4 августа 2010 в 18:34

# «Hello world!» с помощью генетических алгоритмов

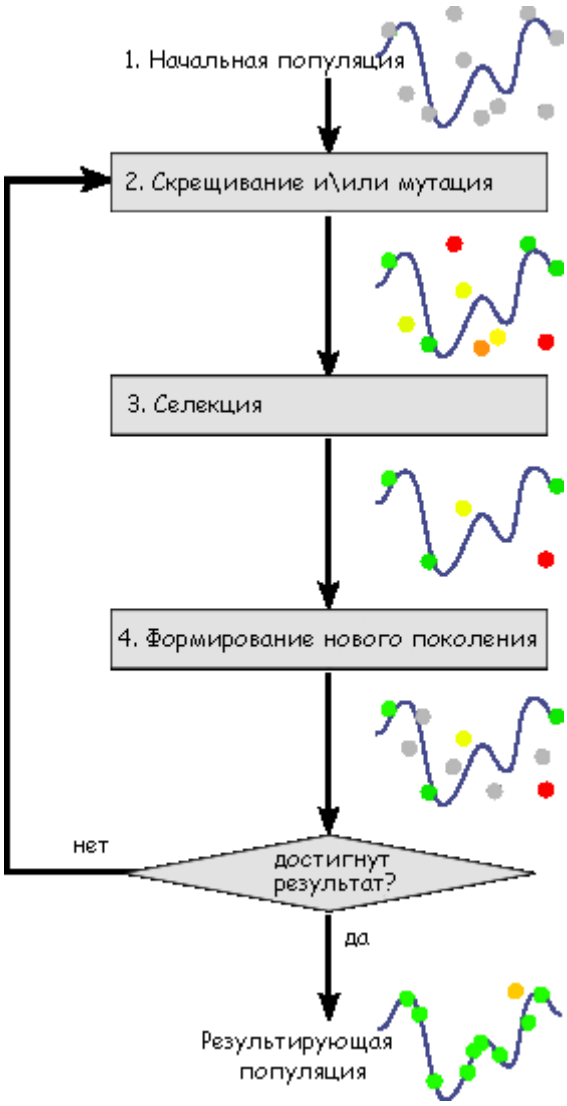
Алгоритмы*

В наше время все большую популярность набирают генетические алгоритмы. Их используют для решения самых разнообразных задач. Где-то они работают эффективнее других, где-то программист просто решил выпендриться…

Так что же такое генетический алгоритм? Если верить википедии, то генетический алгоритм — это эвристический алгоритм поиска, используемый для решения задач оптимизации и моделирования путём случайного подбора, комбинирования и вариации искомых параметров с использованием механизмов, напоминающих биологическую эволюцию. Является разновидностью эволюционных вычислений. Отличительной особенностью генетического алгоритма является акцент на использование оператора «скрещивания», который производит операцию рекомбинации решений-кандидатов, роль которой аналогична роли скрещивания в живой природе.

Т.е. генетический алгоритм работает наподобие нашей с вами эволюции. Сначала создаются начальные популяции, затем они скрещиваются между собой (при этом возможно возникновение мутаций). Популяции выжившие в процессе естественного отбора проверяются на удовлетворение заданным критериям. Если удовлетворяют — все счастливы, если нет — вновь скрещиваются и так до финальной победы.

Как это все выглядит вы можете увидеть на следующем рисунке:




В качестве примера приведу алгоритм, создающий строку «Hello world!»

