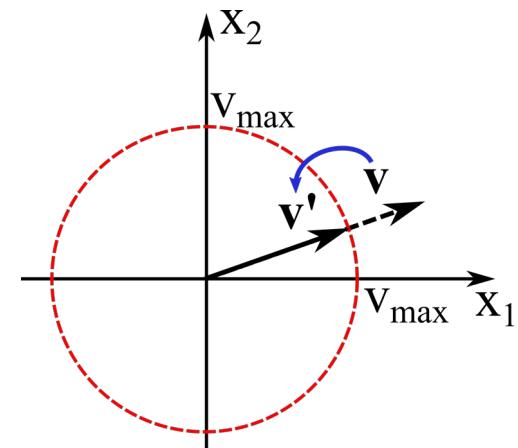
2. Ограничение модуля скорости.

**if**  $|v_i| > v_{\max}$  **then**

$$v_i = v_i \cdot v_{\max} / |v_i|$$

i — номер частицы;

k — номер размерности.



# Перемещение частиц

$$\mathbf{X}_{i,t+1} = \mathbf{X}_{i,t} + \mathbf{v}_{i,t+1}$$

# Реализация алгоритма роя частиц на языке C#

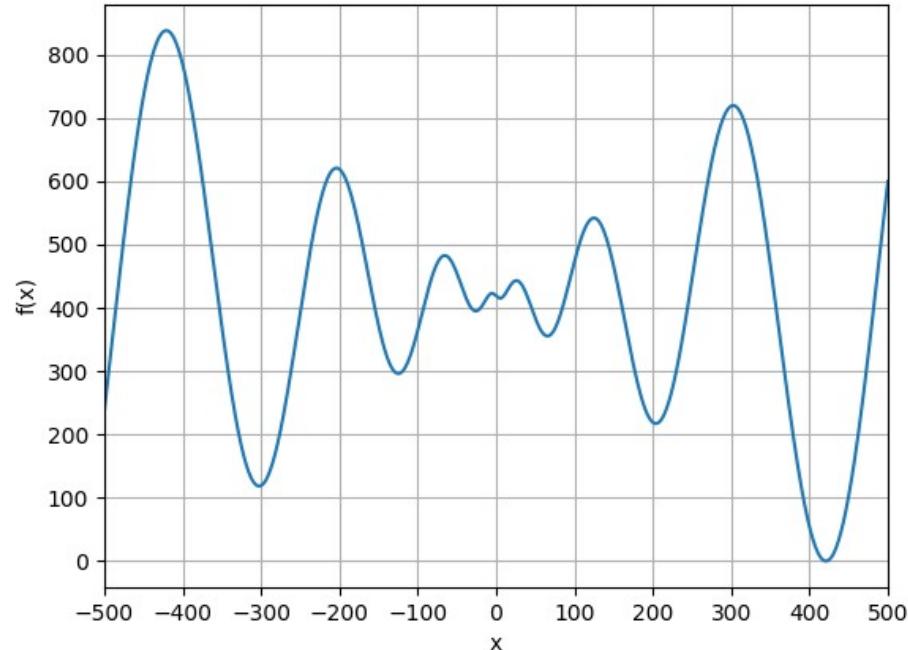
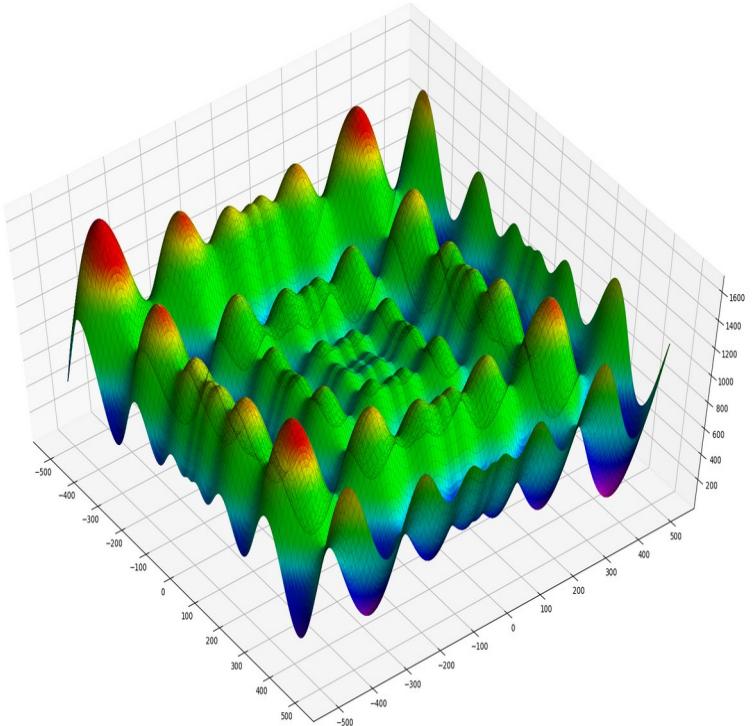
<https://jenyay.net/Programming/ParticleSwarm>



