МЕТАЭВРИСТИКИ

Ю.А.Скобцов

Санкт-Петербургский Государственный Университет Аэрокосмического Приборостроения

В настоящее время бурно развивается новое направление в теории и практике искусственного интеллекта, которое принято называть «метаэвристики»

Эвристика является алгоритмом, который находит «достаточно хорошие» решения сложной проблемы за приемлемое время, теоретически не обосновывая их правильность или оптимальность, т.е. эмпирическим путем.

Метаэвристика расширяет возможности эвристик, комбинируя эвристические методы (процедуры) на основе некоторой высокоуровневой стратегии (приставка «мета»).

Термин «метаэвристика» введен Гловером в 1986 г. и до сих пор является дискуссионным.

Свойства метаэвристик

1. Метаэвристика является стратегией, которая управляет процессом поиска.
2. Целью метаэвристики является эффективное исследование пространства поиска для нахождения (суб)оптимального решения.
3. Методы, используемые метаэвристическим алгоритмом, лежат в диапазоне от простого локального поиска до сложного процесса обучения.
4. Метаэвристический алгоритм является приближенным и обычно не детерминированным.
5. Метаэвристика может использовать механизм, который предотвращает попадание в ловушку в ограниченной области пространства поиска.

Свойства метаэвристик

6. Основные положения метаэвристики допускают абстрактное описание.
7. Метаэвристика не является проблемно-ориентированной.
8. Метаэвристика может использовать проблемно-ориентированное знание в форме эвристик, управляемых высокоуровневой стратегией.
9. Передовые метаэвристики используют опыт, накопленный в процессе поиска и представленный в виде памяти для управления поиском.

Многие метаэвристики в основе указанной высокоуровневой стратегии используют модели некоторых естественных или искусственных систем – биологических, физических, социальных и т.п.

Например, эволюционные метаэвристики (генетические алгоритмы и т.п.) используют модель искусственной эволюции.

Первоначально, как правило, эти модели создаются для формального описания соответствующей системы, а далее они «отрываются» от исходной системы и «живут своей жизнью» - в основном, применяются для решения задач оптимизации.

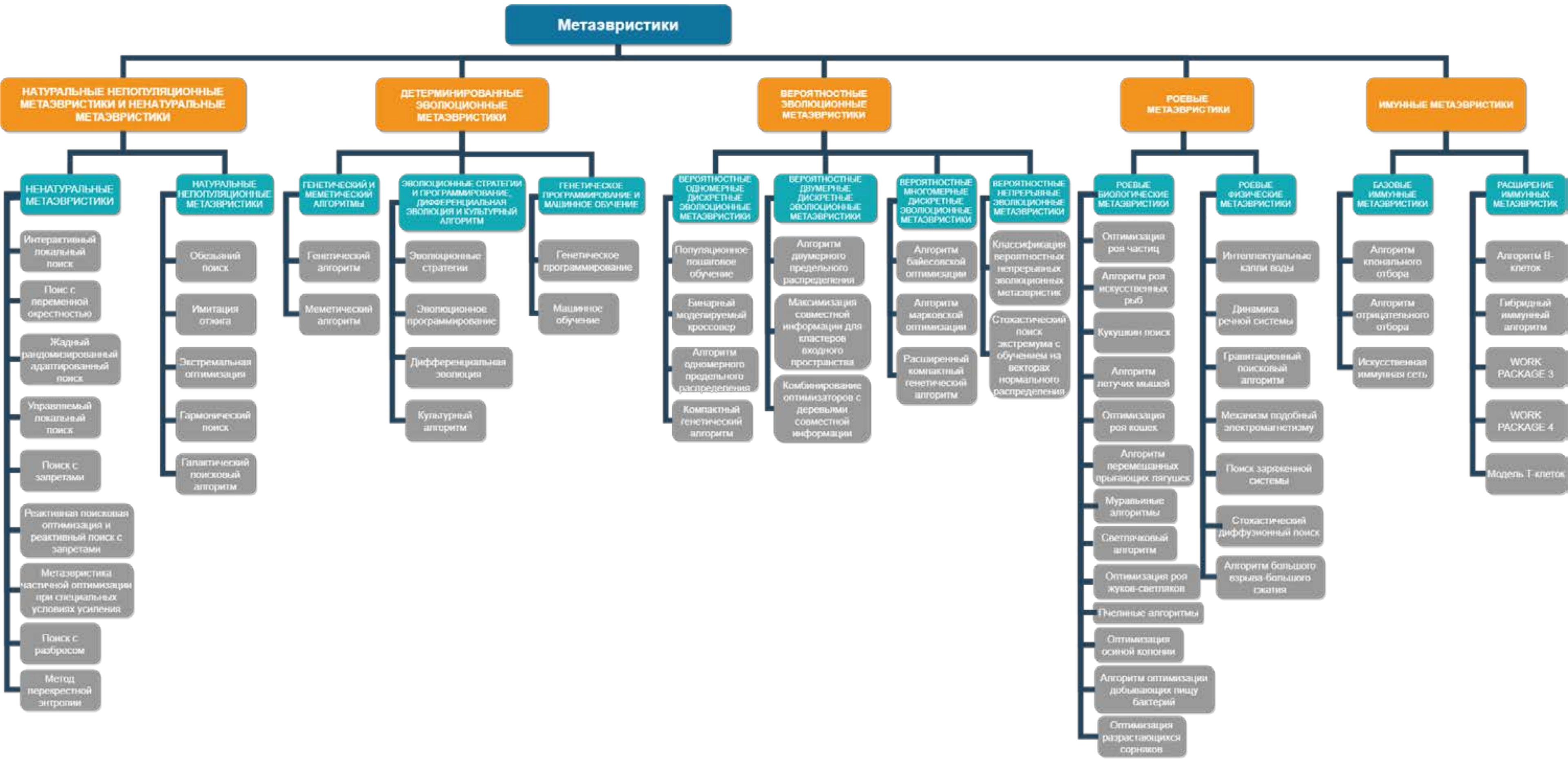
Отметим, что в качестве базовых систем (явлений) часто выступают биологические и социальные системы.

Поэтому часть метаэвристик относят к био-инспирированным алгоритмам (навеянными природой).

В этом случае часто применяются модели биологических или социальных систем.

Классификация метаэвристик

Мэтавристики по количеству используемых решений подразделяются на:
– непопуляционные (используют одно потенциальное решение);
– популяционные (используют множество потенциальных решений).
Непопуляционные метаэвристики подразделяются на:
– натуральные (биологические, социальные и физические);
–ненатуральные метаэвристики, которые не основаны на моделировании процессов и механизмов биологических или физических систем.
Напротив натуральные эвристики построены на базе моделей естественных систем.
Популяционные метаэвристики подразделяются на:
– эволюционные (детерминированные и вероятностные);
– роевые (биологические и физические);
– иммунные;
– ненатуральные.


Метаэвристики
НАТУРАЛЬНЫЕ НЕПОПУЛЯЦИОННЫЕ МЕТАЭВРИСТИКИ И НЕНАТУРАЛЬНЫЕ МЕТАЭВРИСТИКИ
НЕНАТУРАЛЬНЫЕ МЕТАЭВРИСТИКИ
Интерактивный локальный поиск
Поис с переменной окрестностью
Жадный рандомизированный адаптированный поиск
Управляемый локальный поиск
Поиск с запретами
Реактивная поисковая оптимизация и реактивный поиск с запретами
Метаэвристика частичной оптимизации при специальных условиях усиления
Поиск с разбросом
Метод перекрестной энтропии
НАТУРАЛЬНЫЕ НЕПОПУЛЯЦИОННЫЕ МЕТАЭВРИСТИКИ
Обезьяний поиск
Имитация отжига
Экстремальная оптимизация
Гармонический поиск
Галактический поисковый алгоритм
ДЕТЕРМИНИРОВАННЫЕ ЭВОЛЮЦИОННЫЕ МЕТАЭВРИСТИКИ
ГЕНЕТИЧЕСКИЙ И МЕМЕТИЧЕСКИЙ АЛГОРИТМЫ
Генетический алгоритм
Меметический алгоритм
ЭВОЛЮЦИОННЫЕ СТРАТЕГИИ И ПРОГРАММИРОВАНИЕ, ДИФФЕРЕНЦИАЛЬНАЯ ЭВОЛЮЦИЯ И КУЛЬТУРНЫЙ АЛГОРИТМ
Эволюционные стратегии
Эволюционное программирование
Дифференциальная эволюция
Культурный алгоритм
ГЕНЕТИЧЕСКОЕ ПРОГРАММИРОВАНИЕ И МАШИННОЕ ОБУЧЕНИЕ
Генетическое программирование
Машинное обучение
ВЕРОЯТНОСТНЫЕ ЭВОЛЮЦИОННЫЕ МЕТАЭВРИСТИКИ
ВЕРОЯТНОСТНЫЕ ОДНОМЕРНЫЕ ДИСКРЕТНЫЕ ЭВОЛЮЦИОННЫЕ МЕТАЭВРИСТИКИ
Популяционное пошаговое обучение
Бинарный моделируемый кроссовер
Алгоритм одномерного предельного распределения
Компактный генетический алгоритм
ВЕРОЯТНОСТНЫЕ ДВУМЕРНЫЕ ДИСКРЕТНЫЕ ЭВОЛЮЦИОННЫЕ МЕТАЭВРИСТИКИ
Алгоритм двумерного предельного распределения
Максимизация совместной информации для кластеров входного пространства
Комбинирование оптимизаторов с деревьями совместной информации
ВЕРОЯТНОСТНЫЕ МНОГОМЕРНЫЕ ДИСКРЕТНЫЕ ЭВОЛЮЦИОННЫЕ МЕТАЭВРИСТИКИ
Алгоритм байесовской оптимизации
Алгоритм марковской оптимизации
Расширенный компактный генетический алгоритм
ВЕРОЯТНОСТНЫЕ НЕПРЕРЫВНЫЕ ЭВОЛЮЦИОННЫЕ МЕТАЭВРИСТИКИ
Классификация вероятностных непрерывных эволюционных метаэвристик
Стохастический поиск экстремума с обучением на векторах нормального распределения
РОЕВЫЕ МЕТАЭВРИСТИКИ
РОЕВЫЕ БИОЛОГИЧЕСКИЕ МЕТАЭВРИСТИКИ
Оптимизация роя частиц
Алгоритм роя искусственных рыб
Кукушкин поиск
Алгоритм летучих мышей
Оптимизация роя кошек
Алгоритм перемешанных прыгающих лягушек
Муравьиные алгоритмы
Светлячковый алгоритм
Оптимизация роя жуков-светляков
Пчелиные алгоритмы
Оптимизация осиной колонии
Алгоритм оптимизации добывающих пищу бактерий
Оптимизация разрастающихся сорняков
РОЕВЫЕ ФИЗИЧЕСКИЕ МЕТАЭВРИСТИКИ
Интеллектуальные капли воды
Динамика речной системы
Гравитационный поисковый алгоритм
Механизм подобный электромагнетизму
Поиск заряженной системы
Стохастический диффузионный поиск
Алгоритм большого взрыва-большого сжатия
ИМУННЫЕ МЕТАЭВРИСТИКИ
БАЗОВЫЕ ИММУННЫЕ МЕТАЭВРИСТИКИ
Алгоритм клонального отбора
Алгоритм отрицательного отбора
Искусственная иммунная сеть
РАСШИРЕНИЕ ИММУННЫХ МЕТАЭВРИСТИК
Алгоритм B-клеток
Гибридный иммунный алгоритм
WORK PACKAGE 3
WORK PACKAGE 4
Модель T-клеток

- Метаэвристики находят широкое применение для решения различных оптимизационных проблем, машинного обучения, распознавания образов и др.
- Когда задача не может быть решена другими, более простыми методами, метаэвристики часто могут найти оптимальные или близкие к ним решения.
- При этом объем вычислений может оказаться большим, но скорость, с которой он растет при увеличении размерности задачи обычно меньше, чем у других известных методов.
- После того, как компьютерные системы стали достаточно быстродействующими и недорогими, метаэвристики превратились в один из основных инструментов поиска (суб)оптимальных решений задач, которые до этого считались неразрешимыми.