Данный генетический алгоритм имеет такие параметры:

| | |
|---|---|
| Размер популяции: | 2048 |
| Мутации (%): | 25 |
| Элитарность (%): | 10 |

**Исходный код**

```
#include <iostream>                                    // для cout и т.п.
#include <vector>                                      // для класса vector
#include <string>                                      // для класса string
#include <algorithm>                                   // для алгоритма
сортировки
#include <time.h>                                      // для случайных величин
```

```
#include <math.h>                                       // для abs()

#define GA_POPSIZE              2048            // размер популяции
#define GA_MAXITER              16384           // максимальное число итераций
#define GA_ELITRATE             0.10f           // элитарность
#define GA_MUTATIONRATE         0.25f                   // мутации
#define GA_MUTATION             RAND_MAX * GA_MUTATIONRATE
#define GA_TARGET               std::string("Hello world!")

using namespace std;

struct ga_struct
{
        string str;                                                     // строка
        unsigned int fitness;                                           // пригодность
};

typedef vector<ga_struct> ga_vector;                    // для краткости

void init_population(ga_vector &population,
                                        ga_vector &buffer )
{
        int tsize = GA_TARGET.size();

        for (int i=0; i<GA_POPSIZE; i++) {
                ga_struct citizen;

                citizen.fitness = 0;
                citizen.str.erase();

                for (int j=0; j<tsize; j++)
                        citizen.str += (rand() % 90) + 32;

                population.push_back(citizen);
        }

        buffer.resize(GA_POPSIZE);
}

void calc_fitness(ga_vector &population)
{
        string target = GA_TARGET;
        int tsize = target.size();
        unsigned int fitness;

        for (int i=0; i<GA_POPSIZE; i++) {
                fitness = 0;
                for (int j=0; j<tsize; j++) {
```

```
                        fitness += abs(int(population[i].str[j] - target[j]));
                }

                population[i].fitness = fitness;
        }
}

bool fitness_sort(ga_struct x, ga_struct y)
{ return (x.fitness < y.fitness); }

inline void sort_by_fitness(ga_vector &population)
{ sort(population.begin(), population.end(), fitness_sort); }

void elitism(ga_vector &population,
                             ga_vector &buffer, int esize )
{
        for (int i=0; i<esize; i++) {
                buffer[i].str = population[i].str;
                buffer[i].fitness = population[i].fitness;
        }
}

void mutate(ga_struct &member)
{
        int tsize = GA_TARGET.size();
        int ipos = rand() % tsize;
        int delta = (rand() % 90) + 32;

        member.str[ipos] = ((member.str[ipos] + delta) % 122);
}

void mate(ga_vector &population, ga_vector &buffer)
{
        int esize = GA_POPSIZE * GA_ELITRATE;
        int tsize = GA_TARGET.size(), spos, i1, i2;

        elitism(population, buffer, esize);

        // Mate the rest
        for (int i=esize; i<GA_POPSIZE; i++) {
                i1 = rand() % (GA_POPSIZE / 2);
                i2 = rand() % (GA_POPSIZE / 2);
                spos = rand() % tsize;

                buffer[i].str = population[i1].str.substr(0, spos) +
                                population[i2].str.substr(spos, esize -
spos);
```

```
                if (rand() < GA_MUTATION) mutate(buffer[i]);
        }
}

inline void print_best(ga_vector &gav)
{ cout << "Best: " << gav[0].str << " (" << gav[0].fitness << ")" << endl; }

inline void swap(ga_vector *&population,
                                ga_vector *&buffer)
{ ga_vector *temp = population; population = buffer; buffer = temp; }

int main()
{
        srand(unsigned(time(NULL)));

        ga_vector pop_alpha, pop_beta;
        ga_vector *population, *buffer;

        init_population(pop_alpha, pop_beta);
        population = &pop_alpha;
        buffer = &pop_beta;

        for (int i=0; i<GA_MAXITER; i++) {
                calc_fitness(*population);            // вычисляем пригодность
                sort_by_fitness(*population);         // сортируем популяцию
                print_best(*population);              // выводим лучшую
популяцию

                if ((*population)[0].fitness == 0) break;

                mate(*population, *buffer);           // спариваем популяции
                swap(population, buffer);             // очищаем буферы
        }

        return 0;
}
```

Сразу хочется дать несколько комментариев к коду. Во-первых, мы схитрили, когда установили фиксированные размеры строк в генетическом алгоритме, которые равны размеру искомой строки. Это сделано, чтобы обеспечить лучшее понимание кода. Во-вторых, для хранения данных используется STL-класс vector — это сделано для того, чтобы упростить сортировку. Наконец, программа использует два массива для хранения популяций — один для текущей, второй является буфером для следующего поколения. В каждом поколении очки округляются до целого для ускорения вычислений.

