

ІТМО

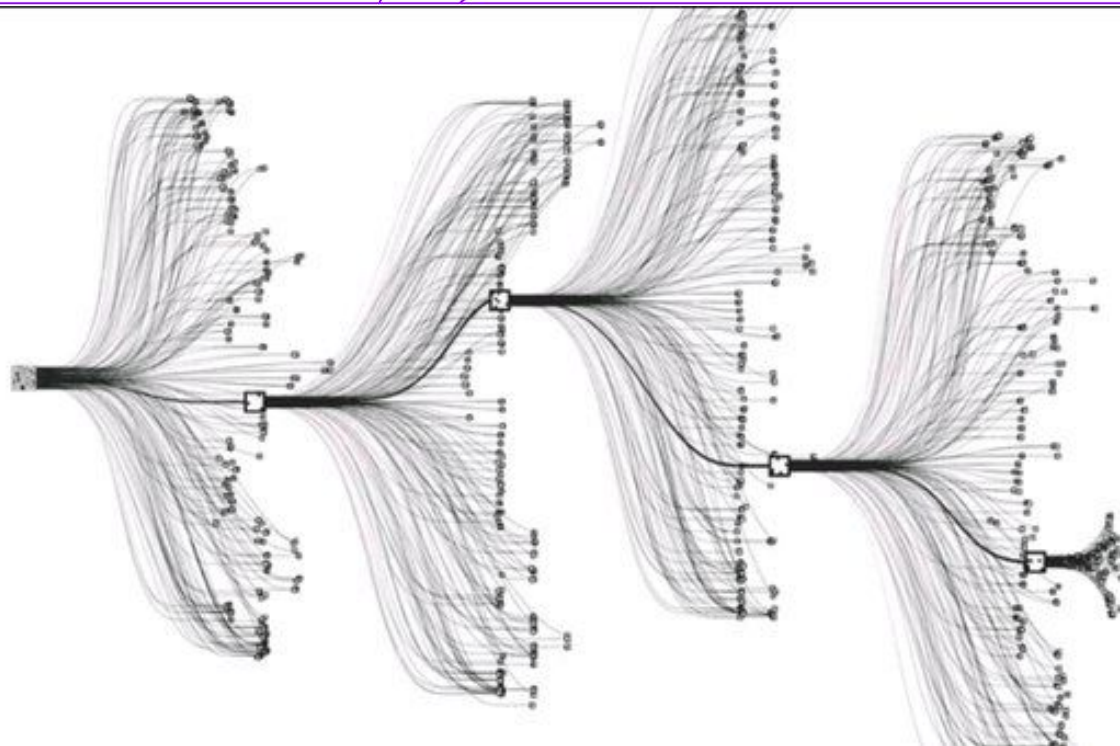
Введение в GameDev: ОСНОВЫ ИГРОВОГО ИИ

Лекция 6. Эволюционные и роевые алгоритмы



1. Оптимизация и эволюционные вычисления
2. Классический генетический алгоритм
3. Комбинаторная оптимизация
4. Роевые алгоритмы
5. Муравьиный алгоритм
6. Генетическое программирование
7. Подведение итогов курса

ГЛАВА 6 _ Оптимизация и алгоритмы ИТМО



Введение в оптимизацию, актуальность для GameDev

ГЛАВА 6 _ Оптимизация и алгоритмы ИТМО

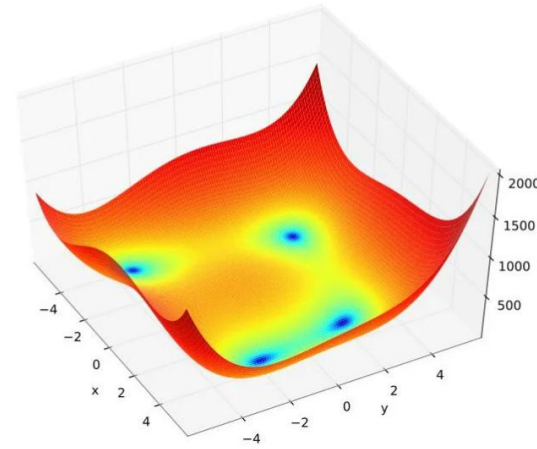
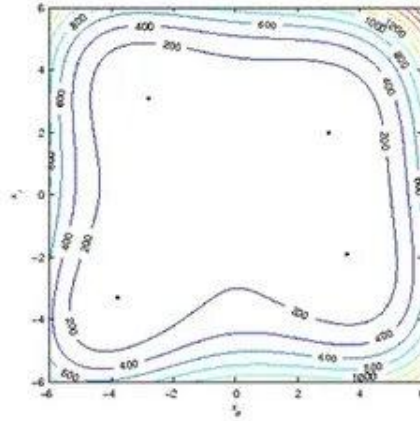
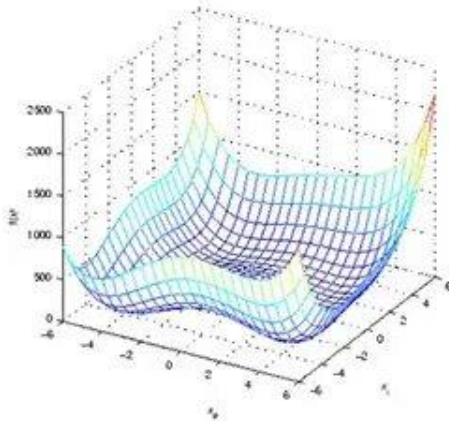
$$F \rightarrow (x) \max \setminus \min$$

Необходимо найти такой x^* ,
чтобы $F(x^*)$ принимала максимальное\минимальное
значение, причем ограничений на ОДЗ нет

Например, популярной функцией для
тестирования методов оптимизации
считается **функция Химмельблау**

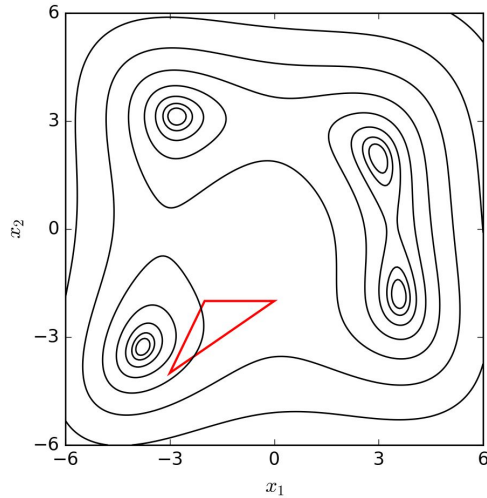


$$f(x, y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2$$



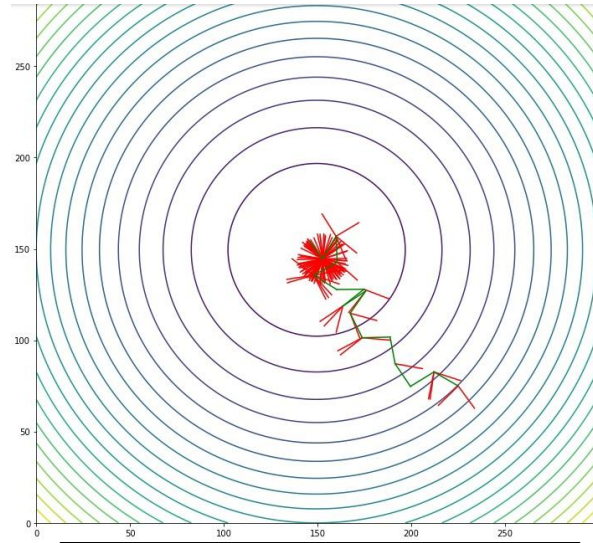
ГЛАВА 6 _ Оптимизация и алгоритмы ИТМО

Детерминированные
методы



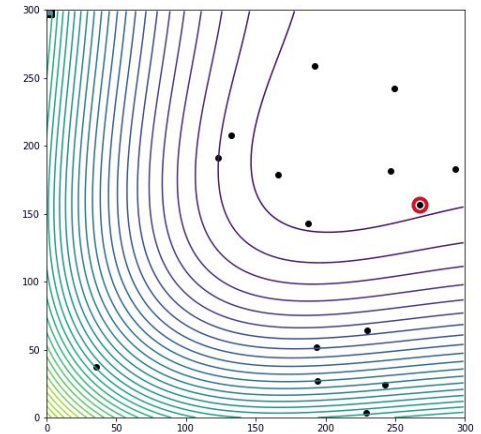
Алгоритм Нелдера-Мида

Стохастические методы



Метод с возвратом при
неудачном шаге

Эвристические методы



Кукушкин поиск

ГЛАВА 6 _ Оптимизация и алгоритмы ИТМО

Алгоритмы, вдохновленные живой природой

- Муравьиный алгоритм
- Кукушкин поиск
- Алгоритм пчелиной колонии
- Обезьяний поиск
- Бактериальная оптимизация
- Алгоритм роя частиц
- Сорняковый алгоритм
- Поиск косяком рыб
- Тасующий алгоритм прыгающих лягушек
- Иммунные системы
- Генетический алгоритм

Алгоритмы, вдохновленные физическими явлениями

- Гравитационный поиск
- Электромагнитный поиск
- Гармонический поиск
- Имитация отжига

Алгоритмы, инспирированные человеческим обществом

- Алгоритм, эволюции разума
- Культурный алгоритм
- Меметические алгоритмы
- Миграционный алгоритм

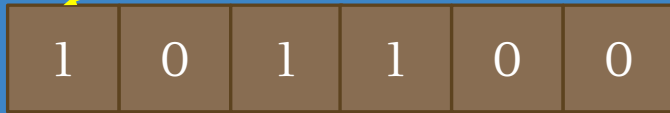
Алгоритмы, вдохновленные природой

ГЛАВА 6 _ Генетический алгоритм

Принципы и терминология — **генетика**. Основные принципы:

1. Выживание сильнейших.
2. Хромосома потомка состоит из частей, полученных из хромосом родителей.
3. Концепция мутации.

Ген = {1, 0}

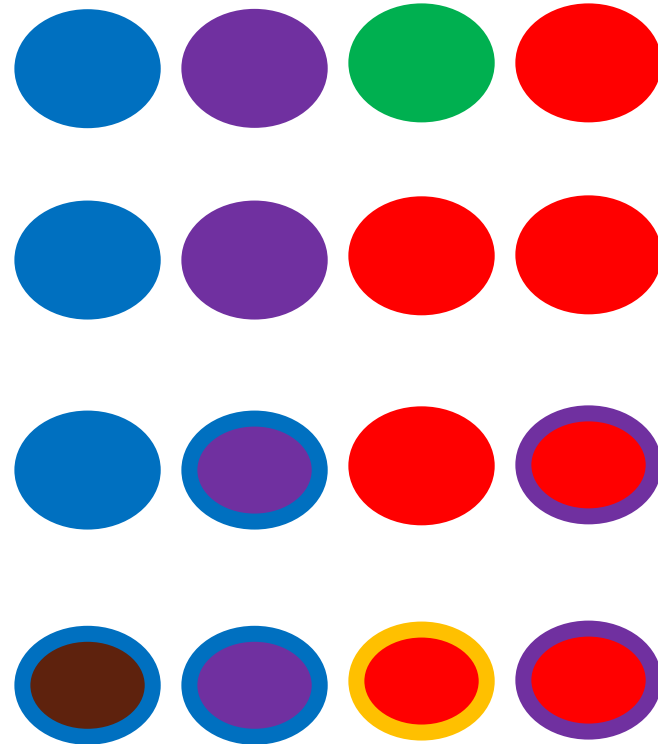
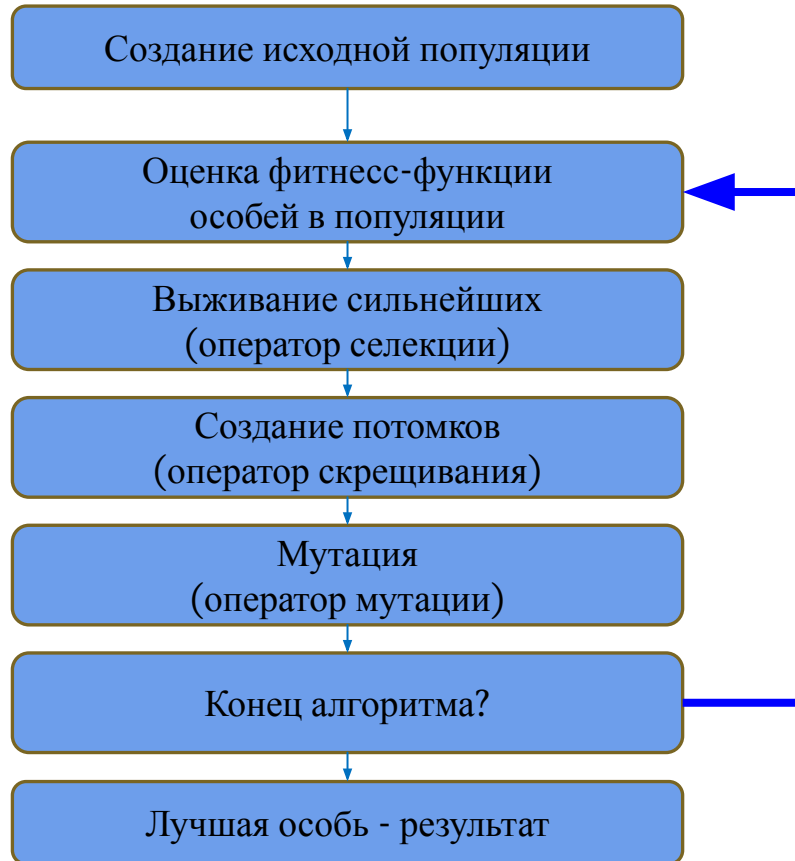


