

带变异特征的捕食搜索算法及其在 TSP 问题实验研究

姚世伟 陈 贤 沈海鸿
(中国地质大学(北京),北京 100083)

摘 要 原捕食搜索算法在 TSP 问题上解的质量与精度上都不高,即其具有容易陷入局部最优的缺点。引入变异特征行为以使得捕食搜索算法能够跳出局部最优。最后对四个常用 TSP 问题进行了仿真,结果表明,新算法在解的质量与精度上比原捕食搜索算法有了明显提高。

关键词 捕食搜索 旅行商问题 变异特征

中图分类号 TP183; **文献标志码** A

捕食搜索算法是一种解决 TSP 问题的有效启发式算法。但近年来国内对捕食算法的研究并不多^[1],主要着重与对它的应用以及与其他算法的拟合。本文主要着重与对捕食搜索算法本身的改进,借鉴了大多数算法借用变异来跳出局部的策略,将变异特征引入到算法中,仿真实验表明了这一改进具有有效性。

1 捕食搜索算法

1.1 捕食搜索算法的基本原理^[2,3]

通过模拟捕食动物的捕食策略,Alexandre Linhares 在 1998 年首次提出了一种新的针对组合优化问题的仿生算法^[4],即捕食搜索(Predatory Search, PS)算法。应用捕食搜索算法寻优时,先在整个搜索空间进行全局搜索,直到找到一个较优解;然后在较优解附近的区域进行集中搜索,直到搜索很多次也没有找到更优解,从而放弃局部搜索;然后再在整个搜索空间进行全局搜索。如此循环,直到找到最优解(或近似最优解)为止,见图 1。在捕食搜索算法中,使用限制(Restriction)来表征较优解的邻域大小。通过限制的调节,实现搜索空间的增大和

减小。

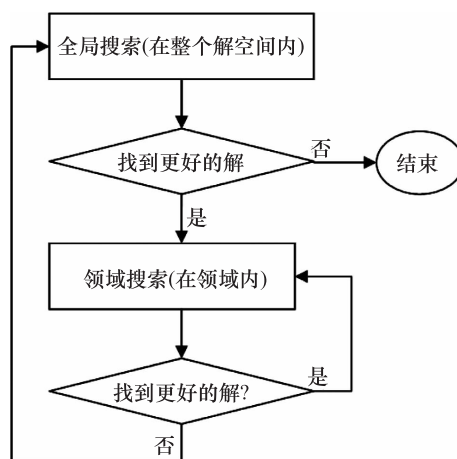


图1 捕食搜索算法

1.2 捕食搜索算法限制的选取^[2,3,5]

限制的选取是捕食搜索算法中非常重要的步骤,针对 TSP,将其定义为一个二元组 (Ω, Z) ,这里 Ω 是巡游路线的集合,函数 $Z: \Omega \rightarrow R$ 代表每个路线到对应的长度的变换,那么寻找路线 b 的限制,就是找到这样的一系列的 $\text{Restriction}(l), l \in \{1, 2, \dots, n\}$,使得 $A(b, \text{Restriction}(1)) \subseteq A(b, \text{Restriction}(2)) \subseteq \dots \subseteq A(b, \text{Restriction}(n))$,这里 l 称为限制的级别, $A(b, \text{Restriction}(l))$ 表示 b 在限制 L 下的邻域。

常用的限制选取策略:

随机取路线 b 通过 2-opt 邻域变换的 n 个路线,根据路线的长度升序排列依次赋值给 $\text{Restriction}(l), l \in \{1, 2, \dots, n\}$, 2-opt 邻域变换是指路线 $b = (x_1, x_2, \dots, x_n)$ 表示访问路线 $x_1 \rightarrow x_2 \rightarrow \dots \rightarrow x_n \rightarrow x_1$, 将其中的子串 $x_q, x_{q+1}, \dots, x_{q+r}$ 倒转得到, $x_{q+r}, x_{q+r-1}, \dots, x_q$ 从而得到新路线, 其中 $1 \leq q, q+r \leq n$ 。

1.3 原捕食搜索算法(PS)^[1,4,6]

1) 首先随机选取一个可行解 $x, x \in X$ (整个解空间), 令 $x_{\text{best}} = x, \text{count} = 0, \text{level} = 0$;

2) 若 $\text{level} < n$, 则随机选取 x 的 2-opt 邻域中 5% 的解 NS, 然后转到第 3 步; 否则结束整个过程;

3) 若选出的邻域解中的最优解 $f(x_{\text{min}}) < \text{Restriction}(\text{level})$, 令 $x = x_{\text{min}}$, 并转到第 4 步; 否则转到第 5 步;

4) 如果 $f(x) < f(x_{\text{best}})$, 令 $x_{\text{best}} = x, \text{level} = 0, \text{counter} = 0$, 重新计算 Restriction, 然后转到第 2 步; 否则转到第 5 步;

