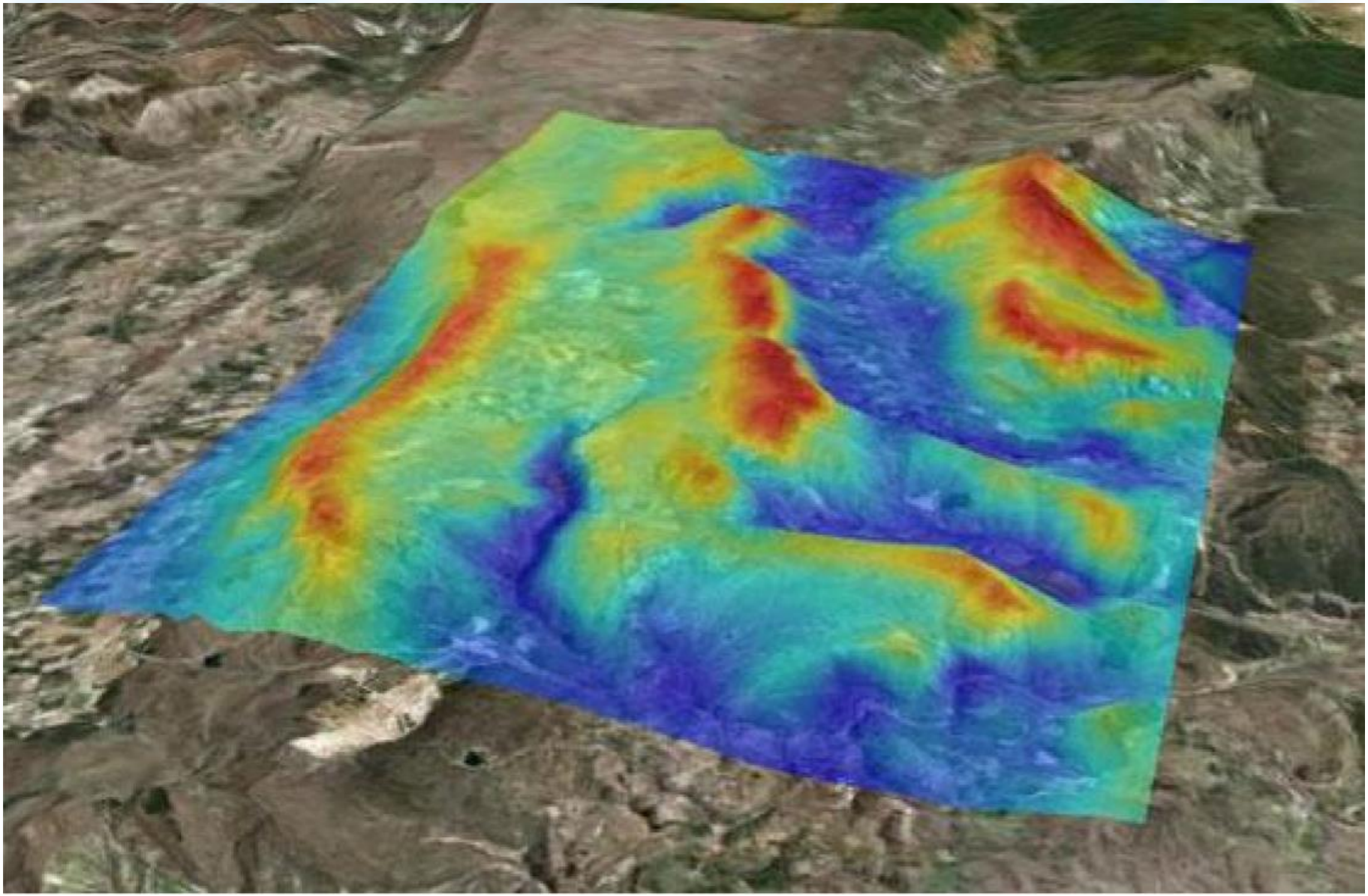
Практическая оптимизационная задача не из области ИТ