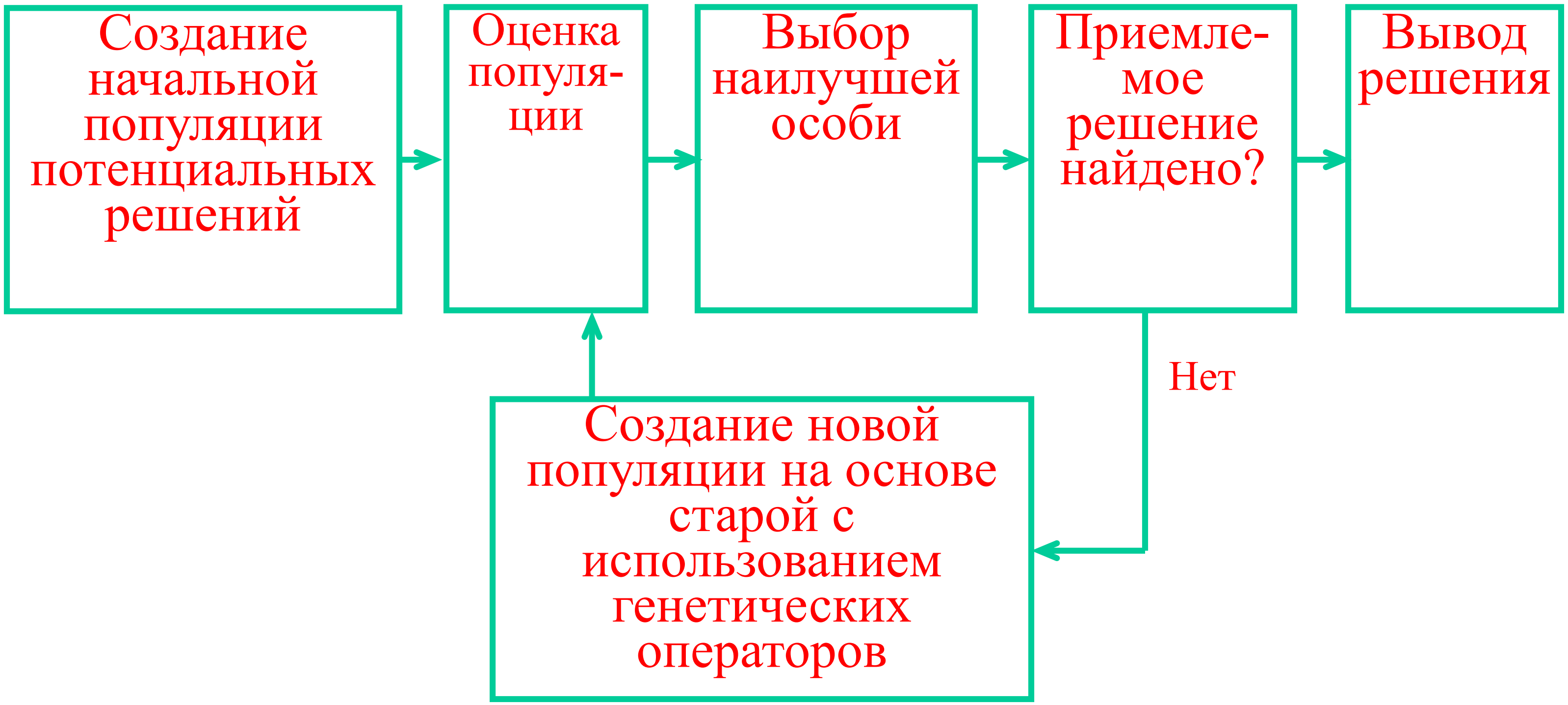
Детерминированные эволюционные метаэвристики

Этот термин обычно используется для общего описания алгоритмов поиска, оптимизации или обучения основанных на некоторых формализованных принципах естественного эволюционного отбора.

Особенности идей эволюции и самоорганизации заключаются в том, что они находят подтверждение не только для биологических систем, развивающихся много миллиардов лет.

Эти идеи в настоящее время с успехом используются при разработке многих технических и программных систем.

Обобщенная блок-схема ЭА



Три источника и три составные части эволюционного подхода

Три источника:

- Чарльз Дарвин, 1859

Принцип выживания сильнейших и естественный отбор

- Мендель, 1865

Основной принцип механизма наследования – хромосомы потомков состоят из частей хромосом их родителей

- де Вре, 1900

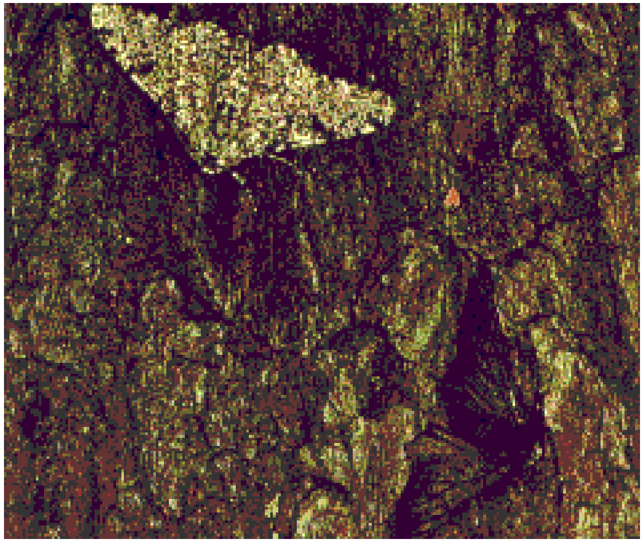
Принцип мутации – существенные (разные) изменения свойств потомков и приобретение ими свойств, отсутствующих у родителей. Подобный механизм используется для изменения свойств потомков и тем самым повышения разнообразия (изменчивость) особей в популяции (множество решений)