# Тестовая функция. Функция Швефеля (Schwefel function)

$$f(\mathbf{x}) = 418.9829n + \sum_{i=1}^n \left( -x_i \sin \left( \sqrt{|x_i|} \right) \right)$$

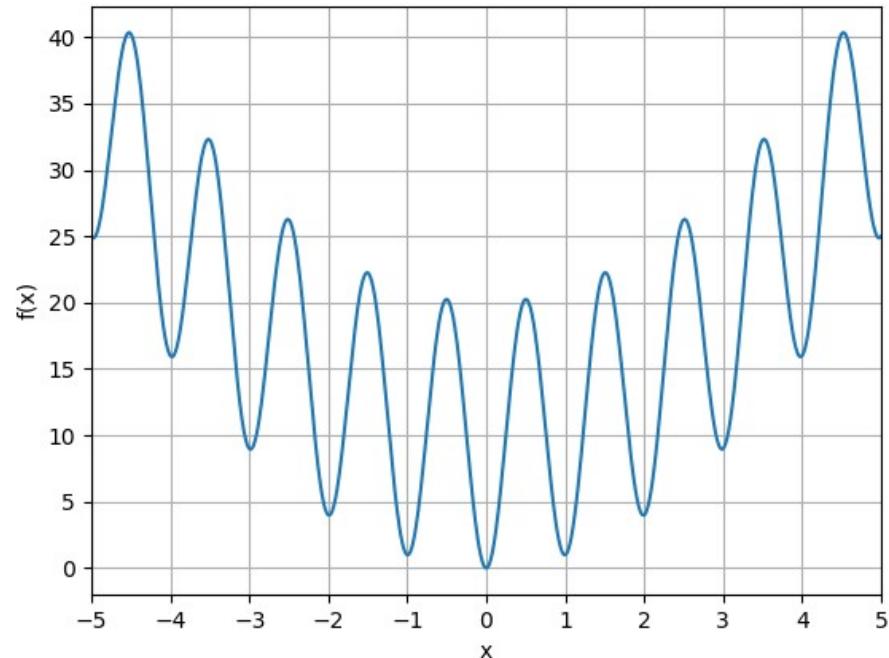
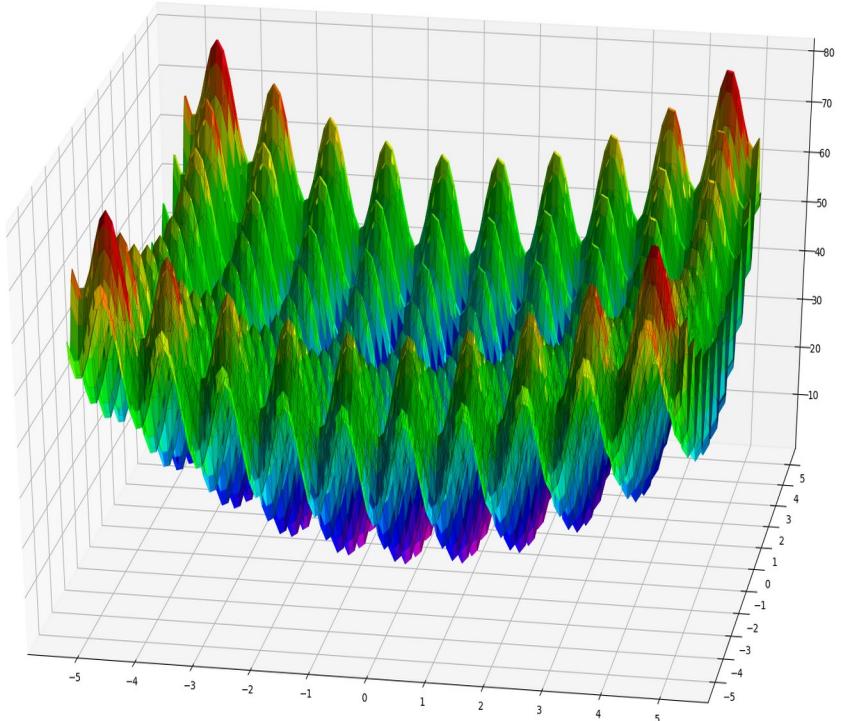
Глобальный минимум:  $f(\mathbf{x}) = 0$  при  
 $x_i = 420.9687, i = 1, \dots, n; -500 \leq x_i \leq 500$