**Определение пригодности**

Давайте, взглянем на функцию определения пригодности популяции:

```
void calc_fitness(ga_vector &population)
{
string target = GA_TARGET;
int tsize = target.size();
unsigned int fitness;

for (int i=0; i<GA_POPSIZE; i++) {
fitness = 0;
for (int j=0; j<tsize; j++) {
fitness += abs(int(population[i].str[j] - target[j]));
}

population[i].fitness = fitness;
}
}
```

В принципе, здесь просто перебирается каждый член популяции и сравнивается с таковым в целевой строке. Разницы между символами складываются и накопленная сумма используется как значение пригодности (таким образом, чем она меньше, тем лучше).

**Результат**

При выполнении, программа выдает на экран лучшую популяцию и ее пригодность (число в скобках).

```
Best: IQQte=Ygqem# (152)
Best: Crmt`!qrya+6 (148)
Best: 8ufxp+Rigfm* (140)
Best: b`hpf"woljh[ (120)
Best: b`hpf"woljh4 (81)
Best: b`hpf"woljh" (63)
Best: Kdoit!wnsk_! (24)
Best: Kdoit!wnsk_! (24)
Best: Idoit!wnsk_! (22)
Best: Idoit!wnsk_! (22)
Best: Idoit!wnsk_! (22)
Best: Idoit!wnsk_! (22)
Best: Ifknm!vkrlf? (17)
Best: Ifknm!vkrlf? (17)
Best: Gfnio!wnskd$ (14)
Best: Ffnjo!wnskd$ (14)
Best: Hflio!wnskd$ (11)
Best: Hflio!wnskd$ (11)
Best: Kflkn wosld" (8)
Best: Ifmmo workd" (6)
Best: Hfljo wosld" (5)
Best: Hflmo workd" (4)
Best: Hflmo workd" (4)
```

```
Best: Hflmo workd" (4)
Best: Iflmo world! (3)
Best: Iflmo world! (3)
Best: Hflmo world! (2)
Best: Hflmo world! (2)
Best: Hflko world! (2)
Best: Hflko world! (2)
Best: Hdllo world! (1)
Best: Hfllo world! (1)
Best: Hfllo world! (1)
Best: Helko world! (1)
Best: Hfllo world! (1)
Best: Hfllo world! (1)
Best: Hfllo world! (1)
Best: Hello world! (0)
```

**Заключение**

Надеюсь, эта небольшая программа способна продемонстрировать, как простой генетический алгоритм может быть использован для решения задачи. Данный алгоритм не является самым эффективным, хотя использование STL должно немного помочь. Алгоритм был протестирован в Visual Studio.

**Пример применения**

Чтобы проверить торгового робота на диапазоне параметров можно «проходить» диапазон не в циклах (как сейчас во всех программах), а при помощи генетических алгоритмов. И время сокращается в 9-12 раз. #

Оригинальная статья на Английском языке.
Что такое генетический алгоритм?

- генетические алгоритмы
- , Hello world

+95

153

Kastrulya0001 38,9

## комментарии (60) отслеживать новые: в почте в трекере

+2


folone,4 августа 2010 в 18:54#

via.

0


senyai,4 августа 2010 в 19:00#

↥

2003 :)

+2


xtender,4 августа 2010 в 19:12#

↥
Да ладно, это ж по-моему заказу :D habrahabr.ru/blogs/algorithm/100953/#comment_3125762
0

Kastrulya0001,4 августа 2010 в 19:16#

↥
Да, признаю ваши заслуги! :D Но думаю, что повторов не будет.
+1

xtender,4 августа 2010 в 19:20#

↥
Надеюсь :) во-всяком случае «Hello world», решенный генетическим алгоритмом, интереснее всяческих print «hello world» :)
+3

sunnybear,4 августа 2010 в 19:02#
видимо, это самый «простой» способ вывести Hello World! :)
+5

