

Поиск оптимальной траектории  
при помощи самоподкорректирующегося  
генетического алгоритма

Формулировка задачи:

$D$  - область поиска  
 $D \in R^n$ ,  $n = 1, 2, 3, \dots$ ,  $S$  - множество  
допустимых кривых

Необходимо подобрать кривую  
 $f(x) \in S: L(f(x)) \rightarrow \min$

$L(f(x))$  - длина кривой  $f(x)$

Алгоритм оптимизации

→ хорошо разработана математическая модель

Требуют точной и полной информации, окруж. средой  
< идеал море окр. среда >

Осн. идеи:

использование принципов естественного отбора и структуризации  
поиска хороших стратегий

Алгоритм ДТ можно  
получить преобразованием  
уравнений Б-матрицы

тепловые. рост объема обработки  
данных

## Тест. алгоритм

- размерность задачи вышка
- на траекторию нам-ся  
ред ограничений
- не редует анализ  
задач с ограничениями

Размерность вышек только  
на продолжительность  
вычислений

Уникальная процедура поиска  
Опт. функ.

$$\text{fitness}(H(x)) = \frac{1}{1 + H(H(x))}$$

стоимость прох  
по кривой  $f(x)$

$$H(H(x)) = L(H(x)) + \sum_{i=1}^n [w_i \cdot I(f(x))]$$

$n$  - кол-во запрещен областей  
в обл. поиска

$w_i$  - вес  $i$ -ой обл.,  $I(f(x))$  - штраф  
запр. обл. на  
кривую

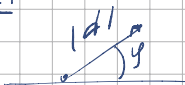
Решение - бинарный код,

исслед-ть параметров генов  
траектории

$n$	$p_1$	$p_2$	...	$p_n$
-----	-------	-------	-----	-------

где  $p_n$  : 

$y_i$	$d_i$
-------	-------



func lambda0 = genetic\_algo(8ys)  
popul = d \* rand(8ys, pop\_size, 5) - 1

# Муравьиный алгоритм

→ маркировка более удачных путей большим количеством феромона

Работа начинается с размещением муравьёв в вершинах графа (городах)

Направление передвижения муравья

$$P_i = \frac{L_i^\alpha \cdot \eta_i^\beta}{\sum_{k=0}^n L_k^\alpha \cdot \eta_k^\beta}$$

$$L_i = \frac{1}{W}$$

$W$  - длина (вес) перехода

$\alpha$  - влияние феромона

$\beta$  - влияние видимости

Пока (усл. выхода не выполнено)

1. Создаём муравьёв

2. Ищем решения

3. Обновляем феромон

4. Возвращаем результат

→

$$P_{ij}/H = \frac{\tau_{ij}/H^{\alpha} \left(\frac{1}{d_{ij}}\right)^{\beta}}{\sum_{\text{je allowed nodes}} \tau_{ij}/H^{\alpha} \left(\frac{1}{d_{ij}}\right)^{\beta}}$$

$\alpha = 0 \rightarrow$  выбор оптимального  
пути наименьшей вероятности  
 $\hookrightarrow$  алгоритм становится  
марковским

$\beta = 0 \rightarrow$  выбор производится только  
на основании вероятности  
что перейдем к субоптимально-  
му решению

$$\tau_{ij}/(t+1) = (1-\rho) \tau_{ij}/H + \sum_{\text{ke delay that used edge (i,j)}} \frac{Q}{L_k}$$

$\rho$  - интенсивность  
испарения

$L_k$  - цена текущего решения  
для  $k$ -го муравья

$Q$  - параметр контроля цены  
 $Q/L_k$  - штраф откляса -  
влияет  $k$ -ым муравьем  
используемым ребро  $(i,j)$

