

遗传算法求解 TSP 问题的 MATLAB 实现

温清芳

(德化陶瓷职业技术学院, 福建 德化 362500)

摘要:以遗传算法求解旅行商问题(TSP)为例,提出一种改进的交叉和变异算子,深入讨论了各个遗传算子的程序实现,并给出其算子的 MATLAB 程序编码,最后用 5 个城市的非对称 TSP 进行仿真分析.结果表明,改进的算法比传统算法收敛速度更快,适应值更优,说明改进算法是有效的,证实 TSP 问题是遗传算法得以成功应用的典型例子.

关键词:旅行商问题; 遗传算法; Grefenstette 编码; MATLAB 程序

中图分类号: O2-2

文献标识码: A

文章编号: 1007 - 5348(2007)06 - 0018 - 05

尽管对遗传算法的研究方兴未艾^[1],但因其基础理论难有突破性进展,要想知道其效果如何就只有利用计算机进行模拟.目前大多采用 C 语言编写这类程序^[2].C 语言作为一种高效的编程语言有它的许多优点,但对于编写遗传算法(GA)程序来说,MATLAB 语言则更为合适.遗传算法是一种全局寻优搜索算法,它首先对问题的可行解进行编码,组成染色体,然后通过模拟自然界的进化过程,对初始种群中的染色体进行选择、交叉和变异,通过一代代进化来找出最优适应值的染色体来解决问题.它具有很强的全局搜索能力和较强的自适应性,适合解决连续变量函数优化问题和离散变量的优化组合问题^[2].它比用 C 等其它高级语言要简单、灵活、快捷,程序篇幅也将缩小许多.

旅行商问题(TSP)是一个典型的优化组合问题,可具体描述如下:已知 n 个城市之间的相互距离,现有一个推销员从某一城市出发,必须遍访这 n 个城市,并且每个城市只能访问一次,最后又必须返回到出发城市.如何安排他对这些城市的访问次序,可使其旅行路线的总长度最短^[2,3].事实上,很多实际问题,如电路布线、输油管路铺设、连锁店的货物配送路线等,经过简化处理后都可建模为 TSP 问题^[4].因而,对求解 TSP 问题方法的研究便具有重要的应用价值.

由于 TSP 问题的可行解数目与城市数目 n 是成指数型增长的,是一个 NP 难问题,因而一般只能近似求解.遗传算法(GA)是求解该问题较为有效的方法之一,目前应用遗传算法解决 TSP 问题,主要解决编码问题和算子的设计问题.采用所遍历城市的顺序排列来表示各个个体的编码串是最自然的编码方法,但它对应的交叉和变异运算实现起来较困难,因为常规的交叉和变异运算会使群体中产生一些不满足问题的约束条件或无意义的巡回路线.为克服这一缺点,可以采用 Grefenstette 编码法^[5].本文主要针对 TSP 问题,讨论自然编码方式下算子的改进及其 MATLAB 的程序实现,较好地解决了群体的多样性和收敛速度的矛盾.

1 Grefenstette 编码法的描述

对于一个 TSP 问题的城市列表 W ,假定每个城市的一个访问顺序为 $T = (t_1, t_2, t_3, \dots, t_n)$,规定每访问完一个城市,就从城市列表中将该城市去掉,则用第 i ($i = 1, 2, 3, \dots, n$) 个访问的城市 t_i 在所有未访问城市列表 $W \setminus \{t_1, \dots, t_{i-1}\}$ 中的对应位置序号 g_i ($1 \leq g_i \leq n - i + 1$) 就可表示具体访问哪个城市,如此这样直到处理完 W 中所有的城市.将全部 g_i 顺序摆列在一起得到一个列表: $G = (g_1, \dots, g_n)$ 就可表示一条巡回路线,它即为遗传算法中的一个个体的基因.例如:

收稿日期: 2006 - 11 - 13

作者简介: 温清芳(1959 -),女,福建德化人,德化陶瓷职业技术学院高级讲师,主要从事高校数学教学与研究.

$$W = (1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ 15),$$

则

$$\text{Path} = (8 \ 15 \ 2 \ 10 \ 7 \ 4 \ 3 \ 11 \ 14 \ 6 \ 12 \ 9 \ 5 \ 13 \ 1),$$

对应

$$\text{Chrom} = (8 \ 14 \ 2 \ 8 \ 6 \ 3 \ 2 \ 5 \ 7 \ 3 \ 4 \ 3 \ 2 \ 2 \ 1).$$

这种编码法的优点在于任意的基因型个体都对应着一条有实际意义的巡回路线,因此可使用常规的交叉算子对其进行操作.

2 TSP 问题的数学模型

假设有一个图 $G = (V, E)$, 其中 V 是顶点集, E 是边集, 设 $D = (d_{ij})_{n \times n}$ 是顶点 i 和顶点 j 之间的距离所组成的距离矩阵, TSP 问题就是求出一条通过所有顶点且每个顶点只通过一次的具有最短距离的回路. 若 $d_{ij} = d_{ji} (i \neq j)$, 称为对称 TSP 问题, 否则称为非对称 TSP 问题.