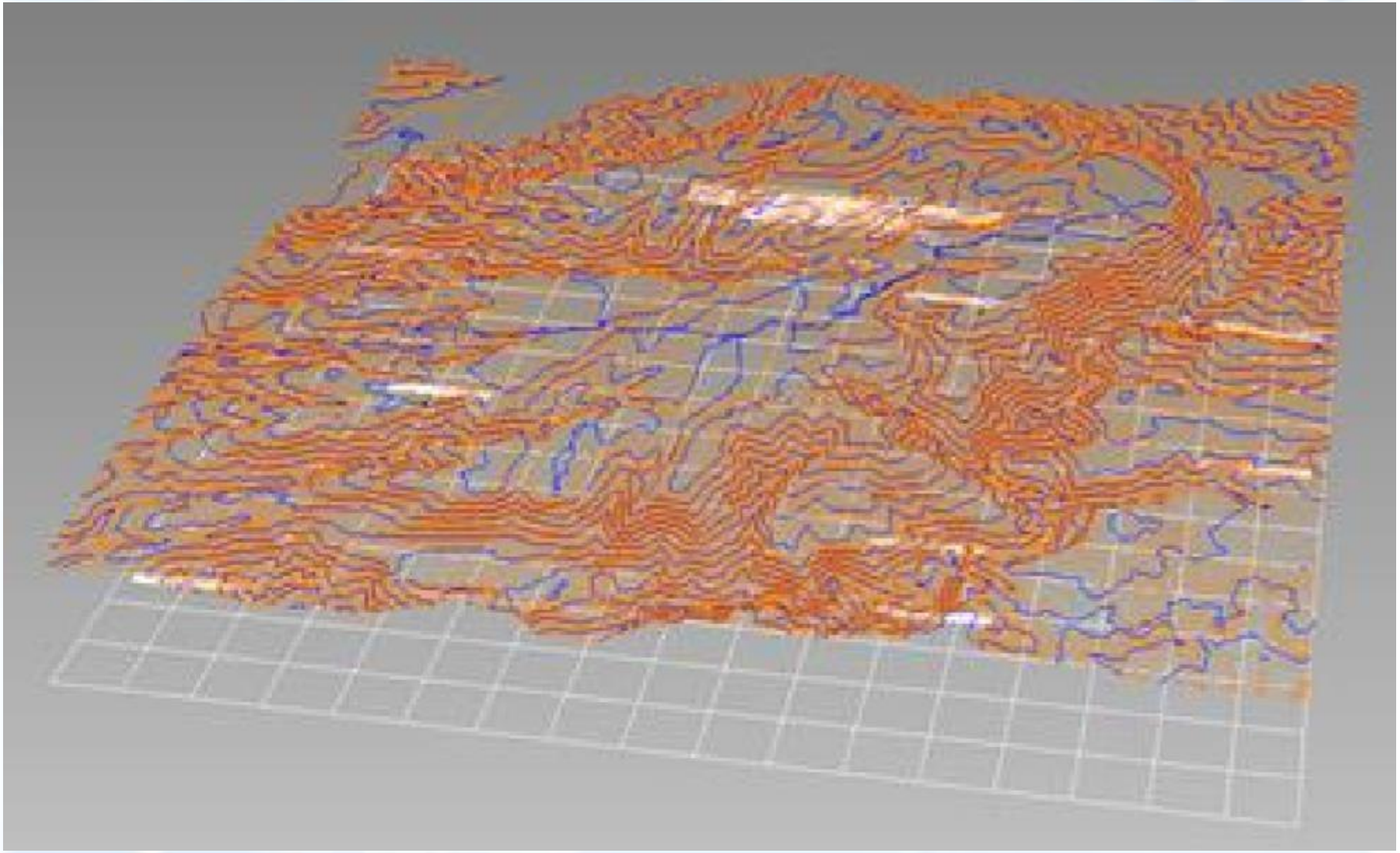
В.В. Балашов, ВМК МГУ

Объект: ветряная электростанция

Ландшафт: здесь ставим башни



Сетка: не более одной башни на квадрат



Формулируем задачу: какая расстановка нужна?