# Тестовые функции. Функция Растигина

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

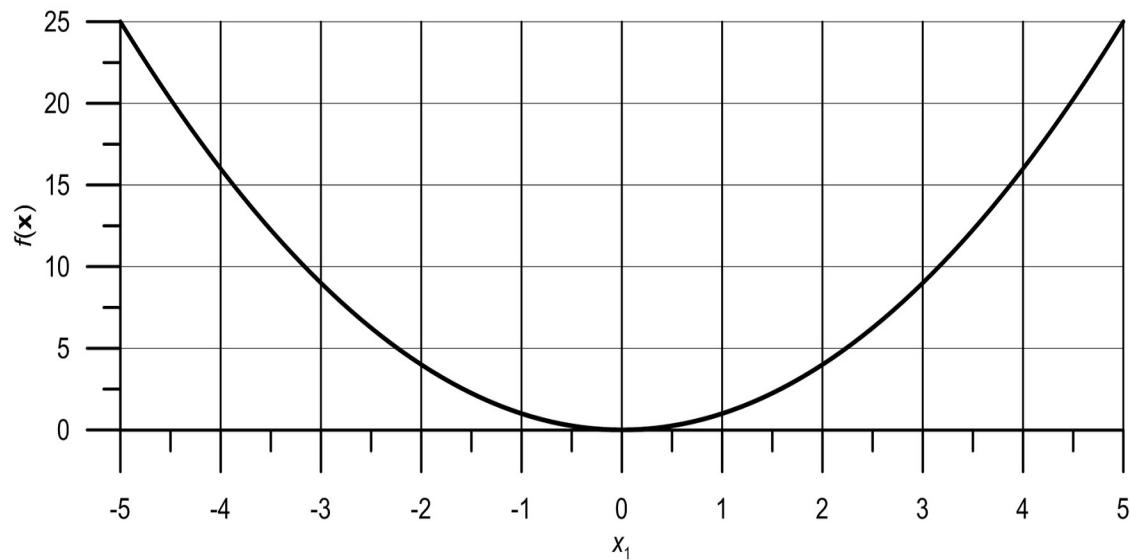
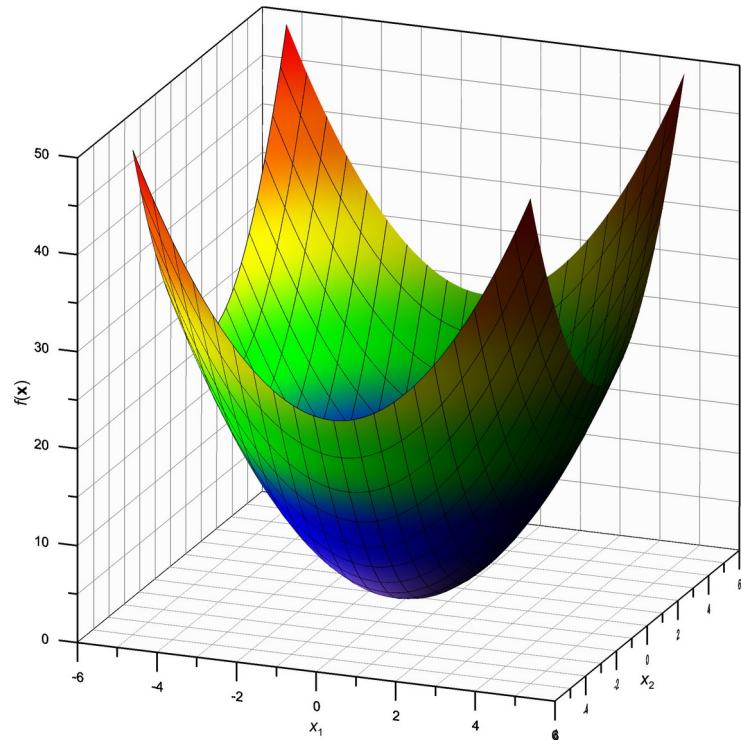
Глобальный минимум:  $f(\mathbf{x}) = 0$  при  
 $x_i = 0, i = 1, \dots, n,$   
 $-5.12 \leq x_i \leq 5.12$