若对于城市 $W = (v_1, v_2, \dots, v_n)$ 的一个访问顺序为 $T = \{t_1, t_2, t_3, \dots, t_n\}$, 其中 $t_i \in V = \{1, \dots, n\}$, 且 $t_{n-1} = t_1$, 则 TSP 问题的数学模型为:

$$\min L = \sum_{i=1}^n d(t_i, t_{i+1}),$$

其中, $d(t_i, t_{i+1})$ 表示顶点 t_i 和顶点 t_{i+1} 之间的距离.

3 MATLAB 6.5 的实现

GA 中的控制参数如下: 染色体数 Popsize, 最大代数 Maxgen, 城市数目 N_{cities} , 交叉率 P_c , 距离矩阵 D , 城市列表 W .

3.1 种群的初始化

编码: Function Chrom = TSPinitialize (Popsize, N_{cities})

```
for i = 1:Popsize
    Chrom(i, 1) = 1; %推销员从城市 1 出发
    for j = 2: Ncities
        Chrom(i, j) = round(rand * (Ncities - j)) + 1;
    end
end
```

解码: Function pop = TSPdecode (Popsize, N_{cities} , Chrom, W)

```
For i = 1:Popsize
    temp W = W;
    for j = 1: Ncities
        pop(i, j) = temp W (chrom(i, j));
        for k = (Chrom(i, j)) : (Ncities - j)
            temp W(k) = temp W(k + 1)
        end
    end
end
end;
```

计算目标函数:Function rpop = rotatepop (pop)

for $i = 1 : \text{Popsize}$

rpop = [pop ($i, 2 : N_{\text{cities}}$), pop ($i, 1$)]

end

Function cost = TSPcost (Popsize, pop, rpop)

For $i = 1 : \text{Popsize}$

Cost (i) = Sum (diag (D (pop ($i, :$)) (pop ($i, :$))));

end;

3.2 遗传操作设计

选择:采用具有排名的转盘式选择算子.按照个体目标函数值从大到小排列 $\text{cost}(1) \quad \text{cost}(2) \quad \dots \quad \text{cost}(\text{Popsize})$,然后根据线性函数计算选择概率:

$$P_i = \frac{1}{\text{Popsize}} \left(a - \frac{b \cdot i}{\text{Popsize} + 1} \right) \quad i = 1, 2, \dots, \text{Popsize},$$

式中: i 为排名序数, a, b 是常数, $1 \leq a \leq 2, b = 2(a - 1)$,通常 $a = 1.1$.

然后按照转盘方式选择父体进行遗传操作:先产生一个 $[0, 1]$ 内的随机数 r ,如果 $P_0 + \dots + P_i - 1 < r$
 $P_1 + \dots + P_i (P_0 = 0)$,则选择 i ,否则随机选一染色体.

Function Chrom = select (Popsize, cost, chrom)

$B = 2^* (a - 1)$;

[fit, I] = sort (- cost);

for $i = 1 : \text{Popsize}$

Pfit (i) = ($a - b^* i / (\text{Popsinx} + 1)$) / Popsize;

end

pfit = cum Sum (pfit);

rs = rand (1, Popsize);

for $i = 1 : \text{Popsize}$

if rs (i) <= pfit (i), Chrom ($i, :$) = Chrom (I(i), :);

else Chrom (I, :) = Chrom ((round * Popsize) + , :);

end

end;

交配:采用常规单点交叉.将选择中产生的种群做为父代,两两组队,以交叉率发生交叉操作:若发生交叉操作,在 $[1, N_{\text{cities}} - 1]$ 区间随机确定一个交叉位置,进行该位后的所有基因对换,否则不做任何处理.

Function Chrom = crossover (Chrom)

For $i = 1 : \text{Popsize}/2$

If rand < P_c

jcross = fix (rand * ($N_{\text{cities}} - 1$)) + 1; % jcross 为交叉位

$P_1 = \text{Chrom} (2^* i - 1, :)$, $P_2 = \text{Chrom} (2^* i, :)$

Chrom ($2^* i - 1, :$) = [$P_1 (1 : \text{jcross})$, $P_1 ((\text{jcross} + 1) : N_{\text{cities}})$];

Else Chrom ($2^* i - 1, :$) = chrom ($2^* i - 1, :$);

```
Chrom (2 * i : ) , = Chrom (2 * i , :)
```

```
end
```

```
end ;
```

变异:传统的遗传算法的基因变异概率很小,在遗传进化过程中,优势个体会越来越多,并在种群中占据大多数位置,到了后期,种群的个体极为相似,就会导致早熟收敛.为解决该弊端,当两个个体太相似时就进行变异操作:随机产生一个染色体替换当作比较的两个个体中的较差者.

```
Function Chrom = popmut (Chrom , cost)
```

```
for i = 1 : Popsizel/2
```

```
sameidx = [ Chrom ( i * i - 1 = = Chrom(2 * i , : ) ] ;
```

```
diffidx = find (sameidx = = 0) ;
```

```
if length (diffidx) < = 2
```

```
if cost (2 * i - 1) < cost (2 * i)
```

```
for j = 2 : Ncities
```

```
Chrom (2 * i , j) = round (rand * ( Ncities - j)) + 1 ;
```

```
else for j = 2 : Ncities Chrom (2 * i - 1 , j) = round (rand * ( Ncities - j)) + 1 ;
```

```
end
```

```
end ;
```

4 仿真试验

以下以 5 个城市的非对称 TSP 问题来说明上述算法的可行性.