1. Расставить N башен так, чтобы приход энергии за год был максимальным
   - критерий оценки: $\sum_{i=1}^{N} E_i \to max$
   - свободные параметры: номера квадратов, куда ставим башни
   - ограничение: номера квадратов не повторяются
2. Найти минимальную по количеству башен расстановку, обеспечивающую заданный приход энергии за год
   - критерий оценки: $N \to min$
   - свободные параметры: количество башен; номера квадратов, куда ставим башни
   - ограничения: $\sum_{i=1}^{N} E_i \geq E_0$; номера квадратов не повторяются

Нужно найти наилучшую или «почти» наилучшую расстановку (оптимизационная задача)

Рассчитываем приход энергии

- Кинетическая энергия воздуха:

$$E_{air} = \frac{1}{2} m \cdot u^2 = \frac{1}{2} (A \cdot u \cdot t \cdot \rho) u^2 = \frac{1}{2} A \cdot t \cdot \rho \cdot u^3$$

  - A: площадь круга лопастей ветряка
  - u: скорость ветра
  - t: длительность ветра с этой скоростью
  - ρ: плотность воздуха
- Приход энергии от ветряка:

$$E_{out} = C_p \cdot E_{air}$$

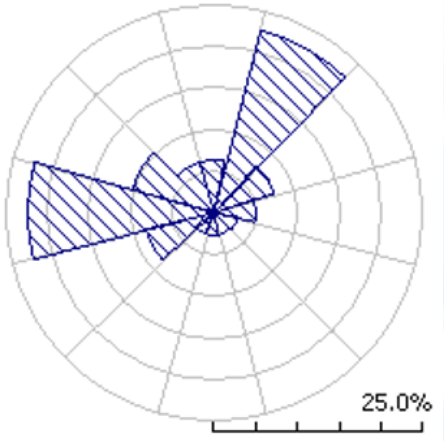
  - C_p: КПД ветряка

Учитываем, что ветер дует по-разному

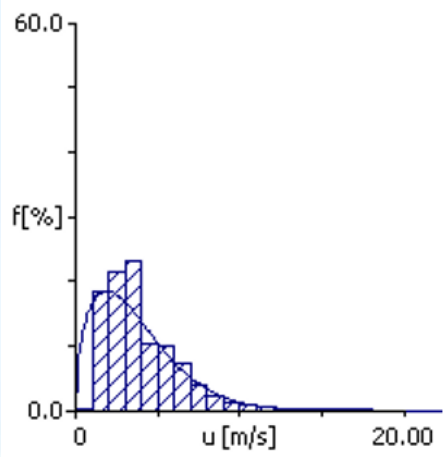
$$E_{out} = C_p \cdot \sum_{u=1}^{u_{max}} H(u) \cdot \frac{1}{2} A \cdot \rho \cdot u^3 =$$

$$E_{out} = \frac{1}{2} C_p \cdot A \cdot \rho \cdot \sum_{u=1}^{u_{max}} H(u) \cdot u^3$$

- $H(u)$: время (за год), которое ветер дует со скоростью u

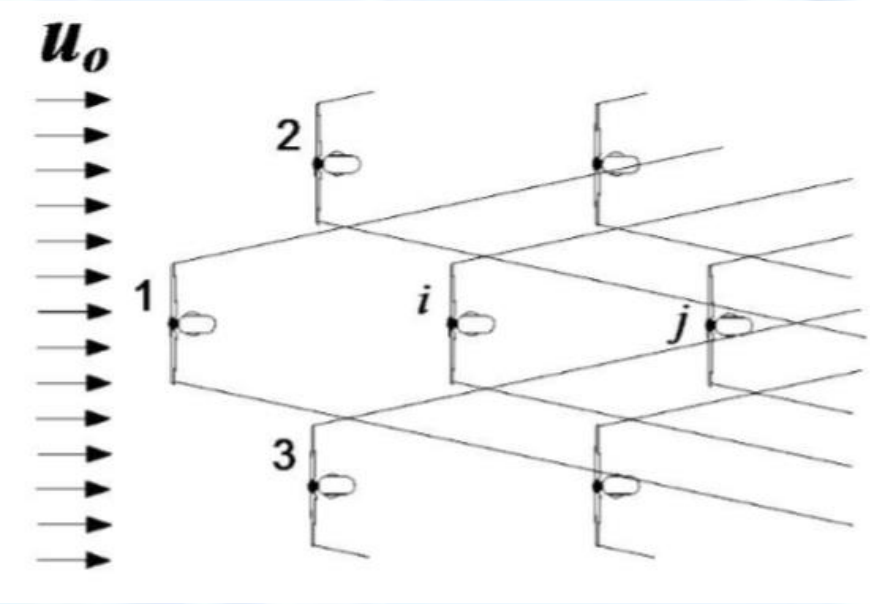


- Нужна модель ветра «за год»:
  - направление, скорость
  - пересчет направления и скорости для каждого квадрата с учетом ландшафта



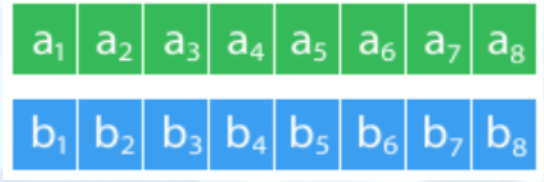
Как расставлять?