Одна особь = Одно решение

Хромосома

ГЛАВА 6 _ Генетический алгоритм

ІІТМО



Оператор Селекции: Колесо рулетки

$$F(x_1) = 32$$

$$F(x_2) = 16$$

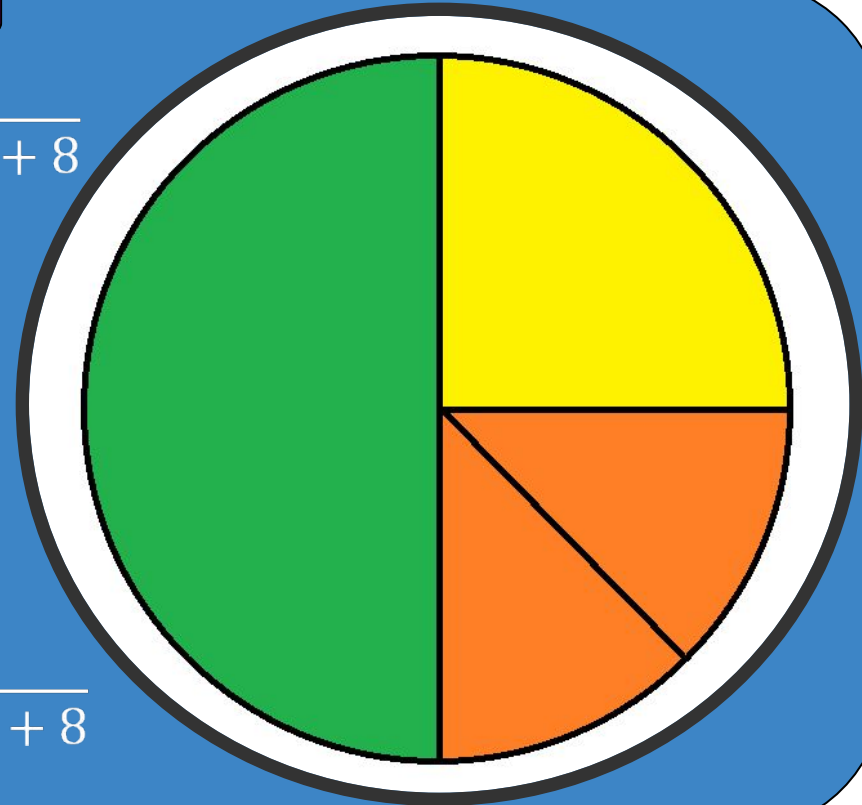
$$F(x_3) = 8$$

$$F(x_4) = 8$$

$$P(x_1) = \frac{32}{32 + 16 + 8 + 8}$$

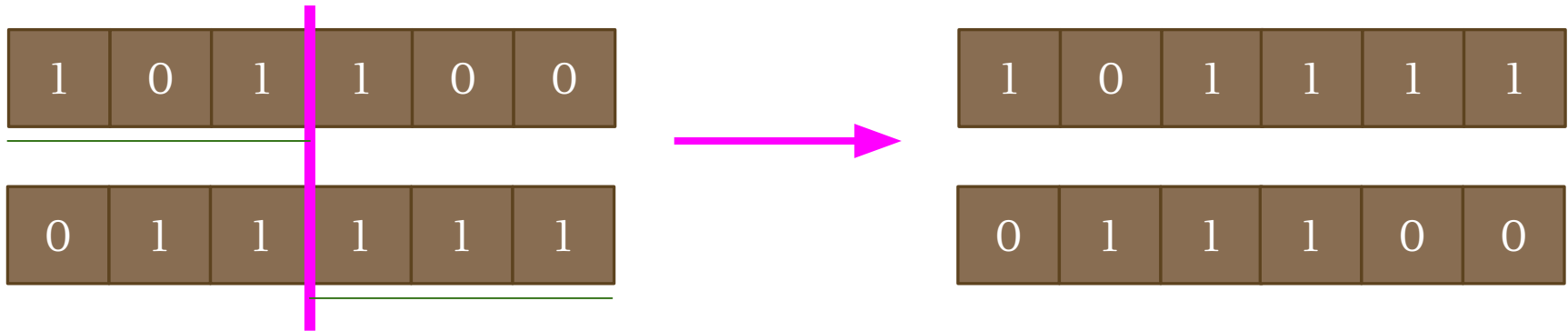
$$P(x_i) = \frac{f(x_i)}{\sum f(x_j)}$$

$$P(x_4) = \frac{8}{32 + 16 + 8 + 8}$$



ГЛАВА 6 _ Генетический алгоритм

Оператор Скрещивания: Одноточечный кроссинговер



Делим по парам особи

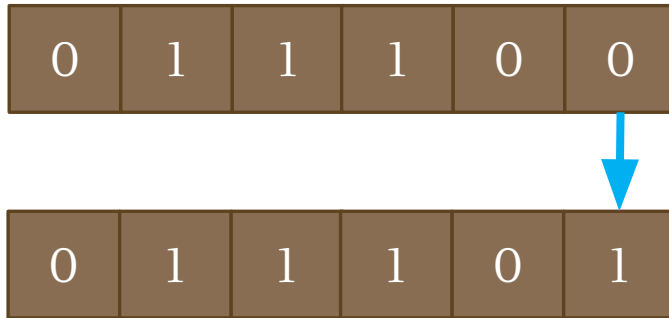
$P_{\text{crossover}}$ (0.5)

Определяем точку
скрещивания

Обмен подстрок

ГЛАВА 6 _ Генетический алгоритм

Оператор Мутации: Классический



Каждая особь

$P_{\text{mutation}} (0.05)$

Выбор случайного
гена

Инвертируем этот ген

ГЛАВА 6 _ Генетический алгоритм

ІІТМО

$$f(x) = 200 - x^2 \rightarrow \max$$
$$x \in [-15, 16]$$

$$N = 6$$

$$P_c = 0.5$$

$$P_m = 0.1$$

$$2^{gens} = 32 \rightarrow gens = 5$$

00000 .. 11111

0 .. 31

0.766
0.989
0.077
0.171
0.160
0.239
0.239
0.792
0.915
0.359
0.114
0.694
0.444
0.849
0.105
0.500
0.973
0.881
0.406
0.029
0.034
0.466
0.038
0.798
0.570
0.449
0.052
0.486
0.364
0.294
0.730
0.873
0.053
0.704
0.574
0.176

ГЛАВА 6 _ Генетический алгоритм

ІІТМО

1. Инициализация популяции

$$f(x) = 200 - x^2 \rightarrow \max$$

$$x \in [-15, 16]$$

$$N = 6$$

$$P_c = 0.5$$

$$P_m = 0.1$$

$$2^{gens} = 32 \rightarrow gens = 5$$

00000 .. 11111

0 .. 31

```
for i in range(6):  
    print(i, ' = ', [random.choice([0, 1]) for _ in range(5)])
```

0 = [0, 1, 1, 0, 0]

1 = [1, 0, 1, 0, 0]

2 = [0, 1, 0, 1, 1]

3 = [1, 0, 1, 1, 0]

4 = [1, 1, 1, 1, 0]

5 = [1, 1, 1, 1, 0]

0.766
0.989
0.077
0.171
0.160
0.239
0.239
0.792
0.915
0.359
0.114
0.694
0.444
0.849
0.105
0.500
0.973
0.881
0.406
0.029
0.034
0.466
0.038
0.798
0.570
0.449
0.052
0.486
0.364
0.294
0.730
0.873
0.053
0.704
0.574
0.176

ГЛАВА 6 _ Генетический алгоритм

ІІТМС

2. Оценка фитнес-функции

$$f(x) = 200 - x^2 \rightarrow \max$$
$$x \in [-15, 16]$$

$$N = 6$$

$$P_c = 0.5$$

$$P_m = 0.1$$

$$2^{gens} = 32 \rightarrow gens = 5$$

00000 .. 11111

0 .. 31

Хромосома
(Генотип)

Фенотип

f(x)

$$0 = 01100$$

$$0 = 12-15 = -3$$

$$0 = 191$$

$$1 = 10100$$

$$1 = 20-15 = 5$$

$$1 = 175$$

$$2 = 01011$$

$$2 = 11-15 = -4$$

$$2 = 184$$

$$3 = 10110$$

$$3 = 22-15 = 7$$

$$3 = 151$$

$$4 = 11110$$

$$4 = 30-15 = 15$$

$$4 = 75$$

$$5 = 11110$$

$$5 = 30-15 = 15$$

$$5 = 75$$

0.766
0.989
0.077
0.171
0.160
0.239
0.239
0.792
0.915
0.359
0.114
0.694
0.444
0.849
0.105
0.500
0.973
0.881
0.406
0.029
0.034
0.466
0.038
0.798
0.570
0.449
0.052
0.486
0.364
0.294
0.730
0.873
0.053
0.704
0.574
0.176

ГЛАВА 6 _ Генетический алгоритм

ІІТМС

3. Оператор селекции

Sum(f(x)) = 851

$$f(x) = 200 - x^2 \rightarrow \max$$

$$x \in [-15, 16]$$

$$N = 6$$

$$P_c = 0.5$$

$$P_m = 0.1$$

$$2^{gens} = 32 \rightarrow gens = 5$$

$$00000 \dots 11111$$

$$0 \dots 31$$

Хромосома
(Генотип)

Фенотип

f(x)

p(x)

0 = 01100

0 = 12-15 = -3

0 = 191

0 = 0.22

1 = 10100

1 = 20-15 = 5

1 = 175

1 = 0.20

2 = 01011

2 = 11-15 = -4

2 = 184

2 = 0.22

3 = 10110

3 = 22-15 = 7

3 = 151

3 = 0.18

4 = 11110

4 = 30-15 = 15

4 = 75

4 = 0.09

5 = 11110

5 = 30-15 = 15

5 = 75

5 = 0.09

0

0.22

0.42

0.64

0.82

0.91

1

$$P(x_i) = \frac{f(x_i)}{\sum f(x_j)}$$

После селекции остаются – 3, 5, 0, 0, 0, 1

0.766
0.989
0.077
0.171
0.160
0.239
0.239
0.792
0.915
0.359
0.114
0.694
0.444
0.849
0.105
0.500
0.973
0.881
0.406
0.029
0.034
0.466
0.038
0.798
0.570
0.449
0.052
0.486
0.364
0.294
0.730
0.873
0.053
0.704
0.574
0.176

ГЛАВА 6 _ Генетический алгоритм

ІІТМО

4. Оператор скрещивания

$$f(x) = 200 - x^2 \rightarrow \max$$

$$x \in [-15, 16]$$

$$N = 6$$

$$P_c = 0.5$$

$$P_m = 0.1$$

$$2^{gens} = 32 \rightarrow gens = 5$$

00000 .. 11111

0 .. 31

Хромосома
(Генотип)

