Прізвище: Метельський

Ім’я: Всеволод

Група: КНМ-14

Дата прийняття роботи

у системі Git: 25.04.2017

Дисципліна: Методи нечіткої логіки та еволюційні алгоритми при автоматизованому проектуванні

Перевірив: Кривий Р.З.

**Звіт до лабораторної роботи № 3**

**«Комбінаторна оптимізація за допомогою еволюційних**

**методів»**

**МЕТА РОБОТИ**

Ознайомитися з основними теоретичними відомостями, вивчити еволюційні оператори схрещування та мутації, що використовуються при розв’язуванні задач комбінаторної оптимізації. Розробити за допомогою пакету Matlab програмне забезпечення для вирішення задачі комівояжера.

**КОРОТКІ 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.

**Впорядковуючий оператор схрещування**

Впорядковуючий оператор схрещування (Order crossover, OX) був запропонований Д. Девісом у 1985 році для негомологічних числових хромосом. Схрещування може виконуватися по одній або по двох точках. Точки схрещування вибираються випадково. При одноточечному схрещуванні в хромосому першого нащадка копіюється хромосома першого батька, а потім гени нащадка, що розташовані правіше точки схрещування, перевпорядковуються у послідовність, що відповідає другому батькові. При цьому другий батько переглядається від початку до кінця, зліва направо, і елементи, яких не вистачає у нащадку, додаються, починаючи від точки схрещування, один по одному.

Крок 1. Випадковим чином вибрати точку схрещування.

Крок 2. Скопіювати в хромосому першого нащадка сегмент хромосоми першого батька, що розташований лівіше точки схрещування.

Крок 3. Інші гени в нащадку копіюються із другого батька в упорядкованому вигляді зліва направо, крім елементів, які вже увійшли до нащадка.

**ІНДИВІДУАЛЬНЕ ЗАВДАННЯ**

Розробити програмне забезпечення для розв’язку задачі комівожера.

|  |  |  |
| --- | --- | --- |
| **№ варіанту** | **Еволюційні оператори** | |
| **Схрещування** | **Мутація** |
| **6** | позиційно впорядковуюче | класична мутація обміну |

**Фрагмент програмного коду:**

nn=10;

asz=10;

ps=3000;

ng=200;

pm=0.01;

pm2=0.02;

pmf=0.08;

r=asz\*rand(2,nn);

dsm=zeros(nn,nn);

for n1=1:nn-1

r1=r(:,n1);

for n2=n1+1:nn

r2=r(:,n2);

dr=r1-r2;

dr2=dr'\*dr;

drl=sqrt(dr2);

dsm(n1,n2)=drl;

dsm(n2,n1)=drl;

end

end

G=zeros(ps,nn);

for psc=1:ps

G(psc,:)=randperm(nn);

end

plot(1);

hpb=plot(NaN,NaN,'r-');

ht=title(' ');

hold on;

for n=1:nn

text(r(1,n),r(2,n),num2str(n));

end

plot(r(1,:),r(2,:),'k.');

axis equal;

xlim([-0.1\*asz 1.1\*asz]);

ylim([-0.1\*asz 1.1\*asz]);

pthd=zeros(ps,1);

p=zeros(ps,1);

for gc=1:ng

for psc=1:ps

Gt=G(psc,:);

pt=0;

for nc=1:nn-1

pt=pt+dsm(Gt(nc),Gt(nc+1));

end

pt=pt+dsm(Gt(nn),Gt(1));

pthd(psc)=pt;

end

ipthd=1./pthd;

p=ipthd/sum(ipthd);

[mbp bp]=max(p);

Gb=G(bp,:);

if mod(gc,1)==0

set(hpb,'Xdata',[r(1,Gb) r(1,Gb(1))],'YData',[r(2,Gb) r(2,Gb(1))]);

set(ht,'string',['generation: ' num2str(gc) ' best distance length: ' num2str(pthd(bp))]);

drawnow;

end

ii=roulette\_wheel\_indexes(ps,p);

Gc=G(ii,:);

Gch=zeros(ps,nn);

for prc=1:(ps/2)

i1=1+2\*(prc-1);

i2=2+2\*(prc-1);

g1=Gc(i1,:);

g2=Gc(i2,:);

cp=ceil((nn-1)\*rand);

g1ch=insert\_begining(g1,g2,cp);

g2ch=insert\_begining(g2,g1,cp);

Gch(i1,:)=g1ch;

Gch(i2,:)=g2ch;

end

G=Gch;

for psc=1:ps

if rand<pm

rnp=ceil(nn\*rand);

rpnn=randperm(nn);

ctp=rpnn(1:rnp);

Gt=G(psc,ctp);

Gt=Gt(randperm(rnp));

G(psc,ctp)=Gt;

end

end

for psc=1:ps

if rand<pm2

cp=1+ceil((nn-3)\*rand);

G(psc,:)=[G(psc,cp+1:nn) G(psc,1:cp)];

end

end

for psc=1:ps

if rand<pmf

n1=ceil(nn\*rand);

n2=ceil(nn\*rand);

G(psc,n1:n2)=fliplr(G(psc,n1:n2));