Три составные части

Закон Дарвина

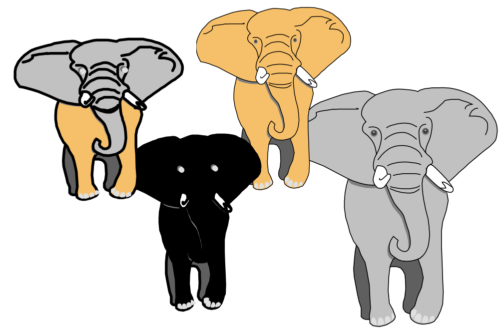
Естественный отбор

Адаптация к окружающей среде бабочки



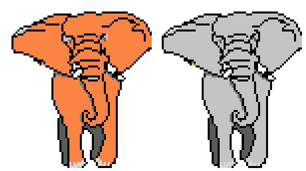
Закон Менделя

Хромосомы потомков состоят из частей хромосом их родителей

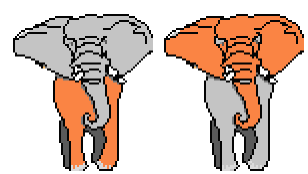


Популяция

Родители

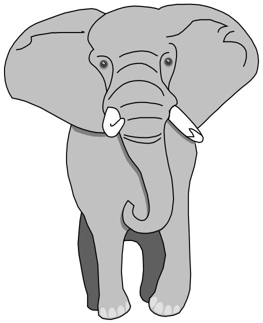


Потомки

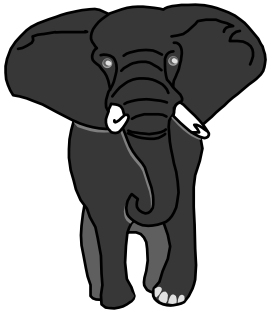


Закон Де Вре
Резкое изменение свойств

До мутации



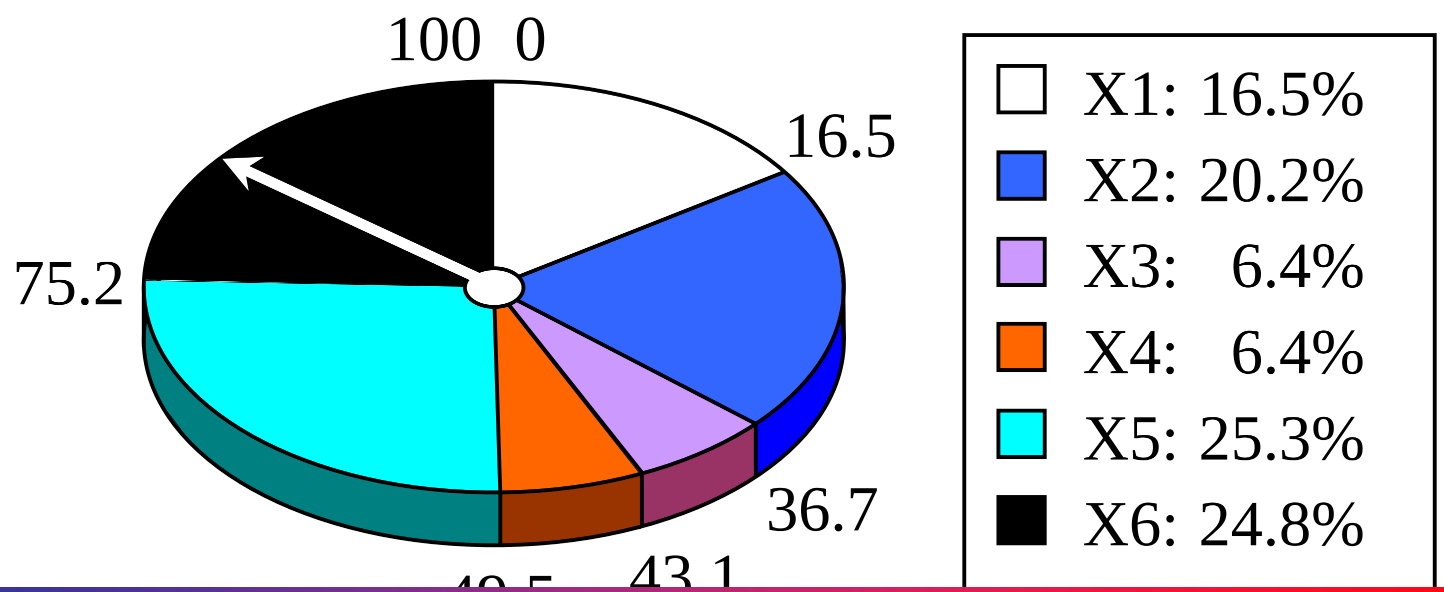
После мутации



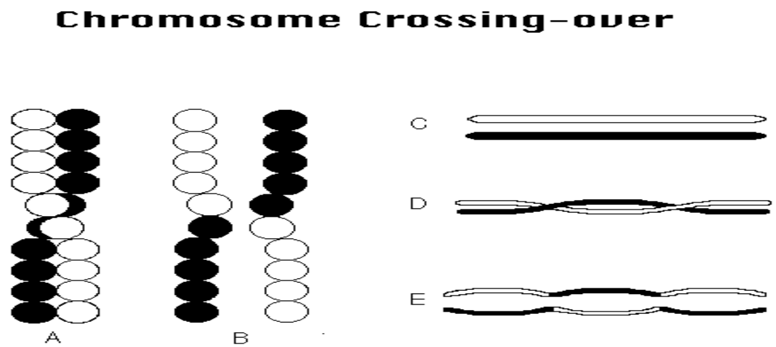
Оператор *селекции* – модель асимметричного колеса рулетки

Популяция отображается на колесо рулетки, где каждая особь представляется сектором, размер которого пропорционален значению фитнесс функции.

Особи выбираются путем повторного вращения на основе стохастического выбора с возвращением.



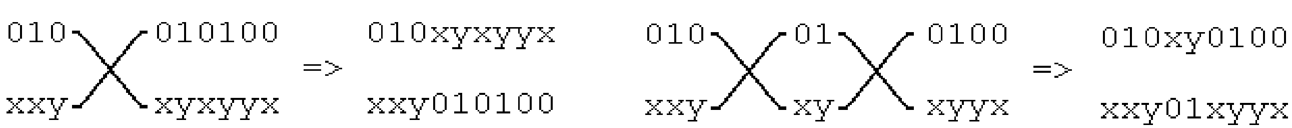
Оператор *кроссинговера*



Одноточечной или простой оператор кроссинговера в выполняется с вероятностью Pc в 2 этапа:

1) целое число – точка кроссинговера k ($1 \le k \le n$-1) выбирается случайным образом для выбранных строк $A=a$1 a2 … an и $B=b$1 b2… bn;

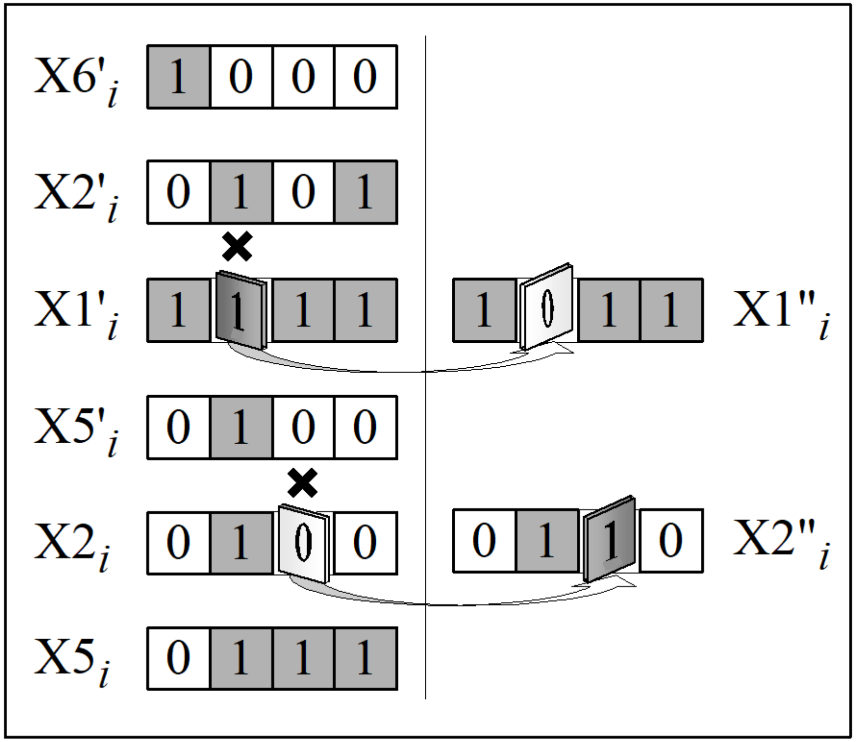
2) особи A и B обмениваются подстроками после позиции k и производят две новые особи $A=a$1 a2 … ak bk+1… bn и $B=b$1 b2… bk ak+1 … an.



Оператор *мутации*

Оператор мутации случайным образом (с малой вероятностью Pm) изменяет значения некоторых генов особей популяции:

01001011 → 01000011



ЭВОЛЮЦИОННЫЕ ВЫЧИСЛЕНИЯ

Основные парадигмы:

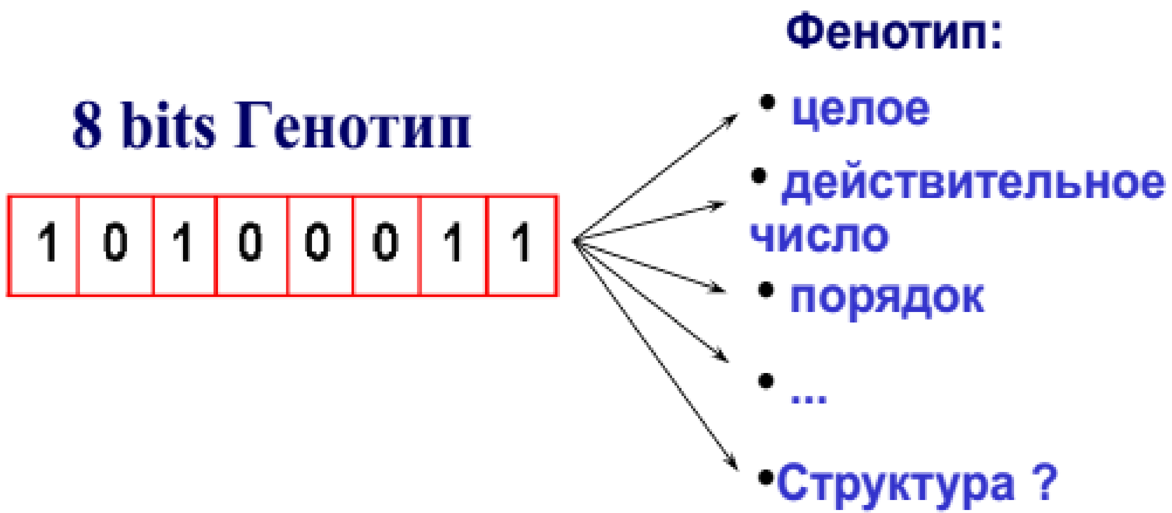
- Генетические алгоритмы
- Эволюционные стратегии
- Эволюционное программирование
- Генетическое программирование

КОДИРОВАНИЕ ХРОМОСОМ

Хромосомы могут представлять:

- Двоичные строки (0101 ... 1100)
- Действительные числа (43.2 -33.1 ... 0.0 89.2)
- Перестановки элементов (E11 E3 E7 ... E1 E15)
- Списки правил (R1 R2 R3 ... R22 R23)
- Элементы программ (genetic programming)
- Матрицы
- Графы
- .. любые структуры данных...

ПРИМЕР: ДИСКРЕТНАЯ РЕКОМБИНАЦИЯ



ПРИМЕР:ДЕЙСТВИТЕЛЬНОЕ ПРЕДСТАВЛЕНИЕ

- Естесственное кодирование в случае решения в виде массива вещественных чисел! (не двоичная строка из1 and 0
- Имеет много приложений
- например, оптимизация с ограничениями

- Особи представляются векторами действительных чисел

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, x_i \in R$$

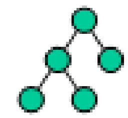
- Фитнесс- функция :

Отображает

$$f : R^n \to R$$

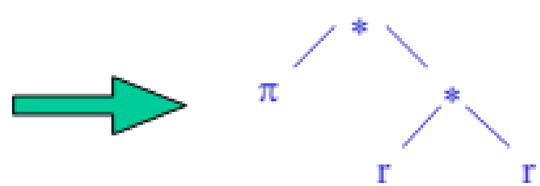
ДРЕВОВИДНОЕ ПРЕДСТАВЛЕНИЕ:

- Особи популяции - деревья.