Фенотип

f(x)

$$0 = 01100$$

$$1 = 10100$$

$$3 = 10110$$

$$5 = 11110$$

$$0 = 12-15 = -3$$

$$1 = 20-15 = 5$$

$$3 = 22-15 = 7$$

$$5 = 30-15 = 15$$

$$0 = 191$$

$$1 = 175$$

$$3 = 151$$

$$5 = 75$$

После селекции остаются – 3, 5, 0, 0, 0, 1

Формируем пары – (3, 0), (5, 0), (1, 0)

Скрещиваем? – (0.24 < 0.5), (0.79 < 0.5), (0.92 < 0.5)

Скрещиваем? – (True), (False), (False)

0.766
0.989
0.077
0.171
0.160
0.239
0.239
0.792
0.915
0.359
0.114
0.694
0.444
0.849
0.105
0.500
0.973
0.881
0.406
0.029
0.034
0.466
0.038
0.798
0.570
0.449
0.052
0.486
0.364
0.294
0.730
0.873
0.053
0.704
0.574
0.176

ГЛАВА 6 _ Генетический алгоритм

ІІТМО

4. Оператор скрещивания

$$f(x) = 200 - x^2 \rightarrow \max$$

$$x \in [-15, 16]$$

$$N = 6$$

$$P_c = 0.5$$

$$P_m = 0.1$$

$$2^{gens} = 32 \rightarrow gens = 5$$

00000 .. 11111

0 .. 31

```
n = 5  
random.randint(1, n-1)
```

2

Хромосома
(Генотип)

0 = 01100

1 = 10100

3 = 10110

5 = 11110

Фенотип

0 = 12-15 = -3

1 = 20-15 = 5

3 = 22-15 = 7

5 = 30-15 = 15

f(x)

0 = 191

1 = 175

3 = 151

5 = 75

После селекции остаются – 3, 5, 0, 0, 0, 1

Формируем пары – (3, 0), (5, 0), (1, 0)

0	1	1	0	0
1	0	1	1	0

0	1	1	1	0
1	0	1	0	0

0.766
0.989
0.077
0.171
0.160
0.239
0.239
0.792
0.915
0.359
0.114
0.694
0.444
0.849
0.105
0.500
0.973
0.881
0.406
0.029
0.034
0.466
0.038
0.798
0.570
0.449
0.052
0.486
0.364
0.294
0.730
0.873
0.053
0.704
0.574
0.176

ГЛАВА 6 _ Генетический алгоритм

5. Оператор мутации

$$f(x) = 200 - x^2 \rightarrow \max$$

$$x \in [-15, 16]$$

$$N = 6$$

$$P_c = 0.5$$

$$P_m = 0.1$$

$$2^{gens} = 32 \rightarrow gens = 5$$

00000 .. 11111

0 .. 31

Хромосома
(Генотип)

0 = 01100

1 = 10100

3 = 10110

5 = 11110

Фенотип

0 = 12-15 = -3

1 = 20-15 = 5

3 = 22-15 = 7

5 = 30-15 = 15

f(x)

0 = 191

1 = 175

3 = 151

5 = 75

Новая популяция

0 = 0` = 01100

1 = 0` = 01100

2 = 1` = 10100

3 = 5` = 11110

4 = ` = 01110

5 = _` = 10100

0 = False

1 = False

2 = False

3 = False

4 = False

5 = False

Никто не мутирует

0.766
0.989
0.077
0.171
0.160
0.239
0.239
0.792
0.915
0.359
0.114
0.694
0.444
0.849
0.105
0.500
0.973
0.881
0.406
0.029
0.034
0.466
0.038
0.798
0.570
0.449
0.052
0.486
0.364
0.294
0.730
0.873
0.053
0.704
0.574
0.176

ГЛАВА 6 _ Генетический алгоритм

ИТМО

Следующая итерация

$$f(x) = 200 - x^2 \rightarrow \max$$

$$x \in [-15, 16]$$

$$N = 6$$

$$P_c = 0.5$$

$$P_m = 0.1$$

$$2^{gens} = 32 \rightarrow gens = 5$$

00000 .. 11111

0 .. 31

Хромосома
(Генотип)

Фенотип

f(x)

$$0 = 01100$$

$$1 = 10100$$

$$3 = 10110$$

$$5 = 11110$$

$$0 = 12-15 = -3$$

$$1 = 20-15 = 5$$

$$3 = 22-15 = 7$$

$$5 = 30-15 = 15$$

$$0 = 191$$

$$1 = 175$$

$$3 = 151$$

$$5 = 75$$

Новая популяция

$$0 = 0' = 01100$$

$$1 = 0' = 01100$$

$$2 = 1' = 10100$$

$$3 = 5' = 11110$$

$$4 = _ = 01110$$

$$5 = _ = 10100$$

$$0 = -3$$

$$1 = -3$$

$$2 = 5$$

$$3 = 15$$

$$4 = -1$$

$$5 = 5$$

$$0 = 191$$

$$1 = 191$$

$$2 = 175$$

$$3 = 75$$

$$4 = 199$$

$$5 = 175$$

0.766
0.989
0.077
0.171
0.160
0.239
0.239
0.792
0.915
0.359
0.114
0.694
0.444
0.849
0.105
0.500
0.973
0.881
0.406
0.029
0.034
0.466
0.038
0.798
0.570
0.449
0.052
0.486
0.364
0.294
0.730
0.873
0.053
0.704
0.574
0.176

ГЛАВА 6 _ Генетический алгоритм

Критерий останова

Число поколений

Ограниченное количество итераций.

$$N^{MAX} = const$$

```
if (n > NMAX) then break
```

Качество

- Найдено удовлетворяющее решение
- Как давно качество не улучшается

```
if (f(x*) better then thresholdvalue)  
then break
```

```
if (n - indexbest) > thresholdtime  
then break
```

Сходимость

Многие агенты нашли похожее решение

$$k_best = const$$
$$err = const$$

```
if |f(xi) - f(xbest)| < err,  
i ∈ 1..kbest  
then break
```

ГЛАВА 6 _ Генетический алгоритм

Вещественные числа

1 – Знаем границы поиска, допустим $[-10, 10]$ $[a, b]$

2 – Знаем необходимую точность решения, допустим **0.01**

Длина = 20

$20 / 0.01 =$ минимум 2000 точек для кодирования

$$2^{gens} \geq 2000, gens \rightarrow \min$$

$$2^{10} = 1024, 2^{11} = 2048$$

11 генов (битов) необходимо

То есть, для обеспечения необходимой точности нужно разбить исходный отрезок на 2048 частей.

Как получить фенотип?

Этап 1. Переводим двоичное число (генотип) в десятичное число X .

Этап 2.

$$x = a + X \frac{b - a}{2^{gens} - 1}$$

ГЛАВА 6 _ Генетический алгоритм

Интерактивные примеры

https://rednuht.org/genetic_walkers/



<https://math.hws.edu/eck/jsdemo/jsGeneticAlgorithm.html>



ГЛАВА 6 _ Генетический алгоритм

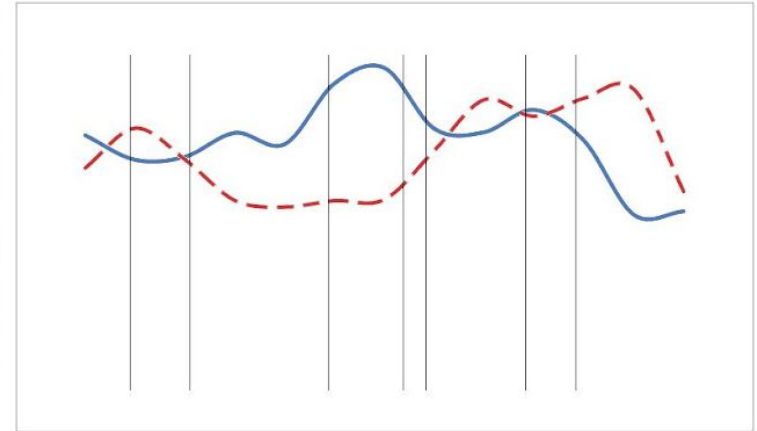
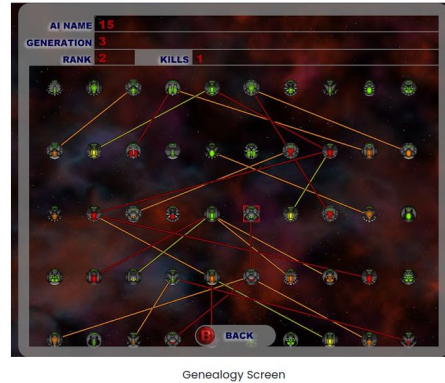
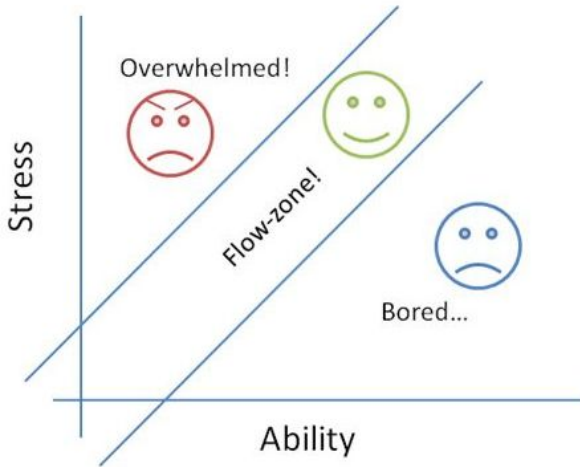
Примеры

В качестве примера рассматривается необходимость настройки параметров производительности для физического моделирования, используемого каждым автомобилем в Gran Turismo 4 в книге (ниже). При моделировании более чем пятисот транспортных средств и десятках настраиваемых параметров для каждой сложной части системы управления автомобилем это было бы действительно сложной задачей для любой компании (по крайней мере, в разумные сроки и при любом денежном бюджете), особенно если цель состояла в том, чтобы точно изобразить реальные характеристики каждого автомобиля.



ГЛАВА 6 _ Генетический алгоритм

Примеры



Changing Fitness Functions

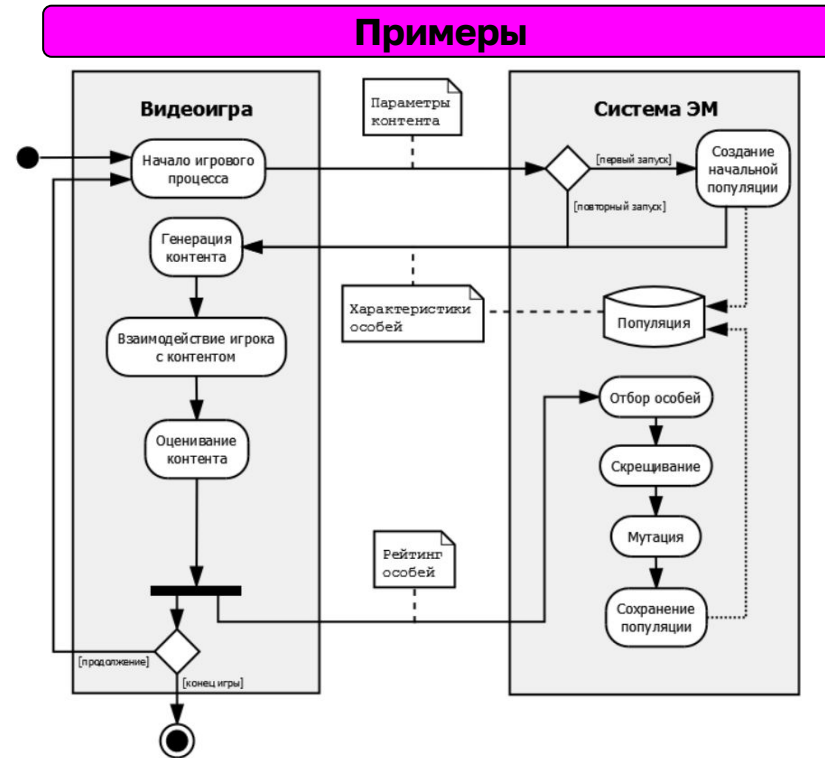


Рис. 4. Схема взаимодействия системы изменения контента с видеоигрой

ГЛАВА 6 _ Генетический алгоритм

Примеры

Hero Academy (Robot Entertainment, 2012)



StarCraft (Blizzard Entertainment, 1998)



