

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»
Кафедра “Системи автоматизованого проектування”



Звіт
до лабораторної роботи №5
з курсу: «Методи нечіткої логіки та еволюційні алгоритми при
автоматизованому проектуванні»
на тему:
**«Програмування ГА для
задачі комівояжера»**

Виконав:
ст. гр. КНСП-11
Гуль О. В.

Перевірив:
Кривий Р. З.

Мета: запрограмувати ГА для задачі комівояжера використовуючи заданий метод селекції.

Теоретичні відомості

Селекція - це вибір тих хромосом, які будуть брати участь в створенні нащадків для наступної популяції, тобто для чергового покоління. Такий вибір проводиться відповідно до принципу природного відбору, за яким найбільші шанси на участь в створенні нових особин мають хромосоми з найбільшими значеннями функції пристосованості. Існують різні методи селекції.

Селекція рулеткою

У методі рулетки (roulette-wheel selection) особини відбираються за допомогою N «запусків» рулетки, де N — розмір популяції. Колесо рулетки містить по одному сектору для кожного члена популяції. Розмір i -го сектору пропорційний ймовірності попадання в нову популяцію. При такому відборі члени популяції з більш високою пристосованістю з більшою ймовірністю будуть частіше вибиратись, ніж особини з низькою пристосованістю.

Турнірна селекція

При турнірному відборі (tournament selection) з популяції, яка складається із N особин, вибираються випадковим чином t особин, і найкраща особина записується в проміжний масив. Ця операція повторюється N раз. Особини в отриманому проміжному масиві потім використовуються для схрещування (також випадковим чином). Розмір групи рядків, що відбираються для турніру, часто дорівнює 2. У цьому випадку говорять про двійковий (парний) турнір. Взагалі ж t називають чисельністю турніру. Перевагою даного способу є те, що він не вимагає додаткових обчислень.

Рангова селекція (селекція усіканням)

При ранговій селекції особини популяції сортуються за значенням їх функції пристосованості. Кількість копій кожної особини, введених в батьківську популяцію, розраховується як відсоток від розміру популяції. Серед особин, що потрапили «під поріг» випадковим чином N раз вибирається найбільш везуча і записується в проміжний масив, з якого потім вибираються особини безпосередньо для схрещування.

Завдання

(Варіант 3)

У вас є безліч міст (представлені у вигляді точок на площині з X і Y координати). Мета полягає в тому, щоб знайти найкоротший маршрут, який

відвідує кожне місто рівно один раз, повертаючись в кінці до своєї відправної точки.

Дано від 10 до 50 точок. Метод селекції — селекція рулеткою.

Хід роботи

Координати точок(міст) читаються з файлу *.csv, де першим значенням є назва міста, другим — координата X, а третім — координата Y.

Для виконання завдання була використана функція ga пакету MatLab. Окремо були реалізовані функції для генерації початкової вибірки, мутації та схрещування. Функція для відбору методом рулетки є стандартною в пакеті MatLab.

Функція для оцінки шляху

```
function [output_args] = FitnessFcn( input_args )
%% Цільова функція. Довжина пройденого шляху
% input_args = [x1, x2, x3, ... ]
% x1, x2 ... – індекси міст, в порядку їх проходження

global WAY_MATRIX;
sum = 0;
way_length = length(input_args);
for i = 1:1:way_length-1
    sum = sum + WAY_MATRIX(input_args(i), input_args(i+1));
end

% Додаємо шлях до повернення в початкове місто
sum = sum + WAY_MATRIX(input_args(way_length), input_args(1));

output_args = sum;

end
```

Функція для генерації початкової вибірки

```
function Population = CreationFcn( GenomeLength, FitnessFcn, options )
%% Функція для створення початкової популяції
негомологічних особин

ret = zeros(options.PopulationSize, GenomeLength);

for i = 1:1:options.PopulationSize
    vars = 1:1:GenomeLength;
    for j = 1:1:GenomeLength
        t = randi(length(vars));
        ret(i, j) = vars(t);
        vars(t) = [];
    end;
end;

end;
```

```
Population = ret;
```

```
end
```