GA 的流程图如下:

MATLAB 主程序:

```
Popsizel = 10 ; Maxgen = 20 ; Ncities = 5 ; Pc = 0.8 ;
```

```
D = [0 6 18 4 8 ; 7 0 17 3 7 ; 4 4 0 4 5 ; 20 19 24 0 22 ; 8 8 16 6 0] ;
```

```
W = [1 : Ncities] ;
```

```
Gen = 0 ;
```

```
Chrom = TSPinitialize (Popsizel , Ncities) ;
```

```
Pop = TSPdecode (Popsizel , Ncities , Chrom , W) ;
```

```
Rpop = rotatepop (pop) ;
```

```
Cost = TSPcost (Popsizel , pop , rpop) ;
```

```
While gen < mazgen
```

```
Gen = gen + 1 ;
```

```
Chrom = select (Popsizel , pfit , Chrom) ;
```

```
Chrom = crossve (Chrom) ;
```

```
pop = TSPdecode (Popsizel , Ncities , Chrom , W)
```

```
rpop = rotatepop (pop) ;
```

```
cost = TSPcost (Popsizel , pop , rpop) ;
```

```
Chrom = popmut (Chrom , cost , Popsizel , Ncities , Pc , D)
```

初始化, 确定遗传参数, 产生初始群体

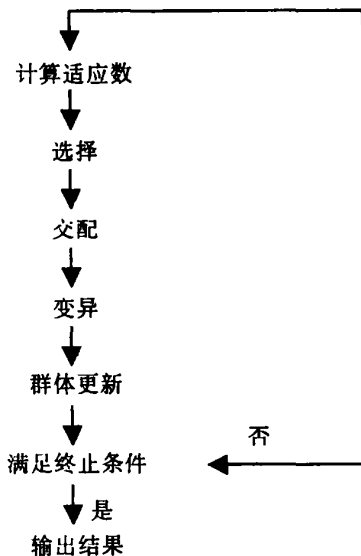


图 1 GA 流程图

```
pop = TSPdecode (Popsizel, Ncities, Chrom, W);  
rpop = rotatepop (pop);  
cost = TSPcost (Popsizel, pop, rpop);  
[costmin, idx] = min (cost);  
tourmin = pop (idx, :);  
tourcost (gen) = costmin;  
disp ([ num2str (gen) 'minimum trip lengths = ' num2str (costmin) ] );  
end;  
x = [ 1 :Maxgen ];  
y = tourmin;  
plot (x, y)
```

得到最优值为 46,最优解为 14325,在第 2 代就达到最优解.

5 结论

本文提出一种基于 TSP 问题的改进的遗传算法,结合 TSP 问题的特点,采用一种受贪婪算法启发的新的交叉算子和变异算子,极大地提高了算法的收敛速度,同时又保证了最优解的得到,这种算法思想不仅可以用于 TSP 问题,还可以很好地应用于其他类似优化问题.

参考文献:

- [1]肖磊,张阿卜,徐文进. 用 MATLAB 求解 TSP 问题的一种改进遗传算法[J]. 厦门理工学院学报,2005, 13(4):38 - 42.
- [2]陈国梁,王煦法,庄镇泉,等. 遗传算法及其应用[M]. 北京:人民邮电出版社,1996:137 - 139.
- [3]Fogel D B. Applying evolutionary programming to selected traveling salesman problems[J]. Cybernetics and systems, 1993, 24: 27 - 36.
- [4]邢桂华. 用 MATLAB 实现中国旅行商问题的求解[J]. 微计算机应用,2004(2): 218 - 222.
- [5]Grefenstette J. Genetic algorithms for the traveling salesman problem in: proc. of 1 int. conf. on genetic algorithms and their applications [J]. Lawrence erlbaum association, 1985, 23: 166 - 168.

Realization of MATLAB on resolving the traveling salesman problem by virtue of genetic algorithms

WEN Qing-fang

(College of Porcelain Vocation Technology, Dehua 362500, Fujian, China)

Abstract: This paper has resolved the traveling salesman problem (TSP) by virtue of genetic algorithms (GA), all improved GA is suggested which gives the MATLAB program its crossover operator and mutation operator. It also solved the question of many GAs on searching TSP problem and gives the relative codes in MATLAB 6.5. Finally, numerical simulations of non-symmetric five-city TSP establish the efficiency of the procedure in this paper. Test shows that the improved GA has better performance than the traditional GA on the convergence speed and the fitness value. The improved GA obtains a better result. TSP problem is a typical problem to which Genetic Algorithm can be successfully used.

Key words: traveling salesman problem; genetic algorithms; Grefenstette codes; MATLAB program

(责任编辑:王桂珍)