AI and Games, 2018, p.114

ГЛАВА 6 _ Генетический алгоритм

Примеры

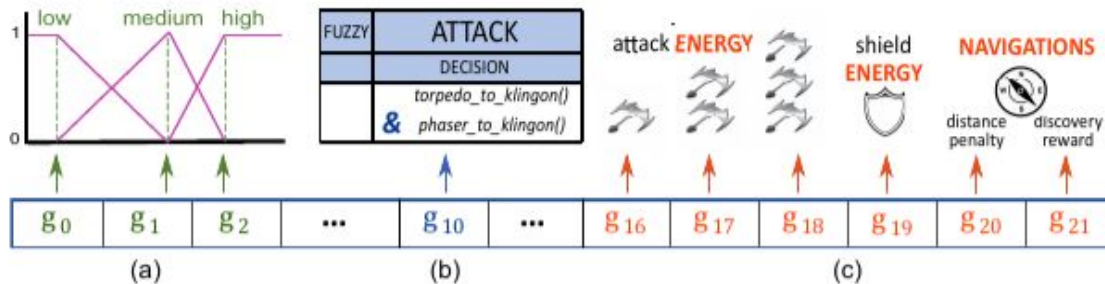


FIGURE 5. Parameter coding in DE population individuals. We implemented our DE optimizer to search for (a) optimal triangular shapes for FS membership functions, (b) optimal decisions for the multiple-output FS rules, and (c) optimal preset values for command parameters.

$$\begin{aligned}
 \text{fitness score} &= \text{average game scores} + \text{win ratio} \\
 \text{game score} &= 0.60 * \# \text{destroyed_enemies} \\
 &\quad + 0.30 * \# \text{found_enemies} \\
 &\quad + 0.10 * (\text{win_game} ? \# \text{remaining_time} : 0.0) \\
 \text{win ratio} &= \frac{\# \text{win_games}}{\# \text{total_games}}
 \end{aligned}$$

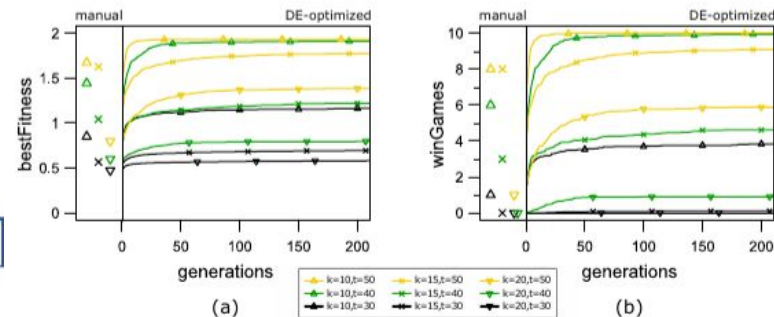


FIGURE 6. (a) The best fitness score and (b) the corresponding maximum numbers of games won by manual tunings (points graph) and DE-optimized tunings (line graph). We conducted experiments with nine game parameter setups, which were obtained by varying the numbers of Klingons (k) and game time (t).

EVOLVING FUZZY LOGIC RULE-BASED GAME PLAYER MODEL FOR GAME DEVELOPMENT

VARUNYU VORACHART¹ AND HIDEYUKI TAKAGI²

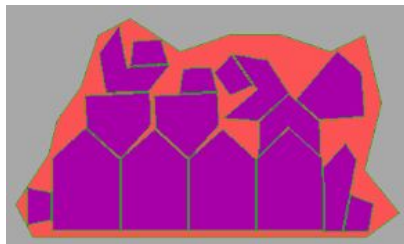
¹Graduate School of Design
Kyushu University
Fukuoka, Japan 815-8540
varunyu@kyudai.jp

²Faculty of Design
Kyushu University
Fukuoka, Japan 815-8540
<http://www.design.kyushu-u.ac.jp/~takagi/>

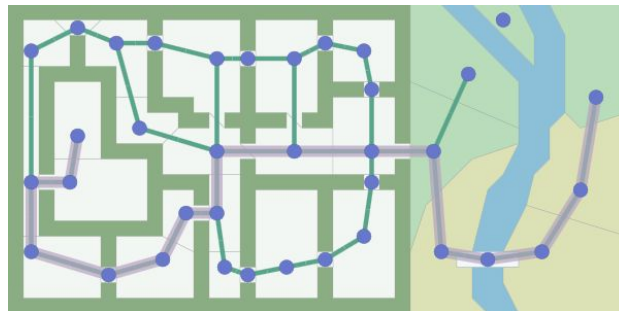
ГЛАВА 6 _ Комбинаторная оптимизация ИТМО



Задача Коммивояжера



Задача раскроя



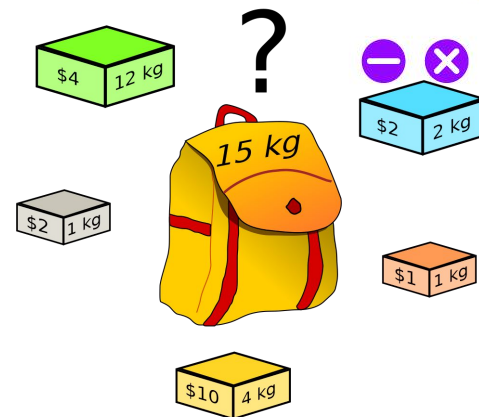
Задача о кратчайшем пути

Расписание уроков

Расписание занятий	Понедельник	Вторник	Среда
1.	1.	1.	1.
2.	2.	2.	2.
3.	3.	3.	3.
4.	4.	4.	4.
5.	5.	5.	5.
6.	6.	6.	6.
7.	7.	7.	7.

Занятия	Четверг	Пятница	Суббота
1.	1.	1.	1.
2.	2.	2.	2.
3.	3.	3.	3.
4.	4.	4.	4.
5.	5.	5.	5.
6.	6.	6.	6.
7.	7.	7.	7.

Сделано в ИТМО - Фабрика для школы



Задача о рюкзаке

Задачи теории расписаний

ГЛАВА 6 _ Комбинаторная оптимизация

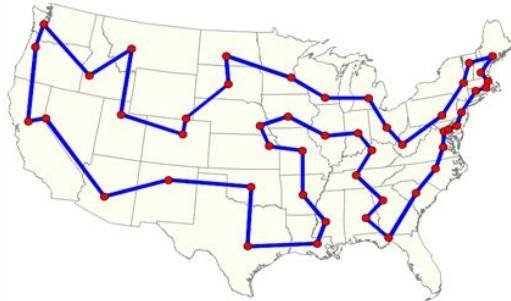


Например, Анно 1404



Задача Коммивояжера

ГЛАВА 6 _ Комбинаторная оптимизация



Генетический алгоритм

Кодирование

1

4

2

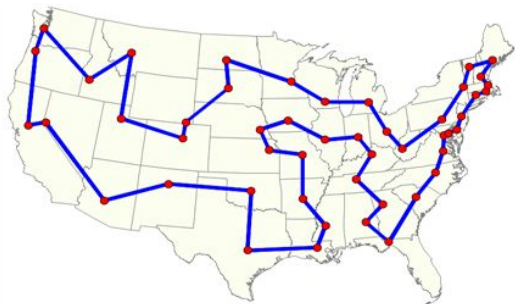
3

5



Задача Коммивояжера

ГЛАВА 6 _ Комбинаторная оптимизация



Генетический алгоритм

Кодирование

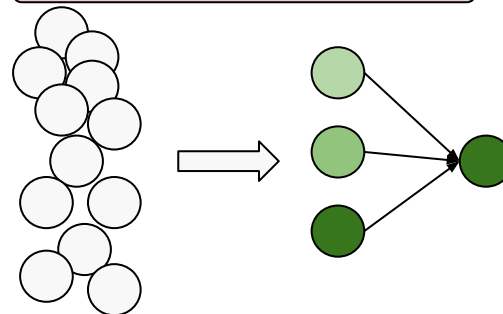
1 4 2 3 5

Селекция

Колесо Рулетки

Стохастическая
универсальная
выборка

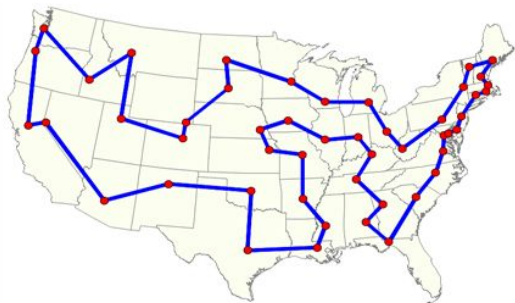
Турнир



Ранговое
преобразование

При близких $f(x)$ используем ранг

Задача Коммивояжера



Генетический алгоритм

Кодирование

1	4	2	3	5
---	---	---	---	---

Селекция

Колесо Рулетки

Турнир

Стох. УВ

Ранг

Скращивание

1	4	2	3	5
---	---	---	---	---

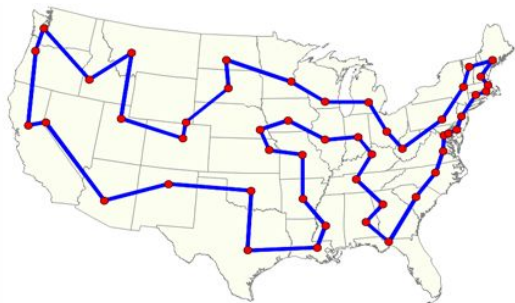
2	5	1	3	4
---	---	---	---	---

1	4	3	2	5
---	---	---	---	---

2	5	3	1	4
---	---	---	---	---

Задача Коммивояжера

ГЛАВА 6 _ Комбинаторная оптимизация



Генетический алгоритм

Кодирование

1 4 2 3 5

Селекция

Колесо Рулетки

Турнир

Стох. УВ

Ранг

Скращивание

1	4	2	3	5
2	5	1	3	4

Мутация

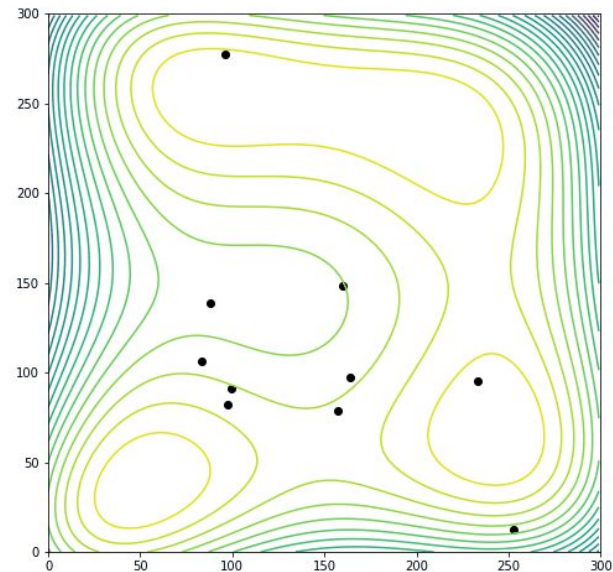
2	5	3	1	4
---	---	---	---	---

1	4	3	2	5
2	5	3	1	4

2	5	4	1	3
---	---	---	---	---

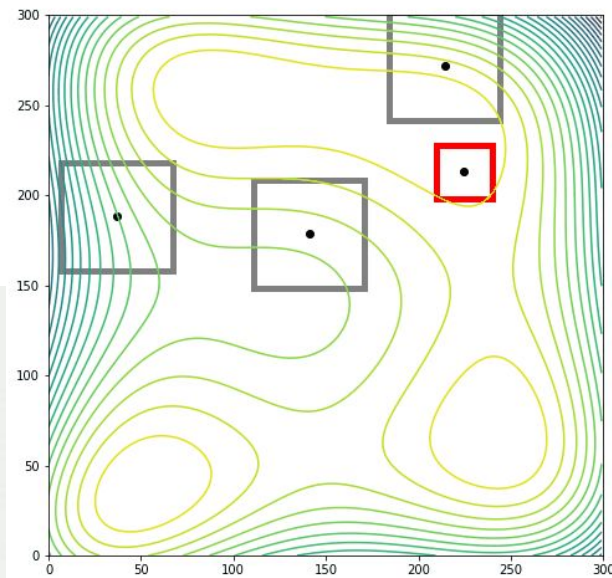
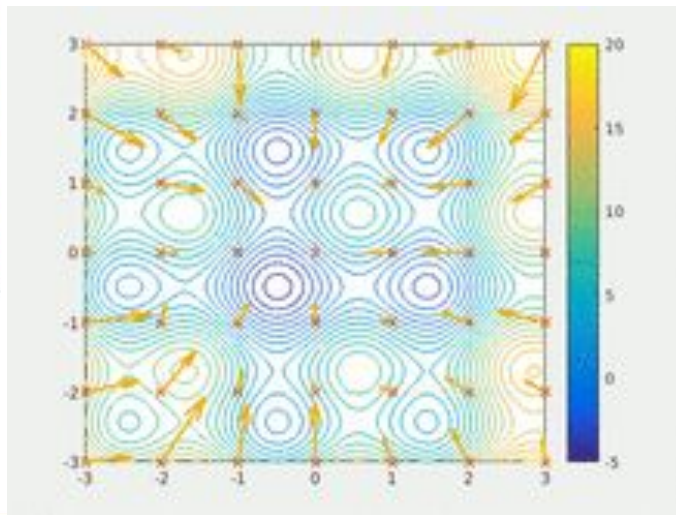
Задача Коммивояжера