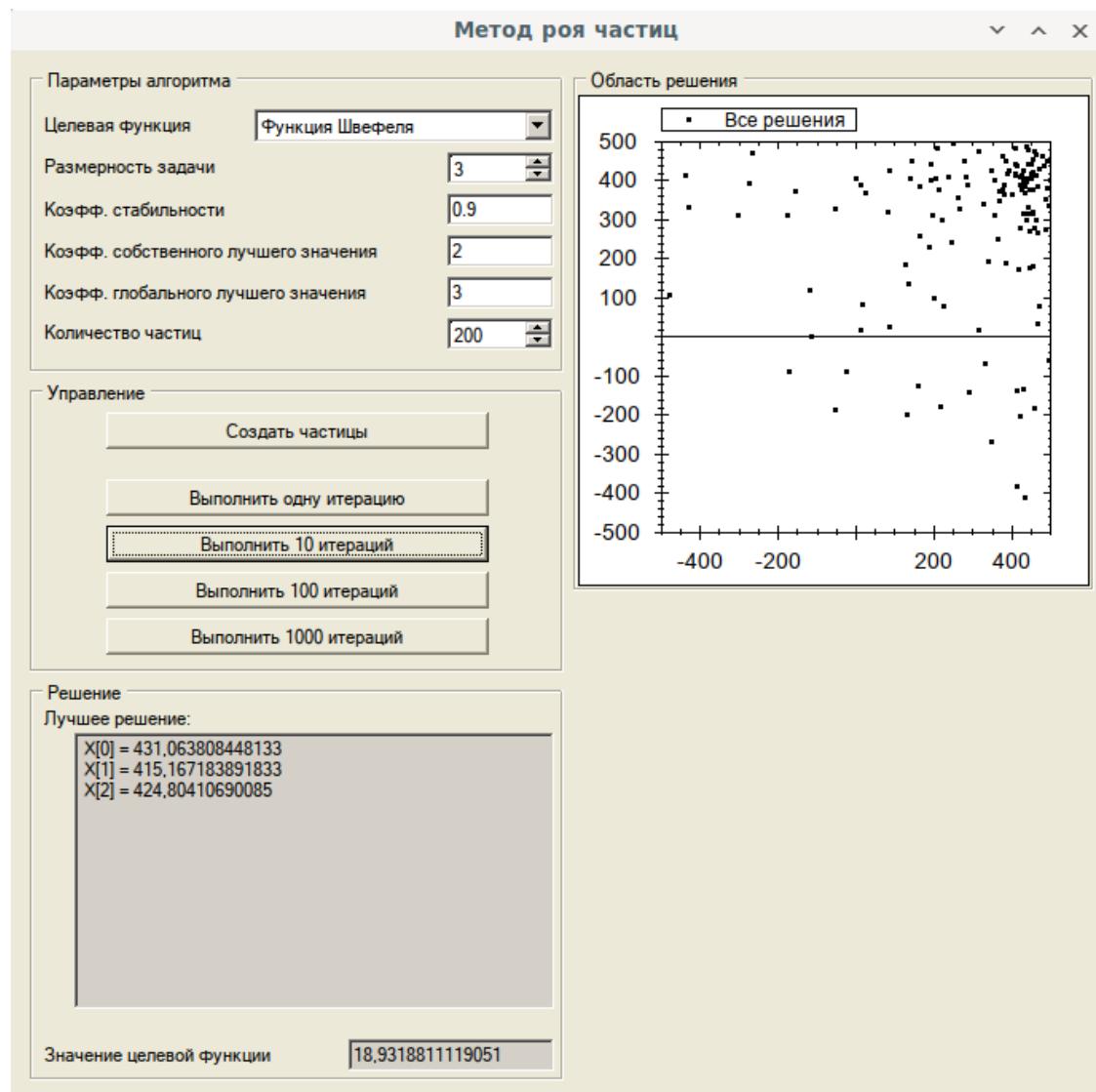
# Тестовые функции

$$f(x) = \sum_{i=1}^n x_i^2$$

Глобальный минимум:  $f(x) = 0$  при  
 $x_i = 0, i = 1, \dots, n$

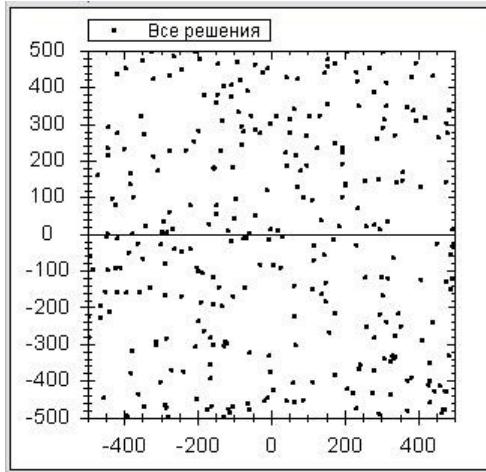


# Демонстрация работы алгоритма роя частиц

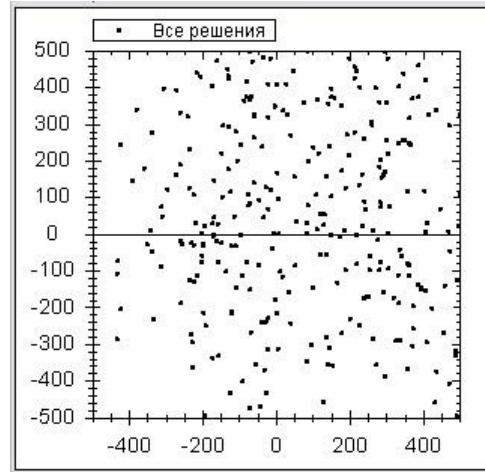


# Сходимость алгоритма роя частиц

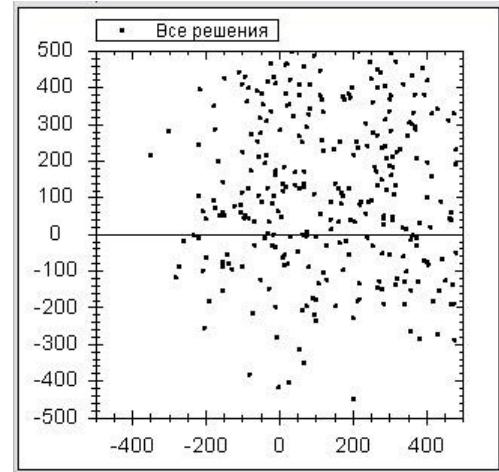
Итерация №0



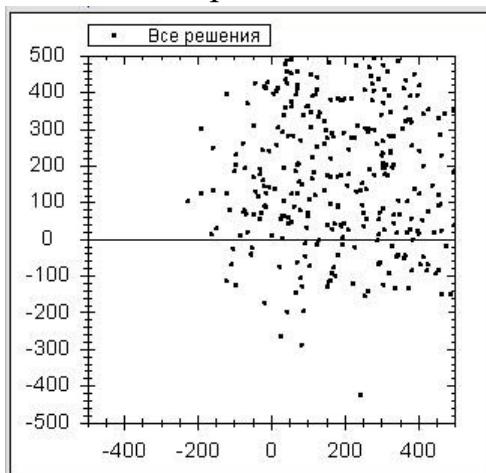
Итерация №1



