## **Что делает программа?**

### **1. Запускает оптимизацию симуляцией отжига:**

initial\_state = np.random.randint(0, 1000, size=(bakery, N, M))

best\_state, best\_value, values = simulated\_annealing(F, initial\_state, temperature)

* Случайным образом генерируется начальное решение initial\_state.
* Затем методом отжига находится приближённый **оптимум** функции F(X), которая моделирует прибыль от поставок пирожных.
* best\_value — минимальное (т.е. **максимальное по исходной задаче**) значение -F(X) — то есть максимальный доход.
* best\_state — соответствующий вектор поставок.

### **2. Сохраняет историю оптимизации**

values # список значений -F(X) на каждой итерации

* Это нужно, чтобы **отследить динамику**: насколько быстро (и стабильно) функция сходится к хорошему значению.
* Используется для визуализации.

### **3. Строит график сходимости**

plt.plot(x, y)

* Каждый график показывает, **как метод отжига приближается к оптимальному решению** во времени.
* Это помогает понять, есть ли прогресс, где происходит "застревание", насколько "разнообразны" траектории в разных запусках.

**Что можно узнать из этой информации?**

### **1. Наблюдать сходимость**

* График покажет, как быстро (или медленно) функция F(X) становится лучше.
* Если на графике видно, что значение почти сразу стабилизировалось — возможно, **поиск застрял в локальном минимуме**.

### **2. Можно увидеть влияние начальной точки**

* При нескольких запусках (с разными initial\_state) можно сравнивать:  
  + Как отличается результат?
  + Есть ли **зависимость от начальной инициализации**?
  + Например, в одном запуске ты получил доход 12000, в другом — 14000 → метод неустойчив, стоит подумать об улучшении new\_x или стратегии.

### **3. Отладка модели**

* Если график скачет дико или идёт вверх (что нелогично) — значит, F(X) или new\_x() работает не так, как ожидалось.
* Это может указать на:  
  + ошибки в расчёте
  + переполнение
  + отсутствие ограничений и т.п.

## **Как интерпретировать график**

* **Плавно убывающая кривая** → хорошо: метод стабильно сходится.
* **Резкие скачки** → случайные отклонения, могут быть полезны (выход из локального минимума).
* **Горизонтальная линия** → метод ничего не улучшает → возможно, плохой new\_x() или слишком низкая temperature.
* **Несколько траекторий сильно отличаются** → стоит запускать метод многократно и выбирать лучший результат.

## **Как это поможет**

| **Цель** | **Как график помогает** |
| --- | --- |
| Найти лучший план поставок | Показывает, какой результат получен |
| Понять, насколько надёжна оптимизация | Сравнивает траектории при разных запусках |
| Подобрать параметры отжига | Видна скорость и качество сходимости |
| Улучшить модель и генератор решений | Видно, где оптимизация "ломается" |
| Убедиться, что модель работает | Видно, что оптимизация делает прогресс |