ГЛАВА 6 _ Роевые алгоритмы



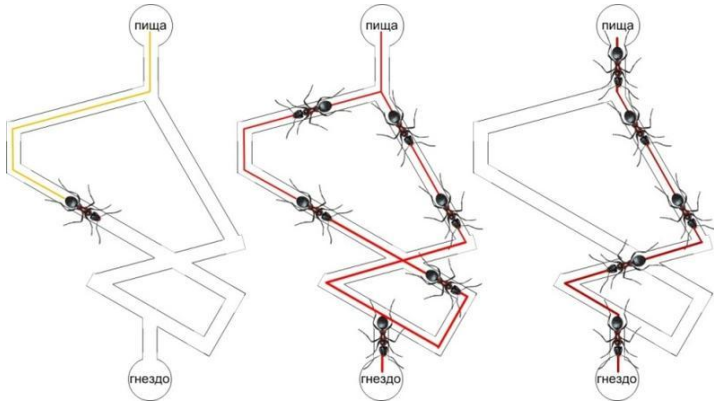
Сорняковый алгоритм

Алгоритм роя частиц

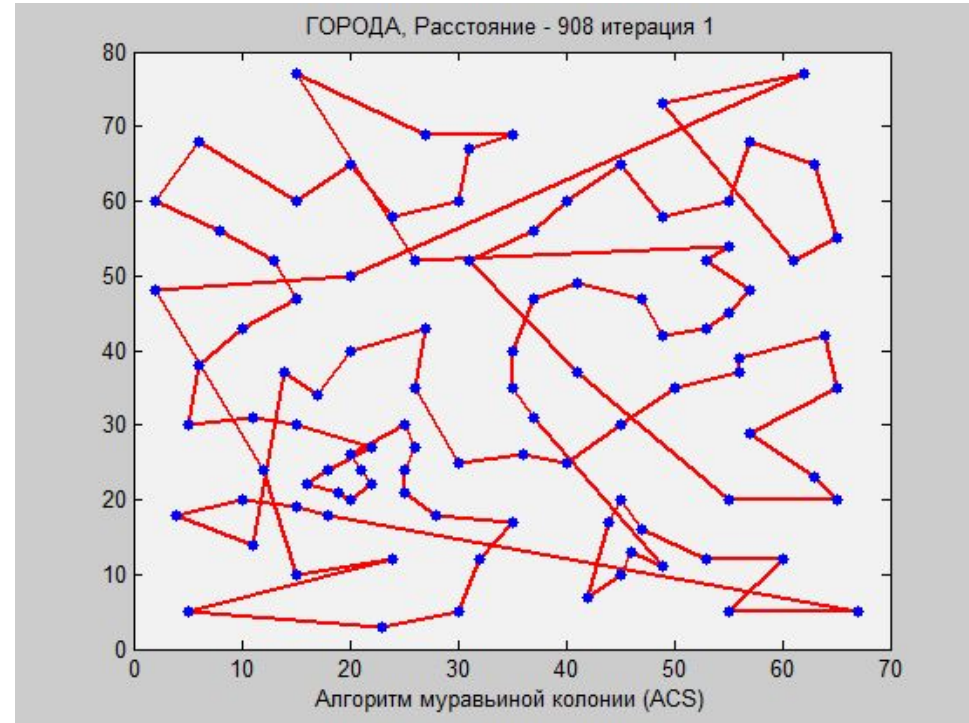


Пчелиный алгоритм

ГЛАВА 6 _ Муравьиный алгоритм

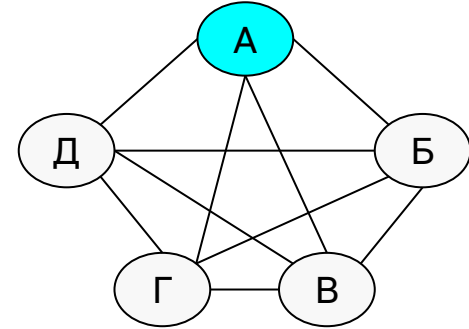


- выделяя феромоны во время перемещения, муравьи изменяют окружающую среду, обеспечивают коммуникацию, а также отыскивают обратный путь в муравейник
- чем больше муравьев используют один и тот же путь, тем выше концентрация феромона на этом пути. Чем ближе внешняя точка к муравейнику, тем чаще к ней перемещались муравьи.



ГЛАВА 6 _ Муравьиный алгоритм

Этап 1. Выбор начальной вершины



ГЛАВА 6 _ Муравьиный алгоритм

Этап 1. Выбор начальной вершины

Этап 2. Распространение муравьев по графу

$$P = \frac{\tau(r,u)^\alpha \times \eta(r,u)^\beta}{\sum_k \tau(r,u)^\alpha \times \eta(r,u)^\beta}.$$

- $\tau(r,u)$ – интенсивность феромона на грани между узлами r и u
 - $\eta(r,u)$ – функция, которая представляет измерение обратного расстояния для грани
 - α – вес феромона
 - β – коэффициент эвристики.
- Параметры α и β определяют относительную значимость двух параметров, а также их влияние на уравнение.

Как с колесом в генетическом алгоритме!

ГЛАВА 6 _ Муравьиный алгоритм

Этап 1. Выбор начальной вершины

Этап 2. Распространение муравьев по графу

Этап 3. Обновление феромона в графе

$$P = \frac{\tau(r,u)^\alpha \times \eta(r,u)^\beta}{\sum_k \tau(r,u)^\alpha \times \eta(r,u)^\beta}.$$

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

Переменная Q является константой.

$$\tau_{ij}(t) = \Delta\tau_{ij}(t) + (\tau_{ij}^k(t) \times \rho).$$

Константа ρ – значение между 0 и 1
Для испарения феромона

ГЛАВА 6 _ Муравьиный алгоритм

Этап 1. Выбор начальной вершины

Этап 2. Распространение муравьев по графу

Этап 3. Обновление феромона в графе

Этап 4. Этапы 1-3 пока критерий позволяет

$$P = \frac{\tau(r,u)^\alpha \times \eta(r,u)^\beta}{\sum_k \tau(r,u)^\alpha \times \eta(r,u)^\beta}.$$

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

$$\tau_{ij}(t) = \Delta\tau_{ij}(t) + (\tau_{ij}^k(t) \times \rho).$$

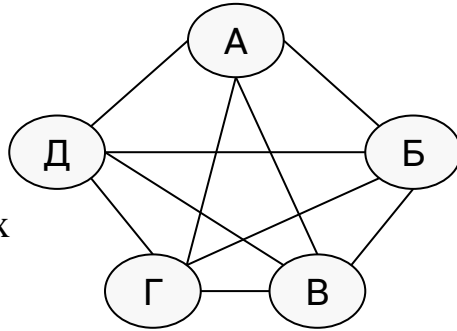
ГЛАВА 6 _ Муравьиный алгоритм

$$\alpha = \beta = 1$$

$$\rho = 0.7$$

$$Q = \text{mean()} * \text{vertex}$$

$$Q = 5.2 * 5 = 26$$



d	А	Б	В	Г	Д	f	А	Б	В	Г	Д
А	-	3	7	1	10	А	-	1	1	1	1
Б	3	-	2	2	8	Б	1	-	1	1	1
В	7	2	-	6	9	В	1	1	-	1	1
Г	1	2	6	-	4	Г	1	1	1	-	1
Д	10	8	9	4	-	Д	1	1	1	1	-

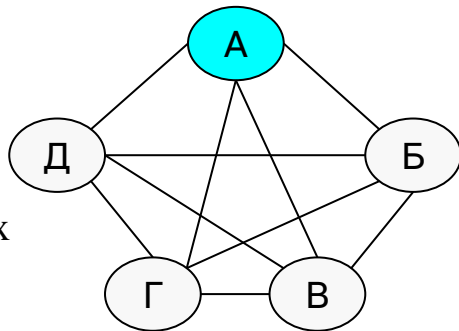
ГЛАВА 6 _ Муравьиный алгоритм

$$\alpha = \beta = 1$$

$$\rho = 0.7$$

$$Q = \text{mean()} * \text{vertex}$$

$$Q = 5.2 * 5 = 26$$



Этап 1. Выбор начальной вершины

random(1-5) = 1

d	А	Б	В	Г	Д	f	А	Б	В	Г	Д
А	-	3	7	1	10	А	-	1	1	1	1
Б	3	-	2	2	8	Б	1	-	1	1	1
В	7	2	-	6	9	В	1	1	-	1	1
Г	1	2	6	-	4	Г	1	1	1	-	1
Д	10	8	9	4	-	Д	1	1	1	1	-

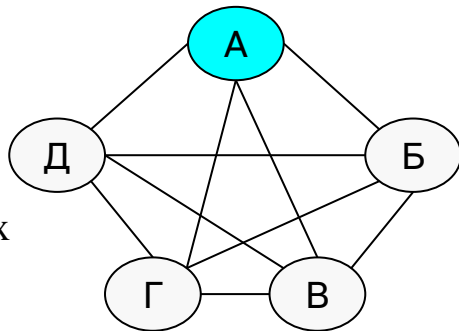
ГЛАВА 6 _ Муравьиный алгоритм

$$\alpha = \beta = 1$$

$$\rho = 0.7$$

$$Q = \text{mean()} * \text{vertex}$$

$$Q = 5.2 * 5 = 26$$



Этап 2. Распространение муравьев по графу

А-Б,
А-В,
А-Г,
А-Д

$$P = \frac{\tau(r,u)^\alpha \times \eta(r,u)^\beta}{\sum_k \tau(r,u)^\alpha \times \eta(r,u)^\beta}.$$

d	А	Б	В	Г	Д	f	А	Б	В	Г	Д
А	-	3	7	1	10	А	-	1	1	1	1
Б	3	-	2	2	8	Б	1	-	1	1	1
В	7	2	-	6	9	В	1	1	-	1	1
Г	1	2	6	-	4	Г	1	1	1	-	1
Д	10	8	9	4	-	Д	1	1	1	1	-