- Любое S-выражение может быть представленодеревом функций и термина
  - Функции: sine, cosine, add, sub, and, If-Then-Else, Turn...
  - Терминалы: X, Y, 0.456, true, false, π, Sensor0…
- Пример: вычисление площади окружности:

$$\pi * r^2$$

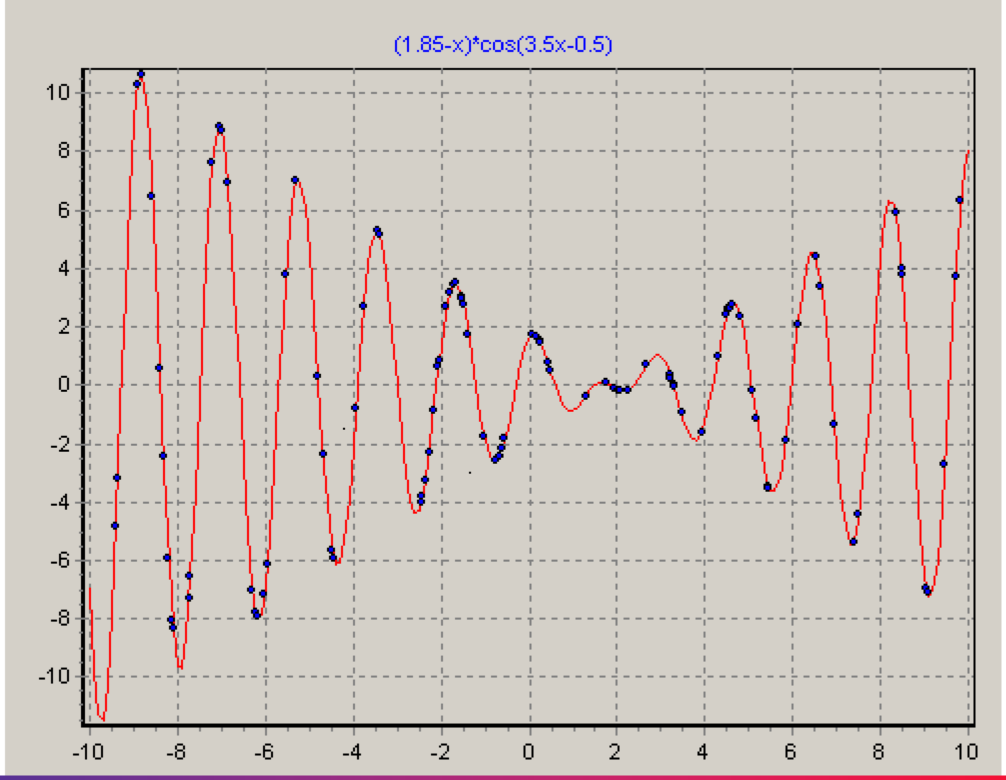


Пример применения га для поиска экстремума мультимодальной функции

Рассмотрим f(x)=(1,85 -x)*cos(3,5x – 0,5),

необходимо найти вещественное x, которое максимизирует f.

Пример функции с популяцией особей в начале эволюции



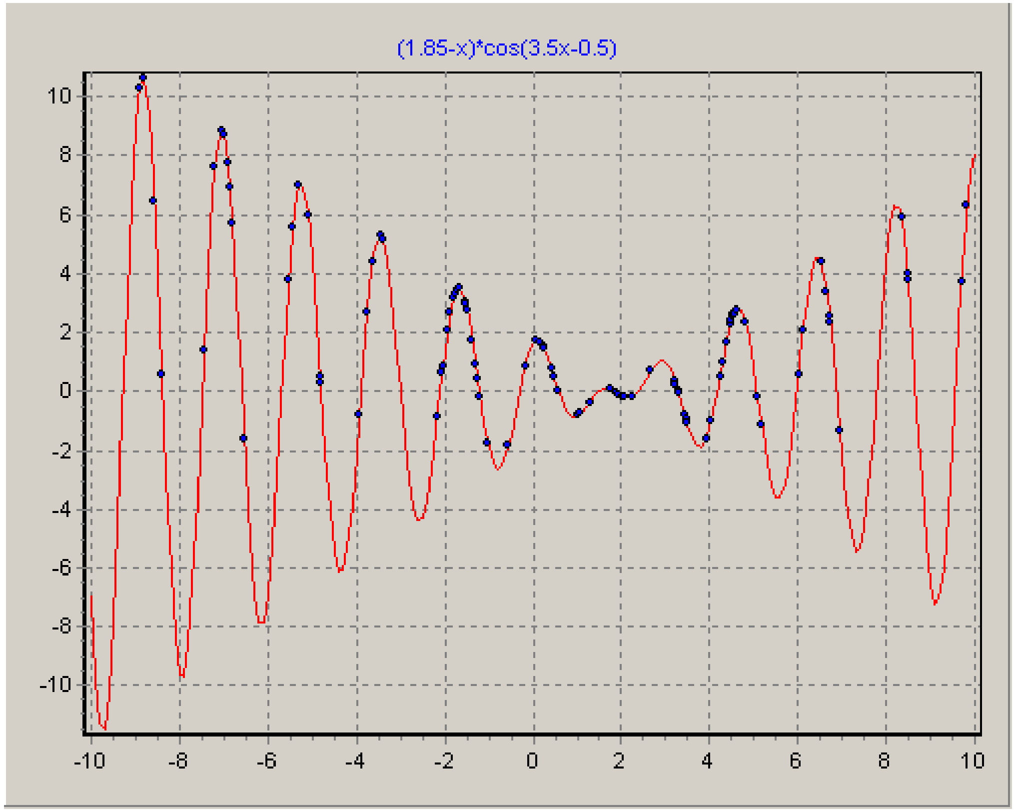
Пример: дискретное представление (двоичный алфавит)

Фенотип может быть действительным числом

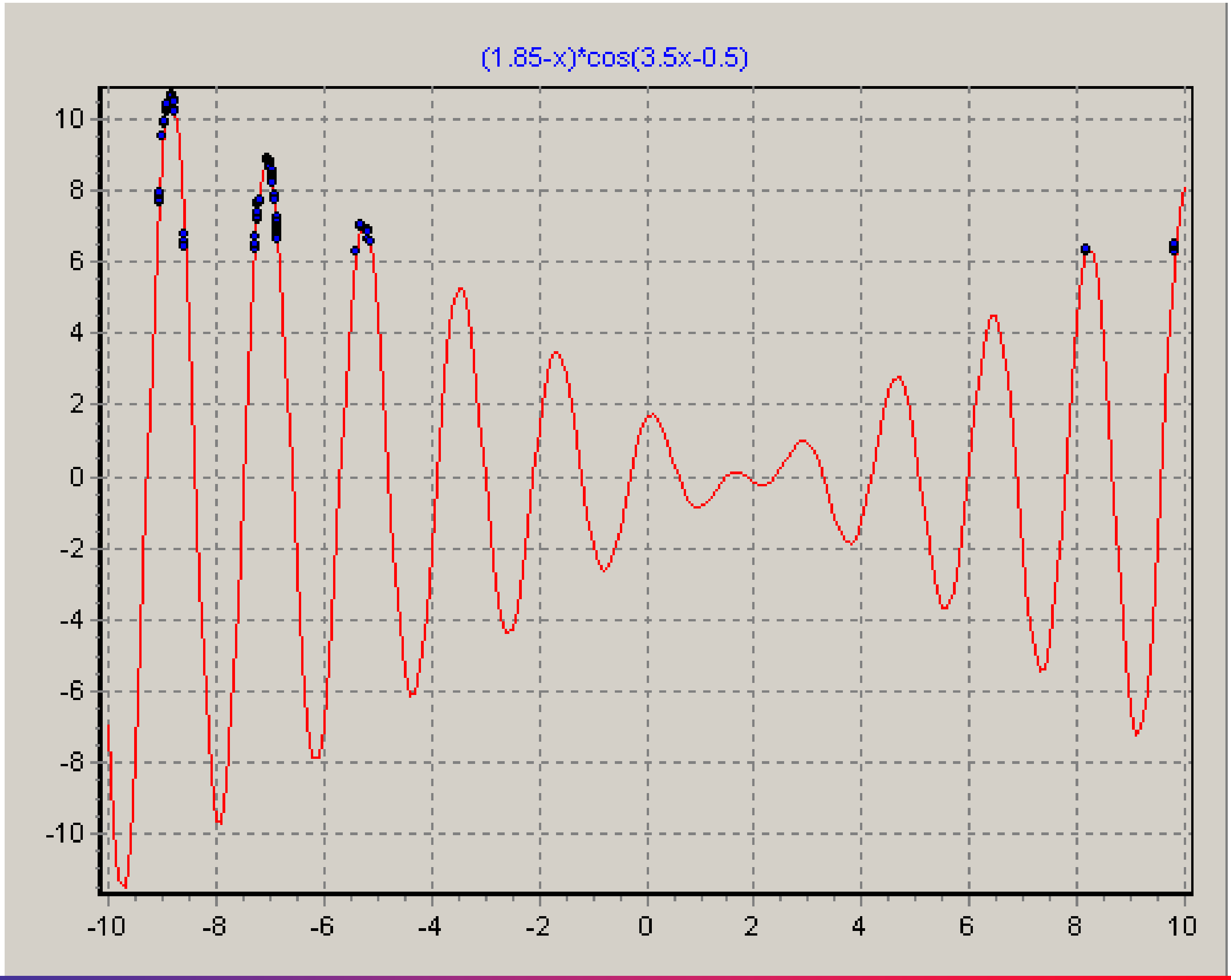
Genotype:									**Phenotype:**
1	0	1	0	0	0	1	1		= 13.9609