- Занимаем все лучшие квадраты подряд?
  - нет: ветряки заслоняют друг друга
- Полный перебор?
  - 289 квадратов, 40 башен: $C_{289}^{40} = \frac{289!}{249! \cdot 40!} \approx 1{,}97 \cdot 10^{49}$ вариантов
- Случайный поиск?
  - много попыток
  - запоминаем лучшую расстановку
  - **а вдруг не повезёт**
- Жадный подход:
  - занять лучший квадрат
  - исключить заслоняющие его квадраты
  - занять лучший из оставшихся
  - исключить заслоняющие
  - …
  - **может упустить лучшее решение:** один «хороший» квадрат может давать меньше энергии, чем два «похуже» рядом с ним

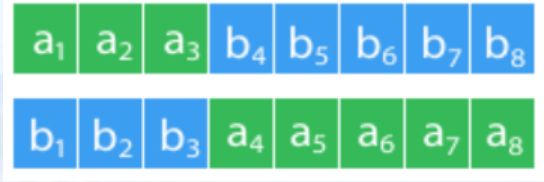


Вспомним про эволюцию

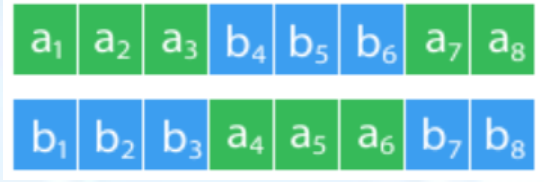
- решение (массив номеров квадратов) = генетический код
- поколение = набор из *K* решений
- отбор решений для скрещивания (селекция)
- скрещивание двух решений => порождение двух новых решений
  - обмен генетической информацией
- мутация нового решения
  - случайное перемещение одной или нескольких башен
- формирование нового поколения