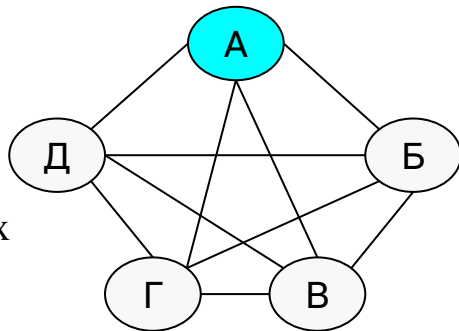
ГЛАВА 6 _ Муравьиный алгоритм

$$\alpha = \beta = 1$$

$$\rho = 0.7$$

Q = mean()*vertex

$$Q = 5.2 * 5 = 26$$



Этап 2. Распространение муравьев по графу

А-Б,
А-В,
А-Г,
А-Д

$$P = \frac{\tau(r, u)^\alpha \times \eta(r, u)^\beta}{\sum_k \tau(r, u)^\alpha \times \eta(r, u)^\beta}$$

$$p(A, Б) = \frac{1^1 \cdot \frac{1}{3}^1}{1^1 \cdot \frac{1}{3}^1 + 1^1 \cdot \frac{1}{7}^1 + 1^1 \cdot \frac{1}{1}^1 + 1^1 \cdot \frac{1}{10}^1} = \frac{0.33}{1.58} = 0.21$$

$$p(A, В) = \frac{1^1 \cdot \frac{1}{7}^1}{1^1 \cdot \frac{1}{3}^1 + 1^1 \cdot \frac{1}{7}^1 + 1^1 \cdot \frac{1}{1}^1 + 1^1 \cdot \frac{1}{10}^1} = \frac{0.14}{1.58} = 0.09$$

$$p(A, Г) = \frac{1^1 \cdot \frac{1}{1}^1}{1^1 \cdot \frac{1}{3}^1 + 1^1 \cdot \frac{1}{7}^1 + 1^1 \cdot \frac{1}{1}^1 + 1^1 \cdot \frac{1}{10}^1} = \frac{1}{1.58} = 0.63$$

$$p(A, Д) = \frac{1^1 \cdot \frac{1}{10}^1}{1^1 \cdot \frac{1}{3}^1 + 1^1 \cdot \frac{1}{7}^1 + 1^1 \cdot \frac{1}{1}^1 + 1^1 \cdot \frac{1}{10}^1} = \frac{0.1}{1.58} = 0.07$$

d	А	Б	В	Г	Д	f	А	Б	В	Г	Д
А	-	3	7	1	10	А	-	1	1	1	1
Б	3	-	2	2	8	Б	1	-	1	1	1
В	7	2	-	6	9	В	1	1	-	1	1
Г	1	2	6	-	4	Г	1	1	1	-	1
Д	10	8	9	4	-	Д	1	1	1	1	-

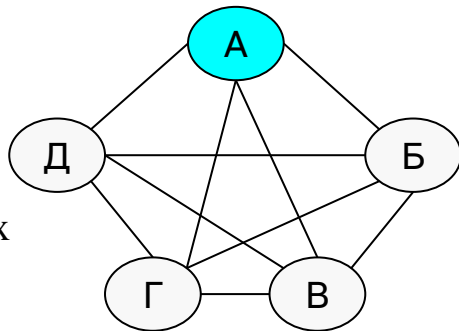
ГЛАВА 6 _ Муравьиный алгоритм

$$\alpha = \beta = 1$$

$$\rho = 0.7$$

$Q = \text{mean()} * \text{vertex}$

$$Q = 5.2 * 5 = 26$$



Этап 2. Распространение муравьев по графу

А-Б,
А-В,
А-Г,
А-Д

$$P = \frac{\tau(r,u)^\alpha \times \eta(r,u)^\beta}{\sum_k \tau(r,u)^\alpha \times \eta(r,u)^\beta}$$



d	А	Б	В	Г	Д	f	А	Б	В	Г	Д
А	-	3	7	1	10	А	-	1	1	1	1
Б	3	-	2	2	8	Б	1	-	1	1	1
В	7	2	-	6	9	В	1	1	-	1	1
Г	1	2	6	-	4	Г	1	1	1	-	1
Д	10	8	9	4	-	Д	1	1	1	1	-

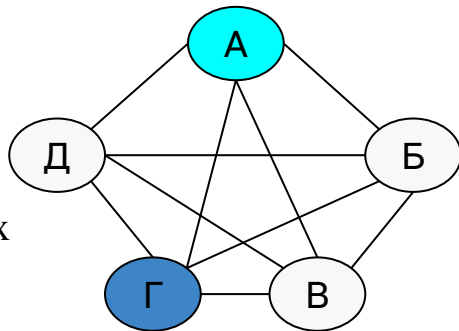
ГЛАВА 6 _ Муравьиный алгоритм

$$\alpha = \beta = 1$$

$$\rho = 0.7$$

$$Q = \text{mean()} * \text{vertex}$$

$$Q = 5.2 * 5 = 26$$



Этап 2. Распространение муравьев по графу

Г-Б
Г-В
Г-Д

$$P = \frac{\tau(r, u)^\alpha \times \eta(r, u)^\beta}{\sum_k \tau(r, u)^\alpha \times \eta(r, u)^\beta}$$

$$1 \cdot \frac{1}{2} + 1 \cdot \frac{1}{6} + 1 \cdot \frac{1}{4} = 0.92$$

d	А	Б	В	Г	Д	f	А	Б	В	Г	Д
А	-	3	7	1	10	А	-	1	1	1	1
Б	3	-	2	2	8	Б	1	-	1	1	1
В	7	2	-	6	9	В	1	1	-	1	1
Г	1	2	6	-	4	Г	1	1	1	-	1
Д	10	8	9	4	-	Д	1	1	1	1	-

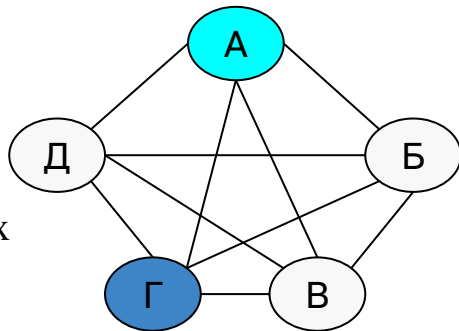
ГЛАВА 6 _ Муравьиный алгоритм

$$\alpha = \beta = 1$$

$$\rho = 0.7$$

$$Q = \text{mean()} * \text{vertex}$$

$$Q = 5.2 * 5 = 26$$



Этап 2. Распространение муравьев по графу

Г-Б
Г-В
Г-Д

$$P = \frac{\tau(r, u)^\alpha \times \eta(r, u)^\beta}{\sum_k \tau(r, u)^\alpha \times \eta(r, u)^\beta}$$

$$1 \cdot \frac{1}{2} + 1 \cdot \frac{1}{6} + 1 \cdot \frac{1}{4} = 0.92$$

$$p(\Gamma, \text{Б}) = \frac{1 \cdot \frac{1}{2}}{0.92} = 0.54$$

$$p(\Gamma, \text{В}) = \frac{1 \cdot \frac{1}{6}}{0.92} = 0.18$$

$$p(\Gamma, \text{Д}) = \frac{1 \cdot \frac{1}{4}}{0.92} = 0.28$$

d	А	Б	В	Г	Д	f	А	Б	В	Г	Д
А	-	3	7	1	10	А	-	1	1	1	1
Б	3	-	2	2	8	Б	1	-	1	1	1
В	7	2	-	6	9	В	1	1	-	1	1
Г	1	2	6	-	4	Г	1	1	1	-	1
Д	10	8	9	4	-	Д	1	1	1	1	-



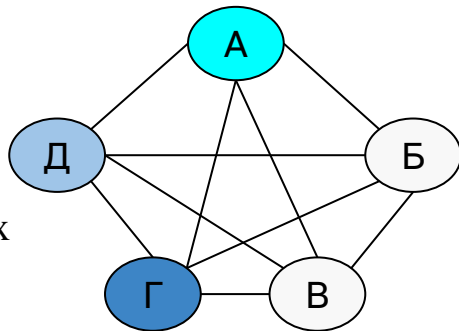
ГЛАВА 6 _ Муравьиный алгоритм

$$\alpha = \beta = 1$$

$$\rho = 0.7$$

$$Q = \text{mean()} * \text{vertex}$$

$$Q = 5.2 * 5 = 26$$



Этап 2. Распространение муравьев по графу

Д-Б
Д-В

$$P = \frac{\tau(r,u)^\alpha \times \eta(r,u)^\beta}{\sum_k \tau(r,u)^\alpha \times \eta(r,u)^\beta}$$

$$p(Д, Б) = \frac{0.13}{0.24} = 0.54$$

$$p(Д, В) = \frac{0.11}{0.24} = 0.46$$

d	А	Б	В	Г	Д	f	А	Б	В	Г	Д
А	-	3	7	1	10	А	-	1	1	1	1
Б	3	-	2	2	8	Б	1	-	1	1	1
В	7	2	-	6	9	В	1	1	-	1	1
Г	1	2	6	-	4	Г	1	1	1	-	1
Д	10	8	9	4	-	Д	1	1	1	1	-



Задача Коммивояжера

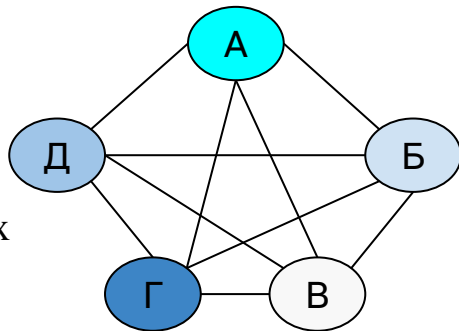
ГЛАВА 6 _ Муравьиный алгоритм

$$\alpha = \beta = 1$$

$$\rho = 0.7$$

$$Q = \text{mean()} * \text{vertex}$$

$$Q = 5.2 * 5 = 26$$



Этап 2. Распространение муравьев по графу

$$P = \frac{\tau(r,u)^\alpha \times \eta(r,u)^\beta}{\sum_k \tau(r,u)^\alpha \times \eta(r,u)^\beta}$$

d	А	Б	В	Г	Д	f	А	Б	В	Г	Д
А	-	3	7	1	10	А	-	1	1	1	1
Б	3	-	2	2	8	Б	1	-	1	1	1
В	7	2	-	6	9	В	1	1	-	1	1
Г	1	2	6	-	4	Г	1	1	1	-	1
Д	10	8	9	4	-	Д	1	1	1	1	-

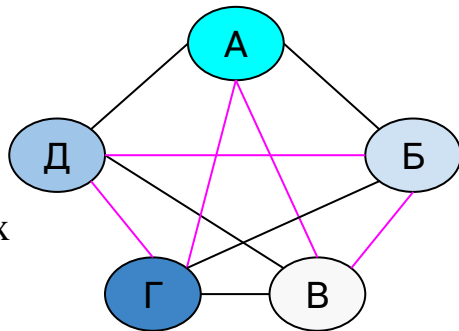
ГЛАВА 6 _ Муравьиный алгоритм

$$\alpha = \beta = 1$$

$$\rho = 0.7$$

$$Q = \text{mean()} * \text{vertex}$$

$$Q = 5.2 * 5 = 26$$



Этап 2. Распространение муравьев по графу

$$P = \frac{\tau(r,u)^\alpha \times \eta(r,u)^\beta}{\sum_k \tau(r,u)^\alpha \times \eta(r,u)^\beta}$$

$$L = 1 + 4 + 8 + 7 = 20$$

d	А	Б	В	Г	Д	f	А	Б	В	Г	Д
А	-	3	7	1	10	А	-	1	1	1	1
Б	3	-	2	2	8	Б	1	-	1	1	1
В	7	2	-	6	9	В	1	1	-	1	1
Г	1	2	6	-	4	Г	1	1	1	-	1
Д	10	8	9	4	-	Д	1	1	1	1	-