$$x = 2.5 + \frac{163}{256}(20.5 - 2.5) = 13.9609$$

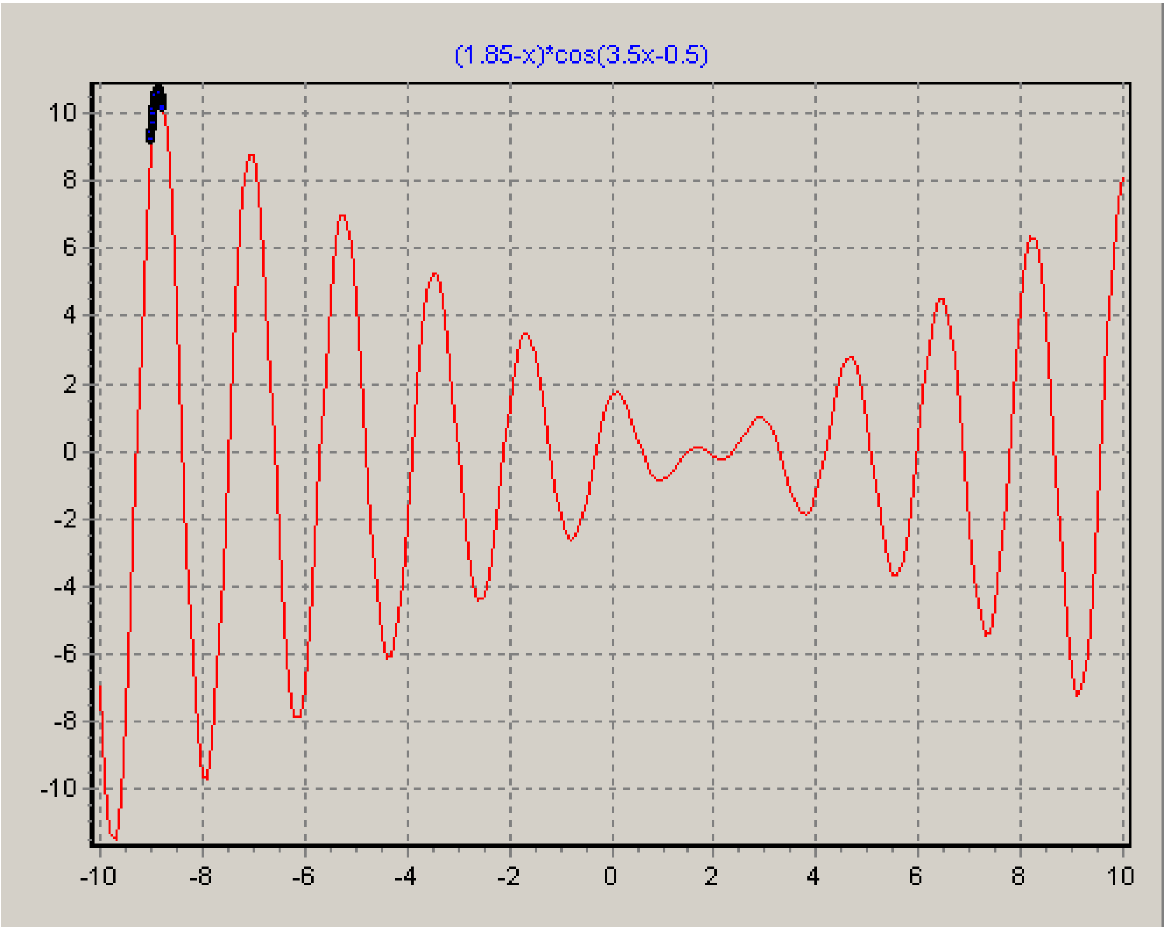
РАСПОЛОЖЕНИЕ ОСОБЕЙ НА РАЗЛИЧНЫХ ЭТАПАХ ГА В ПРОЦЕССЕ ПОИСКА РЕШЕНИЯ



«КОНДЕНСАЦИЯ» ОСОБЕЙ В ЭКСТРЕМУМАХ



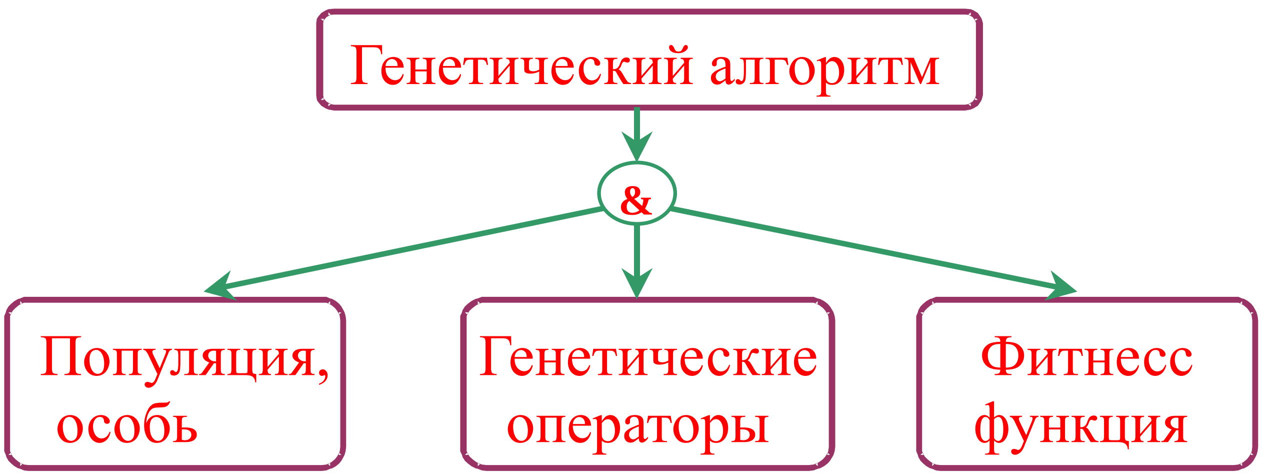
ПОЛОЖЕНИЕ В КОНЦЕ ЭВОЛЮЦИИ



Эволюционный подход

Для решения произвольной задачи с помощью ГА необходимо определить:

1) особь и популяцию;
2) генетические операторы;
3) фитнесс функцию;
4) Параметры.



ВЕРОЯТНОСТНЫЕ ЭВОЛЮЦИОННЫЕ МЕТАЭВРИСТИКИ

В настоящее время кроме детерминированных эволюционных метаэвристик, где популяция (множество потенциальных решений) представляется в явном виде (например, множеством двоичных строк, векторов или деревьев) получили распространение

и **вероятностные эволюционные метаэвристики**, часто называемые алгоритмами оценивания распределения (estimation of distribution algorithm, **EDA**)

или вероятностными моделями построения генетических алгоритмов (probabilistic model building genetic algorithm, **PMBGA**), где популяция представляется, например, некоторым вероятностным распределением.

Представление популяции

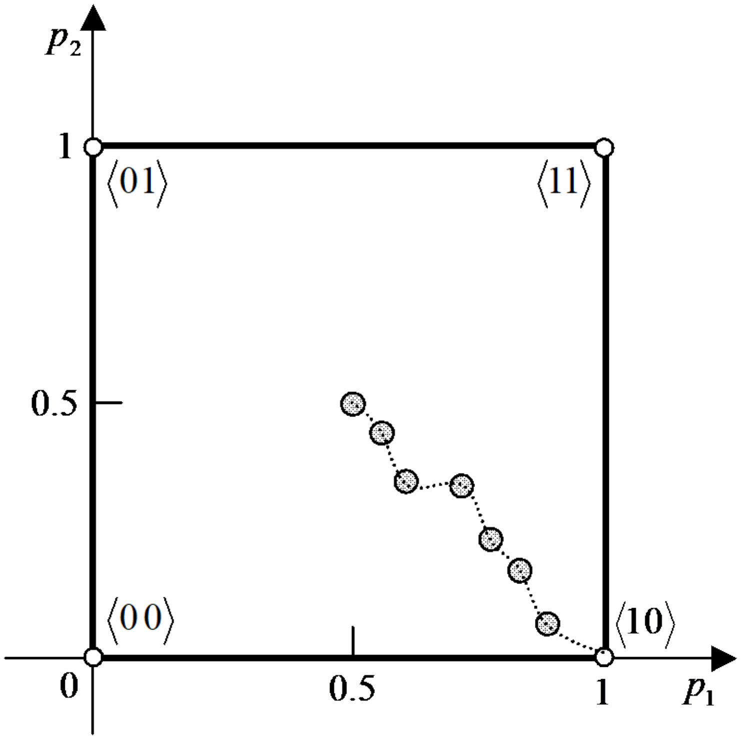
В EDA популяция представляется вектором вероятностей Р

1	0	1	0	1	0	1	0	1	0	1	0
1	1	0	0	0	0	1	1	0	1	1	0
1	0	1	0	1	1	0	0	0	1	1	0
1	0	0	0	0	0	1	0	1	0	1	0

P=(1, 0.25, 0.50, 0)   P=(0.5,0.25, 075, 025)   P=(0.5, 0.5, 1, 0)

Вероятностные генетические алгоритмы

В этом случае эволюция популяции соответствует траектории в гиперкубе пространства $(P_1, P_2,\dots,P_n)$



Репродукция.

В соответствии с текущим распределением вероятностей P генерируется некоторое (относительно небольшое) множество двоичных векторов – особей.

Оцениваются значения фитнесс-функции для всех особей.

Проводится турнир – находится лучшая особь.

Вектор вероятностей P сдвигается в сторону особи υ (вершины гиперкуба), имеющей лучшее значение фитнесс-функции:

$P' = (1-\theta)P + \theta\upsilon$. Здесь $\theta \in (0;1)$ - случайное число.

Мутация. Каждая координата вектора P корректируется случайным образом с заданной вероятностью π: $p_i' = (1-\mu)p_i + \mu z_i$, где $\mu \in (0;1)$ – случайный вещественный параметр, а z_i принимает значения 0 и 1 с равной вероятностью.

Пошаговое обучение на основе виртуальной популяции PBIL (Population-Based Incremental Learning)

Популяционное пошаговое обучение (population-based incremental learning) предложено Балуджей (Baluja). Этот алгоритм состоит из фазы создания новой популяции, фазы модификации вектора вероятности и фазы мутации вектора вероятности.

Укрупненный алгоритм

```
Пошаговое обучение()
    {
Инициализация вектора вероятностей P; // присваивание p_i = 0,5 для всех 1 ≤ i ≤ N

while (критерий_останова <> TRUE) // основной цикл алгоритма
  {
    // генерация особей виртуальной популяции S мощности M
    while ( k ≤ M )
      {
        генерация особи S_k согласно вектору вероятности P;
        оценка значения фитнесс-функции для S_k ;
      }
```

```
// поиск лучшей особи виртуальной популяции
max = поиск_максимума ( s );

// изменение вектора вероятности P
while ( i ≤ N )
   {
      p_i = p_i ·(1.0 – θ ) + max · θ
   }
    // мутация вектора вероятности P
while ( i ≤ N )
   {
      if (random (0,1) < mut_prob)  then
        p_i = p_i · (1.0 – mut_shift) + random (0 or 1) · mut_shift
   }
}
// формирование результирующего вектора X = (x_1, x_2, …, x_N)
while ( i ≤ N )
   {
      x_i = round ( p_i )
   }
  }
```

Компактный генетический алгоритм

Компактный ГА (КГА), как и PBIL, модифицирует вектор вероятности, однако помимо фазы кроссинговера здесь также игнорируется и фаза мутации, что позволяет сконцентрировать все внимание на механизме селекции, который основан на турнирном методе отбора особей. Кроме того, компактный ГА моделирует конечную популяцию и требует меньше вычислительных ресурсов. Псевдокод алгоритма представлен ниже:

Укрупненный алгоритм

```
Компактный ГА
    {
Инициализация вектора вероятностей p;
// присваивание p_i = 0,5 для всех 1 ≤ i ≤ N
// основной цикл алгоритма
while (критерий_останова <> TRUE)
  {
     // генерация двух особей виртуальной популяции мощности M
     a = сгенерировать (p);
     b = сгенерировать (p);
        // определение победителя и проигравшего
     if (фитнесс (a) > фитнесс (b))
```

```
        then   победитель = a
        else   победитель = b
    // изменение вектора вероятности p
    while (i ≤ N)
      {
          if (победитель [i] ≠ проигравший [i]) then
              if (победитель [i] = 1) then p_i = p_i + 1/M  else p_i = p_i − 1/M
      }
  }

return p.
    }
```

Компактный ГА в силу своей простоты допускает эффективную аппаратную реализацию, что позволяет повысить быстродействие на несколько порядков

Использование статистик второго порядка

В идеале при генерации популяции следующего поколения мы должны использовать полное вероятностное распределение лучших М особей подпопуляции.

Однако это требует больших компьютерных ресурсов.

Поэтому рассмотренные ранее методы используют статистику первого порядка.

Можно использовать статистики второго порядка, что требует, естественно, дополнительных вычислений.