Kastrulya0001,4 августа 2010 в 19:04#

↥
Назначаетесь сегодня капитаном!
+3

Grundiss,5 августа 2010 в 08:17#

↥
А представьте, кто-то сегодня решил освоить Си и загуглил «hello world» =)
0

Daemon_Hell,5 августа 2010 в 10:43#

↥
А как же версия на брэйнфаке?
0

sunnybear,5 августа 2010 в 10:46#

↥
в том-то и дело, что она короче :)
+2

conscell,5 августа 2010 в 11:27#

↥
Вот, кстати, вывести с помощью ГА самый короткий код на Брейнфаке, который выводил бы «Hello, World!», было бы намного интереснее и поучительней.
0

build_your_web,7 августа 2010 в 23:52#

↥

Если дать программе возможность генетически генерировать другой код, он сможет сгенерировать генератор ботов и потом захватить мир.

0

 **Xazzzi**,29 ноября 2011 в 05:14#

↥

Есть такой проэкт, называется Avida. Там чем-то похожим занимаются.

+5

 **Vladson**,4 августа 2010 в 19:05#

Много кода, мало текста. Если бы я не знал что такое «генетический алгоритм» я бы честно говоря не понял к чему все эти танцы с брутфорсом.

0

 **Autorun**,4 августа 2010 в 19:12#

интересно, что за текст образовала человеческая популяция?

+1

 **Kastrulya0001**,4 августа 2010 в 19:22#

↥

Боюсь представить, весь текст?

–20

 **timursun**,4 августа 2010 в 19:15#

Из поста понял 2 вещи:

1. генетический алгоритм — штука интетесная, но мутная, надо будет почитать;
2. все-таки за несколько лет работы с PHP стало непривычно и как-то неудобно смотреть на C.

+3

 **SubW**,5 августа 2010 в 14:16#

↥

Вот такая информация поступила…
боюсь вспомнить, как я раньше жил без знания того что вы нам поведали.

+2

 **Levsha100**,4 августа 2010 в 19:27#

Так было уже ж про генетические алгоритмы, даже с видео =)
Пруф:
habrahabr.ru/blogs/edu_2_0/86777/

0

 **Kastrulya0001**,4 августа 2010 в 19:33#

↥

Добавил, спасибо!

+1

 **DIDJER**,4 августа 2010 в 19:31#

— так то же прикольно.

–5

 **KAndy**,4 августа 2010 в 19:50#

Мне всегда казалось что под «генетическим алгоритмом» скривается алгоритм поиска екстремума функции…

0

 **conscell**,5 августа 2010 в 10:07#

↥
А данный пример — это тоже поиск экстремума фитнес-функции. Только вот он единственный и известный заранее, что делает весь пример бесполезным.
0

 **KAndy**,5 августа 2010 в 11:49#

↥
Да, матиматика нинче не в моде…
Вот программирование, это наше все :(
+5

 **sherbacov**,4 августа 2010 в 19:56#
Best: **Idoit***!wnsk_!* (22)

Мне кажется, он пытается нам что-то сказать…
0

 **Kastrulya0001**,4 августа 2010 в 20:01#

↥
Мне вот капча периодически что-то сообщает, порой мне кажется, что она со мной пытается заговорить.
+1

 **sherbacov**,4 августа 2010 в 20:05#

↥
И еще автору:
Я бы описал основное применение, а именно:
Чтобы проверить торгового робота на диапазоне параметров можно «проходить» диапазон не в циклах (как сейчас во всех программах), а при помощи генетических алгоритмов. И время сокращается в 9-12 раз.

0

 **SkywalkerY**,4 августа 2010 в 22:42#

↥
можно трактовать как I do it — я делаю это, а можно подумать, что популяция просто перепутала местами 3 и 4 символ:)
+10

 **root_sashok**,4 августа 2010 в 20:06#
Zetway: Первое существо, которое создал Бог, говорило только «Hello, World!» и умирало...