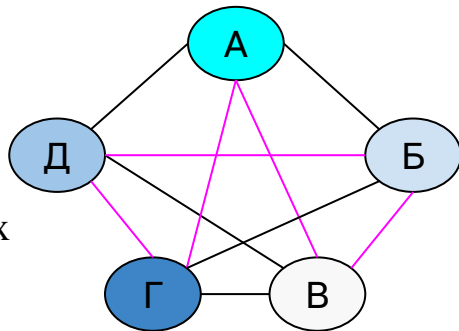
ГЛАВА 6 _ Муравьиный алгоритм

$$\alpha = \beta = 1$$

$$\rho = 0.7$$

$Q = \text{mean()} * \text{vertex}$

$$Q = 5.2 * 5 = 26$$



Этап 3. Обновление феромона в графе

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

$$\tau_{ij}(t) = \Delta \tau_{ij}(t) + (\tau_{ij}^k(t) \times \rho)$$

$$L = 1 + 4 + 8 + 7 = 20$$

d	А	Б	В	Г	Д	f	А	Б	В	Г	Д
А	-	3	7	1	10	А	-	.8	2.1	2.1	.8
Б	3	-	2	2	8	Б	.8	-	.8	.8	2.1
В	7	2	-	6	9	В	2.1	.8	-	.8	.8
Г	1	2	6	-	4	Г	2.1	.8	.8	-	2.1
Д	10	8	9	4	-	Д	.8	2.1	.8	2.1	-

$$(26/20) + (1 * 0.8) = 2.1$$

$$(1 * 0.8) = 0.8$$

Этап 4. Этапы 1-3 пока критерий позволяет

ГЛАВА 6 _ Муравьиный алгоритм

Модификации

Отправляем n муравьев, феромон обновляется у **лучшего** решения

Элитарная муравьиная система с
к элитными муравьями

Max-Min муравьиная система

Ранговая муравьиная система

ГЛАВА 6 _ Муравьиный алгоритм

Примеры

<http://poolik.github.io/visual-aco/#/visualisation>

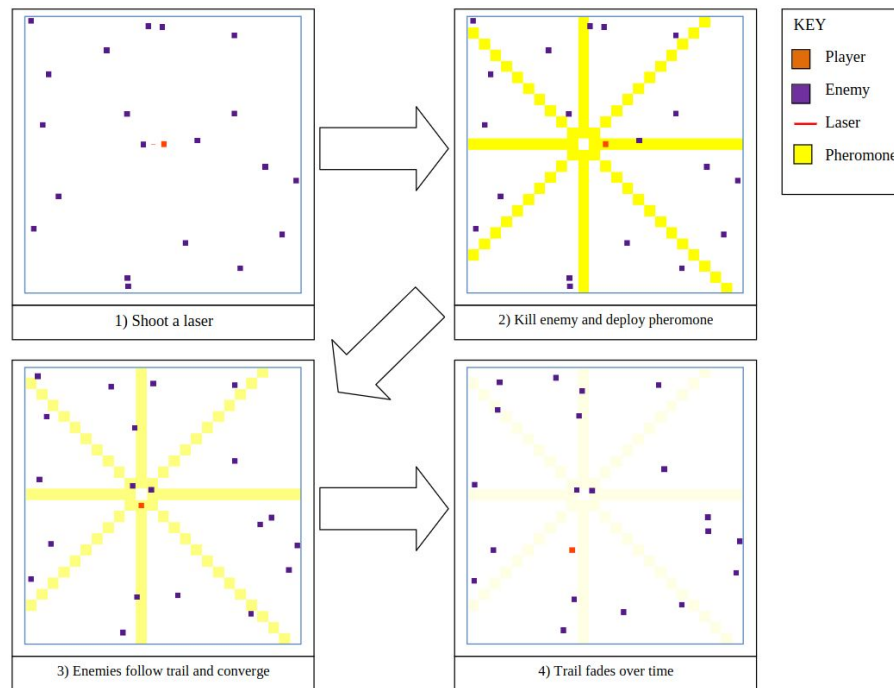


ГЛАВА 6 _ Муравьиный алгоритм

Примеры

	Kills	Deaths	K:D
Long Trails	4000	1907	2.10
Short Trails	3103	1402	2.21
No Trails	1180	587	2.01

Table 1. Test Group Totals



ГЛАВА 6 _ Муравьиный алгоритм

Примеры

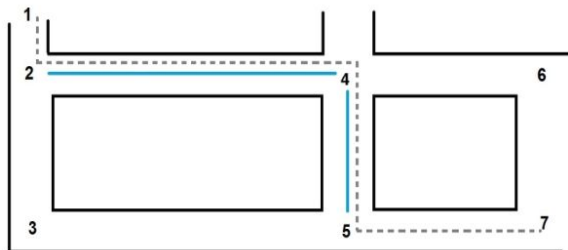
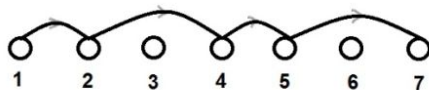


Fig. 1. A sample game environment



$tour_string = [1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1]$

Fig. 2. The corresponding graph for the sample game environment

TABLE I. MARKS'S FUZZY RULE-BASE FOR SELECTING THE NEXT ROAD

	pheromone value is...				
	<i>very high</i>	<i>high</i>	<i>medium</i>	<i>low</i>	<i>very low</i>
line exists	very high	high	high-	medium	low
line does not exist	very high	high	low	low-	very low

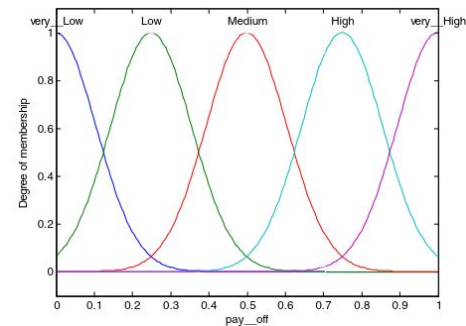


Fig. 3. Membership functions for input pheromones

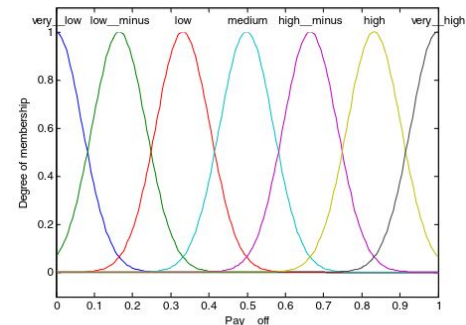
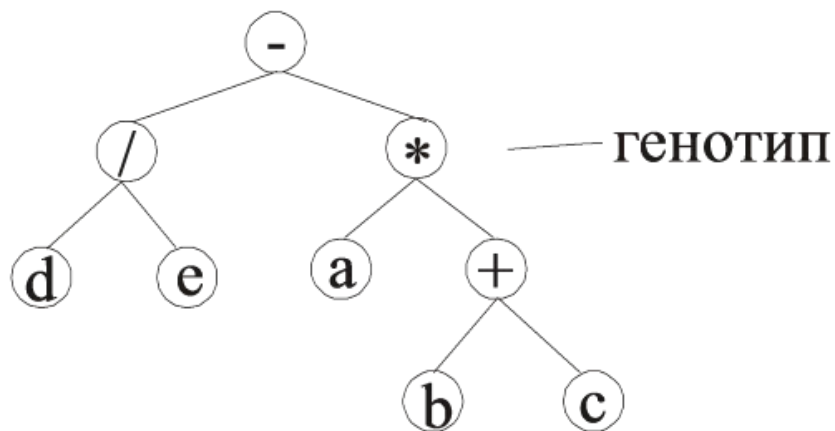


Fig. 4. Membership functions output pay-offs

В **генетическом программировании** (ГП) в качестве особи выступает программа, представленная в определенном формате, которая решает некоторую задачу. Часто это выполняется с использованием обучающих данных и индуктивного вывода. ГП очень близко к машинному обучению и поэтому в качестве фитнес-функции достаточно часто выступают функции ошибки (рассогласования, невязки в различных метриках). Следует отметить, что ГП работает с генетическим материалом переменной длины, что требует нестандартной формы представления генома и соответствующих генетических операторов



Функциональное множество - Операторы и функции

Терминальное множество - переменные и константы

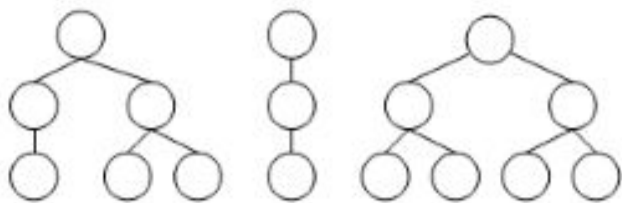
Вопрос: что в деревьях поведения будет считаться терминальным множеством, а что функциональным?

Инициализация начальной популяции

Одним из важнейших параметров в ГП является максимально возможный **размер** (сложность) программы.

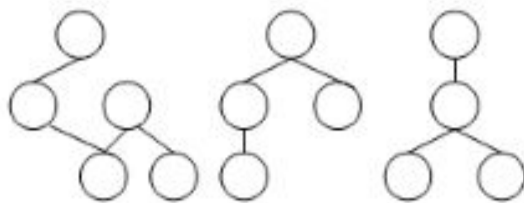
Максимальная глубина

Количество вершин



а)

Полная
инициализация



б)

Растущая
инициализация

На практике часто эти два метода используют одновременно следующим образом. Начальная популяция генерируется так, чтобы в нее входили деревья с разной максимальной длиной примерно поровну (для нашего примера 1-2-3-4) Для каждой **глубины** **первая** половина деревьев генерируется полным методом, а **вторая** – растущей инициализацией.

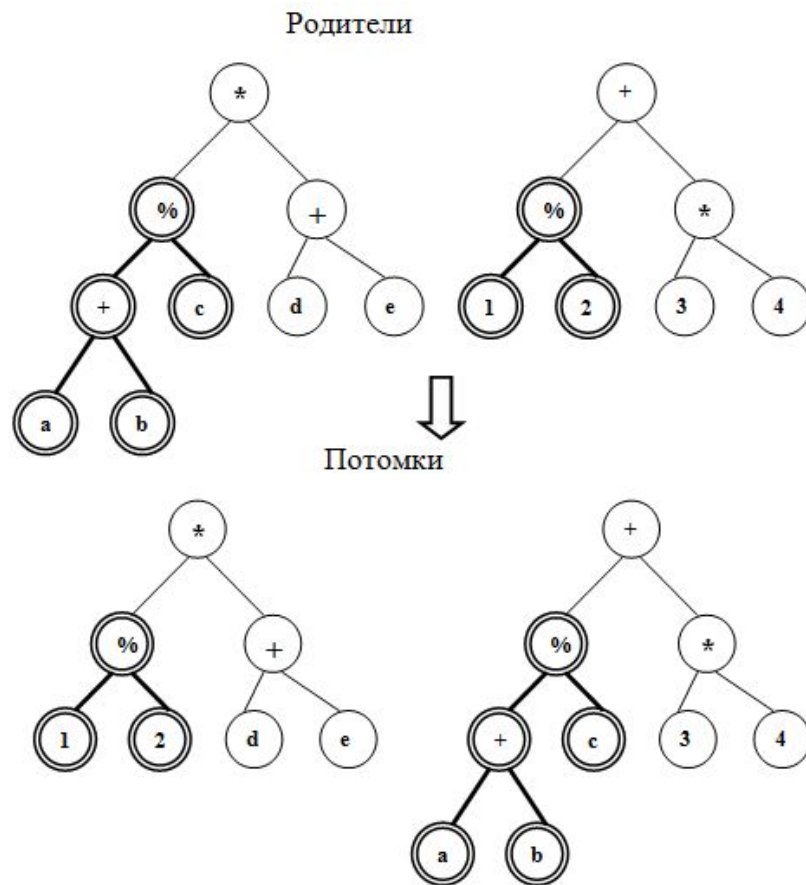
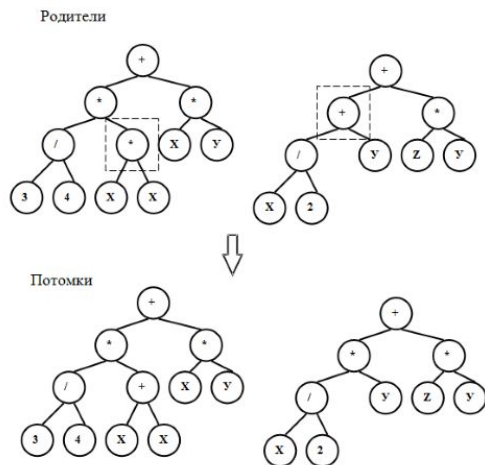
ГЛАВА 6 _ Генетическое программирование

Оператор скрещивания

Узловой кроссинговер

Кроссинговер поддеревьев

Смешанный



ГЛАВА 6 _ Генетическое программирование

Оператор мутации

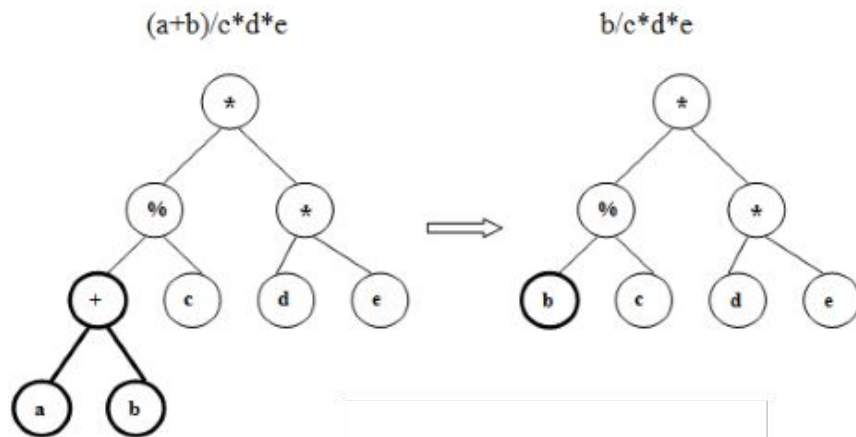
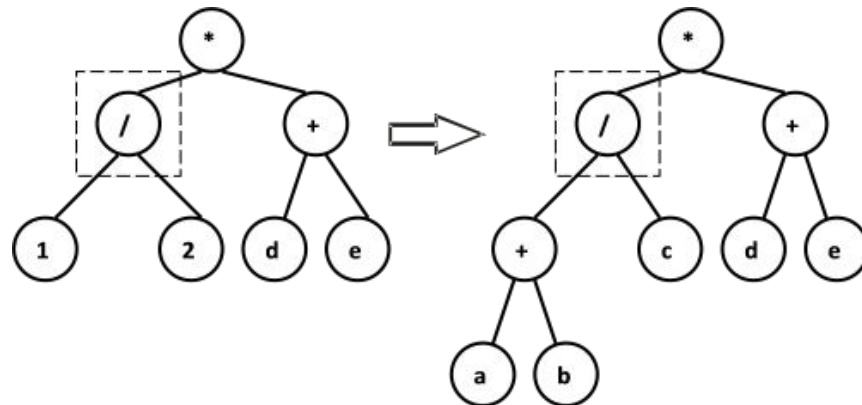
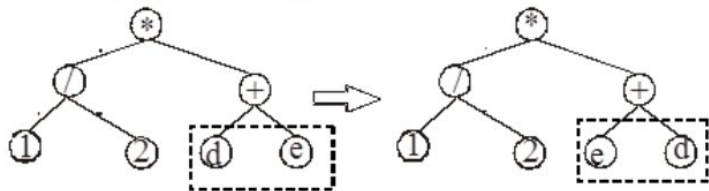
Узловая мутация

Растущая мутация

Усекающая мутация

Таблица 6.6.