1-точечный обмен

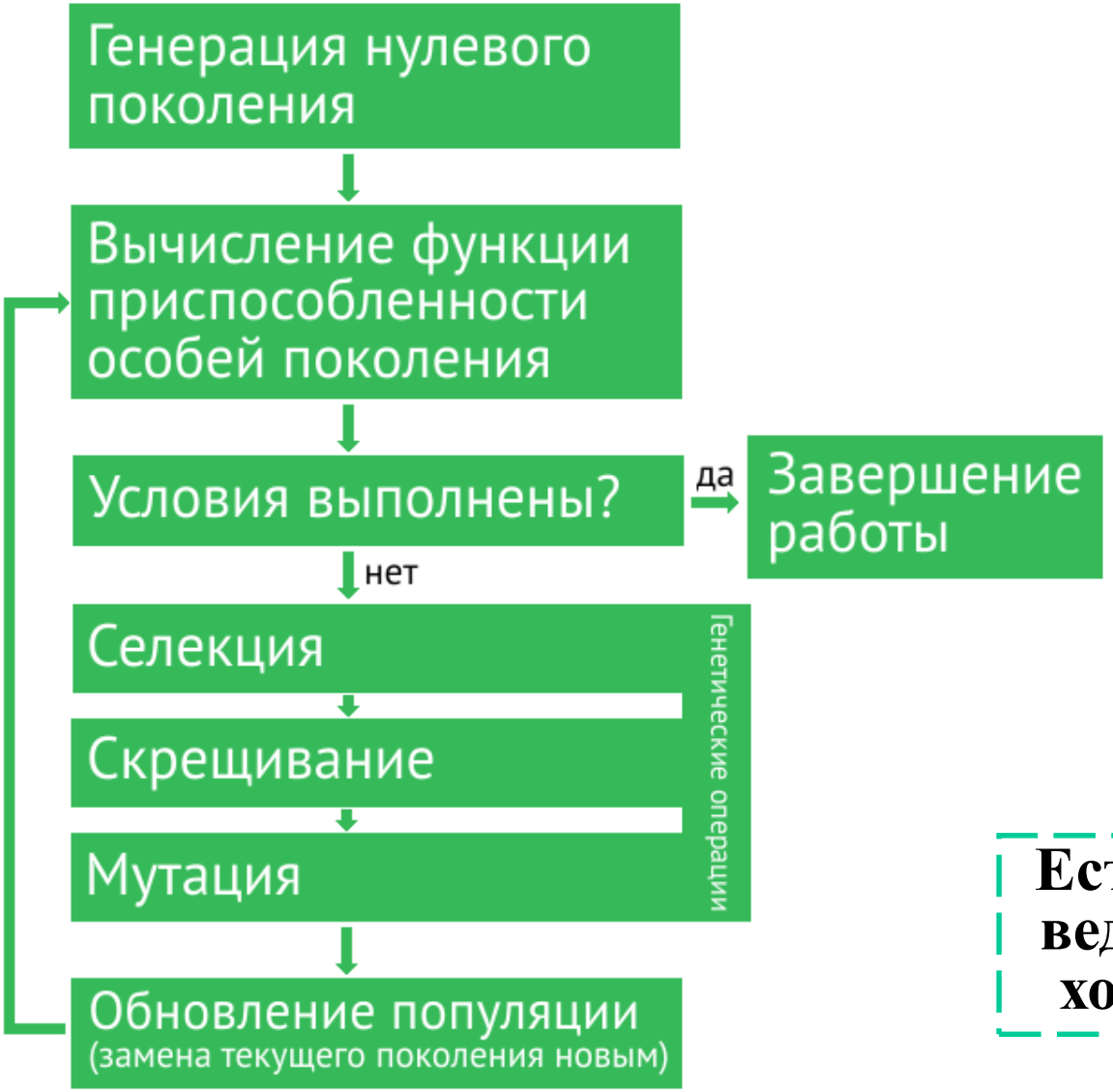


2-точечный обмен



Проверка выполнения ограничений после скрещивания и мутации

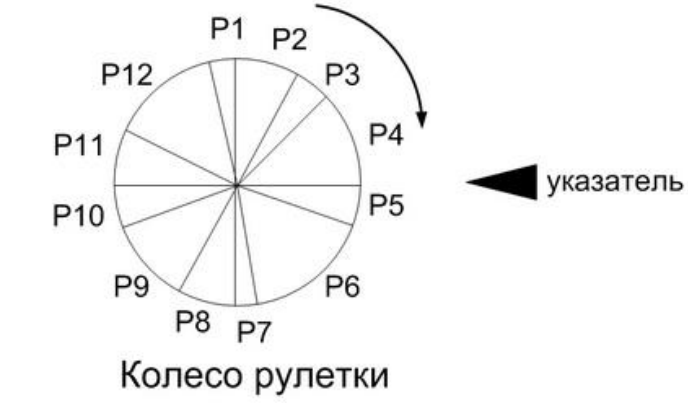
Эволюционный алгоритм



Естественный отбор ведет к нахождению хорошего решения

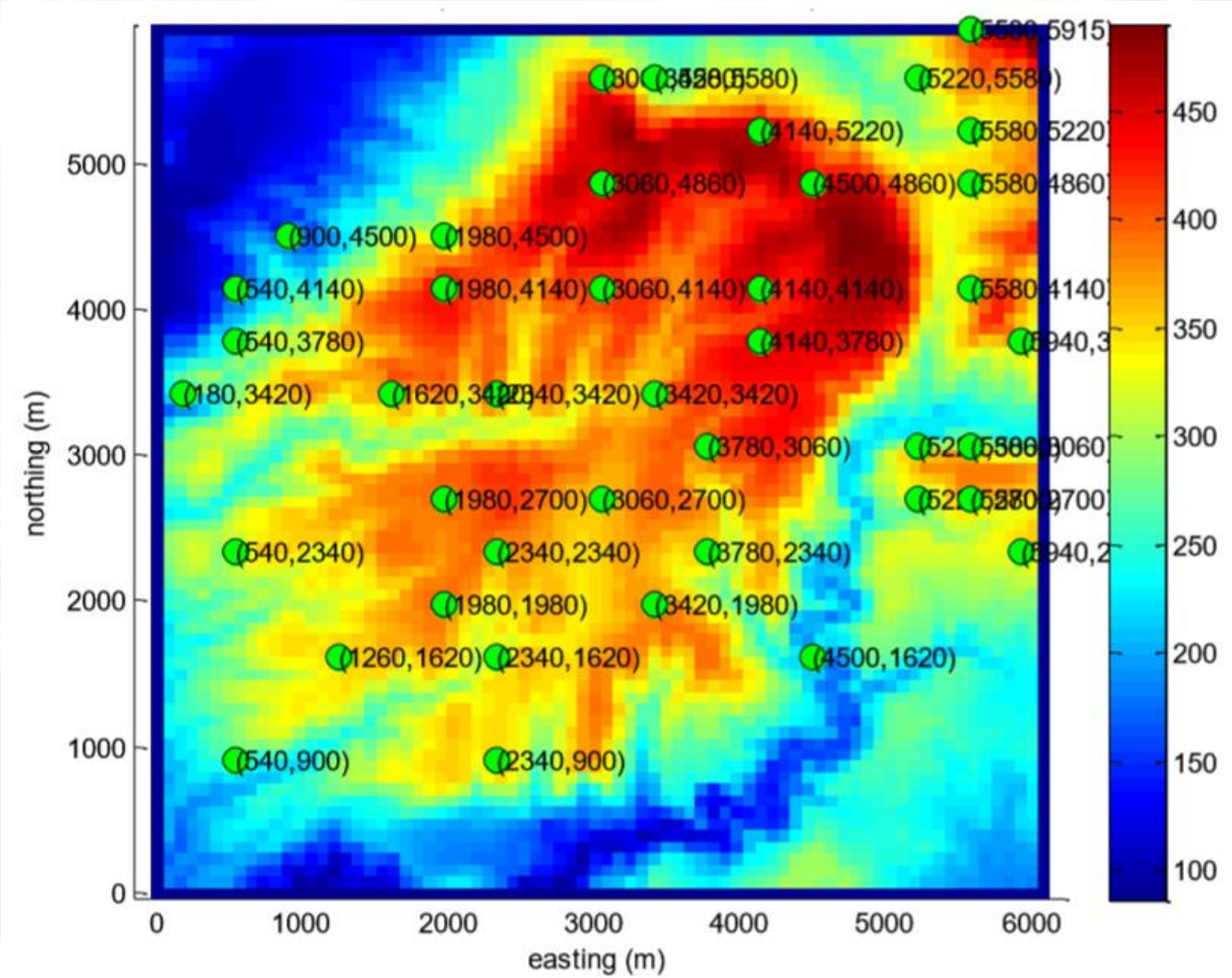
Варианты селекции

- Выбор лучших решений из популяции
  - плохо: можно скатиться к «локально наилучшему» решению
- Рулетка
  - чем лучше решение (по критерию), тем выше *вероятность* его выбора для скрещивания
- Турнирная селекция
  - случайный выбор пар для состязаний
  - победитель выбирается для скрещивания
  - один «сильный» может выбить другого

