|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **НУЛП, САПР, СПК** | | **Тема** | **Оцінка:** | **Підпис:** |
| КНСП-11 | 3 | Комбінаторна оптимізація  за допомогою еволюційних  методів |  |  |
| Янчук Н. Ю. | |
| Варіант 10 | |
| Методи нечіткої логіки та  еволюційні алгоритми | | **Викладач:**  Кривий Р. З. | |

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

**Теоретичні відомості**

При використанні методів еволюційного пошуку для розв’язку задач комбінаторної оптимізації, як правило, застосовуються негомологічні числові хромосоми, тобто такі хромосоми, гени яких можуть приймати значення в заданому інтервалі. При цьому інтервал однаковий для всіх генів, але в хромосомі не може бути двох генів з однаковим значенням.

Комбінаторні задачі оперують із дискретними структурами або розміщенням об'єктів, незначні зміни яких часто викликають стрибкоподібну зміну показників якості (фітнесс- функції). Традиційні оператори еволюційні оператори, що генерують нових нащадків, не можуть бути застосовані при використанні негомологічних хромосом, оскільки внаслідок виконання таких операторів генеруються нащадки, що містять однакові гени і тому не можуть бути інтерпретовані при розв’язку комбінаторної задачі. Тому для розв’язку задач комбінаторної оптимізації були розроблені спеціальні генетичні оператори, що не створюють неприпустимих рішень.

**Завдання**

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

|  |  |  |
| --- | --- | --- |
| № | Еволюційні оператори | |
| Схрещування | Мутація |
| 10 | позиційно впорядковуюче | інвертування із зсувом |

**Хід роботи**

Для виконання завдання була використана функція ga пакету Matlab, і реалізовано власні функції мутації та схрещування, згідно з варіантом для розвязу задачі комівояжера з чотирьма, п’ятьма і шістьма містами (рис. 1,3,5 відповідно).

Функція схрещування

function [xover\_kids] = crossover( parents\_for\_crossover, options, nvars, FitnessFcn, ...

   unused,this\_population )

xover\_kids = zeros(length(parents\_for\_crossover)/2, nvars);

position = 2; %позиція гену для схрещування

for i = 1:2:length(parents\_for\_crossover)-1

   p1 = this\_population(parents\_for\_crossover(i), :);

   p2 = this\_population(parents\_for\_crossover(i+1), :);

   child=zeros(1,nvars);

   for j=1:1:position

       child(j)=p2(j);

   end

   pp1=zeros(1,nvars);

   counter=1;

   tamporalPtr=position+1;

   for j=tamporalPtr:1:nvars

       pp1(counter)=p1(j);

       counter=counter+1;

   end

   for j=1:1:position

       pp1(counter)=p1(j);

       counter=counter+1;

   end

   for j=1:1:position

       for jj=1:1:nvars

           if ( pp1(jj)==child(j))

               pp1(jj)=0;

           end

       end

   end

   for j=1:1:nvars

       if(pp1(j)~=0)

           position=position+1;

           child(position)=pp1(j);

       end

   end

   xover\_kids((i+1)/2,:) = child;

end

end

Функція мутації

function [  mutant ] = mutation( hromosom\_to\_mutate, options, nvars, ...

fitness\_fcn, state, this\_score, population )

current\_hromosome=population(hromosom\_to\_mutate, :);

shift = randi(nvars);

mutant = zeros(1, nvars);

for i=1:1:nvars

    if(i+shift<=nvars)

       mutant(i+shift)=current\_hromosome(i);

    else

        mutant(i+shift-nvars)=current\_hromosome(i);

    end

end\

end

Відповідно до виконання програми відтворено оптимальні маршрути. (рис 2, 4, 6 відповідно).