Наименование	Описание производимых действий
Точечная мутация	Случайное изменение типа одного узла из того же класса
Перестановка	Перестановка аргументов одного узла
"Подъем"	Случайная генерация новой особи из поддерева
Растущая мутация	Замена терминального символа случайным поддеревом
Секущая мутация	Замена поддерева случайным терминальным символом
Мутация поддерева	Замена поддерева случайным поддеревом



Таким образом, для решения задачи с помощью ГП необходимо выполнить описанные выше предварительные этапы:

1. Определить терминальное множество;
2. Определить функциональное множество;
3. Определить фитнес-функцию;
4. Определить значения параметров, такие как мощность популяции, максимальный размер особи, вероятности кроссинговера и мутации, способ отбора родителей, критерий окончания эволюции (например, максимальное число поколений) и т.п

После этого можно разрабатывать непосредственно сам эволюционный алгоритм, реализующий ГП для конкретной задачи. Как и в случае ГА здесь также возможны различные подходы

ГЛАВА 6 _ Генетическое программирование

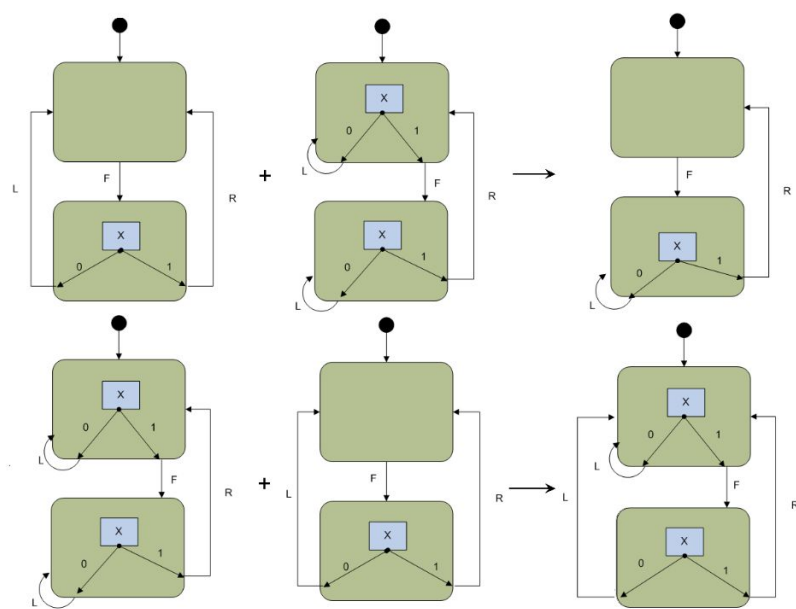


Рис. 16. Порождение потомства

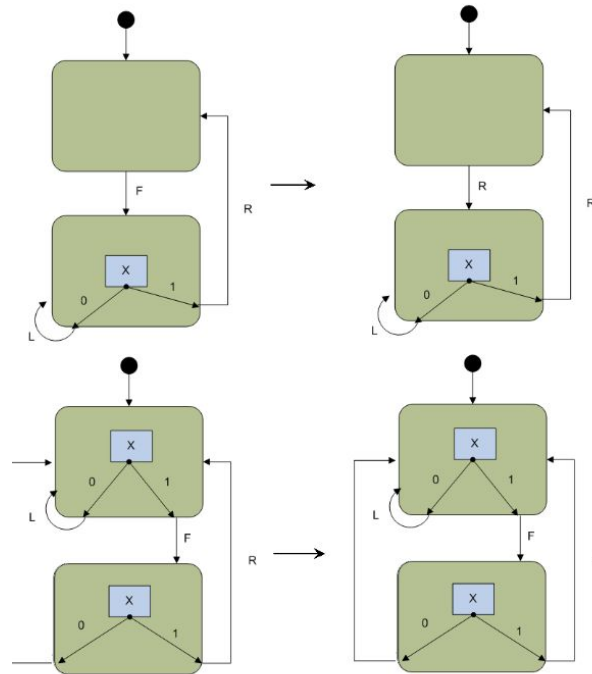


Рис. 17. Мутация потомства

Суть:
построение
автоматов
поведения,
представляя их
как дерево
решений с
помощью ГП.

ГЛАВА 6 _ Генетическое программирование

Результаты

В качестве основных соперников использовались следующие танки из стандартного набора игры «Robocode»: `sample.Fire`; `sample.Tracker`; `sample.Target`; `sample.Walls`.

На рис. 3 представлен граф переходов автомата, построенного с помощью генетического алгоритма.

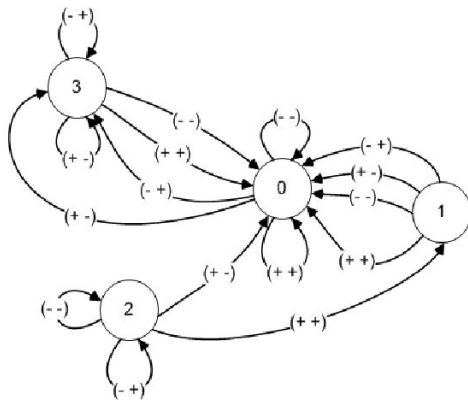


Рис. 3. Граф переходов автомата, полученного в результате работы второго этапа генетического алгоритма

Rank	Robot Name	Total Score	Survival	Surv Bonus	Bullet Dmg	Bullet Bonus	Ram Dmg	* 2 Ram Bonus	1sts	2nds	3rds
1st	core.tank.GVarTa...	4178 (70%)	3050	960	0	0	168	0	96	4	0
2nd	sample.Tracker	1822 (30%)	200	40	1500	35	48	0	4	96	0

Рис. 4. Результаты соревнования

ПРИМЕНЕНИЕ ДВУХЭТАПНОГО
ГЕНЕТИЧЕСКОГО
ПРОГРАММИРОВАНИЯ ДЛЯ
ПОСТРОЕНИЯ МОДЕЛИ ТАНКА
В ИГРЕ «ROBocode»
Д.О. Соколов, ИТМО, 2011

ПР4. Эволюционные и роевые алгоритмы (7 баллов)



Вариант 1. Закрепление материала.

Задание 1. Генетический алгоритм. Исследовать задачу (на выбор) и результаты генетического алгоритма на сервисе <https://geneticalgorithms.online/> . Ожидается таблица с не менее, чем 10 экспериментами их результатами и указанными параметрами (1 балл).

Задание 2. Муравьиный алгоритм. Произвести 2 итерации муравьиного алгоритма для задачи Коммивояжера с 6 городами. Матрицу расстояний и инициализацию матрицы феромона придумать или случайно сгенерировать (2 балла).

Задание 3. Алгоритмы, вдохновленные природой. Выберите любой интересующий вас природный алгоритм, изучите его, опишите: а) биологические предпосылки; б) этапы алгоритма; в) пример одной итерации.

Рекомендуемая литература: Карпенко “Алгоритмы, вдохновленные природой” (4 балла).

Вариант 2. Для тру прогеров.

Реализовать **генетическое программирование** на языке высокого уровня для построения оптимальной вычислительной программы. Пример заданий можно посмотреть тут: <https://intuit.ru/studies/courses/14227/1284/lecture/24178?page=12> **или** применить к созданию деревьев поведения в конкретной ситуации для интеллектуального агента.

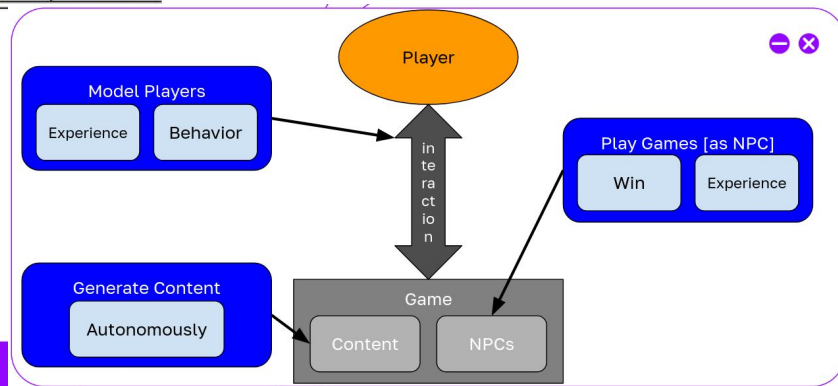
ГЛАВА 6 _ Итоги курса

6.1. Panoramic Views of Game AI

261

Table 6.1 Dominant (●) and secondary (○) AI methods for each of the core AI areas we cover in this book. The total number of methods used for each area appears at the bottom row of the table.

	Play Games		Generate Content		Model Players	
	Winning	Experience	Autonomously	Assisted	Experience	Behavior
Behavior Authoring	●	●				
Tree Search	●	○	○	○		
Evolutionary Computation	●	○	●	●	●	
Supervised Learning	○	●			●	●
Reinforcement Learning	●	○				
Unsupervised Learning				○	○	●
Total (Dominant)	5 (4)	5 (2)	2 (1)	3 (1)	3 (2)	2 (2)



Исследование операций и принятие решений:

- Элементы теории игр
- Поиск по дереву (минмакс алгоритм с альфа-бета отсечениями)
- Поиск кратчайшего пути (Поиск в ширину, Дейкстры, A*)
- Нахождение минимального остовного дерева (Алгоритм Прима)
- Задачи планирования в среде (GOAP и HNT)
- Задачи оптимизации, задачи комбинаторной оптимизации

Машинное обучение и ИИ:

- Агентный подход и основы обучения с подкреплением (Q-learning)
- Мягкие вычисления: нечеткие системы (Системы Мамдани и Ларсена)
- Мягкие вычисления: эволюционные алгоритмы (Генетический и муравьиный)

Программирование поведения и классический ИИИ:

- Системы на основе правил, деревья решений и скриптинг
- Деревья поведений, Конечные автоматы и Иерархические модели
- Utility based AI и технологии программирования ИИИ
- Элементы генерации контента (клеточные автоматы)



ИИИ в рамках Unreal Engine:



- Работа с навигацией
- Работа с деревьями поведений
- Использование blackboards
- Использование и разработка сервисов в BT
- Использование и разработка декораторов в BT
- Использование Perception технологий
- Работа с EQS
- Проектирование и разработка игрового ИИ для агентов с помощью конечных автоматов

И самое главное: **командная работа над собственной игрой, в которой реализован собственно созданный игровой искусственный интеллект.**

**Спасибо
за внимание!**

ITMO *re than a*
UNIVERSITY

tatiana.atyapsheva@mail.ru
abrosimov.kirill.1999@mail.ru