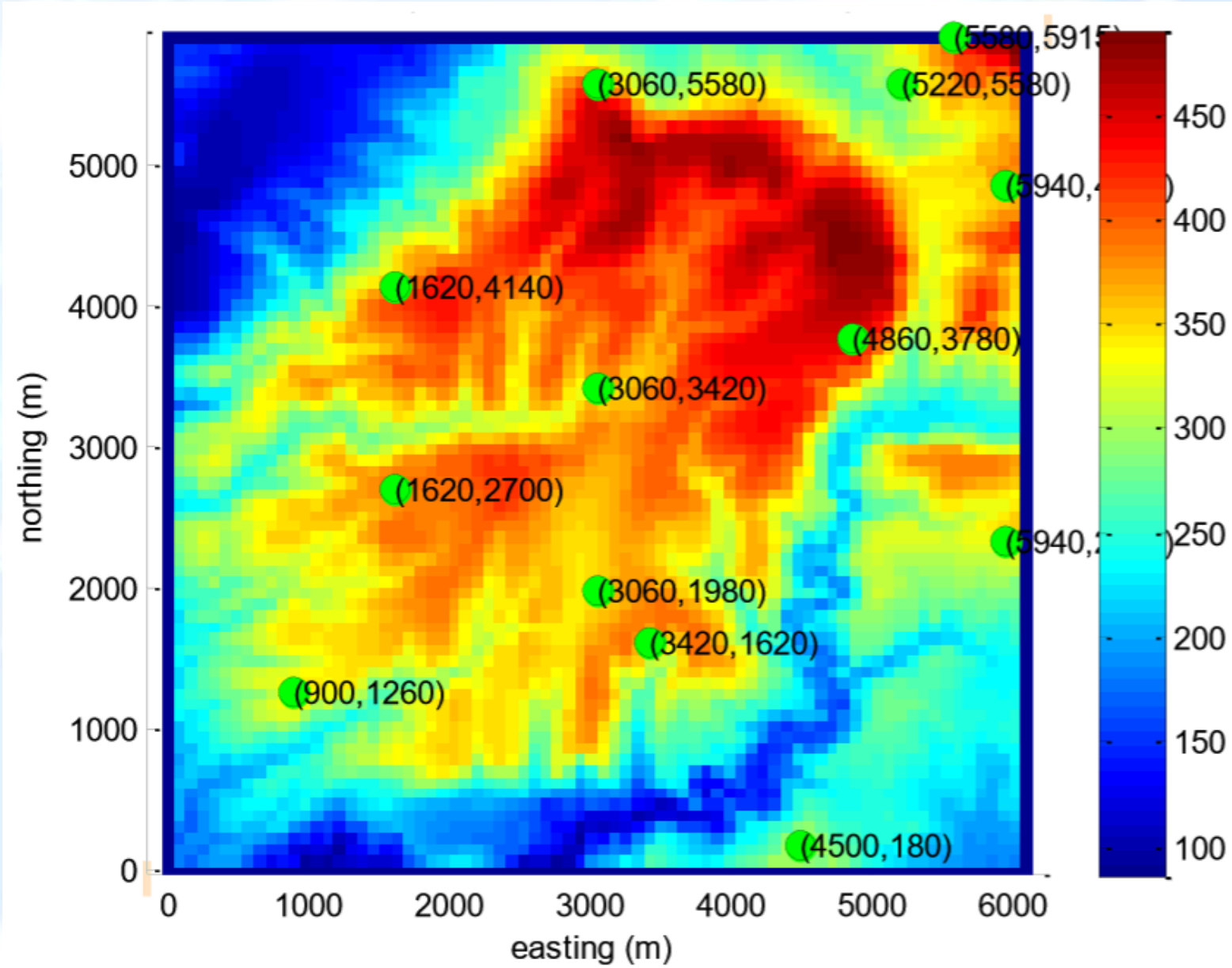


Колесо рулетки

Решение задачи 1



Решение задачи 2



Шаги работы с оптимизационной задачей

- Практическая задача от заказчика
  - четко сформулировать в понятных заказчику терминах
- Математическая формулировка задачи
- Обзор известных методов решения похожих задач
- Построение алгоритма решения задачи
- Исследование алгоритма
- Реализация программной системы для заказчика

Роли участников

1. Аналитик
   - узнаёт у специалистов-«предметников», что им нужно на самом деле
     => содержательная постановка задачи
   - формулирует задачу на языке математики
     => математическая формулировка задачи (представление решения, функция-критерий, ограничения)
2. Специалист по моделированию
   - разрабатывает математические модели, нужные для расчета функции-критерия
   - «красивые» аналитические функции в жизни встречаются редко!
3. Алгоритмист
   - разрабатывает алгоритм решения математической задачи
   - выполняет исследование алгоритма (качество решения, скорость работы)
4. Программист
   - разрабатывает инструментальную систему

1 – 3: важна математика!

Другие задачи оптимизации

- Построение расписаний
  - взлет и посадка самолетов в аэропорту
  - движение электропоездов
  - перевозка контейнеров, включая перегрузку (корабль-поезд-грузовик)
- Построение маршрутов движения
  - курьеры при развозке покупок (ограничения!)
  - грузоперевозки
- Размещение датчиков для мониторинга состояния зданий
- Размещение гасителей колебаний в конструкциях мостов
- Выбор формы деталей для 3D-печати
  - моделирование свойств детали!
  - очень большая гибкость выбора формы
- Выбор режима работы водохранилища
  - цели: подача воды для орошения; компенсация наводнений (резерв емкости); генерация электроэнергии
- Выбор режима работы для системы электростанций
  - наличие запаса мощности
  - отсутствие перерасхода топлива
- Распределение вычислительной нагрузки в ЦОД

В чем выгода?

- Оптимальное решение *правильно поставленной* задачи экономит заказчику массу времени и денег
- Заказчики из областей:
  - энергетика
  - транспорт
  - обработка больших данных (ЦОД)
  - строительство
  - …

Спасибо

Балашов Василий Викторович
ВМК МГУ
hbd@cs.msu.su