+1

 **Bkmz**,4 августа 2010 в 20:43#
Я в генетических алгоритмах не силен, точннее вообще их не знаю и не понимаю. Но решим их изучения наткнулся на странный эффект.
Если добавить символ z в строку, то перебор до этого символа не доходит. Доходит только до «y».
Гдето собака зарыта ;)
0

ZumZoom,5 января 2011 в 03:47#

↥
```
for (int j=0; j<tsize; j++)
    citizen.str += (rand() % 90) + 32;
```

рандом не доходит до буквы z (код 122)
:)
+17

TimTowdy,4 августа 2010 в 20:44#
По-моему пример, использующий в качестве фитнесс функции сравнение с эталоном, ужасен. Генетические алгоритмы используют как раз для того, чтобы вычислить этот эталон. Если эталон известен заранее — в генетическом алгоритме нет смысла, в данном случае он не решает вообще никакую задачу.
–1

qmax,5 августа 2010 в 09:40#

↥
вот мне этот момент тоже непонятен совершенно.
и полностью дискредитирует идею генетических алгоритмов.
+1

conscell,5 августа 2010 в 10:06#

↥
Почему это один некорректный пример «полностью дискредитирует» давно себя отлично зарекомендовавшую идею?

При правильно подобранной фитнес-функции ГА работают очень неплохо, иногда им просто нет альтернативы.
0

qmax,5 августа 2010 в 10:09#

↥
потомучто конерктно в этой статье, рандомный читатель, в соответствии с названием может ожидать «максимально простую реализацию ГА, иллюстрирующую их свойства»
а вместо этого видит максимально корявый пример.
0

conscell,5 августа 2010 в 10:21#

↥
Повторяю вопрос — как один частный корявый пример может дискредитировать идею вообще?
0

kastigar,5 августа 2010 в 11:21#

↥
Вы знаете что такое дискредитация? Вот если б я не осознавал, что пример неудачный, то сказал бы что ГА — это баловство. Хотя сомневаюсь, что автор делает это умышленно.
0

conscell,5 августа 2010 в 11:29#

↥
Вы не чувствуете семантической разницы между «полностью дискредитирует» и «ваш пример дискредитирует»?
0
 qmax,5 августа 2010 в 11:36#

↥
вы искренне полагаете, что повторение вопроса поможет вашему пониманию ответа на него?

поясняю:
этот пример может быть единственно известным читателю, заинтересовавшемуся статьёй.
0
 conscell,5 августа 2010 в 11:46#

↥
Довольно таки странный пример. Мне не известны люди, выстраивающие свое мнение о какой либо теме на основании первой же левой статьи. Тем более что даже эта статья содержит библиографические ссылки.
–1
 qmax,5 августа 2010 в 12:10#

↥
данная статья в той форме как есть у непосвящённого читателя достигает единственной цели — продемонстрировать абсурдность ГА.

в чём ещё смысл этой левой статьи?
0
 conscell,5 августа 2010 в 12:12#

↥
Она демонстрирует непонимание автором смысла ГА, только и всего. Сами то ГА при чем?
0
 qmax,5 августа 2010 в 12:15#

↥
этот смысл доступен только тем, кто понимает ГА.
а не рандомному читателю, с позиций которого я о ней сужу.
0
 conscell,5 августа 2010 в 12:18#

↥
Рандомный читатель, найдя какую либо статью, пройдет по приведенным в статье ссылкам, прочитает соответствующий раздел в Википедии и только после этого составит какое-то мнение. Это если тема хоть немного ему интересна. Если нет, то никакого мнения он составлять и не будет.
0
 qmax,5 августа 2010 в 12:19#

↥

читатель пройдёт по ссылкам, если его заинтересовала статья.

если ему интересна тема сама по себе — он сделает это и без статьи.

0


**wpm1**,5 августа 2010 в 10:41#

↥

Поделитесь нормальным примером. Очень интересно попробовать.

0


**conscell**,5 августа 2010 в 11:31#

↥

Например, вот эта классическая задача (правда, это эволюционный алгоритм, обобщение генетического):

en.wikipedia.org/wiki/The_Evolution_of_Cooperation

Еще очень красивый (но очень сложный) пример:

www.framsticks.com/

0


**wpm1**,5 августа 2010 в 11:55#

↥

мерси. второй пример захватывает. Рассылаю по контакт листу.