Двумерные и многомерные вероятностные метаэвристики

Двумерные дискретные (бинарные), основанные на древовидных моделях:

– алгоритм двумерного предельного распределения;

–максимизация совместной информации для кластеров входного пространства;

– комбинирование оптимизаторов с деревьями совместной информации.

3. Многомерные дискретные (бинарные):

– алгоритм байесовской оптимизации (представляет класс метаэвристик, основанных на байесовских сетях);

– алгоритм марковской оптимизации (представляет класс метаэвристик, основанных на марковских сетях);

– расширенный компактный генетический алгоритм (представляет класс метаэвристик, основанных на кластеризации).

Роевые метаэвристики

Формально, рой (swarm) может быть определен как группа мобильных агентов, которые общаются друг с другом (прямо или косвенно), воздействуя на свое окружение.

Вычислительный роевой интеллект основан на алгоритмических моделях такого поведения.

К биологическим роевым системам относятся системы птиц, рыб, муравьев, пчел, ос, бактерий, лягушек, светлячков, летучих мышей, кукушек, кошек, сорняков и др.

К физическим роевым системам относятся системы частиц (капель) воды;
системы частиц, на которые воздействует гравитация, электромагнитные силы или электрические силы;
системы частиц, участвующих в диффузионных процессах;
системы частиц, основанные на большом взрыве или большом сжатии и др.
В пределах роя особи (или частицы) относительно просты по структуре, но их коллективное поведение обычно достаточно сложно.
Это поведение не является свойством любой одиночной особи (или частицы), а является свойством всего роя, и обычно оно не предсказывается и не выводится из простого поведения особей (или частиц) и называется «эмерджентным» (emergent).
Такое поведение является децентрализованным (не координируется никакой системой управления), самоорганизующимся и распределенным.

Оптимизация роя частиц (particle swarm optimization –PSO применяются при численной оптимизации)

Роевые алгоритмы (РА), также как и эволюционные, используют популяцию особей – потенциальных решений проблемы и метод стохастической оптимизации, который навеян (моделирует) социальным поведением птиц или рыб в стае или насекомых в рое.

В отличие от ГА здесь не используются генетические операторы, в РА особи (называемые частицами - particles) летают в процессе поиска в гиперпространстве поиска решений и учитывают успехи своих соседей.

Если одна частица видит хороший (перспективный) путь (в поисках пищи или защиты от хищников), то остальные частицы способны быстро последовать за ней, даже если они находились в другом конце роя.

С другой стороны, в рое, для сохранения достаточно большого пространства поиска, должны быть частицы с долей «сумасшествия» или случайности в своем поведении (движении).

Основной роевой алгоритм (PSO)

РА использует рой частиц, где каждая частица представляет потенциальное решение проблемы.

Поведение частицы в гиперпространстве поиска решения все время подстраивается в соответствии со своим опытом и опытом своих соседей.

Кроме этого, каждая частица помнит свою лучшую позицию с достигнутым локальным лучшим значением целевой (фитнесс-) функции и знает наилучшую позицию частиц - своих соседей, где достигнут глобальный на текущий момент оптимум.

В процессе поиска частицы роя обмениваются информацией о достигнутых лучших результатах и изменяют свои позиции и скорости по определенным правилам на основе имеющейся на текущий момент информации о локальных и глобальных достижениях.

При этом глобальный лучший результат известен всем частицам и немедленно корректируется в том случае, когда некоторая частица роя находит лучшую позицию с результатом, превосходящим текущий глобальный оптимум.

Каждая частица сохраняет значения координат своей траектории с соответствующими лучшими значениями целевой функции, которые обозначим y_i, которая отражает когнитивную компоненту.

Аналогично значение глобального оптимума, достигнутого частицами роя, будем обозначать , которое отражает социальную компоненту.

Каждая i-я частица характеризуется в момент времени t своей позицией $x_i(t)$ в гиперпространстве и скоростью движения $v_i(t)$.

Позиция частицы изменяется в соответствии со следующей формулой:

$x_i(t+1) = x_i(t) + v_i(t+1)$, *где* $x_i(0) \sim (x_{min}, x_{max})$.

Вектор скорости $v_i(t+1)$ управляет процессом поиска решения и его компоненты определяются с учетом когнитивной и социальной составляющей следующим образом:

$$v_{ij}(t+1) = v_{ij}(t) + c_1 r_{1j}(t)\left[y_{ij}(t) - x_{ij}(t)\right] + c_2 r_{2j}(t)\left[\hat{y}_j(t) - x_{ij}(t)\right]$$

Здесь: $v_{ij}(t)$ - j-ая компонента скорости ($j = 1, \ldots, n_x$) i-ой частицы в момент времени t, $x_{ij}(t)$- j-я координата позиции i-й частицы, c_1 и c_2 – положительные коэффициенты ускорения (часто полагаемые 2), регулирующие вклад когнитивной и социальной компонент,

$r_{1j}(t), r_{2j}(t) \sim (0,1)$ - случайные числа из диапазона [0,1], которые генерируются в соответствии с нормальным распределением и вносят элемент случайности в процесс поиска. Кроме этого $y_{ij}(t)$- персональная лучшая позиция по j-й координате i-ой частицы, а $\hat{y}_j(t)$ – лучшая глобальная позиция роя, где целевая функция имеет экстремальное значение.

При решении задач минимизации персональная лучшая позиция в следующий момент времени (t+1) определяется следующим образом:

$$y_i(t+1) = \begin{cases} y_i(t)\ iff\ f(x_i(t+1)) \geq f(y_i(t)) \\ x_i(t+1)\ iff\ f(x_i(t+1)) < f(y_i(t)) \end{cases}$$

где $f: R^{n_\infty} \to R$ - фитнесс-функция.

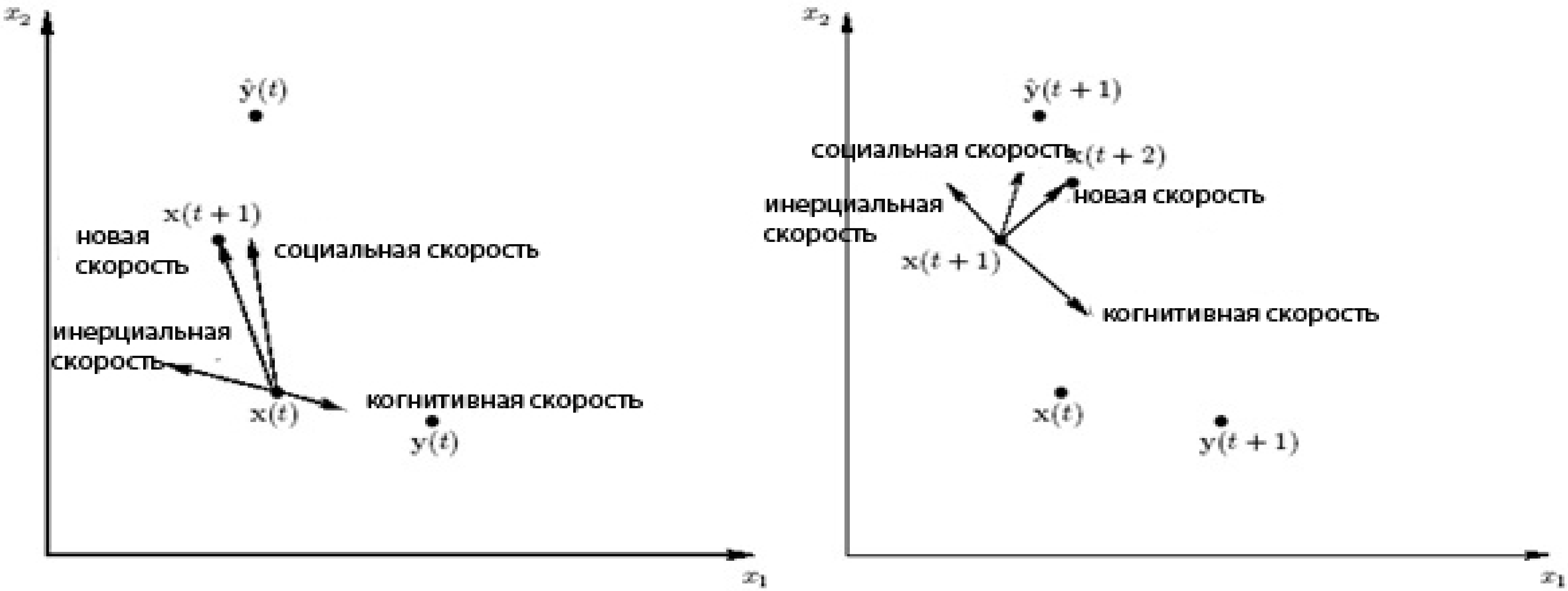
Как и в эволюционных алгоритмах фитнесс-функция измеряет близость текущего решения к оптимуму.

Глобальная лучшая позиция $\hat{y}_j(t)$ в момент t определяется в соответствии с

$$\hat{y}_j(t) \epsilon \{y_0(t), \dots, y_{n_s}(t)\} | f\left(\hat{y}_j(t)\right) = \min\left\{f(y_0(t)), \dots, f\left(y_{n_s}(t)\right)\right\},$$

Где n_s – общее число частиц в рое.

Графическая интерпретация коррекции скорости частицы



(a) скорость во время t

(b) скорость во время t+1

Основной роевой алгоритм

Глобальный роевой

Создание инициализация n_x-мерного роя;
repeat
for каждой частицы i=1,…,n_s **do**
// определить персональную лучшую позицию
If $f(x_i) < f(y_i)$ **then**
$y_i = x_i$;
end
// определить глобальную лучшую позицию
if $f(y_i) < f(\hat{y})$ **then**
$(\hat{y}) = y_i$;
end
end
for каждой частицы i=1,…,n_s **do**
коррекция скорости согласно (11.2);
коррекция позиции согласно (11.1);
end
until критерий останова не выполнен;

В настоящее время РА применяются при решении задач численной и комбинаторной оптимизации (существует дискретный вариант РА), обучении искусственных нейронных сетей, построении нечетких контроллеров и т.д. в различных областях науки техники:

- управление энергетическими системами;
- решение NP-трудных комбинаторных проблем;
- задачи календарного планирования;
- оптимизация в мобильной связи;
- оптимизация процессов пакетной обработки;
- оптимизация многокритериальных задач;
- обработка изображений;
- распознавание образов;
- кластеризация данных;
- биоинформатика;

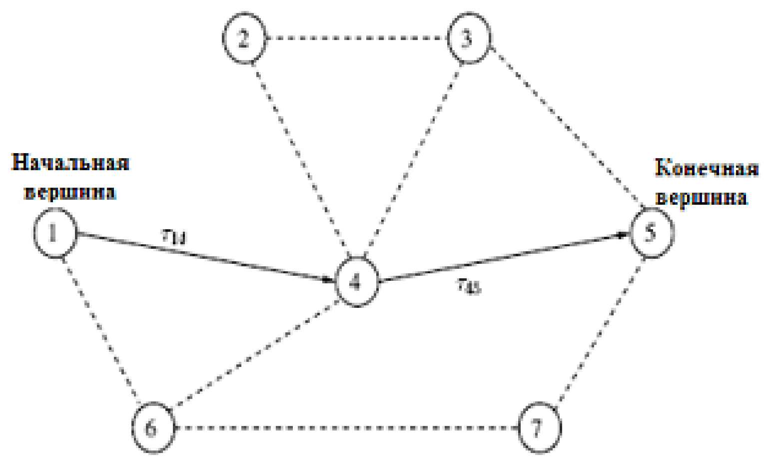
Муравьиные алгоритмы (Ant Colony Optimization -ACO при комбинаторной оптимизации)

В качестве иллюстрации возьмем задачу поиска кратчайшего пути между двумя узлами графа G=(V,E), где V – множество узлов (вершин), а E – матрица, которая представляет связи между узлами.