5) 令 $\text{count} = \text{count} + 1$ 。如果 $\text{count} > \text{CM}$, 那么令 $\text{level} = \text{level} + 1, \text{count} = 0$, 然后转到第 6 步; 否则转到第 2 步;

6) $L_L = \lfloor n/20 \rfloor, G_L = n - \lfloor n/20 \rfloor$ 如果 $\text{level} = L_L$ 则 $\text{level} = G_L$, 然后转到第 2 步; 否则直接转到第 2 步。

参数说明:

n 代表城市个数, C_M 代表局部搜索的最大次数, L_L 代表局部搜索最大 level 层数, N_s 代表领域 5% 的解的个数, G_L 代表全局搜索的起始 level 数, x_{best} 为当前最优解。[X] 代表对 X 取整数。

2 基于变异的捕食搜索算法

2.1 新算法的改进原理

2.1.1 捕食算法的缺点

原捕食算法在全局搜索的深度与广度上还不够大, 因此由局部搜索跳转到全局搜索时, 容易陷入局部最优, 在 TSP 问题上表现为解的质量与精度不高。这是因为整个解空间实在太太大, 只能取很小一部分解来作为全局参考解, 因此不能避免陷入局

部最优的情况, 这是几乎所有智能算法都存在的共有的问题。

2.1.2 改进的策略 1: 增大搜索空间的深度与广度

对于原捕食搜索算法, 在对全局搜索空间搜索时, 也仅是由限制级别 G_L 增大到 n , 搜索的次数较少, 故可以考虑增加全局搜索次数, 以获得更大的解空间, 进而获得更优的解。

2.1.3 改进的策略 2: 变异

对于如何跳出局部最优, 常用的策略是变异, 遗传算法通过变异来跳出局部最优解, 模拟退火也通过 Metropolis 准则(一种以概率接受较次解的策略)来达到跳出局部最优解^[7], 而 Metropolis 准则实际上也是一种变异策略。

本文算法变异的策略选择为在全局最优解的邻域中选取其最好的一个解作为当前的最好解, 变异次数由人为设定, 可以根据问题的规模来设计。

2.1.4 改进策略的可行性论证

对于 TSP 问题而言, 其局部最优解都聚集在全局最优解附近, 在未陷入局部最优时, 增大搜索空间, 可以更大概率发现新的更优解, 使其最后的解达到全局最优解或更接近全局最优解, 这在仿真实验过程中, 可以很直观地看到。

当算法陷入局部最优时, 通过变异就能达到跳出局部最优的目的, Metropolis 准则可以让算法从局部最优解处跳出, 并最终收敛(以概率 1 收敛)到全局最优解^[7], 这已经得到了理论上的证明。因此设定适当的变异次数, 就足以获得更优的解。

2.2 本文提出的捕食搜索算法

1) 首先随机选取一个可行解 $x, x \in X$ (整个解空间), 令 $x_{\text{best}} = x, \text{count} = 0, \text{level} = 0$;

2) 若 $\text{level} \leq n + 2$, 则随机选取 x 2-opt 邻域中 5% 的解 N_s , 然后转到第 3 步; 否则结束整个过程;

3) 若选出的邻域解中的最优解 $f(x_{\text{min}}) < \text{Restriction}(\text{level})$, 令 $x = x_{\text{min}}$, 并转到第 4 步; 否则转到第 5 步;

4) 如果 $f(x) < f(x_{\text{best}})$, 令 $x_{\text{best}} = x, \text{level} = 0, \text{counter} = 0$, 重新设定 Revolution, Maxcount 的值, 重新计算 Restriction, 然后转到第 2 步; 否则转到

第 5 步;

5) 令 $\text{count} = \text{count} + 1$ 。如果 $\text{count} > C_M$, 那么令 $\text{level} = \text{level} + 1, \text{count} = 0$, 然后转到第 6 步; 否则转到第 2 步;

6) $L_L = \lceil n/20 \rceil, G_L = n - \lceil n/20 \rceil$, 如果 $\text{level} = L_L$, 则 $\text{level} = G_L$;

7) 如果 $\text{level} = n + 1$, 则 $\text{level} = n$; $\text{Maxcount} = \text{Maxcount} - 1$; 如果 $\text{Maxcount} = 0$, 则 $\text{level} = n + 1$, 转第 8 步, 否则转第 2 步;

8) 如果 $\text{level} = n + 2$, 则 $\text{Revolution} = \text{Revolution} - 1$ 如 Revolution 不为 0, 则选出 x_{best} 的 5% 邻域解中的最优解 $f(x_{\text{min}})$, 令 $x = x_{\text{min}}, \text{level} = 0, \text{count} = 0$, 并重设定 Maxcount 的值。否则令 $\text{level} = n + 3$, 然后转第 2 步。

符号说明:

Maxcount 为在最高限制级别下的搜索轮次, 每一轮为 CM 次, Revolution 为变异的次数。

当搜索已经达到了最高限制级别时, 程序将继续进行 Maxcount 轮的搜索, 这相当于增大了搜索空间的深度与广度, 对解的改善有积极作用, 此可见步骤 7)。

在步骤 8) 里, 引入了变异, 以使得算法能跳出局部最优。通过设定 Revolution 的值, 来设定变异的次数。

3 仿真结果及其分析

3.1 实验数据背景

分别编程实现原捕食搜索算法与改进算法。

实验使用 att48 berlin52 eil76 eil101 四个旅行商问题进行测试, 对每个程序各独立运行 10 次。

参数设置: $\text{Revolution} = \lceil n/10 + 0.5 \rceil$; $\text{Maxcount} = \lceil n/10 + 0.5 \rceil$; $N_s = \lceil ((n \times n - n) \times 0.05 + 0.5) \rceil, C_M = n; [x]$ 表示对 x 取整。

3.2 实验结果

见表 1。

3.3 实验结果分析

改进算法, 在解的质量以及收敛程度相比原捕

食算法有了明显提高。改进算法在最优解命中次数

表 1 实验结果

用例	TSPLIB 结果	算法	最好解	最坏解	平均解	命中最优 解次数
Att48	33 522	本文算法	33 522	33 723	33 572.8	6
		PS	33 723	34 602	34 090.7	
Berlin52	7 542	本文算法	7 542	7 902	7 601.0	8
		PS	7 751	8 296	7 939.6	
Eil76	538	本文算法	538	540	538.3	8
		PS	538	553	546.4	1
Eil101	629	本文算法	629	638	631.8	3
		PS	640	661	649.9	

方面也十分不错, 同时也看到, 随着问题规模的加大, 算法的最优解命中次数也在减少, 根据 Metropolis 准则的论证, 可以通过加大变异的次数来增加命中率, 但同时增大变异次数就会同时增加运行时间, 这需要根据具体问题来实际调整。

增大与减小 Revolution 、 Maxcount 的值会使算法的运行时间增加与减少, 适当改变其值, 以使得算法运行时间在可接受范围之内。

4 结论

本文算法在原算法的基础上引入了变异特征, 同时在搜索的深度与广度上也做了一些改进, 由仿真结果可以看到新算法解的质量与收敛度较原捕食搜索算法有了明显提高。

在今后的研究过程中, 可以将更多的新元素引入到捕食算法中, 进一步发展捕食搜索算法。

参 考 文 献

1 王有为. 基于禁忌表的捕食搜索算法及其在旅行商问题中的实验研究, 系统工程理论与实践, 2008;2:131—136

2 李国宁. 基于捕食搜索的蚁群算法及其在旅行商问题中的应用研究. 广州: 华南理工大学, 2010

3 Linhares A. Preying on optima: a predatory search strategy for combinatorial problem. Proc of IEEE International Conference on Systems, Man and Cybemetics. Piscataway, NJ: IEEE Press, 1998: 2974—2978

4 Linhares A. State space search strategies gleaned from animal behavior : A travelin salesman experiment. Biological Cybernetics, 1998; 78 :167—173

6 蒋忠中,汪定伟. 车辆路径问题的捕食搜索算法研究,计算机集成制造系统,2006;12(11):1899—1902

5 Linhares A. Synthesizing a predatory search strategy for VLSI layouts. Unpublished Manuscript, National Space Research Institute, S. J. Campos,1999;3(2):147—152

7 冯 剑, 岳 琪. 模拟退火算法求解 TSP 问题,森林工程,2008; 24(1):94—96

The Predatory Search Algorithm Based on Variability and its Experimental on Traveling Salesman Problem

YAO Shi-wei, CHEN Xian, SHEN Hai-hong
(China University of Geosciences(Beijing), Beijing 100083, P. R. China)

[Abstract] The quality and accuracy of solution of original predatory search algorithm is not high on the TSP problem and it is easy to fall into local optimum. So this algorithm is introduced variations to make original predatory search algorithm can jump out of local optimum, By the simulation of four commonly used TSP problem, the results show that the new algorithm are significantly increased in the solution quality and accuracy than the original predatory search algorithm.

[Key words] predatory search algorithm variability TSP

(上接第 6647 页)

Design of Embedded GNSS Receiver Based on FPGA and DSP

YANG Shu-wei¹, ZHANG Bing¹, HU Rui²
(School of Electronics and Information, Jiangsu University of Science and Technology¹, Zhenjiang 212003, P. R. China;
Beijing Navgnss Integration Co. ltd², Beijing 100029, P. R. China)

[Abstract] Along with the deepening of GNSS receiver application, the performance requirements for power consumption and volume is higher and higher, the large-scale integrated circuit chips such as field-programmable gate arrays (FPGA) and high performance digital signal processor (DSP) are widely used in embedded GNSS receiver. A design of baseband signal processing of embedded receiver based on FPGA + DSP is introduced, and one of core design is the digital signal processing unit. FPGA is used to carry correlation. And DSP is designed to achieve processing satellite signal and the navigation data. System static test show that this receiver has characteristic of littler bulk, lower power and real-time.

[Key words] GNSS receiver FPGA DSP static test