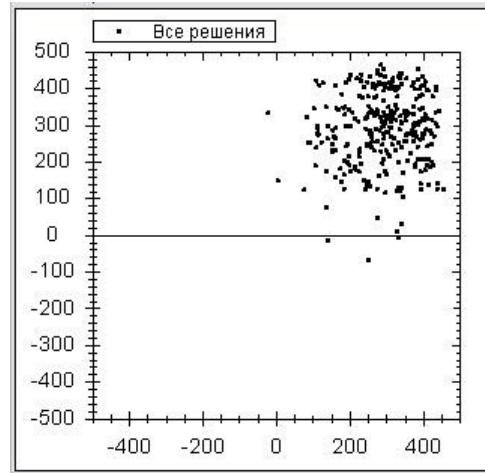
Итерация №2



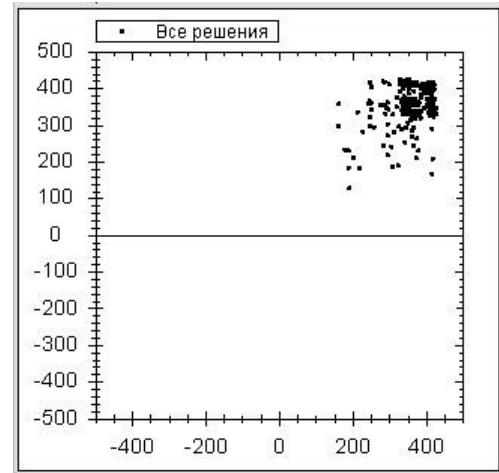
Итерация №3



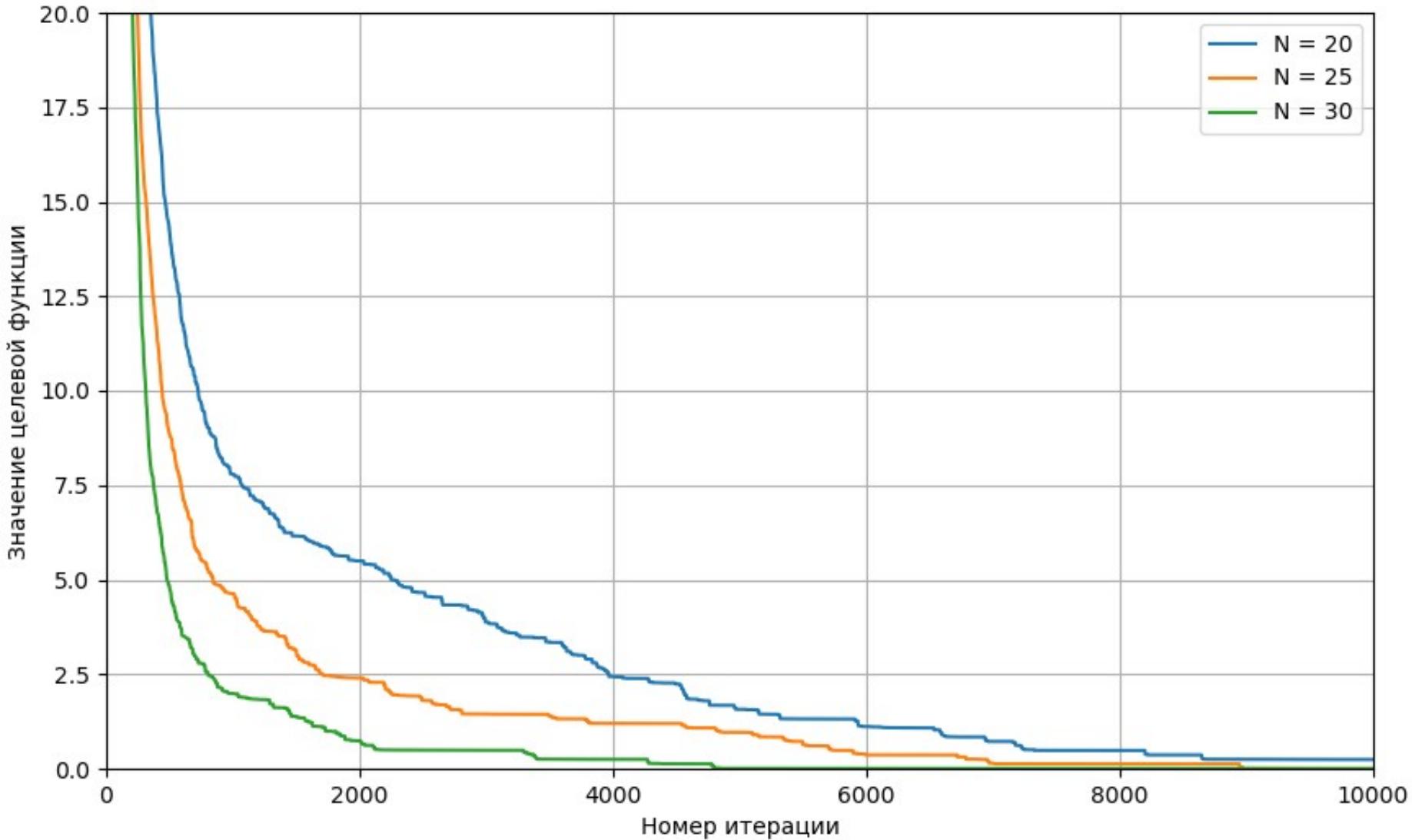
Итерация №10



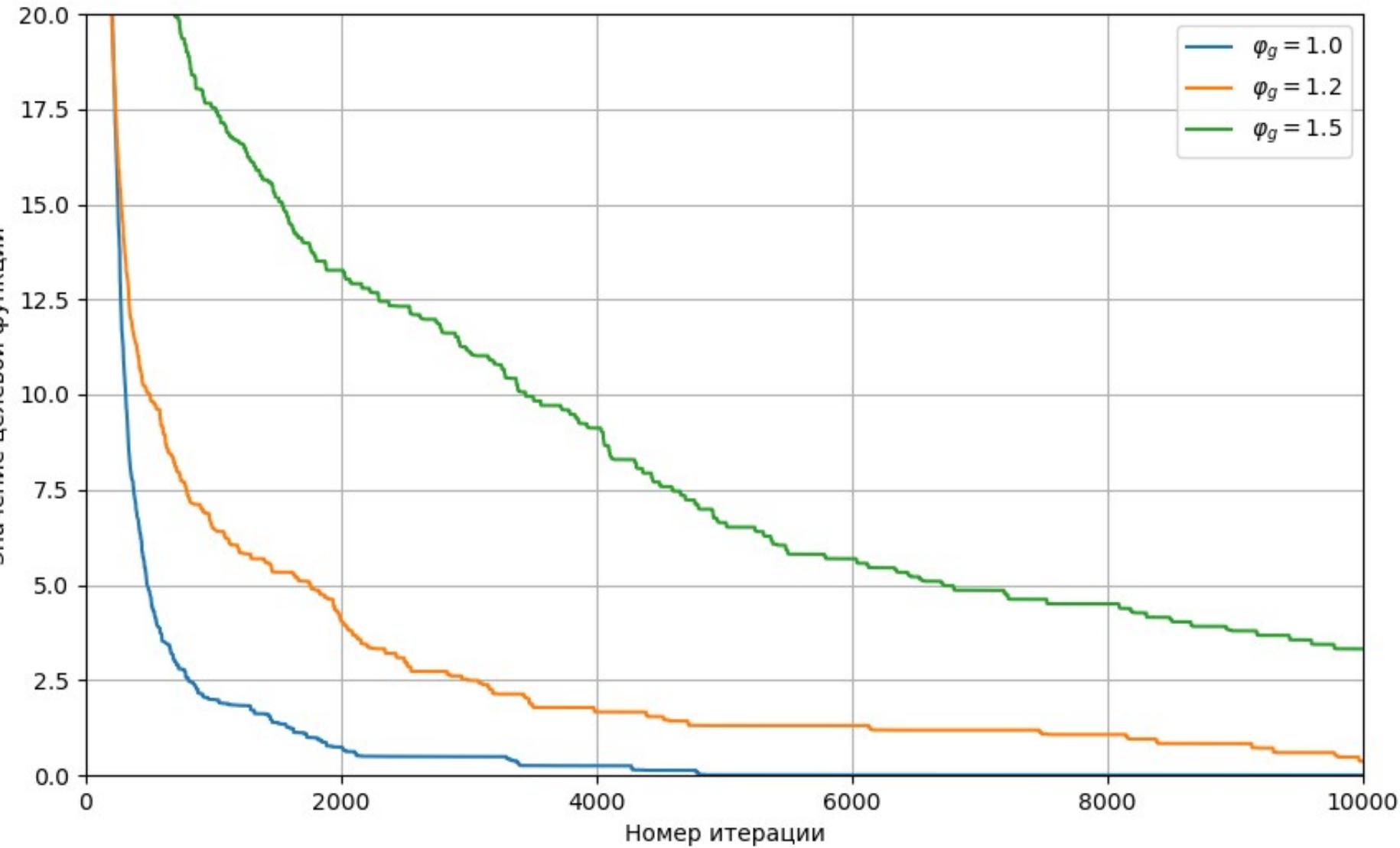
Итерация №30



# Влияние количества частиц на сходимость



# Влияние параметра $\varphi_g$ на сходимость



# Модификации метода роя частиц

1. Добавление случайной телепортации (мутации).
  2. Алгоритм с отрицательным подкреплением.