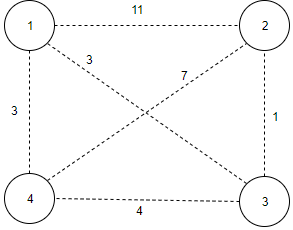


Рис. 1. Постановка задачі з чотирьма містами

Результат виконпння функції пошуку маршруту:

The best hromosome:

    1 4     3 2

The number of generations was : 51

The best function value found was : 7

Last generation:

1 4 3 2 => 7

1 4 3 2 => 7

1 4 3 2 => 7

4 2 3 1 => 12

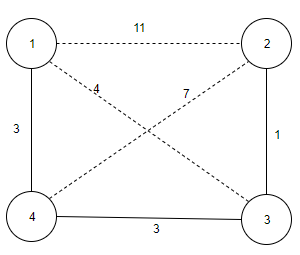


Рис. 2. Розв’язок задачі з чотирьма містами

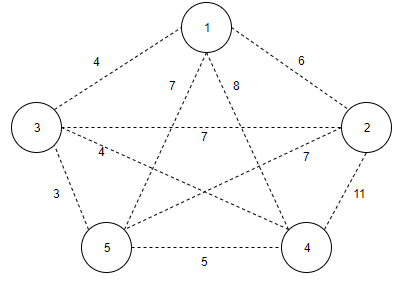


Рис. 3. Постановка задачі з п’ятьма містами

Результат виконпння функції пошуку маршруту:

The best hromosome:

    4 5     3 1 2

The number of generations was : 63

The best function value found was : 18

Last generation:

4 5 3 1 2 => 18

4 5 1 2 3 => 25

1 2 4 5 3 => 25

4 5 3 1 2 => 18

4 5 3 1 2 => 18

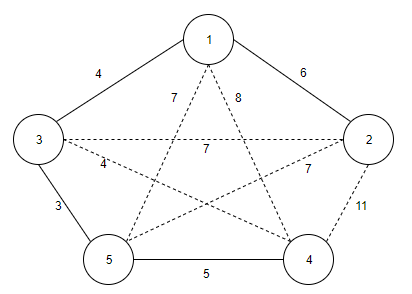


Рис. 4. Розв’язок задачі з п’ятьма містами

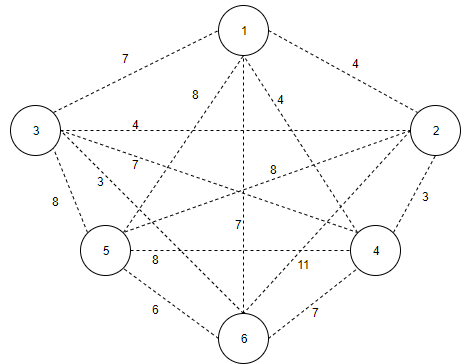


Рис. 5. Постановка задачі з шістьма містами

Результат виконпння функції пошуку маршруту:

The best hromosome:

    2 4     1 3 6     5

The number of generations was : 68

The best function value found was : 20

Last generation:

2 4 1 3 6 5 => 20

2 4 1 5 3 6 => 26

2 4 1 3 6 5 => 20

2 4 1 3 6 5 => 20

3 6 5 2 4 1 => 24

4 1 3 6 5 2 => 25

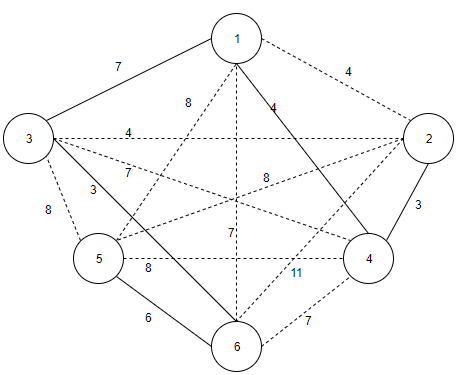


Рис. 5. Розв’зок задачі з шістьма містами

**Висновок**

На відміну від класичних методів розв’язання задачі комівояжера, використовуючи генетичні алгоритми, ми зразу отримуємо декілька оптимальних варіантів. Але такий підхід не гарантує, що результат є найоптимальніший. Чотири з десяти запусків пргграми показують оптимальні рішення задачі комівояжера