Точка входу в програму

```
function main()
[ names, x, y ] = textread('cities.csv', '%s %d %d', 'delimiter', ',');
global WAY_MATRIX;
WAY_MATRIX = getWayMatrix(x, y, 1000);

population_size = ceil(length(x)^(1/2)); % до більшого цілого
nvars = length(x);

global RET;
RET = struct('generation', 0, 'population', struct, 'fvals', struct);

options = gaoptimset(...
    'CreationFcn', @CreationFcn, ...
    'PopulationSize', population_size, ...
    'MutationFcn', @MutationFcn, ...
    'CrossoverFcn', @CrossoverFcn, ...
    'OutputFcns', {@OutputFcn}, ...
    'SelectionFcn', @selectionroulette, ...
    'PlotFcns', {@gaplotbestf, @gaplotdistance} ...
);

[xval, fval, exitflag, output, population, scores] = ga(@FitnessFcn, nvars, options);

% графічне представлення шляху
XX = x; YY = y; NNames = names;
for i = 1:1:nvars
    XX(i) = x(xval(i));
    YY(i) = y(xval(i));
    NNames(i) = names(xval(i));
end
XR = [XX(nvars), XX(1)];
YR = [YY(nvars), YY(1)];

figure
plot(XX, YY, '-*', XR, YR, '—')
text(XX+1, YY+1, NNames)

% вивід результатів
disp('Початкова популяція:');
PrintIter(RET.population.s0, RET.fvals.s0, nvars, population_size);
disp('Покоління 1:');
PrintIter(RET.population.s1, RET.fvals.s1, nvars, population_size);
disp('Покоління 2:');
PrintIter(RET.population.s2, RET.fvals.s2, nvars, population_size);
```

```

disp('Результат:');
PrintIter(population, scores, nvars, population_size);
fprintf('Best:¥n');
PrintOne(xval, fval, nvars);
PrintOneStr(NNames, fval, nvars);

end

function way_matrix = getWayMatrix(X, Y, M)
%% Формує матрицю з відстаннями між кожним містом
ret = zeros(length(Y), length(X));
for i = 1:1:length(Y)
    for j = 1:1:length(X)
        if (i == j)
            ret(i, j) = M;
        else
            v = ((X(i) - X(j)).^2 + (Y(i)-Y(j)).^2).^(1/2);
            ret(i, j) = v;
        end
    end
end
end
way_matrix = ret;
end

```

Результати виконання:

	Стандарт	Стандарт	Стандарт
1 A	12	40	
2 B	41	50	
3 C	26	8	
4 D	3	3	
5 E	15	26	
6 F	10	1	
7 G	18	7	
8 K	50	32	
9 L	20	22	
10 M	8	3	

Command Window

New to MATLAB? See resources for [Getting Started](#).

```

>> main
Optimization terminated: maximum number of generations exceeded.
Початкова популяція:
[ 9, 10, 2, 8, 5, 1, 4, 6, 7, 3, ] => 228.51
[ 2, 10, 9, 5, 7, 1, 4, 8, 6, 3, ] => 345.05
[ 7, 1, 9, 10, 6, 5, 4, 3, 8, 2, ] => 256.34
[ 8, 1, 4, 2, 3, 10, 7, 5, 9, 6, ] => 310.91

```

Рис. 1. Початкові дані

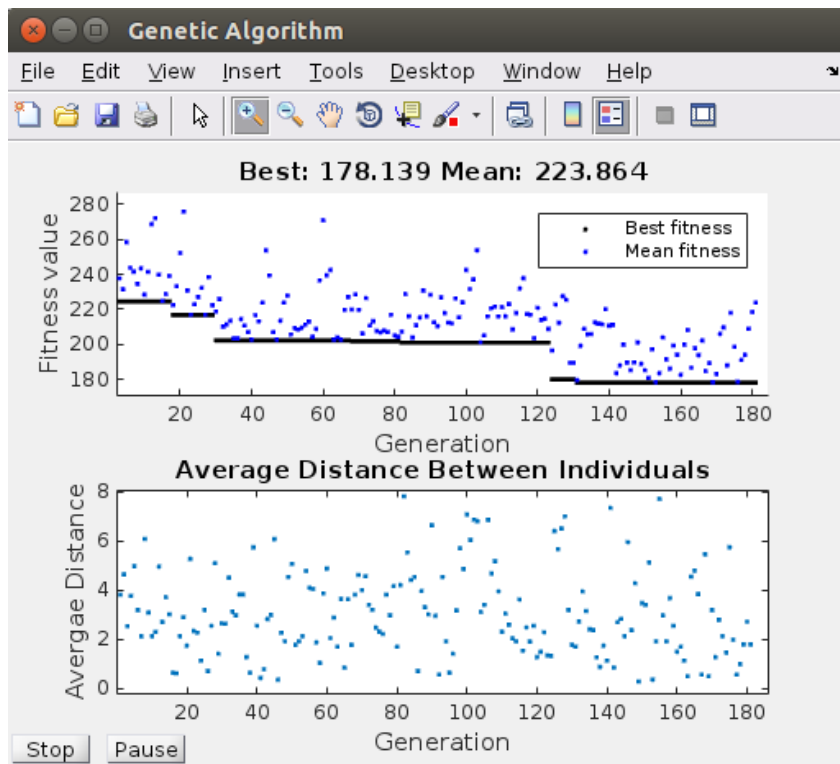


Рис. 3. Сходимость функції

Результат:

```
[ 9,  6, 10,  4,  7,  3,  8,  2,  5,  1, ] => 178.14
[ 9,  6, 10,  2,  7,  3,  8,  4,  5,  1, ] => 289.47
[ 9,  6, 10,  4,  7,  3,  8,  2,  5,  1, ] => 178.14
[ 9,  6, 10,  5,  7,  3,  8,  2,  4,  1, ] => 249.71
```

