|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **НУЛП, САПР, СПК** | | **Тема** | **Оцінка:** | **Підпис:** |
| КНСП-11 | 5 | Запрограмувати ГА для  задачі комівояжера (tsp) |  |  |
| Дитиняк В. П. | |
|  | |
| Методи нечіткої логіки та  еволюційні алгоритми | | **Викладач:**  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 рангової 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 Х, 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 ... - інденкси 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 негомологічих 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;    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  way\_matrix = ret;  end |

Результати виконання:

|  |  |
| --- | --- |
|  |  |

Рис. 1. Початкові дані

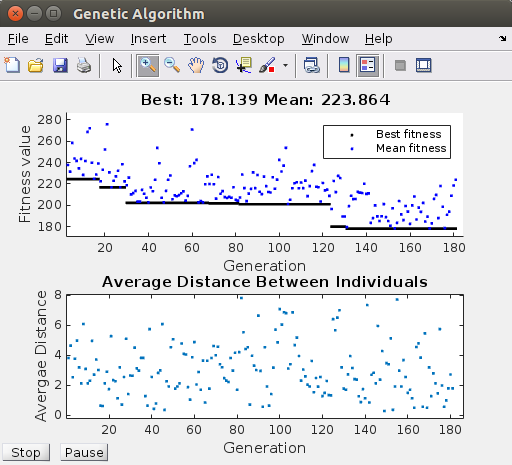


Рис. 3. Сходимість функції

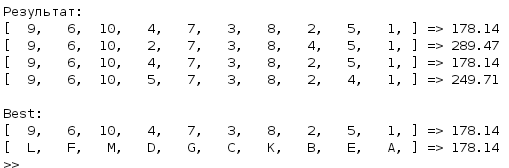


Рис. 4. Результат виконання алгроритму

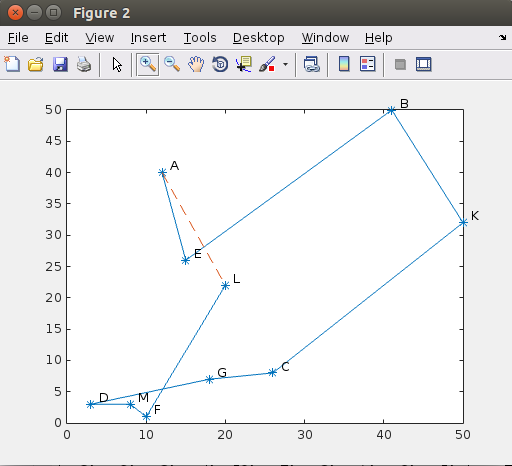


Рис. 5. Графічне представлення маршруту

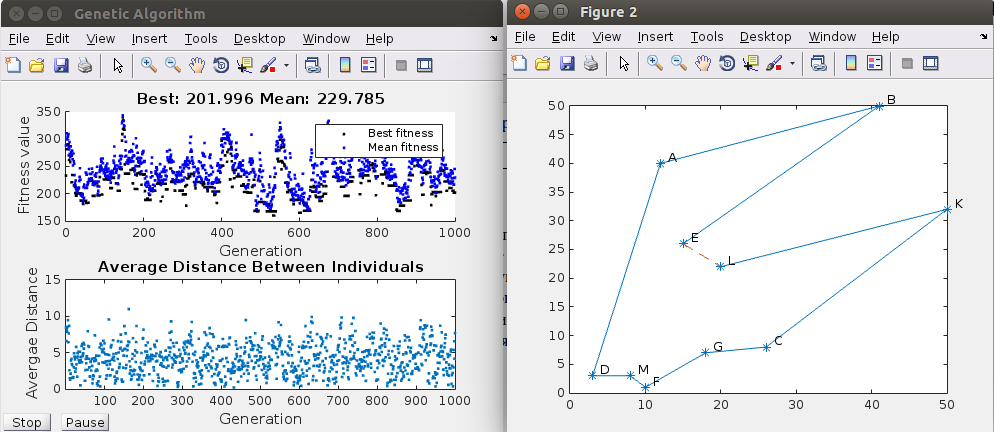


Рис. 6. Результат виконання без елітних нащадків

**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.