  3. Совместное использование с другими алгоритмами оптимизации [1].
  4. Разделение роя частиц на несколько групп.
- 
1. Saptarshi Sengupta, Sanchita Basak, Richard II. (2018). **Particle Swarm Optimization: A survey of historical and recent developments with hybridization perspectives.** 10.3390/make1010010.

# Преимущества и недостатки метода роя частиц



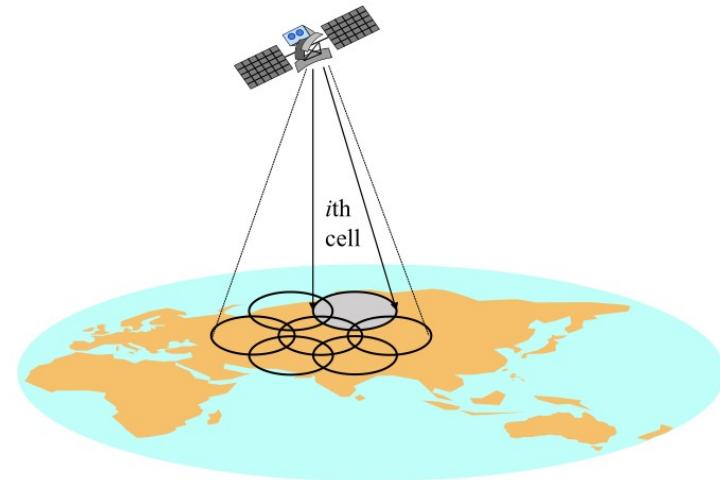
Простота реализации



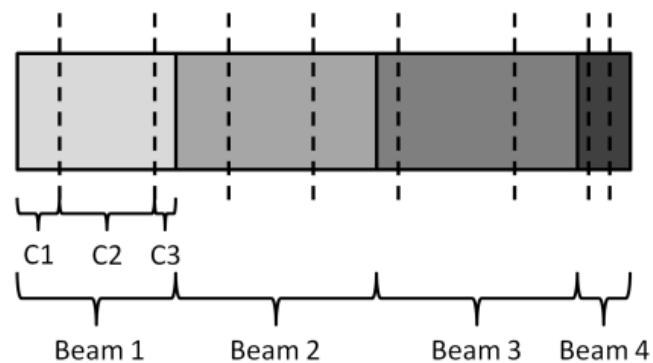
Сильная зависимость качества  
сходимости от выбранных коэффициентов

# Примеры применения в радиотехнике

1. Fabio Renan Durand, Taufik Abrão.  
**Power allocation in multibeam satellites based on particle swarm optimization.**  
AEU - International Journal of Electronics and Communications. August 2017



2. Pachler, Nils & Garau Luis, Juan Jose & Guerster, Markus & Crawley, Edward & Cameron, Bruce. **Allocating Power and Bandwidth in Multibeam Satellite Systems using Particle Swarm Optimization.** In 2020 IEEE Aerospace Conference, 2020.



## Ссылки

**Алгоритм роя частиц.**

**Описание и реализации на языках Python и C#:**

<https://jenyay.net/Programming/ParticleSwarm>

**Реализация алгоритмов оптимизации на языке Rust:**

<https://github.com/Jenyay/rust-optimization>

**Выступление в Московском клубе программистов с  
докладом про алгоритм роя частиц:**

<https://www.youtube.com/watch?v=57YBBIwnkQU>