Пусть $n_G = |V|$ - число узлов в графе.

Обозначим L^k – длину пути в графе, пройденного k-м муравьем, которая равна числу пройденных дуг (ребер) от первой до последней вершины пути.

С каждой дугой, соединяющей вершины (i,j), ассоциируем концентрацию феромона τ_{ij} .



Муравей выбирает следующую дугу пути случайным образом фактически в соответствии с алгоритмом 12.1 .

Множество муравьев k={1,…,n_k} помещаются в начальную вершину.

В каждой итерации ПМА каждый муравей пошагово строит путь до конечной вершины. При этом в каждой вершине каждый муравей должен выбрать следующую дугу пути.

Если k-й муравей находится в i-ой вершине, то он выбирает следующую вершину $j \in N_i^k$ на основе вероятностей перехода

$$p_{ij}^k(t) = \begin{cases} \dfrac{\tau_{ij}^{\alpha}(t)}{\sum_{j \in N_i^k} \tau_{ij}^{\alpha}(t)}, & \text{если } j \in N_i^k \\ 0, & \text{если } j \notin N_i^k \end{cases},$$

Здесь N_i^k представляет множество возможных вершин, связанных с i-й вершиной, для k-го муравья.

Если для любого i-го узла и k-го муравья $N_i^k = \varnothing$, тогда предшественник узла i включается в N_i^k. В этом случае в пути возможны петли. Эти петли удаляются при достижении конечного города пути.

Когда все муравьи построили полный путь от начальной до конечной вершины, удаляются петли в путях, и каждый муравей помечает свой построенный путь, откладывая для каждой дуги феромон в соответствии со следующей формулой

$$\Delta\tau_{ij}^{k}(t) = \frac{1}{L^{k}(t)}$$

Здесь $L^k(t)$ – длина пути, построенного k-м муравьем в момент времени t.

Таким образом, для каждой дуги графа концентрация феромона определяется следующим образом

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \sum_{k=1}^{n_k} \Delta\tau_{ij}^{k}(t),$$

где n_k- число муравьев.

Простой муравьиный алгоритм

Инициализация $\tau_{ij}(0)$ малыми случайными значениями;
t=0;
поместить n_k муравьев на начальную вершину;
repeat
for каждого муравья $k=1,\ldots,n_k$ **do**
// построение пути $x^k(t)$
$x^k(t)=0$;
repeat

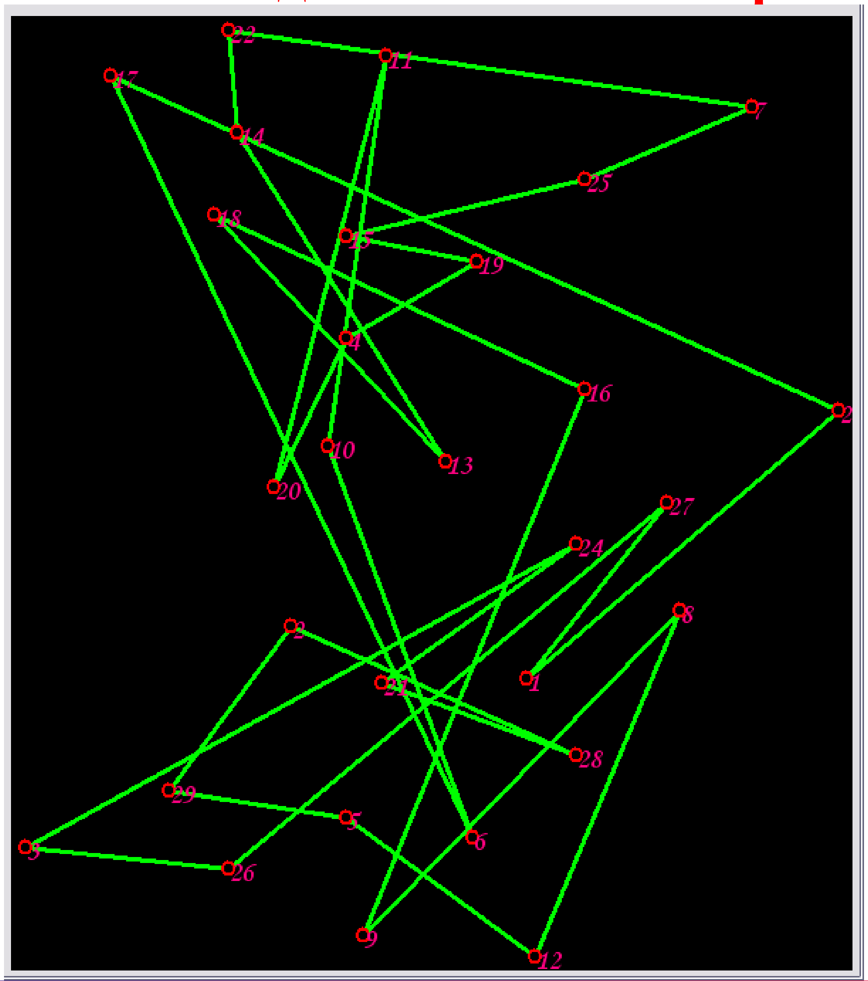
выбрать следующую вершину согласно вероятности, определяемой выражением (12.2);
добавить дугу (i,j) в путь $x^k(t)$;
until конечная вершина не достигнута;
удалить петли из $x^k(t)$;
вычислить длину пути $f(x^k(t))$
end

```
        for каждой дуги графа (i,j) do
        //испарение феромона
        Уменьшить концентрацию феромона согласно выражению(12.5);
        end
        for каждого муравья k=1,…,n_k do
                for каждой дуги(i,j) пути x^k(t) do
```

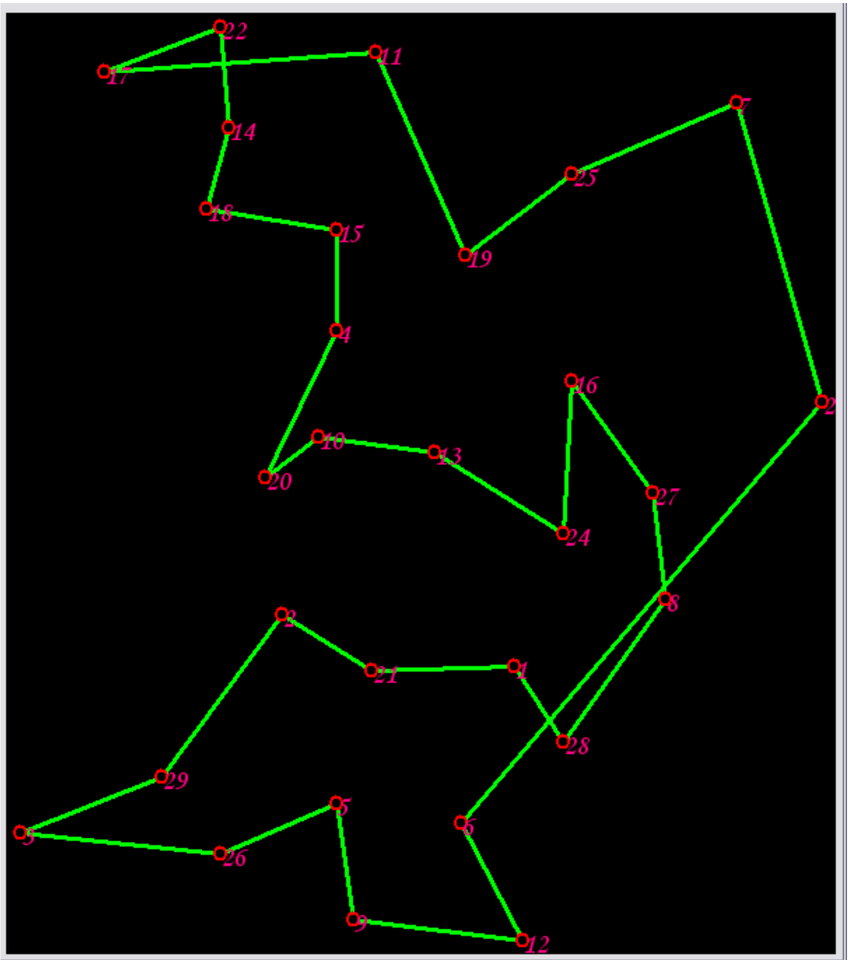
$$\Delta\tau^k = \frac{1}{f(x^k(t))};$$

```
                        Коррекция τ_ij согласно (12.5);
                End
        end
        t=t+1;
until   не выполнен критерий останова;
Возврат решения – пути с наименьшим значением f(x^k(t));
```