Best:

```
[ 9,  6, 10,  4,  7,  3,  8,  2,  5,  1, ] => 178.14
[ L,  F,  M,  D,  G,  C,  K,  B,  E,  A, ] => 178.14
>>
```

Рис. 4. Результат виконання алгоритму

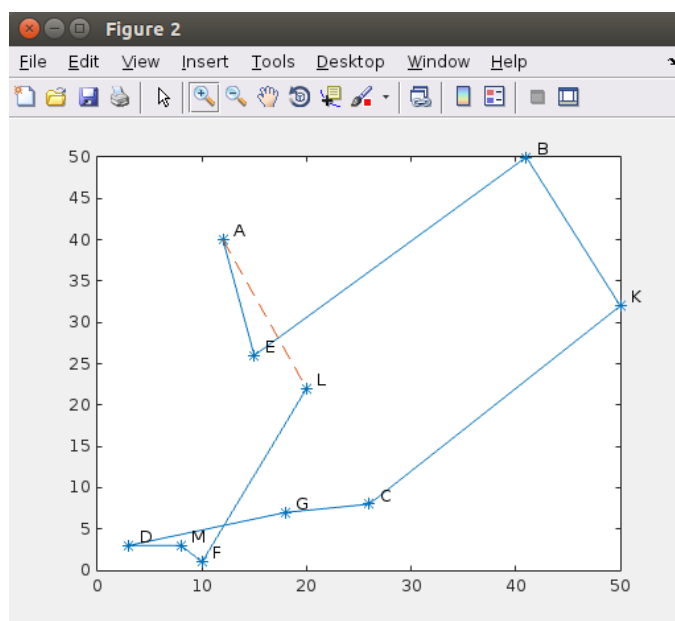


Рис. 5. Графічне представлення маршруту

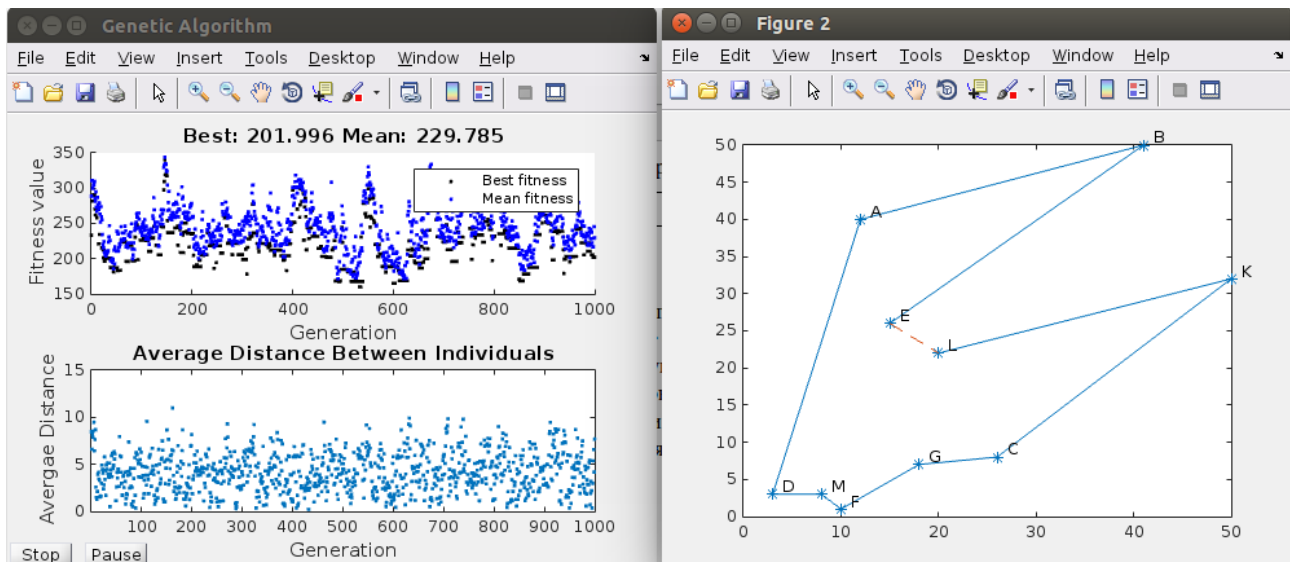


Рис. 6. Результат виконання без елітних нащадків

Висновок: На даній лабораторній роботі, при використанні генетичного алгоритму, була використана селекція рулеткою. Сходимість функції з обраним методом селекції і без визначення елітних потомків є надзвичайно низька.