МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет “Львівська політехніка”
Інститут комп’ютерних наук та інформаційних технологій

Звіт
до лабораторної роботи № 3

на тему:

"КОМБІНАТОРНА ОПТИМІЗАЦІЯ ЗА ДОПОМОГОЮ ЕВОЛЮЦІЙНИХ МЕТОДІВ"

з курсу “ Методи нечіткої логіки та еволюційні алгоритми при автоматизованому проектуванні ”

Виконав:
студент групи КНСП-11
Дербіж А. В.

Перевірив: викладач каф. САП,
асист. Кривий Р.З

Львів - 2020

МЕТА РОБОТИ

Ознайомитися з основними теоретичними відомостями, вивчити еволюційні оператори схрещування та мутації, що використовуються при розв’язуванні задач комбінаторної оптимізації.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Інвертування

Оператор інвертування (інверсії) змінює порядок генів у ділянках хромосом. Його мета полягає в тому, щоб спробувати знайти порядок генів, що має кращий еволюційний потенціал. Перев порядкування також значно розширює область пошуку. Мало того, що генетичний метод намагається знаходити гарні множини значень генів, він також одночасно пробує знаходи- ти гарне впорядкування генів.

Найчастіше використовуються наступні види інвертування:

Інвертування із зсувом: випадковим чином вибирається елемент, який переміщується на випадково обирану позицію, зсуваючи інші елементи вправо по циклу. Іншою формою такого виду інвертування є випадковий ви-14 бір підстроки й переніс її на випадково обирану позицію із зсувом елементів, що залишилися.

Як правило, задається ймовірність інвертування P in , конкретне значення якої залежить від розв'язуваної задачі й у загальному випадку перебуває в інтервалі [0,001;0,01].

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

Розробити за допомогою пакету Matlab програмне забезпечення для вирішення задачі комівояжера. Параметри еволюційного методу обрати з таблиці 1 відповідно до варіанту.

Варіант 5

5	двохточечне впорядковуюче	інвертування із зсувом

Інші параметри, необхідні для еволюційного пошуку, обрати самостійно. Вибір параметрів обґрунтувати.

Виконати тестування розробленого програмного забезпечення за допомогою вирішення конкретних прикладів задачі комівояжера. Задачі (не менше трьох) для виконання тестування програми сформулювати самостійно. Вибір тестових задач обґрунтувати.

Порівняти одержані результати вирішення різних прикладів задачі комівояжера. Результати порівняльного аналізу звести до таблиці, попередньо розробивши систему критеріїв порівняння результатів вирішення задачі комівояжера.

РЕЗУЛЬТАТИ ВИКОНАННЯ РОБОТИ

Для виконання завдання була використана функція ga пакету Matlab, і реалізовано власні функції мутації та схрещування, згідно з варіантом.