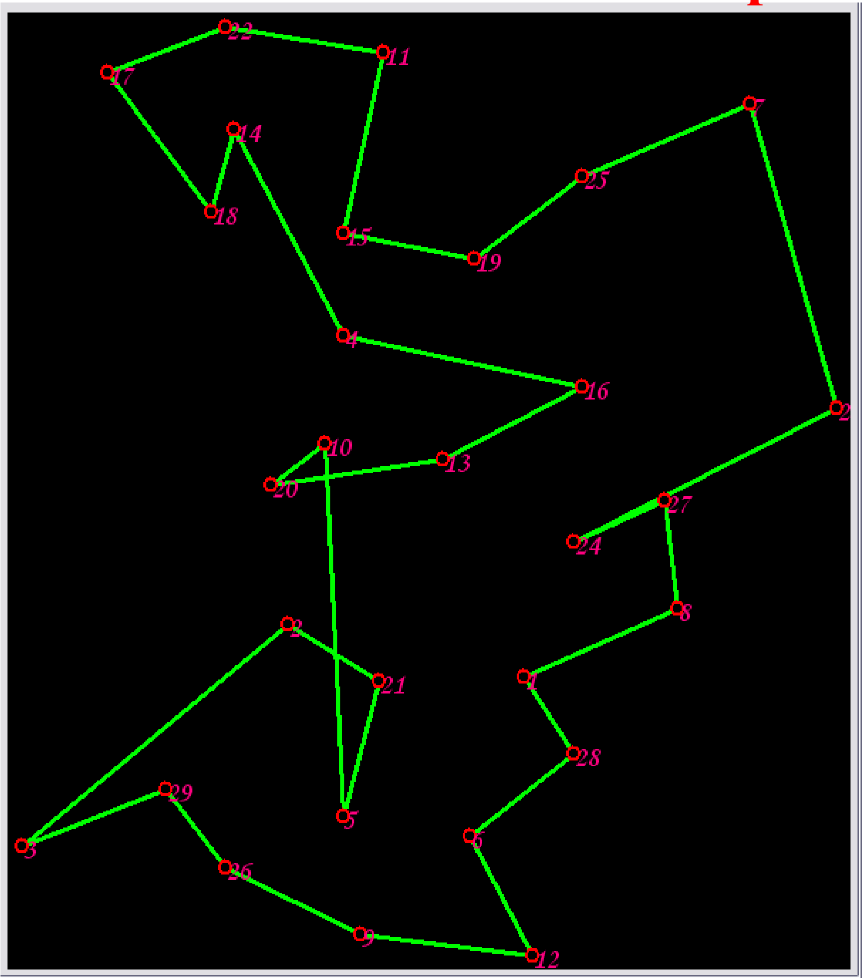
Муравьиные алгоритмы для решения задачи коммивояжера

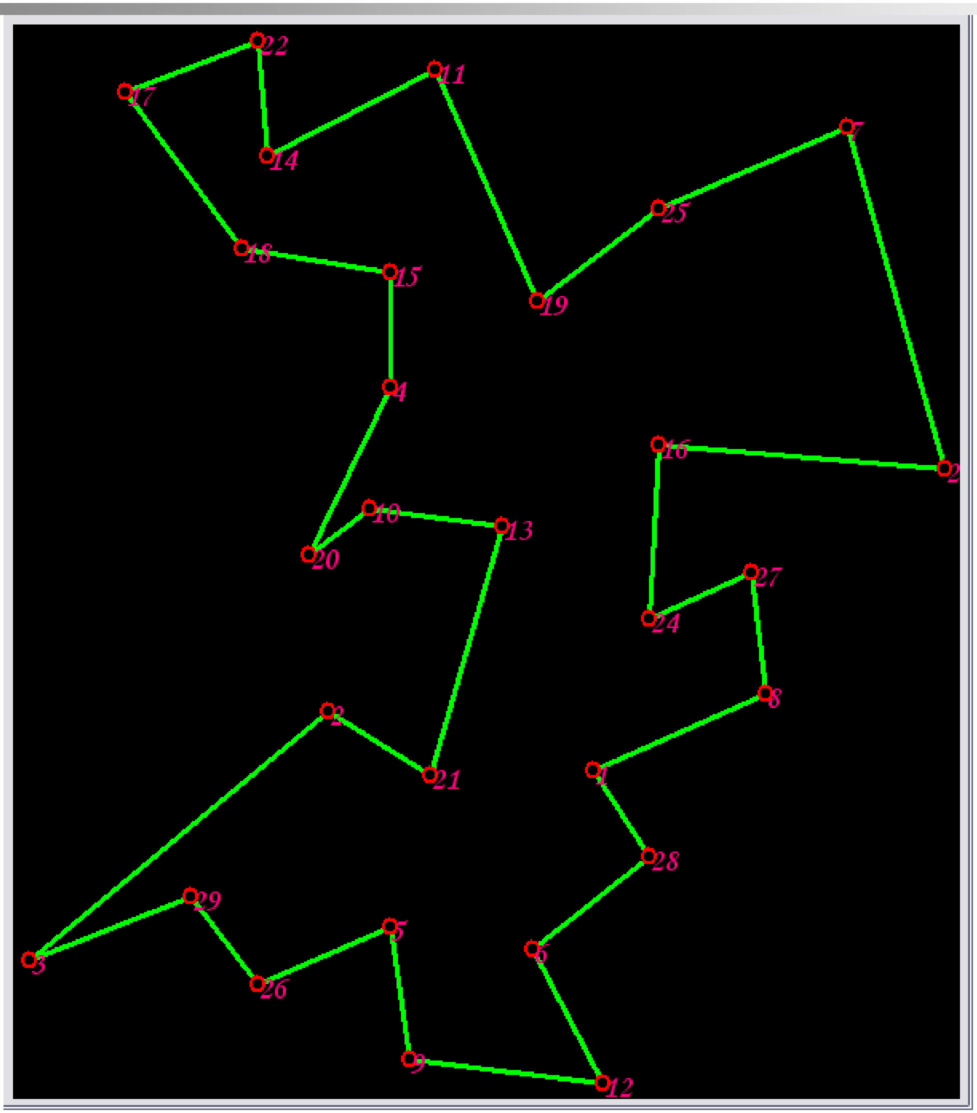


Муравьиные алгоритмы для решения задачи коммивояжера



Муравьиные алгоритмы для решения задачи коммивояжера




22
17
11
14
7
25
18
15
19
4
16
2
10
13
20
27
24
8
3
21
1
28
29
5
3
26
6
9
12

В настоящее время муравьиные алгоритмы получили применение при решении следующих практических задач :

1. маршрутизация (прежде всего, в сетях – networkrooting);
2. задачи назначения (quadratic assignment problem, graph coloring, generalized assignment, frequentially assignment);
3. машинное обучение (classification rules, Bayesian networks,
4. кластеризация данных;
5.эволюционная роботика;
6. календарное планирование и составление расписания;

max independent set, reduncy allocation, set covering, maximum clique, weight constrained graph tree partition, bin packing);

8. биоинформатика;
9. обработка текстов.

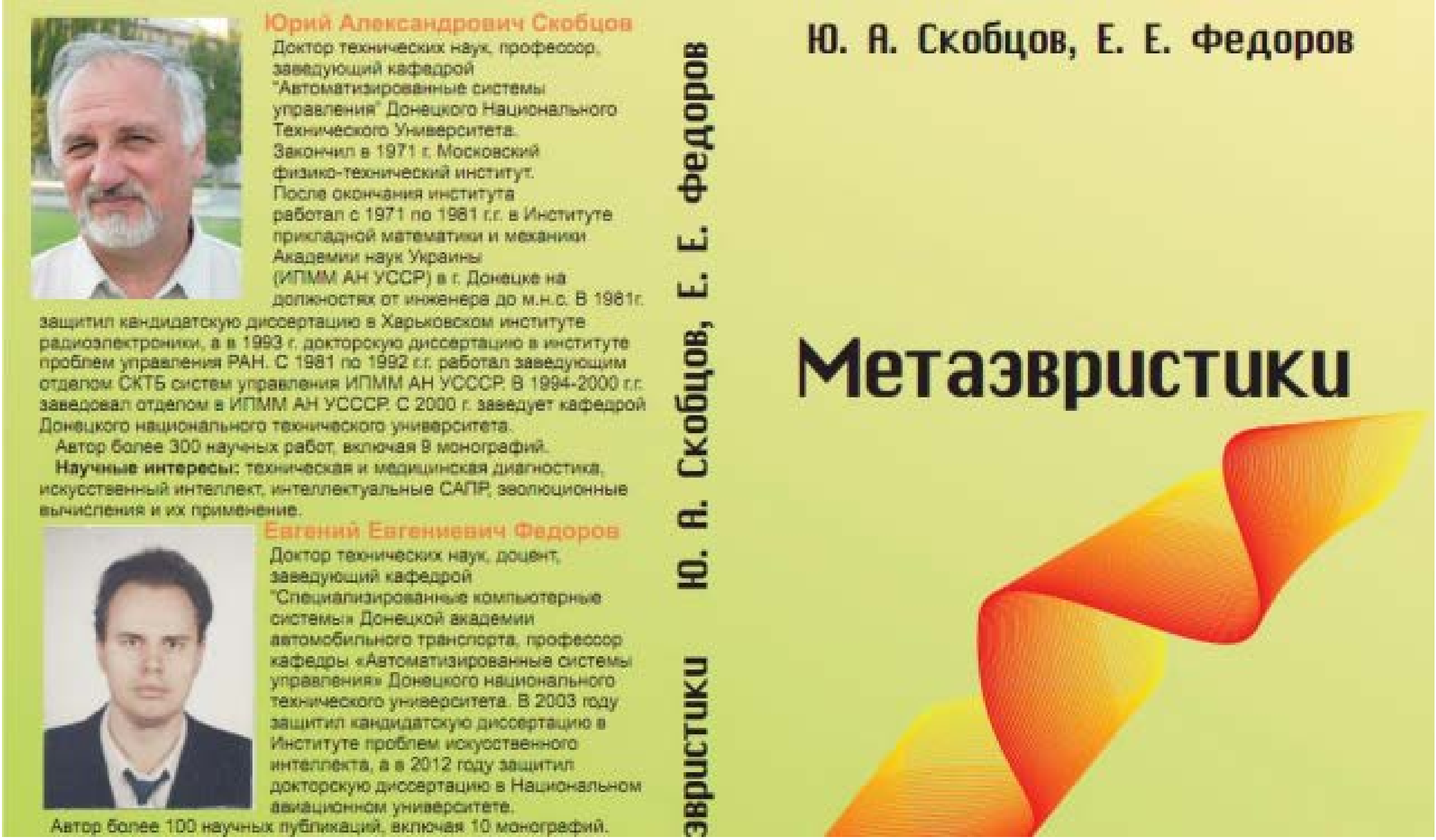
Этот список можно продолжить, поскольку число публикаций с использованием МА последние десять лет быстро растет.

СПАСИБО ЗА ВНИМАНИЕ!

Вы можете связаться с автором по следующему e-mail адресу
ya_skobtsov@list.ru

С основами метаэвристики можно ознакомиться в:

Скобцов Ю.А., Федоров Е.Е.Метаэвристики.-Донецк: изд-во «Ноулидж», 2013.-426с.



Скобцов Ю.А., Сперанский Д.В. **Эволюционные вычисления. М: Национальный Открытый Университет ИНТУИТ,2015.- 326 с.**

ИНТУИТ
НАЦИОНАЛЬНЫЙ ОТКРЫТЫЙ УНИВЕРСИТЕТ

Основы информационных технологий

Ю. А. Скобцов, Д. В. Сперанский

Эволюционные вычисления

Учебное пособие

М о с к в а
Национальный Открытый Университет «ИНТУИТ»
2 0 1 5