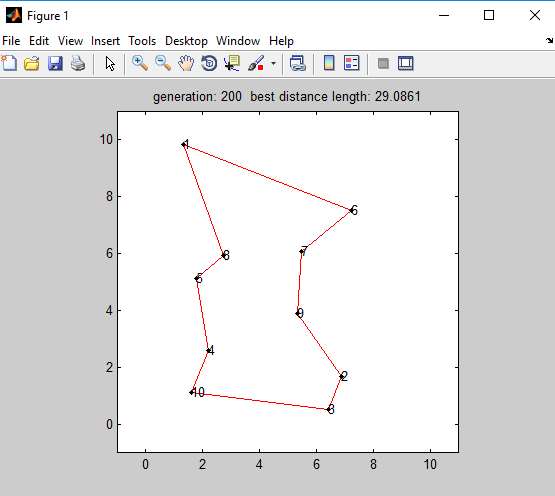
end

end

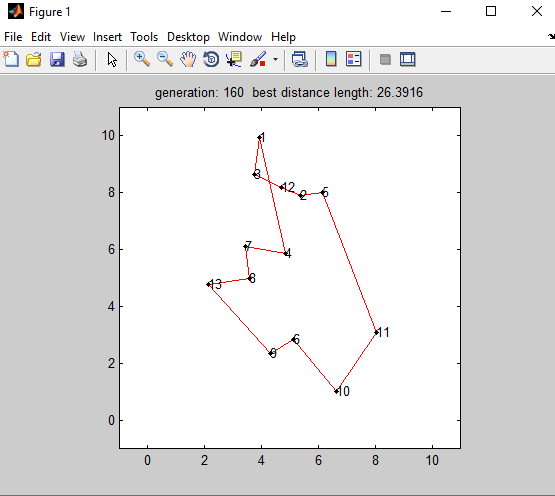
G(1,:)=Gb;

End

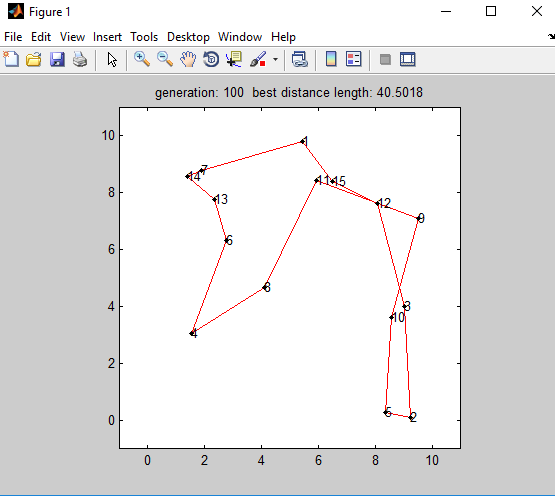
**Отримані результати:**



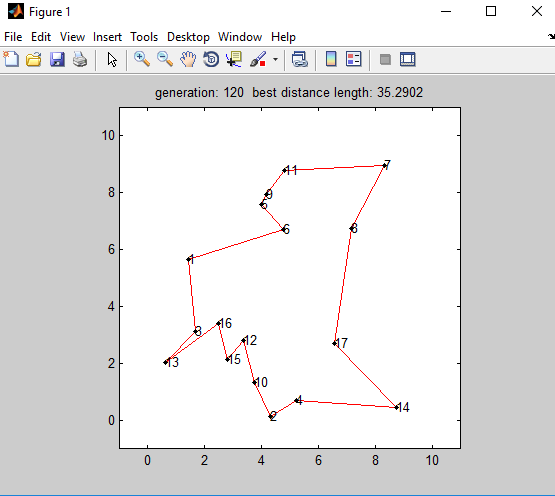
**Рис 1.** Результат, отриманий при 10 містах та 200 поколінь



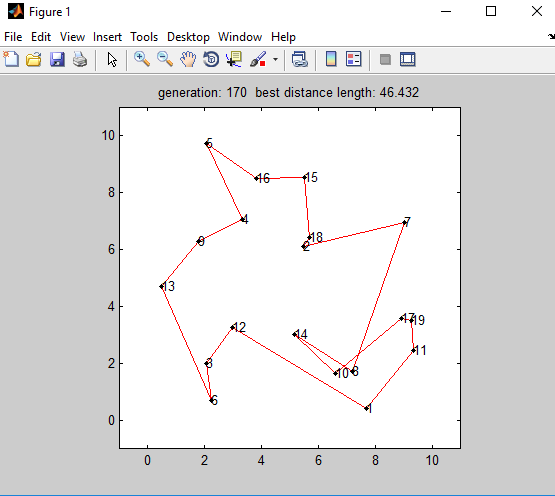
**Рис 2.** Результат, отриманий при 13 містах та 160 поколінь



**Рис 3.** Результат, отриманий при 15 містах та 100 поколінь



**Рис 4.** Результат, отриманий при 17 містах та 120 поколінь



**Рис 5.** Результат, отриманий при 19 містах та 170 поколінь

**Висновок:** виконуючи дану лабораторну роботу я ознайомився з основними теоретичними відомостями, вивчив еволюційні оператори схрещування та мутації, що використовуються при розв’язуванні задач комбінаторної оптимізації та розробив за допомогою пакету Matlab програмне забезпечення для вирішення задачі комівояжера.