0


**kastigar**,5 августа 2010 в 11:16#

↥

Во, и я задлася тем же вопросом. Есть эталон, программа пытается его найти брутфорсом, хотя вот он лежит в памяти. Сразу зреет вывод, что либо пример бестолковый, либо генетические алгоритмы туфта. Не увидел реального толку от ГА в этом примере. Хотя очень хотелось…

**+11**


**nsinreal**,4 августа 2010 в 21:04#

Перенесите в «Ненормальное программирование». Ибо это самый извращенный способ вывода «hello world»

**+1**


**Kastrulya0001**,4 августа 2010 в 21:12#

↥

Так и сделаю, если хабровчане поддержат ваше предложение.

0


**Tar**,4 августа 2010 в 21:29#

В коде, имху, ошибка.

```
...
buffer[i].str = population[i1].str.substr(0, spos) +
population[i2].str.substr(spos, esize - spos);
...
```

Вместо substr(spos, **esize** - spos) надо substr(spos, **tsize** - spos)

**+16**


**Aquahawk**,4 августа 2010 в 21:58#

#pragma warning(disable:4786) // отключаем отладочные предупреждения

А за такое предлагаю бить по рукам, сильно и больно.

0


**AlMag**,5 августа 2010 в 23:36#

↥

да, нельзя, что бы такое вошло в привычку, на сколько бы ты ни был в себе уверен.

**–1**


**RinOS**,5 августа 2010 в 14:44#

```
inline void swap(ga_vector *&population, ga_vector *&buffer)
{
ga_vector *temp = population; population = buffer; buffer = temp;
}
```

**swap**(population, buffer); // **очищаем буферы**

Нестыковка… population копируется в buffer что бы потом скрестить с предыдущей популяцией.

0


**Kastrulya0001**,6 августа 2010 в 00:40#

Спасибо всем, кто помог сделать статью лучше и полнее. Больше сотни в избранном, значит результат есть, а это главное.

0


**wpm1**,6 августа 2010 в 12:11#

Мы вот с другом задумались. У вас ведь задача получается с одним глобальным экстремумом? И локальные экстремумы совпадают с глобальным.

Можно ли вашу целевую функцию сделать функцией с двумя глобальными экстремумами. Чтобы задача интереснее была.

0


**wpm1**,6 августа 2010 в 12:45#

↥

Ну и мы сделали так:

1) ввели вторую цель той же длины #define GA_TARGET2 std::string(«Hello gnome!»)

2) изменили целевую функцию

```
population[i].fitness = 0;
fitness = 0;
for (int j=0; j<tsize; j++) {
fitness += abs(int(population[i].str[j] — target[j]));
}
population[i].fitness = fitness;
fitness = 0;
for (int j=0; j<tsize; j++) {
fitness += abs(int(population[i].str[j] — target2[j]));
}

population[i].fitness *= fitness;
```

Т.е. определили её как произведение модулей разности целевых векторов и особи.
И теперь алгоритм падает то к одному то ко второму экстремуму.
0
 wpm1,6 августа 2010 в 12:47#

↥
Ну и мы сделали так:
1) ввели вторую цель той же длины #define GA_TARGET2 std::string(«Hello gnome!»)
2) изменили целевую функцию
```
population[i].fitness = 0;
fitness = 0;
for (int j=0; j<tsize; j++) {
fitness += abs(int(population[i].str[j] — target[j]));
}
population[i].fitness = fitness;
fitness = 0;
for (int j=0; j<tsize; j++) {
fitness += abs(int(population[i].str[j] — target2[j]));
}

population[i].fitness *= fitness;
```

Т.е. определили её как произведение модулей разности целевых векторов и особи.
И теперь алгоритм падает то к одному то ко второму экстремуму.
0
 wpm1,6 августа 2010 в 12:47#

↥
тьфу. «Some error… We know...»