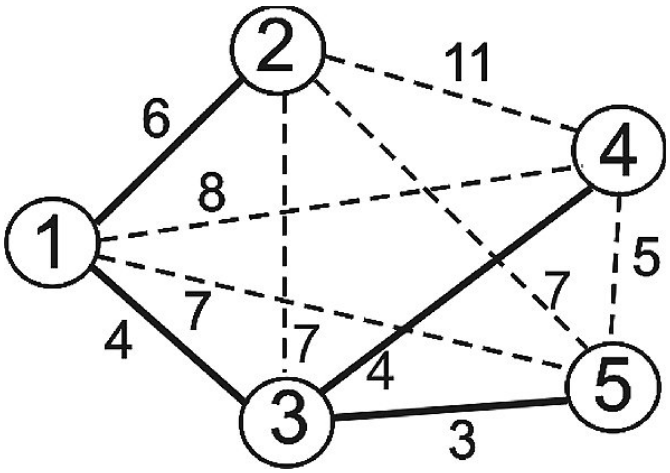

Рис. 1. Графічне представлення задачі комівояжера

Функція для схрещування

```
function [xoverKids] = CrossoverFcn( parents, options, nvars, FitnessFcn, ...
   unused,thisPopulation )
%% Реалізація функції для схрещування потомків (циклове схрещування)
% parents - індекси батьків в поточній популяції, що беруть участь у
%         схрещуванні. вектор з парною кількістю елементів
% nvars   - кількість змінних (генів)
% unused  - вектор-стовбець із оцінкою кожної особини
% thisPopulation - поточна популяція (матриця)

ret = zeros(length(parents)/2, nvars);

for i = 1:2:length(parents)-1
   p1 = thisPopulation(parents(i), :);
   p2 = thisPopulation(parents(i+1), :);

   % генеруємо цикл
   t = randi(nvars);     % початок циклу (індекс)
   cycle = zeros(1,nvars);
   for j = 1:1:nvars
      cycle(1,j) = t;
      nv = p2(t);
      t = find(p1==nv);
      if (p1(cycle(1,1)) == nv)
         break;  % цикл замкнувся
      end;
   end;

   % елементи, що не попали в цикл успадковуються від іншого батька
   child = p2;
   for j = 1:1:nvars
      if (cycle(1,j) ~= 0)
         child(1,cycle(1,j)) = p1(cycle(1,j));
      end;
   end;
   ret((i+1)/2,:) = child;
end;
xoverKids = ret;

end
```

Функція мутації

```
function [  mutationChildren ] = MutationFcn( parents, options, nvars, ...
FitnessFcn, state, thisScore, thisPopulation )
% Проводить мутацію методом золотого січення

% parents - номер особини в популяції, що мутує
% nvars - кількість змінних
% state - інформація про поточну популяцію
% thisScore - оцінки поточної популяції
% thisPopulation - поточна популяція

 k = 0.62;            % k=62%
 t = ceil(k*nvars);     % точка розриву

 mutant = thisPopulation(parents, :);
 d = mutant(t);
 d1 = mutant(t+1);
 mutant(t) = d1;
 mutant(t+1) = d;

 mutationChildren = mutant;

end
```

Точка запуску програми

```
%% точка запуску програми
%% Варіант 3
%% Схрещування: циклове
%% Мутація: золотого перетину

% Задача: знайти найвигідніший маршрут,
% який проходить через кожне місто по одному разу
% одною особою є послідовність обходу міст
% значення генів не можуть повторюватися
% а довжина хромосоми рівна кількості міст
% для матриці з 5 міст можливо всього 5! = 120 різних способів обходу
% тому розмір популяції візьмемо рівний кількості міст (5)

startPopulation = [
   1, 2, 3, 4, 5;
   2, 3, 4, 5, 1;
   3, 4, 5, 1, 2;
   4, 5, 1, 2, 3;
   5, 1, 2, 3, 4
];

options = gaoptimset(...
   'EliteCount', 0, ...
   'PopulationSize', 5, ...
   'InitialPopulation', startPopulation, ...
   'MutationFcn', @MutationFcn, ...
   'CrossoverFcn', @CrossoverFcn, ...
   'TimeLimit', 3 ...
);
[x,fval,exitflag,output,population,scores] = ga(@optim_function, 5, options);
```

```
disp('Найращей потомок:'); disp(x);
fprintf('f(x) = %d\n', fval);
disp('Остання популяція:');
for i=1:1:5
   for j=1:1:5
      fprintf('\t%d', population(i,j));
   end;
   fprintf('\t=>\t%d\n', scores(i));
end;
```

Результат виконання

```
>> main
Optimization terminated: average change in the fitness value less than options.TolFun.
Найращей потомок:
     2     1     3     5     4

f(x) = 18
Остання популяція:
        2       1       3       5       4       =>      18
        2       1       3       5       4       =>      18
        2       1       3       5       4       =>      18
        2       1       3       5       4       =>      18
        2       1       3       4       5       =>      19
>>
```

Рис.1

```
New to MATLAB? See resources for Getting Started.
>> main
Optimization terminated: average change in the fitness value less than options.TolFun.
Найращей потомок:
     1     2     3     5     4

f(x) = 21
Остання популяція:
        1       2       3       4       5       =>      22
        1       2       3       5       4       =>      21
        1       2       3       4       5       =>      22
        1       2       3       5       4       =>      21
        1       2       3       4       5       =>      22
fx >>
```

Рис.2

```
New to MATLAB? See resources for Getting Started.
>> main
Optimization terminated: average change in the fitness value less than options.TolFun.
Найращей потомок:
     1     2     3     5     4

f(x) = 21
Остання популяція:
        1       2       3       4       5       =>      22
        1       2       3       5       4       =>      21
        1       2       3       5       4       =>      21
        1       2       3       5       4       =>      21
        1       2       3       5       4       =>      21
fx >>
```

Рис.3

ВИСНОВОК

На цій лабораторній роботі, я ознайомився з основними теоретичними відомостями, вивчив еволюційні оператори схрещування та мутації, що використовуються при розв’язуванні задач комбінаторної оптимізації. Розробив за допомогою пакету Matlab програмне забезпечення для вирішення задачі комівояжера.